Geoinformatica
DOI 10.1007/s10707-009-0090-7

openModeller: a generic approach to species' potential distribution modelling

Mauro Enrique de Souza Muñoz · Renato De Giovanni · Marinez Ferreira de Siqueira · Tim Sutton · Peter Brewer · Ricardo Scachetti Pereira · Dora Ann Lange Canhos · Vanderlei Perez Canhos

Received: 19 December 2007 / Revised: 15 June 2009
Accepted: 22 July 2009

Abstract Species' potential distribution modelling is the process of building a representation of the fundamental ecological requirements for a species and extrapolating these requirements into a geographical region. The importance of being able to predict the distribution of species is currently highlighted by issues like global climate change, public health problems caused by disease vectors, anthropogenic impacts that can lead to massive species extinction, among other challenges. There are several computational approaches that can be used to generate potential distribution models, each achieving optimal results under different conditions. However, the existing software packages available for this purpose typically implement a single algorithm, and each software package presents a new learning curve to the user. Whenever new software is developed for species' potential distribution modelling, significant duplication of effort results because many feature requirements are shared between the different packages. Additionally, data preparation and comparison between algorithms becomes difficult when using separate software applications, since each application has different data input and output capabilities. This paper describes a generic approach for building a single computing framework capable of handling different data formats and multiple algorithms that can be used in potential distribution modelling. The ideas described in this paper have been implemented in a free and open source software package called openModeller. The main concepts of species' potential distribution modelling are also explained and an example use case illustrates potential distribution maps generated by the framework.

M. E. de Souza Muñoz · R. De Giovanni (✉) · M. F. de Siqueira · R. S. Pereira · D. A. L. Canhos · V. P. Canhos
Centro de Referência em Informação Ambiental, Campinas, SP, Brazil
e-mail: renato@cria.org.br

T. Sutton
PO Box 942, Lanseria, Gauteng, South Africa, 1748

P. Brewer
Malcolm and Carolyn Wiener Laboratory for Aegean and Near Eastern Dendrochronology,
B-48 Goldwin Smith Hall, Cornell University, Ithaca, NY 14853-3201, USA

Published online: 18 August 2009

Keywords Potential distribution modelling · Ecological niche modelling · Predicting species distribution · openModeller

1 Introduction

Knowledge of the potential distribution of species is of great importance for developing strategies for conservation, public health and sustainable development. Example applications include: searching for rare or endangered species, planning new conservation areas, assessing the impact of human activities on biodiversity, predicting the impacts of climate change on species' distribution, preventing the spread of invasive species, identifying disease vectors, understanding the abiotic needs of species, increasing agricultural productivity, and others [1]. However, there are often insufficient biodiversity data to support these activities [2, 3]. In response to this problem, many modelling techniques have been used and developed in an attempt to calculate the potential distribution of species as a proxy for actual observations. Potential distribution modelling is the process of combining occurrence data (locations where the species has been identified as being present or absent) with ecological and environmental variables (such as temperature, precipitation, and vegetation) to create a model of the species' requirements [4]. Although there are many algorithms that can be used to create such models, unfortunately no algorithm is suitable in all situations. Algorithm suitability is determined by a variety of factors including the number of occurrence points, availability of absence data, type and number of environmental variables, and purpose of the experiment [5].

There has been considerable activity in the field of species' potential distribution modelling. New algorithms and tools are frequently created and compared with existing ones [6–9]. However, there is a practical gap between devising a new algorithm and implementing it as a usable software package. This gap is due to the different types of expertise required for the various areas involved in developing the software. Algorithms for modelling species' potential distribution are usually created by people with a strong background in mathematics and ecology. However, developing a usable and robust application for a new algorithm involves considerable additional effort and also requires a deep understanding of geospatial data. Ideally, such applications should be able to deal with a range of tasks such as transforming between different geospatial reference systems, handling geospatial data in different scales and extents, reading and writing geospatial data in different file formats, and facilitating data visualization. The final software package should also ideally offer pre-analysis and post-processing tools, providing support for a range of protocols and data standards for sharing and retrieving occurrence and environmental data.

These tasks are common to any implementation of potential distribution modelling software, but not directly related to the algorithm itself. To date, most software development has been carried out in isolation, producing separate software packages that are targeted to a single algorithm. This is the case for Domain [10], DesktopGARP [11] and Maxent [9] among others.

There are drawbacks in having a different software package for each algorithm. In particular, users need to learn multiple software applications to use different algorithms. Getting familiar with the parameters and methodology of an algorithm is something unavoidable for users wishing to make proper use of the algorithm itself. However, it should not be necessary to require that users learn the specific data preparation procedures and nuances of each different interface simply to run the same experiment using different algorithms.

Additionally, since each software package has different input requirements, it is usually necessary to perform specific data conversions to run the same experiment across multiple packages. This makes it difficult to compare results. Fair comparisons should ideally be performed using identical input data and an identical computational environment.

Although literature mentions potential distribution modelling frameworks that can run different algorithms [12, 13] none seem to be open source and easily available. Openness and accessibility are important features to allow better evaluation by the scientific community, estimulate collaboration, and improve research productivity.

The issues described here provided the main motivation for a research effort whose aim was to create a generic open source framework for potential distribution modelling.

This paper is organized as follows: Section 2 describes the main concepts of potential distribution modelling. Section 3 presents the scope, framework design, main components, and a sample use case of openModeller. Section 4 discusses concerns and limitations about potential distribution models. Section 5 contains the main conclusions of this work.

2 Species' potential distribution models

Species' distributions are influenced by many factors [14]. While suitable environmental conditions determine a species' fundamental niche [15], biological factors such as competition tend to reduce the fundamental niche into the realized niche [15]. The potential distribution of a species can be seen as the geographical expression of its realized niche at a particular time (i.e., where there is a fulfillment of both abiotic and biotic requirements [1]). It is important to note that actual distributions often do not correspond to the modelled potential distribution. Stable populations can only be found in regions that have been accessible to the species since its origin (via natural, anthropogenic or other means of dispersal) [16]. Despite this, potential distribution models provide a powerful tool for predicting species distribution in different geographical and temporal contexts, as well as for studying other aspects of evolution and ecology [1].

There are many methods that can be used to model the potential distribution of species. Most are data-driven methods based on a correlative approach. The correlative approach tries to build a representation of the fundamental ecological requirements of a species based on the environmental characteristics of known occurrence sites [17]. In this case, three types of input data are required to generate a model: occurrence data, environmental data, and algorithm-specific parameters. When projected into a geographical region, the resulting map will typically show areas that are ecologically similar to those where the species is known to occur [16]. Community models follow a similar approach, except that they include data from other species belonging to the same biological group as the species being modelled [6, 18]. Another method known as mechanistic modelling is based on the physiological responses and constraints of organisms to the environment [19]. Since this method requires an in depth understanding of the species' physiology, its use is still limited. This article focuses on correlative modelling methods.

2.1 Species' occurrence data

Species' occurrence data are records of where individuals of a certain species were collected or observed. This kind of data is typically comprised of a unique identifier (e.g., an accession code), taxonomic identification, location (including longitude, latitude, and ideally, datum and the associated error), abundance, and a corresponding date. Occurrence

data is normally obtained from biological collections or field observations. Biological collections hold approximately 2.5 billion records [20] collected over three centuries. These records are maintained by institutions such as natural history museums, herbaria and culture collections. Observation data are usually gathered during surveys by researchers and also by citizen volunteers that participate in numerous biodiversity monitoring projects.

Of these 2.5 billion estimated records, a concerted digitization effort is under way and the data is quickly becoming available over the Internet [21]. The exchange and sharing of such data is facilitated by the adoption of common data formats [2]. One of the most prominent sources of species occurrence data is the Global Biodiversity Information Facility (GBIF), currently serving more than 150 million records from several institutions around the world. However, it should be noted that much of this data may still not be usable for potential distribution modelling due to data quality issues [3].

2.2 Environmental data

Environmental data is mostly available in the form of georeferenced raster [22] layers. These environmental layers typically represent abiotic conditions that will be used to determine the potential distribution of the species being modelled. They may represent, for example, temperature, precipitation, radiation, wind, evaporation, topography, soil moisture, and vegetation coverage. Geospatial raster data are delineated by an extent of coverage (coordinates of its corners), an associated geospatial reference system and a matrix of cells containing the actual data for the region. The region covered by one cell is represented by a single number (cell value).

