uncert|web
uncertainty-enabled model web

SEVENTH FRAMEWORK PROGRAMME

Proposal / Contract no.: 248488
Project start: February 1, 2010
Project end: January 31, 2013

UncertWeb

The *Uncert*ainty Enabled Model *Web*

SEVENTH FRAMEWORK PROGRAMME

THEME FP7-ICT-2009-4

ICT for Environmental Services and Climate Change Adaptation

Deliverable 4.3

A habitat modelling UncertWeb service

Title of Deliverable	A habitat modelling UncertWeb service
Deliverable reference number	D4.3
Related WP and Tasks	WP4 (Task 4.3)
Type of Document	Public Report
Authors	Jon Skøien, Michael Schulz, Gregoire Dubois
Date	12/10/2012
Version	1.1

Project coordinator

Dr. Dan Cornford

Aston University, United Kingdom

E-mail: d.cornford@aston.ac.uk

http://www.uncertweb.org

Revision History

Version	Date	Changes	Authors
1.0	20/09/2012	First draft before revisions	Jon Olav Skøien, Michael Schulz, Gregoire Dubois
	02/10/2012	Revision	Lorenzo Bigagli
1.1	12/10/12	Final Version	Jon Olav Skøien, Michael Schulz

D4.3 A habitat modelling UncertWeb service

An UncertWeb service derived from the APAAT to compute habitats and associated uncertainties when chained with remote sensing data and the associated uncertainties.

Related task(s):

Task 4.3 Development of a habitat modelling Web service

This task will focus on setting up a service for modelling habitats using the HRI. This will be developed as a model Web service that will

• allow the exchange of new habitat maps and their uncertainties;

• incorporate a number of basic climatic parameters and their uncertainties that affect the environmental variables used by the HRI (e.g. % of water bodies, aridity index). The service will thus, where appropriate, allow the fusion of land cover data collected from the land cover validation tool with other environmental variables to compute the HRI and its associated uncertainties. It will be developed as an UncertWeb prototype service that can generate new habitat maps and simulate habitat changes when coupled with the simple prototype climate change model data service developed in WP2.

Active partners: JRC, AST

Legal Notices

Executive Summary

We have developed a Web Processing Service (WPS) for assessing habitat similarity which we refer to as eHabitat. The main idea behind eHabitat is to provide a service allowing end-users to identify areas having similar ecological properties compared to a reference location. This approach is typically used for ecological niche modelling in which a spatial prediction model for a given species is computed from a set of environmental parameters [see e.g. *Clark et al.*, 1993; *Knick and Dyer*, 1997; *Rotenberry et al.*, 2006]. To compute the similarity to a reference location for each pixel of the domain under study, a popular approach is based on the Mahalanobis distance [*Mahalanobis*, 1936]. The method is mathematically simple and fairly easy to understand, performs relatively well compared with most other models [*Tsoar et al.*, 2007] and computationally fast compared with more complex methods such as Maxent [*Phillips et al.*, 2006]. These and other methods are generally referred to as Species Distribution Models (SDMs)

Common to all methods is that they, based on a set of training points (typically species observations, and in the case of Maxent, also absence observations), compute an optimal range of environmental conditions, and a relative distance to this optimum. The methodology and the crispness of the results (gradient from suitable to unsuitable habitat) differ, but they all share a common principle in the use of rasters of environmental indicators and reference geometry (points, polygons) defining the samples from the environmental indicator.

Whereas the methods are simple, the choice of an optimal set of environmental indicators is not. This will for the desktop application mean that the user needs to download a large number of possible rasters, extract the right extent and resolution, before applying the SDM. By setting up the methods as a WPS with standardized OGC (Open Geospatial Consortium) input and output, eHabitat can interact with other web services. When the environmental indicators are available as Web Coverage Services, these can easily be ingested by eHabitat, using the right extent and resolution.

Traditional Species Distribution Modelling has had a limited focus on uncertainty, with the exception of bootstrapping in the Maxent method, which means resampling of point observations. As we are mostly dealing with polygons instead of points in this study, we have rather focused on the uncertainty of the environmental indicators. Ideally, some of this uncertainty should have been available from the data producers. But unfortunately, there are hardly any such data sets with a quantified uncertainty available. We have therefore developed our service for propagating uncertainty also in cases where it is not available. A simple method is used for approximating the uncertainty of each input data set, which is then propagated through the service. One could say that this is closer to scenario modeling than true uncertainty propagation, however, it will at least give some indication on the effect of uncertainty on the final similarity. Additionally, it means that we already have the necessary infrastructure in place for the time when real uncertainty is available.

We have implemented the computational backend of eHabitat in R [*R Development Core Team*, 2012], as an add-on package called eHab. This assures easy installment of the system also in different environments. PyWPS [*Cepicky and Becchi*, 2007] was chosen as a lightweight WPS implementation in the Python programming language. This also enables easy use of additional packages like e.g. GDAL (Geospatial Data Abstraction Library) for necessary data pre- and postprocessing steps in the WPS.

Since the OGC WPS specification is rather abstract, the best way to demonstrate the usage of the eHabitat process was to build a web application that enables users to simply select a reference location (in our example either a protected area or a species range), set an area of interest and to choose from pre-prepared indicator layers (bioclimatic variables) to collect the necessary inputs to execute the process. When the results are computed they can be directly visualized in this web application, that is reachable at this URL:

http://ehabitat-wps.jrc.ec.europa.eu/

The simulated uncertainty can also be visualized directly in this client. The UncertML encoded output can also be propagated and visualized using the visualization tools developed in WP3. Therefore a Python wrapper library for the encoding of UncertML using the NetCDF-UW convention was developed.