Environmental rasters are typically produced from satellite data, weather station data (by interpolation of raw data) or some other set of measurements. The environmental coverages currently available in digital format encompass the key physical variables that commonly influence macro-distributions of species [17]. These data are produced and made available by many different sources, ranging from local initiatives to international efforts, from companies and non-government organizations to government agencies. Examples include NASA, USGS, INPE, IPCC, CIAT, the UK Met Office and WorldClim among others.

2.3 Algorithms for modelling

There are many different methods that have been used to model the potential distribution of species. Some were developed specifically for this purpose, such as Bioclim [23] and GARP (Genetic Algorithm for Rule-Set Production) [24]. Others are general methods that are already widely used in other areas and have been applied to species' distribution models. These include various statistical approaches, like regression analysis [25–31], discriminant analysis [32, 33], and machine learning techniques [7, 9, 34–36].

To use these algorithms, researchers have either developed their own software applications or used general packages for statistics and numerical computation, such as R, GNU Octave and MATLAB. Software applications that have been specifically developed for species' distribution modelling tend to offer a better experience for users, although even these bespoke packages are limited to a single algorithm and were not designed for extensibility. Generic statistical packages are usually very flexible, with many plug-ins available and sometimes a graphical user interface. However, in this case usability tends to be compromised by factors that are unrelated to the modelling process, such as the need to learn a programming language or to use complex statistical software.

3 A generic potential distribution modelling framework

The openModeller framework was developed to perform the most common tasks related to species' potential distribution modelling based on the correlative approach. Despite this original purpose, the framework follows a generic design that may allow other areas such as archaeological research and geological prospecting to make use of it. Virtually any problem that seeks patterns for the geographical distribution of an entity based on environmental attributes could make use of the framework described here. This possibility is also enabled by the fact that openModeller can run different algorithms, read and write different data formats, and be used by different types of interfaces.

Temporal data is not handled by the framework, therefore it can only produce static models. When users have time series of environmental layers, openModeller can be used to generate independent potential distribution maps for each scenario. The result can then be post-processed by other tools.

3.1 System requirements

The following requirements were considered to be most important during the software design process. A generic potential distribution modelling framework should be able to:

- Read a set of georeferenced points (presence or absence) with support for different coordinate systems.
- Load attributes for each point from a list of specified raster geospatial datasets.
- Read and write different raster geospatial formats.
- Permit the use of rasters with different cell sizes and extents in the same experiment.
- Permit the use of different modelling algorithms, isolating algorithm logic from other issues related to geospatial data, input and output formats.
- Project resulting models back into geographical space.
- Allow for the existence of multiple end-user interfaces.
- Run on commonly used operating systems including Microsoft Windows™, Mac OS X™, and GNU/Linux.
- Run with optimal performance.

Additionally, to encourage external review and possible new partnerships, it was decided that the development process should be collaborative and transparent. As a consequence, openModeller is free and open source software.

3.2 Architecture

The openModeller framework follows a common approach taken by other similar generic libraries. For optimal performance and cross platform portability, it was written in ANSI C++. The framework contains a set of classes that are focused on the core modelling functions and delegates the task of creating user interfaces to other high level applications. This architecture allows openModeller to be used by and embedded in different applications such as the openModeller Desktop graphical user interface.

The framework was designed in a modular fashion making use of other external libraries where available. Proj.4 is used to perform transformations between different cartographic projections and GDAL is used to read and write different GIS raster formats. TerraLib, an open source GIS library, can optionally be used to read and write both raster and point data that are stored in TerraLib relational databases.

To facilitate the implementation of new algorithms, the framework takes care of all common tasks and treats algorithms as plugins (dynamic libraries). This means that their implementation must conform to an abstract interface that specifies all common methods that are needed to interact with the core modelling components. Using this approach, new algorithms can be loaded at run-time. Algorithms can also make use of additional external libraries when necessary, such as the GNU Scientific Library (GSL), which is already being used by one of the algorithms. Figure 1 shows a diagram representing the basic architecture of openModeller.

The main modelling functions are exposed through specialized classes and a more generic controller class. This enables other languages and applications (like GIS software) to make use of openModeller via its programmatic interface.

Besides the C++ API that can be directly used from openModeller's framework, an additional binding is provided for the Python programming language. This was achieved using the SWIG interface translator, which can also be used to produce bindings for other languages in the future.

The following interfaces were developed on top of the framework:

- ***Console and command-line interface:*** A set of shell programs including *om_console* (both a console and command-line application that can be used to create and project models based on a parameters file with key-value pairs), *om_model* (command-line application to

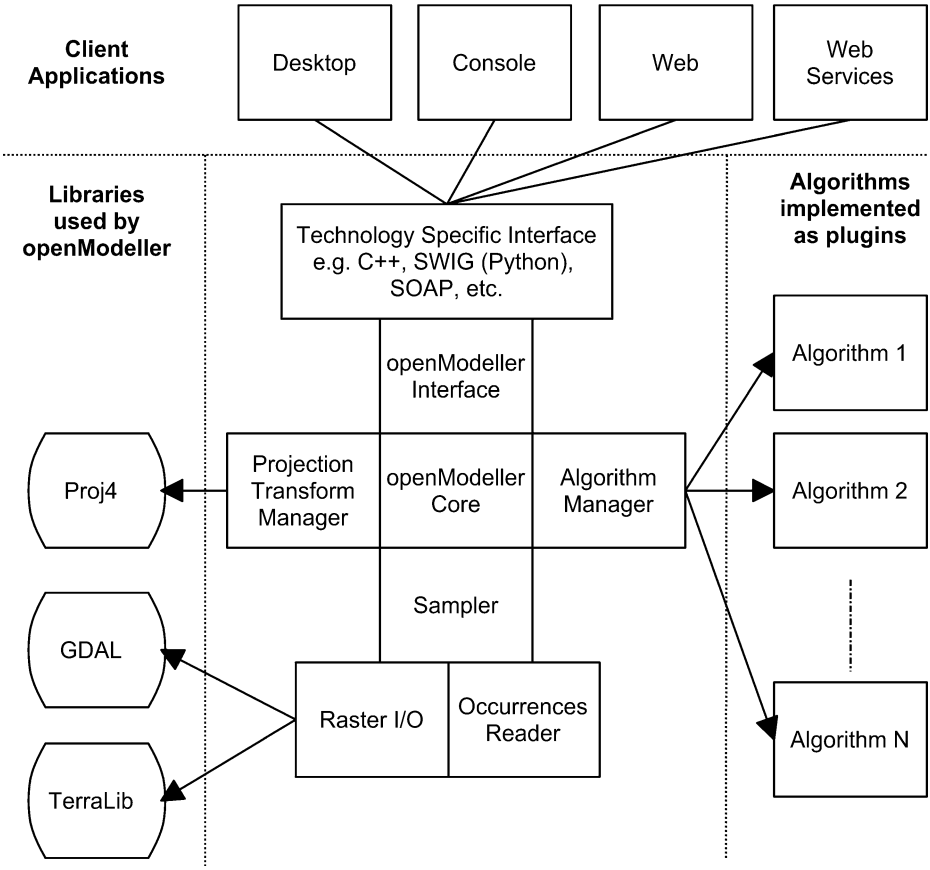


Fig. 1 openModeller's architecture

create models, also accepting parameters in XML), *om_project* (command-line application to project models, also accepting parameters in XML), *om_test* (command-line application to test models, also accepting parameters in XML), *om_algorithm* (command-line application to display information about algorithms), *om_sampler* (command-line application to dump environmental data), *om_points* (command-line application to read occurrence data from various formats, including remote sources) and *om_pseudo* (command-line application to generate random points inside a specified mask region). In addition to being useable as standalone programs, these applications can also be used by shell scripts to run complex experiments in batch mode or by other types of applications that can interact with scripting environments. Two other related programs, *om_viewer* and *om_niche*, depend on the X11 library and can be used to visualise distribution maps and models, respectively. All these applications are distributed with the openModeller framework.

- ***Graphical User Interface:*** openModeller Desktop is a separate software application, also free and open source, that provides a user friendly graphical front-end to the framework. The current version (1.0.9) has been designed to support large experiments where multiple species can be modelled using multiple algorithms. It can use the locally installed openModeller library bundled with the application or it can interact with a modelling service, in this case sending requests to a remote modelling server responsible for running the experiment and then retrieving the results.
- ***Web Services:*** Web Services interfaces allow remote program interaction between client and server software through a specific protocol. Such a protocol has been defined as part of the openModeller project making use of HTTP and SOAP. A modelling server has been developed in C++ as well as two clients: one in Perl and the other in C++ as part of the openModeller Desktop application.
- ***Web Interface:*** A prototype Web interface was developed so that users can run experiments using an Internet connection and a web browser. At the moment this interface is still experimental and unavailable to the general public.