For a sensible interpretation of the similarity results the raw input values of the indicator layers and their variability inside the reference location are necessary. During a process execution these values are gathered and returned to the user. They can be seen as sample phenomena directly linked with the reference location and thus be encoded using the UncertML Observation and Measurements (O&M) profile and visualized as histograms also in the visualization client.

Contents

1 Introduction

The use of distributed computing technology is revolutionizing the way we deal with information, and international initiatives, such as the Group on Earth Observations (GEO), are encouraging different communities to make their systems and applications interoperable. Biodiversity is one of the social benefit areas that is likely to benefit most from this initiative because of the nature of the datasets required for monitoring and strategy evaluation; they are huge in their spatio-temporal scope and dimensionality, while at the same time they are often documented and managed in a very fragmented and inconsistent manner.

When we consider ecological modelling, better results have traditionally been achieved either by improving existing models or by developing new ones. Chaining interoperable model components is now a third alternative that is particularly interesting because such a chain can potentially answer more questions than the individual models alone, allowing users to address complex questions in a variety of different contexts. It is still a challenge to set up a computing infrastructure where models can be easily plugged and played [*Service*, 2011]. However, the proposed "Model Web" [*Geller and Turner*, 2007] envisages exactly such an environment, and encourages the practical development of a distributed, multidisciplinary network of independent, interoperating models (and datastores) communicating with each other using Web services. Beyond the simple sharing of information, the Model Web conceives increasing access to models and their outputs, and aims to facilitate greater model-model interaction, resulting in webs of interacting models, databases, and websites [*Nativi et al.*, 2012].

A first effort in this direction has been described by Best et al. [2007] with OBIS-SEAMAP[1], the Ocean Biogeographic Information System - Spatial Ecological Analysis of Megavertebrate Populations, a spatially referenced online database, aggregating marine mammal, seabird and sea turtle observation data from across the globe. Another milestone was the setting up by Nativi et al. [2009] of an ecological niche modelling framework built around the popular OpenModeller[2] [*Muñoz et al.*, 2011]. This modelling framework successfully employed a Service Oriented Architecture (SOA) even though at the time OpenModeller was still a stand-alone application, by making the modelling kernel accessible through external interfaces like SOAP and SWIG (Simplified Wrapper and Interface Generator).

An example of an interoperable biodiversity information system where models are chained, is the Digital Observatory for Protected Areas (DOPA)[3] that is currently being developed at the Joint Research Centre of the European Commission in collaboration with other international organizations, including the Global Biodiversity Information Facility (GBIF), the UNEP-World Conservation Monitoring Centre (WCMC), Birdlife International and the Royal Society for the Protection of Birds (RSPB). DOPA is conceived as a set of distributed databases combined with open, interoperable Web services to provide end-users, from park managers to scientists and decision-makers, with the means to assess the state of protected areas (PAs) at the global scale [*Dubois et al.*, 2010]. On top of the need to easily exchange

[1] http://seamap.env.duke.edu/

[2] http://openmodeller.sourceforge.net/

[3] http://dopa.jrc.ec.europa.eu/

information with a number of reference spatial data infrastructures (SDIs) for computing the indicators involved in the assessments, DOPA must also rely on automated services for monitoring purposes. Ultimately, when used in conjunction with other environmental services which can supply information on phenomena such as predicted climate change, DOPA should be flexible enough to allow ecological forecasting and consideration of alternative future scenarios.

This last objective has been partly achieved through the development of eHabitat[4], DOPA's core modelling service that is made available to the community by means of a Web processing service (WPS). In this report we give some more details on the development and properties of eHabitat and discuss its use in an environment of interoperable data and model services. Being a simple habitat modelling service for the ecologists, it can be chained with other services offering ancillary information on predicted flooding, population and agricultural pressures or land prices. The flexibility is considerably higher than for the model described by Nativi et al. [2009].

The largest potential benefit from the Model Web is likely to be the practical and easy re-use of basic modelling components for different purposes. We believe that the granularity of the models expected to interact with each other is a critical factor in any operational version of the Model Web. A higher granularity is likely to generate more reusable elementary services, greater control for the users composing those services and thus, ultimately, more complex and useful modelling chains. The reusability of results from eHabitat is assured by wrapping the statistical habitat modelling with the standardised OGC (Open Geospatial Consortium) WPS interface.A further step towards interoperability is the ability to use OGC standards like WCS (Web Coverage Service) with GeoTIFF and NetCDF encoding for input and output data and e.g WMS (Web Mapping Service) for easy integration of outputs in web mapping applications.

2 Ecological modeling using the Mahalanobis distance and similar models

The main idea behind eHabitat is to provide a service allowing end-users to find areas that have similar ecological properties to a reference location. This approach is typically used for ecological niche modelling in which a spatial prediction model for a given species is computed from a set of environmental parameters [see e.g. *Clark et al.*, 1993; *Knick and Dyer*, 1997; *Rotenberry et al.*, 2006]. In this context, Geographic Information Systems (GIS) have proven to be very useful tools for conservation because of the ease of handling various thematic layers and using multi-criteria decision trees for extracting information. To compute similarity to a reference location for each pixel of the domain under study, a popular approach is based on the Mahalanobis distance [*Mahalanobis*, 1936]. The method is mathematically simple and fairly easy to understand, relatively good compared with most other models [*Tsoar et al.*, 2007] and computationally fast compared with more complex methods such as Maxent [*Phillips et al.*, 2006]. We have therefore chosen mostly to focus on this method below, although also other methods have been implemented through the WPS.