3.3 Modelling approach

The framework builds on the notion that a *Potential Distribution Model* (**PD-Model**) is a mathematical model generated by an algorithm based on the correlative approach (see Section 2). This process receives as input a set of georeferenced occurrence points (presence or absence), a set of georeferenced environmental rasters and a set of algorithm-specific parameters.

To create a PD-Model, openModeller reads the corresponding environmental values for each occurrence point from the input rasters. The occurrence points are transformed into data structures called samples, whose elements represent the environmental conditions at each location. The set of all samples is then used by an algorithm to find a representation of the species' niche in the environmental space. The result is a PD-Model, which can be a probabilistic data model or a mathematical function that relates environmental conditions to the suitability for species existence. Algorithms are free to internally make use of additional species-specific data - an approach taken, for example, by the AquaMaps [37] algorithm. They are also free to make use of other algorithms internally, eventually combining multiple PD-Models into a single one. This allows community models to be implemented in different ways.

For some algorithms, the mathematical representation of the PD-Model can be easily interpreted. Simple envelope-based algorithms like Bioclim or some of the machine learning algorithms like Decision Trees [38] fall into this category. This allows PD-Models to be used for special purposes such as predicting the distribution of hypothetical species

ancestors by fitting PD-Models to phylogenetic trees [39, 40]. Most types of PD-Models, however, are difficult to interpret. This is the case for PD-Models generated by GARP, SVM (Support Vector Machines) [41] and ANN (Artificial Neural Networks) [42].

However, PD-Models are typically not interpreted, but rather used to project the model back into a target geographical region. In this process, the environmental conditions are iteratively read from a set of rasters for each cell position associated with a target region. This set of rasters can be the same as used to generate the PD-Model (native projections). Alternatively, an equivalent set of rasters related to a different period in time or different geographical region could be used. The set of environmental conditions for each cell is then passed as a parameter to the algorithm. The algorithm returns a prediction value corresponding to the suitability of this environment for the species. Each prediction is then written to a corresponding cell in the georeferenced output map, and the final result is a *Potential Distribution Map* (**PD-Map**). A PD-Map represents the potential distribution of a species in a particular geographic region at a particular time. Prediction values can be real probabilities or can be the result of a function that returns continuous or categorical values depending on the algorithm used. The meaning of prediction values on a PD-Map can only be correctly interpreted by understanding the algorithm that produced the PD-Model.

A more precise description of the whole process is illustrated by the following pseudo code:

```
01. Let O = the set of n occurrence points;
02. Let E = the set of m environmental rasters;
03. Let V = the set of n environmental conditions related to O;

04. // Find the set of environmental conditions for each occurrence.
05. For each occurrence o in O do:
06.  Let (lat, long) = location of occurrence o;
07.  Let v = a vector of environmental conditions;
08.   For i = 1 to m, do:
09.    Let v[i] = value of raster E[i] at (lat,long);
10.   done
11.  Insert v into V;
12. done

13. // Generate PD-Model based on V and on algorithm parameters.
14. Let alg = new Algorithm object instance;
15. Set alg specific parameters;
16. If alg.needsNormalization() Then
17.  Normalize V;
18. Feed alg with V;
19. alg.initialize();
20. While Not alg.done() do:
21.  alg.iterate();
22. done
23. alg.finalize();

24. // Generate PD-Map (PD-Model projection).
25. Let Model = alg.getModel();
26. Let Map = the resulting potential distribution map;
27. For each cell c in Map do:
28.  // Find the environmental conditions at the location of c.
29.  Let (lat, long) = position of cell c (geometric center);
30.  For i = 1 to m, do:
31.   Let v[i] = value of raster E[i] at (lat, long);
32.  done
33.  Let Prediction = Model.getValue(v);
34.  Set the value of cell c with Prediction;
35. done
```

3.4 Occurrences

openModeller has a generic *OccurrencesFactory* component that can be used to instantiate different drivers to read point data. Currently two drivers can be used to read local data: a driver for reading delimited text files and a TerraLib driver that can read point data from TerraLib relational databases. With the growth of biodiversity Web Portals and the establishment of protocols and data standards for biodiversity data [2, 43–45], two other drivers were implemented to retrieve occurrence data from the following remote data sources: GBIF's database (through GBIF's occurrence record data service) and TAPIR/DarwinCore providers. All drivers implement the openModeller *OccurrencesReader* API, and any number of additional drivers can be developed in the same way. These drivers provide openModeller with a set of occurrence locations, each characterised by an identification (unique identifier), label (normally the species determination), latitude, longitude, and abundance. An abundance of zero can be used to indicate absence. A coordinate system must also be specified for the set of occurrence locations using the Well-Known Text (WKT) descriptive format as defined by the Open Geospatial Consortium (OGC) specification for coordinate transformation services. When occurrences are loaded, openModeller converts them to an internal coordinate system (Lat/Long WGS84) using the Proj.4 library.

3.5 Rasters

openModeller has a generic *RasterFactory* component that can be used to instantiate the appropriate *Raster* driver to read or write raster data. Currently there are two implementations of the *Raster* API: a GDAL driver and a TerraLib driver. These drivers are responsible for reading raw raster data and delivering it to openModeller as a cell matrix, as well as for producing the final PD-Maps. GDAL itself is a generic raster library that supports a large number of different formats. Currently, when rasters have multiple bands openModeller will only access the first band. Environmental data access is also limited to raster data. In the future this could be expanded to vector (polygon) coverages.

The framework can deal with rasters with different cell sizes. To enable this during projection, one raster is nominated by the user as a template layer. When sampling rasters, the cell size of the template layer is used to determine the sampling interval. If raster layers with different cell sizes are encountered, the *Sampler* component interpolates data from each raster to return a single number for the areal extent represented by a single cell of the template layer. The Sampler implements the *nearest neighborhood* algorithm, but any other bidimensional interpolation algorithm such as *linear* and *bi-cubic* interpolation can be easily implemented through programmatic class extension. openModeller performs all necessary transformations when reading from rasters with different geographical projections. From the algorithm developer's point of view, all environmental raster data is always transparently read and resampled.

Environmental rasters can be located on a host that is remote from the computer carrying out the modelling tasks. However, generating PD-Maps requires reading iteratively each cell from each environmental raster. Since raster files can be large, currently this process rapidly becomes prohibitive over an Internet (and in some cases even an Intranet) connection. Where bandwith allows, remote rasters can be read using the GDAL driver through WCS (Web Coverage Service). If necessary, more drivers can also be developed based on the openModeller *Raster* I/O interface in the future.

3.6 Algorithms

openModeller can deal with different types of algorithms, ranging from simple and intuitive ones to more complex, sometimes non-deterministic algorithms. The framework also serves as a platform to experiment with the development of new approaches. Currently, algorithms must be written in the same language as the framework: ANSI C++. Since C++ allows integration with statistical packages such as R, it may be possible in the future to also run algorithms written directly in the R language.

In the framework, each algorithm has its own class extending an abstract *AlgorithmImpl* class. This abstract class facilitates the implementation of new algorithms by providing a standard way to:

- *Define and expose algorithm's metadata:* Metadata is used to identify the algorithm, distinguish between different versions, provide information about algorithm functionality, give credit to authors and developers, and also indicate what kind of input data it accepts (whether it accepts categorical maps, and whether it requires absence points).
- *Define and expose parameter metadata:* Algorithms indicate which parameters are needed, providing clients with additional descriptive information. This information can be passed to an interface from where parameter values can be specified by users. Examples of parameter metadata are: *name*, *data type*, *description*, *valid range*, and *default value*.
- *Get parameter values:* For each parameter described by the algorithm metadata, openModeller can retrieve the corresponding value in a computational way.
- *Get occurrences data:* All input points (presence and/or absence) are directly available to algorithms by means of a *sampler* object. When absence points are not available but required by the algorithm, the same *sampler* object can be used to generate pseudo-absence points by sampling random points from the background.
- *Get environmental data:* Algorithms can use the same *sampler* object to read environmental conditions at some desired coordinate (latitude, longitude). Usually this is used to get the conditions for each input occurrence. By using the *sampler*, algorithms do not need to worry about sampling methods, GIS map formats, scale problems (including matching between environmental rasters with different scales), or cell value normalization (for example, algorithms can instruct openModeller to return environmental conditions scaled to a certain range).