2.1 The Mahalanobis distance

[4] http://ehabitat.jrc.ec.europa.eu/

For a set of environmental variables available for the region of interest, there are different ways of modelling the environmental similarity between this region and a reference geometry, typically a set of points referring to presence observations or the points within a protected area. The Mahalanobis distance D_i is here used as a measure of the similarity of a set of environmental variables between a pixel i and the averages of these environmental variables for the reference geometry, and is defined as:

$$D_i^2 = [\boldsymbol{x_i} - \boldsymbol{\mu}]^T [\boldsymbol{C}]^{-1} [\boldsymbol{x_i} - \boldsymbol{\mu}] \qquad \textbf{[1]}$$

where $\boldsymbol{x_i}$ is the vector of the values of the environmental variables for pixel i, $\boldsymbol{\mu}$ is the mean of the environmental variables for the reference geometry, and $[\boldsymbol{C}]$ is the covariance matrix of the environmental variables for the reference geometry. The covariance matrix for n variables is given by

$$[\boldsymbol{C}] = \begin{bmatrix} \mathrm{cov}(\boldsymbol{x_1}, \boldsymbol{x_1}) & \mathrm{cov}(\boldsymbol{x_1}, \boldsymbol{x_2}) & \cdots & \cdots & \mathrm{cov}(\boldsymbol{x_1}, \boldsymbol{x_n}) \\ \mathrm{cov}(\boldsymbol{x_2}, \boldsymbol{x_1}) & \mathrm{cov}(\boldsymbol{x_2}, \boldsymbol{x_2}) & \cdots & \cdots & \mathrm{cov}(\boldsymbol{x_2}, \boldsymbol{x_n}) \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & & \ddots & \vdots \\ \mathrm{cov}(\boldsymbol{x_n}, \boldsymbol{x_1}) & \mathrm{cov}(\boldsymbol{x_n}, \boldsymbol{x_2}) & & & \mathrm{cov}(\boldsymbol{x_n}, \boldsymbol{x_n}) \end{bmatrix} \qquad \textbf{[2]}$$

and the covariance between any two variables, $\boldsymbol{x}_k$ and $\boldsymbol{x}_l$, with means μ_k and μ_l and number of points in the reference geometry J is given by

$$\mathrm{cov}(\boldsymbol{x_k}, \boldsymbol{x_l}) = \sum_{j=1}^{J} \left(\frac{(x_{kj} - \mu_k)(x_{lj} - \mu_l)}{J} \right) \qquad \textbf{[3]}$$

The use of the inverse of the covariance matrix makes the Mahalanobis distance scale-independent, i.e., it is not affected by the different scales of the variables. Highly correlated variables will have less effect on D_i than uncorrelated variables.

In ecology, it has traditionally been assumed that D_i^2 is distributed according to a χ^2 distribution with n-1 degrees of freedom (df), as long as the input variables have a normal distribution. This way D_i^2 can be converted into p-values ranging from 0-1. The p-values (or probability values) range from 0.0 representing no similarity to 1.0 for areas which are identical to the mean of the reference area. If the predictor variables are not normally distributed, the conversion is still useful as it rescales the unbounded D_i^2 values to a [0-1] range.

This rescaling is very useful, however, we have discovered the need to make a correction to the degrees of freedom that is being used in the transformation. It seems that the use of df = n-1 was an error by Clark [*Clark et al.*, 1993], and that D_i^2 rather has a χ^2 distribution with n degrees of freedom. The author independently discovered this himself recently [*Clark*, 2012]. The use of df = n is then according to other uses of the Mahalanobis distance.

The p-value can be seen as the probability that the pixel i has a similar set of environmental variables as the ones found for the reference area, or of the probability that a pixel in the future has a similar set of environmental variables. Figure 1 illustrates the use of Mahalanobis distances for identifying areas that are presenting ecological characteristics similar to those found in a protected area, the Kafue National Park, in Zambia. A set of ecological variables are used as input data and a map of probabilities to find similar values taken by these variables can be generated.

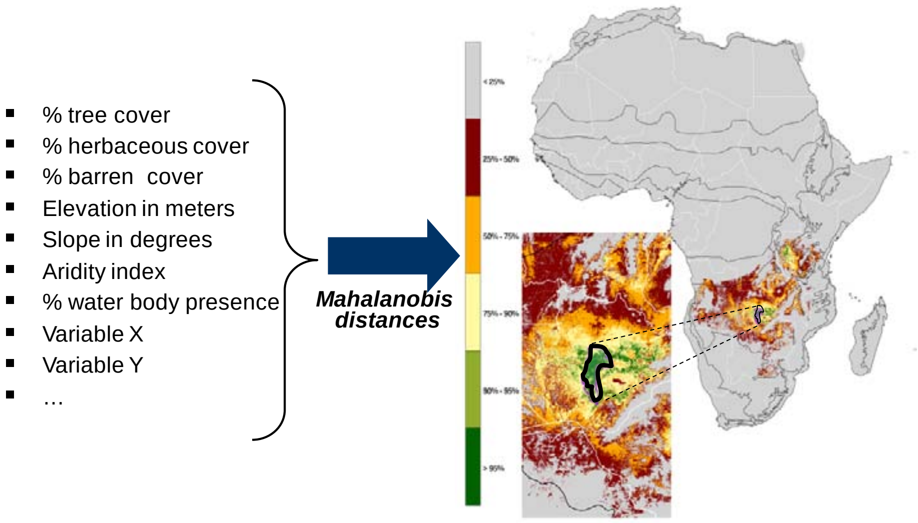

Figure 1. Map of habitats that are similar in the Zambezian ecoregion to the protected area of Kafue in Zambia, and associated scale of similarity.

The traditional use of the Mahalanobis distance is for identifying habitats for a given species [*Clark et al.*, 1993; *Rotenberry et al.*, 2006]. Instead of computing the mean and the variances, $\boldsymbol{\mu}$ and [$\boldsymbol{C}$], over a whole protected areas, one is computing these statistics for the ecological variables measured at all the locations where a species has been observed. The Mahalanobis distance can so be computed to produce a map of probabilities of finding habitats that are specific to the given species, a map that is thus a probabilistic representation of the species' ecological niche.

For our purpose, where we use the Mahalanobis distance for predicting the similarities between forecast climatic conditions for PAs, we can also compute a habitat replaceability index (HRI) after computing the *p*-value for all pixels. This is the size of the area with a p-value above a certain limit, which typically can be set to 0.5, divided by the area of the PA. The HRI is then an indication of the size of the area with environmental variables relatively similar to the PA itself.

2.2 Other methods

There are several other models that are also used for species distribution modelling. The most commonly used during the last years is the Maxent (Maximum Entropy) method [*Phillips et al.*, 2006], which is based on a machine learning technique. Maxent estimates the likelihood of a species being present by finding the distribution of maximum entropy (i.e., that is closest to uniform) subject to the constraint that the expected value of each environmental variable under this estimated distribution matches its empirical average [*Hijmans and Graham*, 2006; *Phillips et al.*, 2006]. An option to use this is also included in the WPS (see below), together with methods such as Bioclim [*Busby*, 1991; *Nix*, 1986]and Domain [*Carpenter et al.*, 1993].

Common to all models is that they, based on a set of training points (typically species observations, and in the case of Maxent, also absence observations), compute an optimal range of environmental conditions, and a relative distance to this optimum. The methodology and the crispness of the results (gradient from suitable to unsuitable habitat) differ, but they all share a common principle in the use of rasters of environmental indicators and reference geometry (points, polygons) defining the samples from the environmental indicators. We will in the remaining mainly focus on the Mahalanobis distance due to speed of the method for repeated computations, and because of the easy traceable equations.

2.3 Uncertainty of Mahalanobis distance

Despite the common usage of these methods both for deriving species range maps and for ecological forecasting, little attention has been paid to the mathematical characteristics of the methods. Of particular interest within the UncertWeb project is of course the uncertainty propagation of the methods.

There is a range of uncertainties that can be propagated in habitat modeling. In the traditional approach where the Mahalanobis Distance or similar methods are used for modeling the ranges of species, [*Rocchini et al.*, 2011] list a range of uncertainties that in theory can affect the results. Their focus was first of all on the species presence (and to some degree absence) observations. Sources of uncertainty for presence observations range from the typical measurement errors to more specific biodiversity errors such as bias towards biodiversity hotspots and more accessible areas, records of species wandering from their natural habitat and records from abandoned regions. Absence records can be affected by lack of sampling strategies, and failure of the observer to actually detect present species. These errors are challenging to model, although they can be approached my randomization of the point locations.

However, our main focus is on protected areas, where the locational uncertainty is either insignificant (park boundaries are where they should be) or impossible to quantify as a function of political disputes or erroneous reporting. We will therefore focus on the uncertainty of the environmental indicators in the following. A follow up could be to look at locational errors for species observations, similar to what was done by Graham et al. [*Graham et al.*, 2008].

2.3.1 Analytical uncertainty

The Mahalanobis distance for a pixel *i*, given in Equation 1 is a quadratic form, which can be generalized as (note that we for simplicity have not used the subscript *i* for the pixel to be predicted in the equations below):

$$D^2 = \mathbf{X}^{\mathbf{T}} \Lambda\, \mathbf{X} \qquad \textbf{(4)}$$

where **X** is (**x-m**) and Λ is $\mathbf{C}^{-1}$. The expectation of this when **X** is a random vector with mean $\boldsymbol{\mu}$ and covariance matrix Σ is [*Searle*, 1971]:

$$E(D^2) = tr(\Lambda\Sigma) + \boldsymbol{\mu}^T \Lambda \boldsymbol{\mu} \qquad \textbf{(5)}$$

where tr denotes the trace of the matrix (i.e. the sum along the diagonal). The variance of D is given by:

$$Var(D^2) = 2tr(\Lambda\Sigma\Lambda\Sigma) + 4\boldsymbol{\mu}^T \Lambda\Sigma\Lambda\boldsymbol{\mu} \qquad \textbf{(6)}$$

In the subsequent analysis we assume that **x** is random, not **m**. If we can assume stationarity, then the covariance matrix Σ of **x** at location *s* is composed of:

$$\Sigma_{ij} = Cov(x_{is}, x_{js}) = \sigma_{is}\sigma_{js} - \gamma_{ij}(0) \qquad \textbf{(7)}$$

where σ_{is} and σ_{js} are the local variances of the indicators and $\gamma_{ij}(0)$ is the cross-variogram between the two indicators at zero distance. The first term of Equation (5) always increases D^2 for increasing variances in Σ. This means that the expectation of D^2 increases hence and

the expectation of similarity decreases if we assume that the values of the indicators are uncertain.
The above gets more complicated when also **m**, the means of the variables at the reference locations, are uncertain. In this case we are interested in the covariance matrix of **x-m**. The variance of indicator *i* is found as:

$$Var(x_i - m_i) = Var(x_i) + Var(m_i) - 2Cov(x_i m_i) \quad \textbf{(8)}$$

whereas the covariance between different indicators can be found as

$$Cov(x_i - m_i, x_j - m_j) \quad \textbf{(9)}$$
$$= Cov(x_i, x_j) - Cov(x_i, m_j) - Cov(x_j, m_i) + Cov(m_i, m_j)$$

All variances and covariances involving the means can be found through integration of the cross-covariance between the individual points and all reference points, such as:

$$Var(m_i) = \frac{1}{J^2}[J + 2\sum_{k=1}^{J-1}\sum_{l=k+1}^{J} Cov_i(h_{kl})] \quad \textbf{(10)}$$

where $Cov_i(h_{kl})$ is the covariance function of variable *i* as a function of the separation distance h_{kl} between two different reference points *k* and *l*. The cross-covariance between one variable at prediction location *l* and the mean of another variable *j* can be found as:

$$Cov(x_i, m_j) = \overline{m_i}\overline{x_j} + \frac{1}{J}\sum_{k=1}^{J} Cov_{ij}(h_{kl}) \quad \textbf{(11)}$$

where Cov_{ij} refers to the cross-covariance between two variables as function of separation distance h_{kl}. We do not give the remaining parts of the covariance matrices but these can be calculated in a similar way.