Algorithm execution is controlled by openModeller, which calls a sequence of methods available from the abstract *AlgorithmImpl* class. Real algorithms extending this class must implement the methods *initialize* (used for general initialization procedures) and *getValue* (used to return a prediction value given an environmental condition after PD-Model creation). The purpose of an algorithm is to create PD-Models. Non-iterative algorithms can create the PD-Model in a single step, which can take place during initialization. Iterative algorithms can use the *iterate* method to create the PD-Model. In this case, openModeller keeps calling the *iterate* method until the *done* method returns true. Algorithms can also implement the *getProgress* method allowing users to keep track of the execution progress. This standardized way of interacting with algorithms does not allow user interaction during PD-Model creation. The only opportunity for user interaction is to specify parameter values before algorithm initialization.

PD-Models are stored as an XML file including data about environmental layers, occurrence points and algorithm parameters that were originally used as input. This data is stored in a standard way regardless the algorithm used. The XML also includes a specific

section to store PD-Model properties calculated by the algorithm. In this case, each algorithm defines its own PD-Model representation and must be able to encode and decode that specific XML section accordingly. Methods *_setConfiguration* and *_getConfiguration* must be implemented by algorithms for this purpose. The following XML illustrates a simple PD-Model generated with the Bioclim algorithm. The PD-Model was created with two layers and five presence points. Inside the <Model> element, each algorithm will represent the PD-Model in a different way:

```
<SerializedModel>
 <Sampler>
  <Environment NumLayers="2">
   <Map Id="precipitation.tif" IsCategorical="0"/>
   <Map Id="temperature.tif" IsCategorical="0"/>
   <Mask Id="precipitation.tif"/>
  </Environment>
  <Presence Label="some species" Count="5">
   <CoordinateSystem>
GEOGCS[&quot;WGS84&quot;,DATUM[&quot;WGS84&quot;,SPHEROID[&quot;W
GS84&quot;,6378137.0,298.257223563]],PRIMEM[&quot;Greenwich&quot;
,0.0],UNIT[&quot;degree&quot;,0.01745329251994295],AXIS[&quot;Lo
ngitude&quot;,EAST],AXIS[&quot;Latitude&quot;,NORTH]]
   </CoordinateSystem>
   <Point Id="1" X="-68.849999" Y="-11.150000"/>
   <Point Id="2" X="-67.379999" Y="-14.320000"/>
   <Point Id="3" X="-64.700000" Y="-15.970000"/>
   <Point Id="4" X="-71.269999" Y="-11.919999"/>
   <Point Id="5" X="-72.829999" Y="-12.330000"/>
  </Presence>
 </Sampler>
 <Algorithm Id="BIOCLIM" Version="0.2">
  <Parameters>
   <Parameter Id="StandardDeviationCutoff" Value="0.674"/>
  </Parameters>
  <Model>
   <Bioclim Mean="207.80 2458.81" StdDev="70.51 136.79" Minimum="90
2113.01" Maximum="319 2589.48"/>
  </Model>
 </Algorithm>
 <Statistics/>
</SerializedModel>
```

Once created, PD-Models can be used to generate one or more PD-Maps, each covering a different region and/or a different period in time. In each case, the same environmental variables that were used to generate the PD-Model need to be used to generate the PD-Map. The matching in this case is semantic—the same data sets do not need to be used, but the data sets used in the projection should represent the same environmental variables used during PD-Model generation. PD-Models can also be stored on disk for later usage or be transmitted to other computers to parallelize PD-Map generation. Each cell of a PD-Map contains a prediction value calculated by the algorithm and returned by the *getValue* method.

The current openModeller version (1.0.0) includes the following algorithms: ANN, AquaMaps, Bioclim, two different implementations of GARP (for individual runs and GARP Best Subsets [4]), CSM (Climate Space Model) [46], a generic distance-based algorithm (Environmental Distance), SVM, and Envelope Scores.

Figures 2, 3, 4 and 5 show different PD-Models in the environmental space for different algorithms using the same input data: 65 presence points from *Thalurania furcata boliviana*

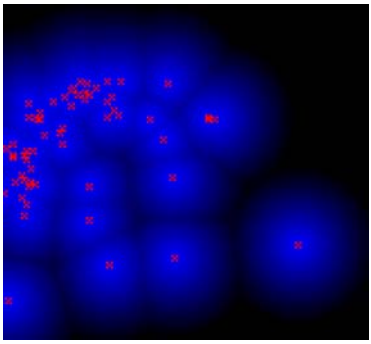


Fig. 2 PD-Model generated with Environmental Distance to the nearest point using the Euclidean metric

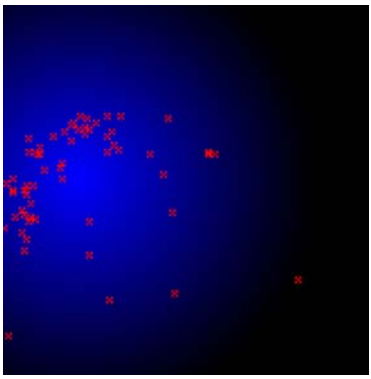


Fig. 3 PD-Model generated with Environmental Distance to the average point using the Euclidean metric

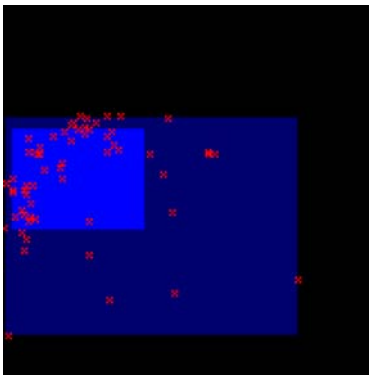


Fig. 4 PD-Model generated with Bioclim

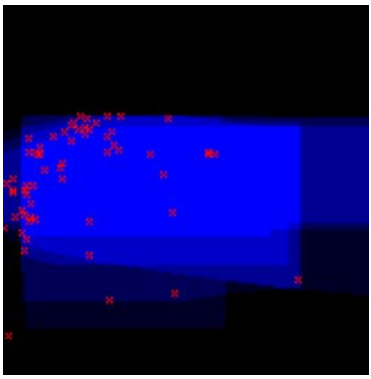


Fig. 5 PD-Model generated with GARP Best Subsets

(Boucard, 1894) and two environmental layers, precipitation in the "x" axis and temperature in the "y" axis. Red dots represent the original presence points and the blue density represents prediction values.

In Fig. 2, the prediction value is inversely proportional to the Euclidean distance to the nearest presence point. In this example the maximum distance parameter was set to "0.1". In Fig. 3, the prediction value is inversely proportional to the Euclidean distance to the average presence point. In this example the maximum distance parameter was set to "0.3". Figure 4 shows the bioclimatic envelopes calculated with Bioclim using a cutoff parameter of "0.99". In this case the space is divided into three areas: a) Suitable: when all environmental values fall within the interval [m−c*s, m+c*s] where "m" is the mean, "s" is the standard deviation and "c" is the cutoff parameter; b) Marginal: if at least one environmental value falls outside the suitable envelope but still inside the minimum and maximum range for the corresponding variable; c) Unsuitable: if at least one environmental value falls outside the marginal envelope. Figure 5 shows a model generated by the GARP Best Subsets algorithm. GARP is a genetic algorithm that creates a set of rules based on the environmental variables to determine suitable areas for the species. The GARP Best Subsets procedure runs a number of GARP models, chooses the best models based on omission and commission errors and then overlays the best models. In this example 20 GARP models were run to select the five best rulesets. The other parameters were: Training proportion (0.5), hard ommission threshold (100), models under ommission threshold (20), commission threshold (50), commission sample size (10,000), maximum generations (400), convergence limit (0.1), population size (50) and resamples (2,500).

3.7 Sample use case

Mathematical models are an approximation of reality based on assumptions. PD-Models are no exception. Although there are many issues that can affect results, one of the advantages of using generic modelling software is that researchers can use an experimental approach and try many algorithms. Various PD-Models can be produced using different algorithms with different parameters and input values enabling researchers to compare results and then decide if and how to use them.

One of the interfaces developed on top of openModeller, known as openModeller Desktop, enables users to set up complex experiments involving multiple sets of occurrence points, algorithms and projection scenarios. Figure 6 shows a screenshot of the "Experiment designer" window. The application also includes a simple panel that can be used to visualise PD-Maps (Fig. 7).

Figures 8 and 9 show two PD-Maps that were generated by openModeller using different algorithms and the same input data (occurrence points for *Caryocar brasiliense* Cambess. and a set of environmental layers that may influence the species' distribution). Both PD-Maps are displayed in pseudocolor with the test points as a dot overlay.