2.3.2 Simulation of uncertainties

The equations above are useful for interpretations, but they are not really feasible for computation of the Mahalanobis distance for a large number of points, typically up to a million points when estimating the similarity of a PA. Instead in the following we show how simulations can be used for estimating the uncertainty. This is implemented as an optional choice in the eHabitat WPS, the deterministic approach still being the default. The approach we have chosen is to generate simulations of the input data, from a predefined distribution of the uncertainty. The simulations must reflect the spatial correlation of the observed data, particularly if we are treating the reference variables as uncertain. Otherwise the variances in the covariance matrix inverted in Equation 1 might be too large, which would decrease the Mahalanobis distance.
Ideally, the simulations should be based on known uncertainty. However, many indicators come without any information about the associated uncertainty. The second issue is that indicators are usually cross-correlated, and hence their errors are also likely to be cross-correlated. This information is rarely available. We have therefore chosen to guesstimate the uncertainty in the following way:

- For each pixel and indicator, the standard deviation is estimated as 10% of the smallest value of the indicator above zero plus 5% of the indicator value at the pixel itself
- The correlations between the errors of the different indicators are assumed to be equal to the correlations between the different indicators
- A cross-correlogram of the uncertainties is based on the correlations and:
    - o zero nugget effect (but easy to adapt for particular applications)
    - o range found as the mean of the ranges of variograms automatically fitted to sample variograms of the individual indicators. Individually fitted ranges with higher value than the diagonal of the bounding box of the data set are set equal to the diagonal

From the cross-correlograms, we created sets of unconditional realizations of normalized errors of the indicators through sequential Gaussian simulation. The set of realizations is multiplied with the standard deviations given above, before adding the result to the initial data set. Realizations outside the ranges of the individual indicators are set equal to their minimum and maximum, respectively, to avoid impossible data, such as negative biotemperature or precipitation.

With a given number of simulations, we compute a set of possible maps of the similarities. From these we can either look at single realizations or different types of summary statistics. We present some possibilities below in the results section. All computations are done in R [*R Development Core Team*, 2012], the variogram fitting was done with automap [*Hiemstra et al.*, 2008], whereas the simulations were done with the package gstat [*Pebesma*, 2004].

3 Implementation of computational backend – R package eHab

The implementation of the computational backend of eHabitat is completely done within the statistical environment/language R [*R Development Core Team*, 2012] which is an easily extendible interpreted language where new functionality can be implemented as functions of existing functions. We have chosen to implement the functionality as a package, which makes it easy to create and add documentation and run unit tests during the implementation phase, and to load the correct versions of all functions when run in the operational stage. We will most likely not upload the package to The Comprehensive R Archive Network (CRAN: http://cran.r-project.org), as the addition to existing libraries is limited, and because the functionality is very much oriented to our needs and is probably less useful to other groups. The package will still be available through the eHabitat web client when the online documentation of the service has reached a more mature stage.

The eHab package is relying on a series of other packages in R, to take advantage of existing developments. First of all these are GIS-related packages, such as sp[*Bivand et al.*, 2008] for spatial classes, rgdal[*Keitt et al.*, 2011] for simple GIS capabilities, rgeos[*Bivand and Rundel*, 2012] for some more advanced GIS capabilities and raster [*Hijmans and van Etten*, 2012] for rastered information, particularly suitable for large input data sets. The package can also use automap [*Hiemstra et al.*, 2008] as a simple interface to gstat [*Pebesma*, 2004] for filling in missing values, if necessary, and dismo[*Hijmans et al.*, 2012], which is an existing package for distribution modelling.

Although the dismo package does offer a lot of useful functionality for distribution modelling, it was decided not to rely only on this package. There were several reasons for that.

1) Although not part of the web service, it was convenient for the precomputation of a set of reference results to make the input of the reference data more flexible than for dismo. Hence, for the Mahalanobis distance it is possible to pass this data as a spatial data set, but also just as the reference data, or as the mean and the covariance of the reference data.

2) It was necessary for us to control the behaviour of the method when the indicator layers were categorical (such as land cover classes), or if the indicator layers are constant within the reference geometry (species observations or protected area). The solution to this issue was to only treat as similar the areas that have the same categorical variable as the reference geometry, or to treat the variable as a categorical variable if it was constant within the reference geometry. For reference geometries where different values of a categorical variable were observed, all areas with one of these values were treated as possibly similar (before computing the similarity from the continuous variables), as long as the variable covered more than 5% of the pixels in the reference geometry. In this way we avoid that bare rock is treated as a suitable area for replacement if it actually only covers a minor part of a PA.

3) We also developed an additional package as eHab offers a range of features additional to the ones offered by dismo. These include

    a. Computation of Habitat replaceability index

    b. Uncertainty propagation of uncertain input

    c. Possibility for weighting the observations, and the use of multipolygons for alternative methods for finding the similarities between a PA and the surroundings.

The dismo package is still used, but mainly for taking advantage of its implementation of three other methods for species distribution modelling. These methods are Maximum Entropy or Maxent [*Hijmans and Graham*, 2006; *Phillips et al.*, 2006], Bioclim [*Busby*, 1991; *Nix*, 1986] and Domain [*Carpenter et al.*, 1993]. For using the Maxent method, it is necessary to explicitly copy the Maxent java library into the java folder of dismo. This is because redistribution of the Maxent library is not legal according to the Maxent license. However, it is allowed to use the method as a part of a Web Service, given that the proper usage restrictions are passed to the end user.