Caryocar brasiliense Cambess. is a tree characterized by it's twisted trunk covered by a thick rough bark. It occurs in the Cerrado (Brazilian savannah) open field, typical Cerrado and Cerradão physiognomies [47] across a wide area in the central region of Brazil. The PD-Maps illustrated here were produced using 114 presence points that were collected using a rapid survey technique [48, 49]. From these points, 64 were collected throughout Brazil, mainly during the 1990's [50]. These points are estimated to have a precision of 1 Km², which is still acceptable for regional-scale modelling experiments. The other 50 points were collected in the State of São Paulo, Brazil, during the period of 1999–2001, and were obtained using a Global Positioning System (GPS) unit with precision of approximately 100 m² [49].

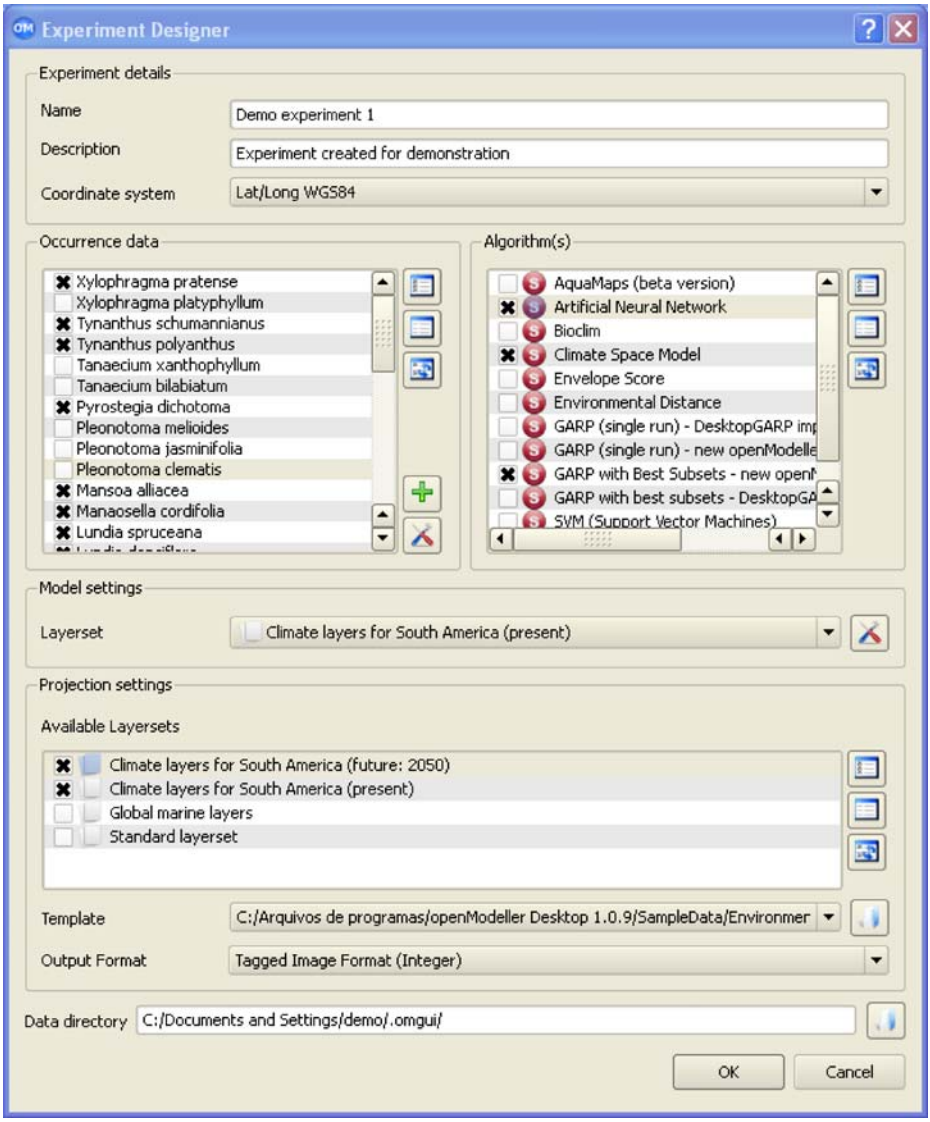


Fig. 6 Screenshot of the Experiment designer window (openModeller Desktop version 1.0.9)

The following environmental data layers were used:

1) Average rainfall in January (resolution: 2.5 min);
2) Average rainfall in July (resolution: 2.5 min);
3) Maximum temperature in January (resolution: 2.5 min);
4) Minimum temperature in July (resolution: 2.5 min);
5) Elevation (resolution: 30 s).

The climate data layers (1 to 4) came from the WorldClim [51] current conditions dataset (1960–1990), while the topographic layer (5) came from the HYDRO1 k project.

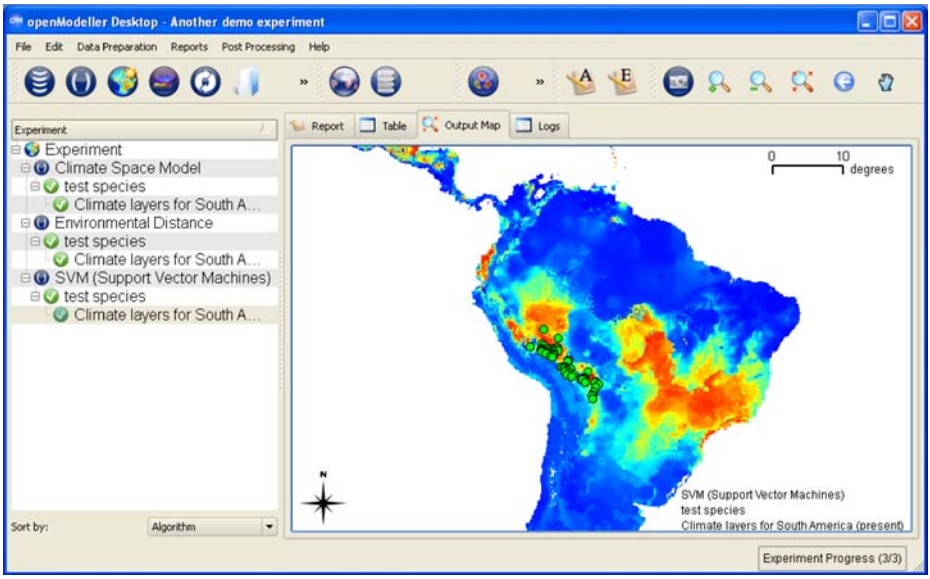


Fig. 7 Screenshot of window showing experiment results (openModeller Desktop version 1.0.9)

The aim of this experiment was to illustrate PD-Maps generated by openModeller and demonstrate the capacity of different algorithms to represent the potential distribution of the species. The 114 occurrence points were randomly separated into test and training datasets, each one with 57 points. The training dataset was then used to generate two PD-Models using different algorithms: Bioclim and GARP Best Subsets. Both PD-Models were tested with the second dataset and the statistical results were highly significant (binomial, one-tailed, and chi-square tests, $p<0.0001$). The PD-Models were then projected back into geographical space using the same environmental layers to produce the PD-Maps (Figs. 6 and 7). The Bioclim implementation produces only three discrete prediction values—1.0, 0.5 and 0.0—based on its own notion of bioclimatic envelopes. Brown areas represent the suitable envelope (highest prediction value) and orange areas represent the marginal envelope (medium prediction value) while remaining areas represent unsuitable regions for the species. The GARP Best Subsets implementation produces a wider range of prediction values based on the number of models that were overlayed. In this case the colors are ramped based on the number of GARP models predicting presence, where brown represents high prediction values and yellow represents low prediction values. These scaling differences between the two algorithms are intrinsic to their nature. However, in terms of predicted area, both algorithms produced similar results that approximately coincide with the Cerrado biome in central Brazil. Despite not having an input layer related to vegetation coverage, both algorithms were able to predict the Cerrado biome region using points from a single species along with a few climatic and topographic layers.