3.1 Function structure

There are four main functions in eHab, currently three of these are available to the external users. These are:

habMeanCovPoly – This is the function that extracts the habitat characteristics from the indicator layers. The function also checks if some of the layers are completely correlated, if there are categorical variables, or if any of the variables are constant within the reference geometry. If variables are perfectly correlated, one of them is removed, and this information is also stored in the object returned. Constant variables are treated as categorical variables, and for both of these, the value within the reference geometry is added to the object returned.

pHabitat – The function which either computes the similarities, or that call the necessary functions in the dismo package. Two columns are added to the input – the Mahalanobis distance and the similarity. In addition the results from habMeanCovPoly are returned as attributes, accessible as attr(result, “attributeName”).

hri – This is the function that computes the habitat replaceability index. It can take population density of species into account, although we have not used this option yet. The function can compute the hri for a range of similarity levels, both inside and outside the reference geometry. This is also returned as an attribute of the result, that is otherwise similar to the result from pHabitat. The hri-function returns the computed hri as an attribute to the result.

mhri – This is the function that can compute uncertain hri and uncertain similarity maps. The inputs can be the same as above, indicators and a spatial reference geometry. The function will then first compute a linear model of covariance (LMC), which is a set of variograms and cross variograms between the indicator variables, as described above. Also an assumed standard deviation will be estimated for each layer. Alternatively, if such information exist, the function can also take the LMC and the uncertainty as input arguments. The uncertainty can be passed as a WCS to the WPS, but there is currently no standardized XML-extension that can encode the LMC. The mhri-function returns the computed hri for all realizations as an attribute to the result, in addition to a summary of the hri (mean and standard deviation).

Also other functions exist, although these are mostly for desktop analysis and for other computational necessities for the Web Services under construction at the JRC.

The help files for all functions are given in the Annex.

3.2 Package structure

Through the implementation of eHab as an R package, it is possible to take advantage of the package system of R. First of all, this means that all functions exposed to the end user (or to a WPS) has a separate help file that is easily accessible through the R GUI. The collection of these help files is also available as a pdf manual, which is attached in the Annex.

A second advantage is that R has a check function, which first of all checks the integrity of the package. Second, it can run predefined test scripts, which, if they succeed, are compared with earlier results. In this way we can control that developments do not change earlier results, unless this was the purpose.

4 Implementation of eHabitat WPS

4.1 Architecture

The first version of eHabitat (eHabitat 1.0) was designed as a proof of concept to compute the PoHS (Probability of Habitat Similarity) for a given Protected Area using only three predefined thematic maps. However, the need to monitor ecosystems outside of protected areas, whether terrestrial or marine, is stronger than ever, if only to assess connectivity between protected areas and the external pressures caused by competition for land and water. Providing the scientific community with the means to compute the PoHS anywhere on the globe, using their own thematic ingredients, is therefore an interesting option.

The current version of eHabitat (eHabitat 2.0) therefore allows for an arbitrary number of input indicators along with the definition of an area of interest, which serves as a bounding box constraint for further processing [*GEOSS AIP-3 Engineering Report*, 2011].

Technically, PyWPS [*Čepický and Becchi*, 2007] was chosen as the WPS implementation. It is a lightweight Python based server that easily integrates with the Apache Webserver, e.g. using the Common Gateway interface (CGI). The WPS serves the habitat modelling as one process, which is also implemented using Python.

The process expects several mandatory and some optional parameters (see Table 1). Using the Python-bindings for GDAL (Geospatial Data Abstraction Layer)[5] and OWSlib[6] the process can ingest and output a variety of different geospatial data formats (e.g. GeoTIFF, NetCDF) as well as different OGC web services, like Web Feature Service (WFS), Web Coverage Service (WCS) or Catalog Service for the Web (CSW).

A schema showing the architecture and the dataflow in eHabitat 2.0 is shown in Figure 2.