4 Discussion

Although openModeller provides a powerful tool for potential distribution modelling, care must be taken when interpreting results. Users should be familiar with the underlying logic

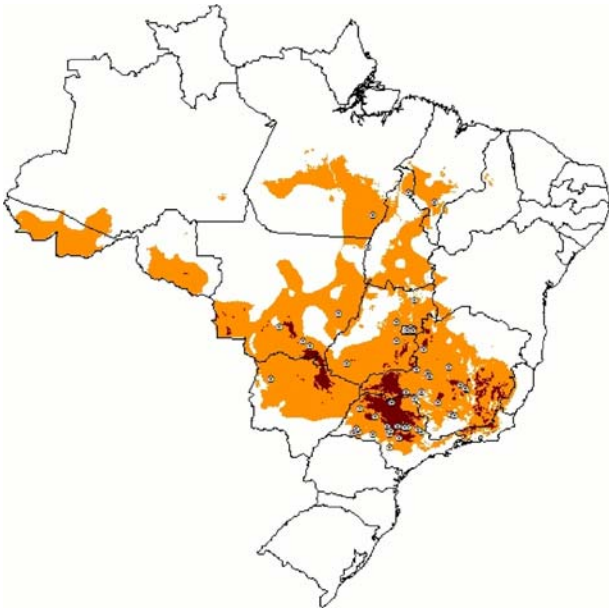


Fig. 8 PD-Map for *Caryocar brasiliense* Cambess. generated with the Bioclim algorithm and projected into Brazil

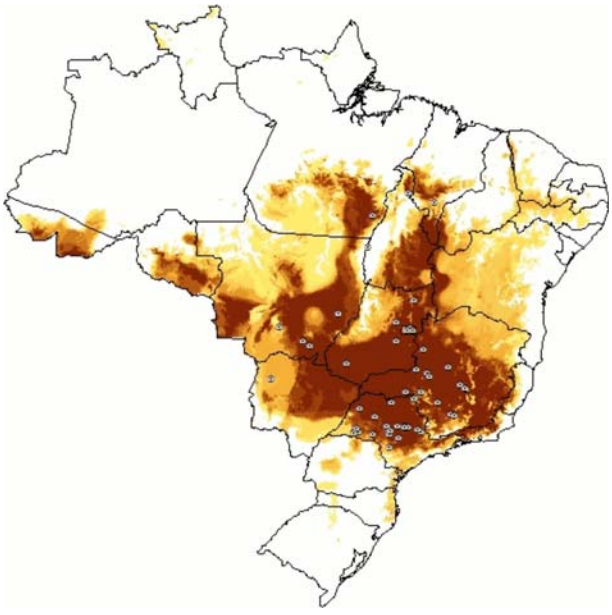


Fig. 9 PD-Map for *Caryocar brasiliense* Cambess. generated with the GARP Best Subsets algorithm and projected into Brazil

of the algorithm being used so as to understand its advantages and limitations. With this background knowledge, the input requirements, parameter meanings, provenance, and accuracy of input data can then be properly considered before trying to interpret a result.

When a resulting PD-Map is interpreted, it is important to remember that algorithms will usually only find regions that "resemble" those where the occurrence points are located, based on the layers provided [16].

Other factors such as biotic conditions (interactions with other species), ecological barriers (regions that are inaccessible for dispersal by the species) and evolutionary aspects are also not necessarily taken into account by the algorithm. One must always bear in mind that the tools are intended to facilitate the work of specialists and not to substitute them.

4.1 Concerns about input data

Accuracy of PD-Models and the corresponding PD-Maps is directly related to the accuracy of input data. There are important aspects concerning quality and accuracy of occurrence and environmental input data that must be considered when used in the context of PD-Modelling [52].

4.1.1 Occurrence data

- *Coordinate precision:* In biological collections, specimen location is often only described by the county and/or locality where the specimen was found. In these cases, locations need to be translated into geographical coordinates if they are to be used by PD-Modelling algorithms. This translation process can be carried out in different ways [53] but it inherently entails a geographical uncertainty, sometimes as large as a county radius. It is important to note that even when a location is determined with a GPS there is an associated error that must be taken into account. Ideally, uncertainty values should always be provided as part of location data, so that applications or researchers can decide if the corresponding occurrence record is suitable for a particular analysis.
- *Misidentification:* Specimens are sometimes misidentified [54], either because they were labelled with an invalid or outdated name, or because they belong to groups of organisms that are taxonomically difficult to identify. It has been estimated that specimen misidentification rates are in the range of 5% to 60% [21]. In these cases an incorrect niche may be modelled.
- *Digitization error:* There can be errors during the digitization process or even during label transcription. One of the most frequent errors when recording coordinates in decimal degrees is to omit the '−' sign for latitudes in the southern hemisphere and for longitudes in the eastern hemisphere. Another common error is to switch latitude and longitude.
- *Lack of absence points:* Although *presence*-only data satisfies the mathematical requirements of clustering algorithms [55] like Bioclim and CSM; classification algorithms [56], such as GARP, depend on *absence* data. If absence points are not given, classification algorithms can ask openModeller to produce *pseudo-absence* points. Usage of pseudo-absence points tends to reduce model accuracy, especially when they are randomly generated without any previous knowledge about the species distribution. Currently, openModeller provides only a random generator, but other pseudo-absence sampling strategies [57] can be implemented to minimize the possibility of feeding classifiers with incorrect data. Therefore, when real species absence data is available, it should always be used when running a classification algorithm.

4.1.2 Environmental layers

- *Resolution:* Resolution is the geographical size of each raster cell. Choosing the appropriate resolution of environmental layers in a modelling experiment is a critical step and also depends on the goal of the experiment [58]. In general, resolution must be compatible with occurrence data uncertainty. Too fine a scale can lead to errors if the corresponding occurrence data uncertainty is greater than the cell size, while too coarse a scale may produce results that fail to distinguish between suitable and unsuitable areas for the species [59]. A similar problem may happen with categorical data, where cell values represent the most abundant value for the corresponding area. Since openModeller does not require input layers to have the same resolution, care must be taken not to mix layers with significantly different scales. In such cases, at each point the current version of openModeller will simply retrieve the nearest neighbor value for each environmental layer.
- *Accuracy:* Environmental rasters are generated in different ways, but the process always involves a sample of reality. For example, rainfall coverages are generated by interpolation of raw data captured by rainfall stations that are distributed across the target region. The accuracy of this kind of process depends on the accuracy of the sampled data, the number of samples, and the mathematical method (approximation or interpolation) [60].
- *Spatial Reference System:* A georeferenced vector or raster coverage is a two dimensional representation of a region. To achieve this representation, a mathematical shape for the earth must be assumed. Different spatial reference systems assume different approximated earth shapes and each assumption leads to different kinds of imprecision. There is also an intrinsic distortion related to the projection itself. This distortion can also compromise coverage accuracy in different ways for different regions [61].
- *Cell value:* Each cell is represented by a numerical value. If the cell value is not categorical and represents a numerical range, then that range will influence map accuracy. For example, if cell values need to be stored in one byte, the values can only represent a total of 256 possible values, which may not be accurate enough to express certain types of data.
- *Aggregation:* Sometimes a single environmental raster is actually the result of processing and aggregating data from several other rasters. Examples include mean annual rainfall or mean lower temperature during the coolest months of a particular year. Experience demonstrates that such combinations may have more environmental relevance and often produce better PD-Models than just using raw monthly values [23]. In these cases, the act of post-processing environmental data can introduce further errors in the final result.

5 Conclusions

Although it has been demonstrated by several studies that species' potential distribution modelling can be successfully used in many important areas [1], only a limited number of experts have been able to use the existing tools so far. Wide and effective use of species' potential distribution models will depend on the availability of new tools and interfaces that can improve users' experience. New programmatic interfaces should also facilitate the processing of massive modelling experiments so that species' distribution model repositories can be created and made available. The framework approach presented in this paper has been created to help address these issues.

Model results produced by two independent software packages usually include differences that are related to different sampling strategies, data conversions and limitations

in data input/output. Detecting such differences and measuring how much they can affect results is a difficult task. Generic frameworks like openModeller enable researchers to generate PD-Models and PD-Maps using different algorithms with the guarantee that any observed differences are exclusively due to algorithm logic and not to other internal computational decisions made by application developers of different software packages.

The plugin architecture described here provides a robust platform for researchers to write new algorithms. By separating data handling from the implementation of algorithms, it is hoped that openModeller will stimulate future algorithm development and increase productivity in the field. The flexibility of the architecture presented here has already enabled novel applications for potential distribution modelling [39, 40].

Finally, the open source nature of openModeller also provides for transparency of the underlying modelling algorithms. The authors feel this is essential to encourage scientific debate by allowing thorough peer review of algorithm logic and implementation.

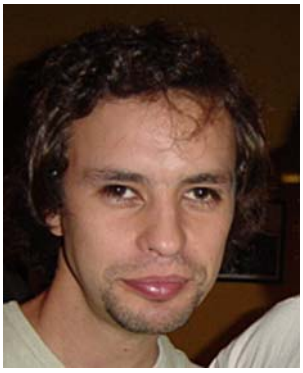
Acknowledgements The openModeller framework was originally developed by CRIA with support from FAPESP during the speciesLink project. After being released as a free and open source software package, other projects and institutions started to collaborate. The University of Kansas Natural History Museum & Biodiversity Research Center, the University of Reading and also individual contributors helped to significantly further develop the framework. In the beginning of 2005, openModeller received additional support from FAPESP as part of a new thematic project to be carried out by three Brazilian institutions: CRIA, INPE and Poli/USP.