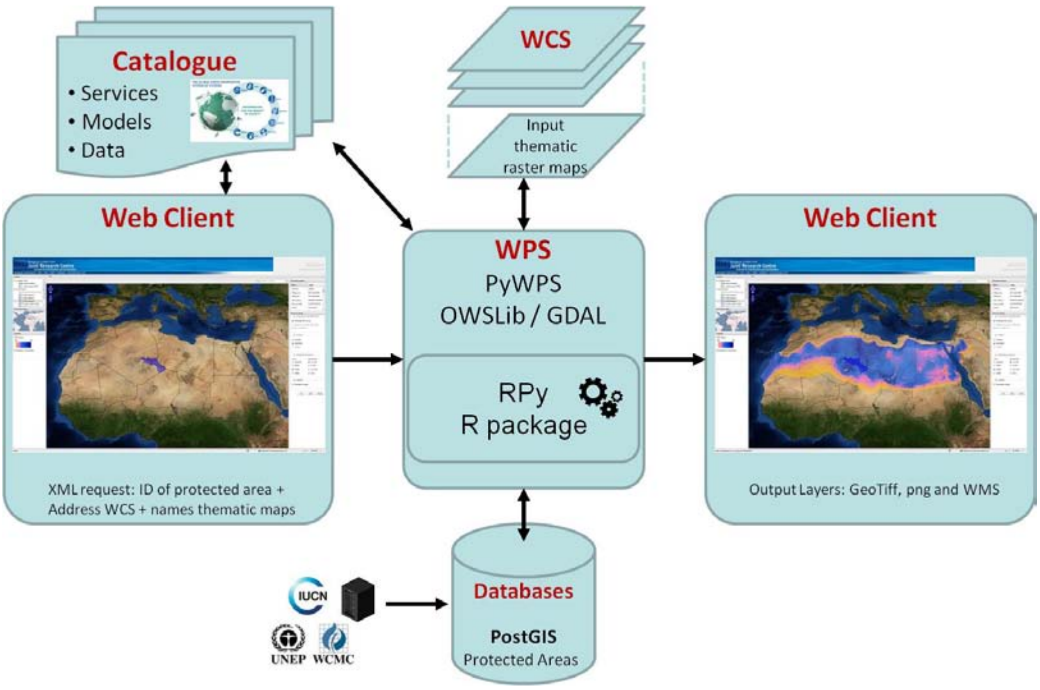

Figure 2. Design of the architecture of eHabitat WPS

[5] http://www.gdal.org

[6] http://owslib.sourceforge.net/

Name	Card.	Description	Type[b]	Format
indicators	3..*	Multiple WCS/CSW URLs pointing to indicator coverages	ComplexData	Geotiff NetCDF
siteID[a]	0..1	WDPA site identification number, resulting in the reference geometry	LiteralData	Integer
sitePolygon[a]	0..1	Well-Known-Text representation of a user defined polygon	LiteralData	WKT
aoi	1	Bounding box defining the area of interest	BoundingBoxData	
method	0..1	Statistical backend method (Mahalanobis, BioClim, Domain, Maxent)	LiteralData	String
forecast	0..1	Enable forecasting, default false (if true the forecasted indicators have to be provided as well)	LiteralData	Boolean
numRealisations	0..1	Number of realizations, to calculate uncertainty	LiteralData	Integer
[a] exactly one of these parameters has to be submitted [b] these types refer to the InputFormChoice data structure (Table 20) as defined in Schut, 2007				

Table 1: Overview of the main mandatory and optional input parameters for the eHabitat process (for details see DescribeProcess.xml in the annex).

4.2 Sample execution and results

The process is initiated by sending an WPS Execute request to the WPS server (see Annex for an example Execute.xml). This request describes all the required input parameters and desired outputs.

Indicator datasets have to be referenced in the request using WCS DescribeCoverage or a GDAL specific XML structure. The datasets are accessed using the provided area of interest and the default spatial resolution of the WCS layer with a GetCoverage request. All indicator layers that are requested must share similar geospatial properties (coordinate reference system (CRS), resolution). It is not in the scope of the habitat modelling to provide resampling or reprojection.

The reference geometry for which to compute the Mahalanobis distances (i.e. the boundary of the Protected Area) is referenced using either a unique id specific to this PA as defined by the World Database on Protected Areas (WDPA) or by a Well-Known-Text (WKT) representation of the polygon geometry (see Table 1).

The Execute request defines in its Response section also the details of the outputs the WPS should generate and return to the caller (see table 2 for a selection of possible outputs).

When the Execute request is received and the requested process was started, the input parameters are validated and checked for consistency. Eventually, e.g. if a mandatory parameter is missing, a WPS exception is returned to the caller. If all provided inputs are correct, the data for the indicators are downloaded, transformed in R data structures and the R computation is initiated.

The results returned from the R code are processed to generate different output formats depending on the required further use of those outputs. If the user wants to perform processing on the output, a GeoTiff, NetCDF (Network Common Data Format) or OGC WCS reference can be requested. For visualization, PNG (Portable Network Graphics) images may be sufficient. For visualization in Web mapping clients, the user may request the output as an OGC-WMS reference (see example of ExecuteResponse.xml in the Annex).

Name	Description	Type[a]	Format
MahalDist	Raw similarity data as computed by the chosen method	ComplexData	Geotiff NetCDF
layerMahalDist	Reference to an OGC-WMS layer serving the result (GteMap)	ComplexData	URL
hri	Computed HRI value	LiteralData	Float
uhri	Computed mean and standarddeviation for HRI value	ComplexData	uncertML
MahalDistSummary Statistics	summary statistics for the similarity	ComplexData	uncertML netCDF-U
omIndicatorValues	Raw values of indicator values inside the PA OM encoded	ComplexData	XML JSOM
PNGoutput	Rendered image of the result with country borders background, legend and scale	ComplexData	URL
[a] These types refer to the OutputData data structure (table 60) and DataType data structure (table 46) as defined in [*Schut*, 2007]			

Table 2: Selection of requestable outputs of the eHabitat WPS (for details see DescribeProcess.xml in the annex).

4.3 Propagating uncertainty in the eHabitat WPS

Skøien et al. [*Skøien et al.*, 2011] have shown how to assess possible uncertainties in eHabitat using geostatistics and Monte Carlo simulations when these uncertainties have not been documented in the input data, a case that is the most frequently encountered.

When mainly dealing with gridded datasets, propagating uncertainty using XML encoding in the WPS response has some drawbacks. The connection between the model result and propagated uncertainties is only referenced in the WPS response document. Propagating multiple model results with correlated uncertainties is complicated due to the simple structure of the WPS response.