References

1. Peterson AT (2006) Uses and requirements of ecological niche models and related distributional models. Biodiversity Informatics 3:59–72
2. Canhos VP, Souza S, De Giovanni R, Canhos DAL (2004) Global Biodiversity Informatics: setting the scene for a “new world” of ecological forecasting. Biodiversity Informatics 1:1
3. Yesson C, Brewer PW, Sutton T, Caithness N, Pahwa JS, Burgess M, Gray WA, White RJ, Jones AC, Bisby FA, Culham A (2007) How global is the Global Biodiversity Information Facility. PLoS ONE 2(11):e1124
4. Anderson RP, Lew D, Peterson AT (2003) Evaluating predictive models of species’ distributions: criteria for selecting optimal models. Ecol Model 162:211–232
5. Peterson AT, Papes M, Eaton M (2007) Transferability and model evaluation in ecological niche modeling: a comparison of GARP and Maxent. Ecography 30(4):550–560
6. Elith J, Graham CH, Anderson RP, Dudík M, Ferrier S, Guisan A, Hijmans RJ, Huettmann F, Leathwick JR, Lehmann AL, Li J, Lohman LG, Loiselle BA, Manion G, Moritz C, Nakamura M, Nakazawa Y, Overton JMcC, Peterson AT, Phillips SJ, Richardson K, Scachetti-Pereira R, Schapire RE, Soberón J, Williams S, Wisz MS, Zimmermann NE (2006) Novel methods improve prediction of species’ distributions from occurrence data. Ecography 29:129–151
7. Manel S, Dias JM, Buckton ST, Ormerod SJ (1999) Alternative methods for predicting species distribution: an illustration with Himalayan river birds. J Appl Ecol 36:734–747
8. Johnson CJ, Gillingham MP (2005) An evaluation of mapped species distribution models used for conservation planning. Environ Conserv 32:117–128
9. Phillips SJ, Anderson RP, Schapire RE (2006) Maximum entropy modeling of species geographic distributions. Ecol Model 190:231–259
10. Carpenter G, Gillison AN, Winter J (1993) DOMAIN: a flexible modeling procedure for mapping potential distributions of animals and plants. Biodivers Conserv 2:667–680
11. Scachetti-Pereira R (2002) Desktop GARP, http://www.nhm.ku.edu/desktopgarp, October, 24, 2007
12. Thuiller W (2003) BIOMOD—optimizing prediction of species distributions and projecting potential future shifts under global change. Glob Chang Biol 9:1353–1362
13. Garzón MB, Blazek R, Neteler M, de Dios RS, Ollero HS, Furlanello C (2006) Predicting habitat suitability with machine learning models: the potential area of Pinus *sylvestris* L. in the Iberian Peninsula. Ecol Model 97:383–393
14. MacArthur RH (1972) Geographical ecology: patterns in the distribution of species. Harper and Row, New York
15. Hutchinson GE (1957) Concluding remarks. Cold Spring Harb Symp Quant Biol 22:415–442

16. Soberón J, Peterson AT (2005) Interpretation of models of fundamental ecological niches and species' distributional areas. Biodiversity Informatics, https://journals.ku.edu/index.php/jbi/article/view/4
17. Anderson RP, Laverde M, Peterson AT (2002) Using niche-based GIS modeling to test geographic predictions of competitive exclusion and competitive release in South American pocket mice. Oikos 93:3–16
18. Ferrier S, Drielsma M, Manion G, Watson G (2002) Extended statistical approaches to modelling spatial pattern in biodiversity in northeast New South Wales. II. Biodivers Conserv 11(12):2309–2338
19. Kearney M, Porter W (2009) Mechanistic niche modelling: combining physiological and spatial data to predict species' ranges. Ecol Lett 12(4):334–350
20. Beaman R, Conn B (2003) Automated geoparsing and georeferencing of Malesian collection locality data. Telopea 10:43–52
21. Guralnick RP, Hill AW, Lane M (2007) Towards a collaborative, global infrastructure for biodiversity assessment. Ecol Lett 10(8):663–672
22. Longley PA, Goodchild MF, Maguire DJ, Rhind DW (2005) Geographic information systems and science, 2nd edn. John Wiley & Sons, Chichester, 517 p
23. Nix HA (1986) A biogeographic analysis of Australian elapid snakes. In: Longmore R (ed) Atlas of Australian elapid snakes. Australian Flora and Fauna Series 8:4–15
24. Stockwell DRB, Noble IR (1992) Induction of sets of rules from animal distribution data: a robust and informative method of analysis. Math Comput Simul 33:385–390
25. Mladenoff DJ, Sickley TA, Haight RG, Wydeven AP (1995) A regional landscape analysis and prediction of favorable greywolf habitat in the northern Great Lakes region. Conserv Biol 9:279–294
26. Bian L, West E (1997) GIS modeling of Elk calving habitat in a prairie environment with statistics. Photogramm Eng Remote Sensing 63:161–167
27. Frescino TS, Edwards TC, Moisen GG (2001) Modeling spatially explicit forest structural attributes using generalized additive models. J Veg Sci 12:15–26
28. Kelly NM, Fonseca M, Whitfield P (2001) Predictive mapping for management and conservation of seagrass beds in North Carolina. Aquatic Conservation: Marine and Freshwater Ecosystems 11(6):437–451
29. Guisan A, Edwards TC, Hastie T (2002) Generalized linear and generalized additive models in studies of species distributions: setting the scene. Ecol Model 157:89–100
30. Felicísimo AM, Francés E, Fernández JM, González-Díez A, Varas J (2002) Modeling the potential distribution of forests with a GIS. Photogramm Eng Remote Sensing 68:455–462
31. Fonseca MS, Whitfield PE, Kelly NM, Bell SS (2002) Statistical modeling of seagrass landscape pattern and associated ecological attributes in relation to hydrodynamic gradients. Ecol Appl 12(1):218–237
32. Livingston SA, Todd CS, Krohn WB, Owen RB (1990) Habitat models for nesting bald eagles in Maine. J Wildl Manage 54(4):644–653
33. Fielding AH, Haworth PF (1995) Testing the generality of bird-habitat models. Conserv Biol 9(6):1466–1481
34. Pearson RG, Dawson TP, Berry PM, Harrison PA (2002) SPECIES: a spatial evaluation of climate impact on the envelope of species. Ecol Model 154(3):289–300
35. Guo Q, Kelly M, Graham CH (2005) Support vector machines for predicting distribution of sudden oak death in California. Ecol Model 182(1):75–90
36. Leathwick JR, Elith J, Francis MP, Hastie T, Taylor P (2006) Variation in demersal fish species richness in the oceans surrounding New Zealand: an analysis using boosted regression trees. Mar Ecol Prog Ser 321:267–281
37. Kaschner K, Ready JS, Agbayani E, Rius J, Kesner-Reyes K, Eastwood PD, South AB, Kullander SO, Rees T, Close CH, Watson R, Pauly D, Froese R (2007) AquaMaps: predicted range maps for aquatic species, http://www.aquamaps.org, December, 2007
38. Quinlan JR (1986) Induction of decision trees. Mach Learn 1:81–106
39. Yesson C, Culham A (2006) Phyloclimatic modelling: combining phylogenetics and bioclimatic modelling. Syst Biol 55(5):788–802
40. Yesson C, Culham A (2006) A phyloclimatic study of cyclamen. BMC Evol Biol 6:72
41. Vapnik V (1995) The nature of statistical learning theory. Springer-Verlag, New York
42. Ripley BD (1996) Pattern recognition and neural networks. Cambridge University Press, Cambridge
43. Peterson AT, Vieglais DA, Navarro-Sigüenza AG, Silva M (2003) A global distributed biodiversity information network: building the world museum. Bull Br Ornithol Club 123A:186–196
44. Soberón J, Peterson AT (2004) Biodiversity informatics: managing and applying primary biodiversity data. Philos Trans R Soc Lond, B 359:689–698
45. Stein BR, Wieczorek JR (2004) Mammals of the world: MaNIS as an example of data integration in a distributed network environment. Biodiversity Informatics
46. Robertson MP, Caithness N, Villet MH (2001) A PCA-based modelling technique for predicting environmental suitability for organisms from presence records. Divers Distrib 7:15–27

47. Durigan G, Baitello JB, Franco GADC, Siqueira MF (2004) Plantas do cerrado paulista: imagens de uma paisagem ameaçada. Páginas & Letras Editora e Gráfica, São Paulo, 475 p
48. Ratter JA, Bridgewater S, Ribeiro JF, Dias TAB, Silva MR (2000) Distribuição das espécies lenhosas da fitofisionomia cerrado sentido restrito nos estados compreendidos no bioma cerrado. Bol Herb Ezechias Paulo Heringer 5:5–43
49. Durigan G, Siqueira MF, Franco GADC, Bridgewater S, Ratter JA (2003) The vegetation of priority areas for cerrado conservation in São Paulo state, Brazil. Edinb J Bot 60:217–241
50. Ratter JA, Bridgewater S, Atkinson R, Ribeiro JF (1996) Analysis of the floristic composition of the Brazilian cerrado vegetation II: comparison of the woody vegetation of 98 areas. Edinb J Bot 53:153–180
51. Hijmans RJ, Cameron SE, Parra JL, Jones PG, Jarvis A (2005) Very high resolution interpolated climate surfaces for global land areas. Int J Climatol 25:1965–1978
52. Chapman AD (2005) Principles of data quality, version 1.0 report for the Global Biodiversity Information Facility. Copenhagen, Denmark, pp1–58, http://www2.gbif.org/DataQuality.pdf
53. Guralnick RP, Wieczorek JR, Beaman R, Hijmans RJ, the BioGeomancer Working Group (2006) Biogeomancer: automated georeferencing to map the world's biodiversity data. PLoS Biol 4(11):e381
54. Wheeler QD, Raven PH, Wilson EO (2004) Taxonomy: impediment or expedient? Science 303:285
55. Jain AK, Murty MN, Flynn PJ (1999) Data clustering: a review. ACM Comput Surv 31(3):264–323
56. Michie D, Spiegelhalter DJ, Taylor CC (1994) Machine learning, neural and statistical classification. Ellis Horwood, New York
57. Stockwell DRB, Peterson AT (2002) Predicting species occurrences: issues of accuracy and scale, controlling bias in biodiversity data. Island, Washington, pp 537–546
58. Johnson CM, Johnson LB, Richards C, Beasley V (2002) Predicting species occurrences: issues of accuracy and scale, predicting the occurrence of amphibians: an assessment of multiple-scale models. Island, Washington, pp 157–170
59. Chapman AD, Muñoz MES, Koch I (2005) Environmental information: placing biodiversity phenomena in an ecological and environmental context. Biodiversity Informatics, https://journals.ku.edu/index.php/jbi/article/view/5
60. Hartkamp AD, De Beurs K, Stein A, White JW (1999) Interpolation techniques for climate variables, NRG-GIS Series 99-01, CIMMYT, Mexico D.F.
61. Bannerman BS (1999) Positional accuracy, error and uncertainty in spatial information. Geoinnovations, Howard Springs, NT, Australia. http://www.geoinnovations.com.au/posacc/default.htm. Accessed 12 Jun 2009



Mauro Enrique de Souza Muñoz graduated in Computer Engineering at the State University of Campinas (UNICAMP), Brazil, in 1996. He completed his M.Sc in Artificial Intelligence at the Federal University of Rio Grande do Sul, UFRGS, in 1999. After this he worked for 2 years in telecommunication, smart card and telephone voice portals. Since the beginning of 2002 until the end of 2004 he worked at the Reference Center on environmental Information (CRIA), Brazil, as a software developer in biodiversity informatics.

Renato De Giovanni has a BsC in Mechanical Engineering from the Aeronautics Institute of Technology (ITA, 1993, Brazil). After this he accumulated 14 years of experience in software development, working mainly at local consulting companies. Since 2002 he participated in several projects at the Reference Center on Environmental Information (CRIA), Brazil, working with distributed searches on biological collections' databases, data quality tools, systems for species interaction data, and potential distribution modelling. From 2004 to 2007 he took part in the Subcommittee for Data Access and Database Interoperability of the Global Biodiversity Information Facility (GBIF). He is also an active member of the Biodiversity Information Standards (TDWG) organization, being involved in the DarwinCore and TAPIR task groups.

Marinez Ferreira de Siqueira Biologist, graduated in 1989 at the State University of Campinas (UNICAMP), Brazil. She holds a Ph.D. in Science of Environmental Engineering in 2005 from Centre of Water Resources and Applied Ecology (CRHEA) - Engineering School of São Carlos - University of São Paulo (USP) and a Master Degree in Ecology in 1994 from the State University of Campinas (UNICAMP), Campinas, Brazil. She works with species distribution modeling based on ecological niche since 2002. Her main research line is in ecology of tree species from Cerrado and Atlantic Rain Forest.

Tim Sutton completed his MA in GIS Analysis and Decision Making at the University of Stellenbosh, South Africa in 2004. He has accumulated over 15 years of experience in applied conservation management and developing GIS based biodiversity information systems. He is a keen advocate of open source software. He is involved in a number of open source projects, including being a co-developer and Project Steering Committee member of the Quantum GIS (QGIS) open source GIS application, and the principle developer of the openModeller Desktop application.

Peter Brewer completed his Ph.D. in Bioinformatics at the University of Reading, UK, in 2003. Between 2003 and 2007 he held Post doctoral positions at the University of Reading, first on the BiodiversityWorld e-Science project and then as the Systems Manager of the Species 2000 and ITIS Catalogue of Life. In July 2007 he moved to Cornell University, USA where he now leads IT development at the Cornell Tree-Ring Laboratory and the global tree ring data standards initiative (TRiDaS) for the dendrochronology community.

Ricardo Scachetti Pereira graduated in Computer Engineering at the State University of Campinas (UNICAMP), Brazil, in 1997. He holds a M.Sc in Theory of Computing in 1999 and a specialization in Business Management in 2001, all from the same University. In 1999, he started to work in several Biodiversity Informatics projects in affiliation with Universities and non-governamental organizations around the world, including the Reference Center for Environmental Information, Brazil, and the University of Kansas, U.S.A.. Currently he is based in Brasilia, Brazil, working as a consultant.

Dora Ann Lange Canhos is the Project Director of the Reference Center on Environmental Information (CRIA), Brazil. She has been working with databases and online information systems since 1985. She has been involved with biodiversity information networks since 1992, as a member of BIN21 (Biodiversity Information Network - Agenda 21) and responsible for its web site, and as technical coordinator of the project BINbr (Biodiversity Information Network - Brazil) for the Ministry of the Environment from May 1997 to April 2001. She is a member of the following international working groups: Clearing-House Mechanism Informal Advisory Committee of the Convention on Biological Diversity, since 2001; Codata Task Group on Preservation and Archiving of Scientific and Technical Data in Developing Countries, since 2003; Conservation Commons Interim Steering Committee, since 2004.

Vanderlei Canhos has a BsC degree (1971) and MsC degree (1975) in Food Science from the State University of Campinas (UNICAMP), and a PhD degree from the Oregon State University (1980). He acted as assistant professor at the Food Science Department at UNICAMP from 1980 to 2001. For the last 20 years Dr. Canhos has been involved with the development of strategies and policies for the consolidation of biological collections and biodiversity databases in Brazil. He acted as scientific director of the Tropical Culture Collection (CCT) and Tropical Data Base (BDT) from 1985 to 2001. He is currently the President Director of the Brazilian Reference Center on Environmental Information and the team leader of a number of projects including openModeller. He has acted as a member of the Task Force on Biological Resource Centers at the Organization for Economic Development and Cooperation - OECD (2001 - 2006); a member of the Steering Committee of the Program Biota-FAPESP The Virtual Institute of Biodiversity (1999 -2006); a member of the Subcommittee on Capacity Building and Outreach (OCB) of the Global Biodiversity Information Facility, GBIF (2001 - 2003); a member of the Project Team of the Inter-American Biodiversity Information Network - IABIN (1999 - 2003); the former President of the Executive Board of the World Federation for Culture Collections - WFCC (1996 - 2000); a member of the Informal Advisory Committee of the Clearing-House Mechanism to the Convention on Biological Diversity (1998 - 2001). He is currently a member of the Subcommittee for Digitisation of Natural History Collection Data (DIGIT) of the Global Biodiversity Information Facility, GBIF (since 2003); a member of the Species 2000 Board of Directors (since March 2003); and a member of the Board of Directors of the Expert Taxonomy Information - ETI, University of Amsterdam (since January 2000).