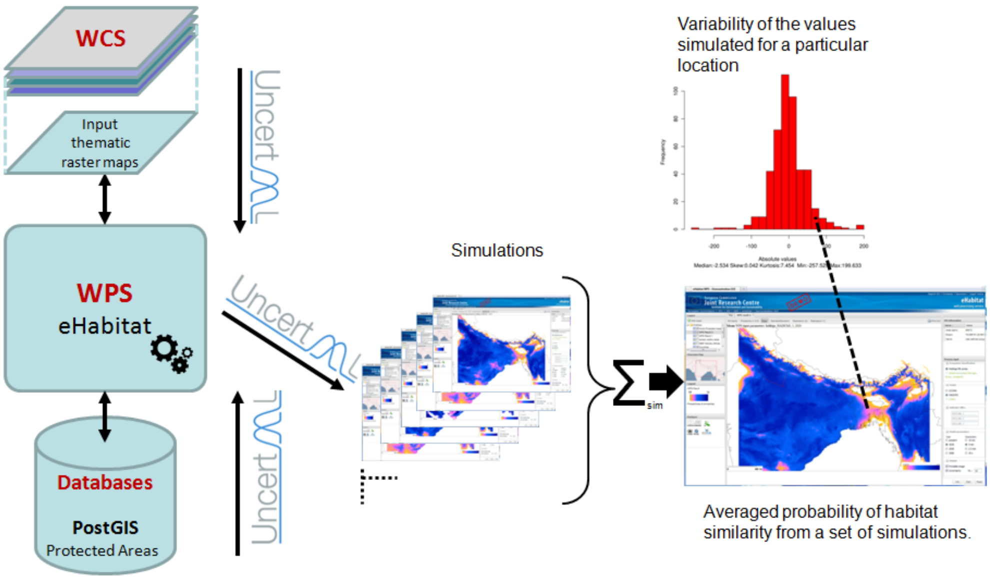

Figure 3. Uncertainty propagation and modelling in eHabitat.

In addition to propagating the uncertainties using UncertML XML-encoding, eHabitat can also make use of the NetCDF-U encoding as proposed in the UncertWeb project [*Bigagli and Nativi*, 2011]. Therefore a NetCDF-U Python-wrapper library is being developed, that is based on the generic Python-NetCDF library netdf-4 python[7].

In collaboration with WP3 and their development of the uncertainty visualisation client VISS, the propagation of the uncertainty results from the eHabitat computation to the VISS client using the demonstration web application[8] is established.

Besides visualisation of model outputs, the communication of intermediate results from the model is important as well. For the case of eHabitat, the spatial pattern of the similarity is directly linked to the data distribution within the protected area. Examining intermediate results can help in understanding the final result, particularly in cases where it differs from the expectations [*Schulz et al.*, 2012].

Therefore the eHabitat accumulates all observed values for all input indicator datasets inside the reference geometry (aka. the PA) and encodes them as JSON UncertML O&M (Observation & Measurement). The reference to this O&M result can be loaded by the VISS client and thus visualises the boundary of the protected area and a histogram of the observed indicator values.

[7] http://code.google.com/p/netcdf4-python/

[8] http://ehabitat-wps.jrc.ec.europa.eu/ehabitat/

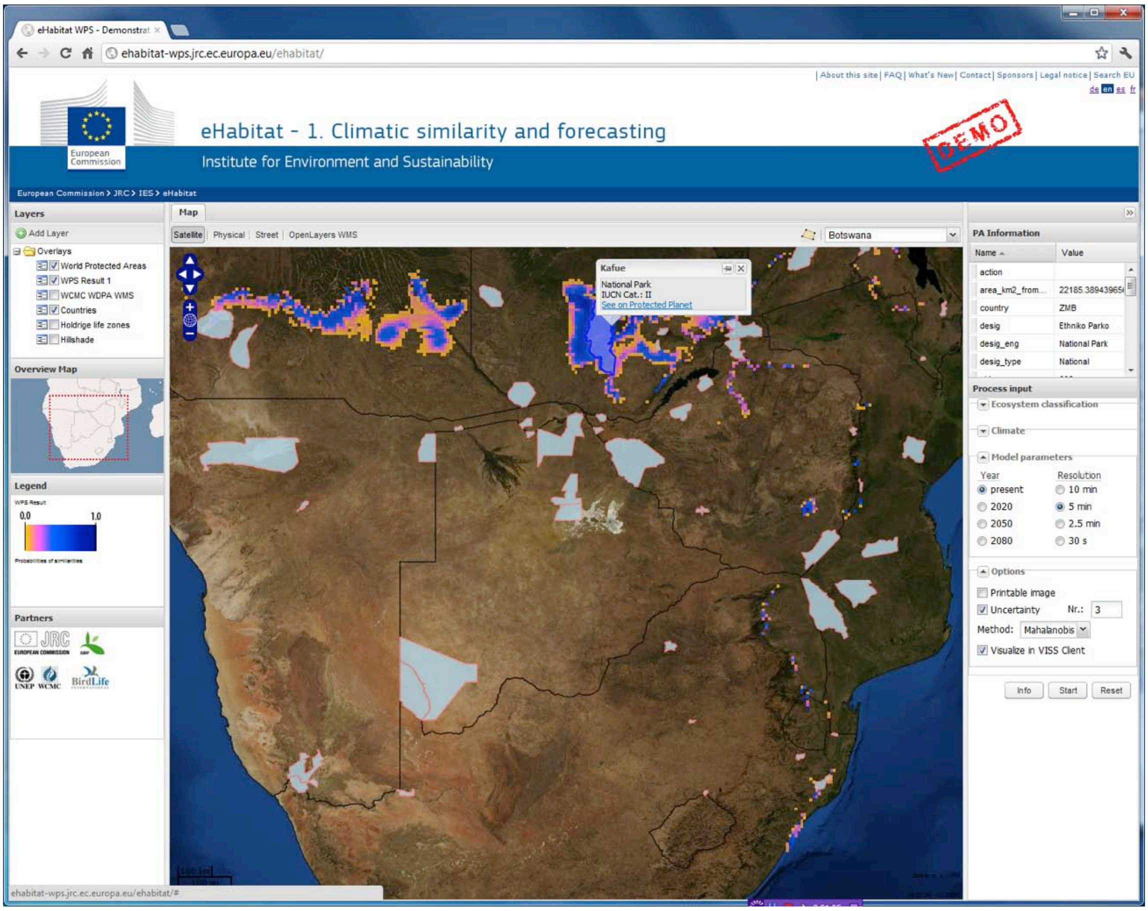

Figure 4. Similarity result in eHabitat web application.

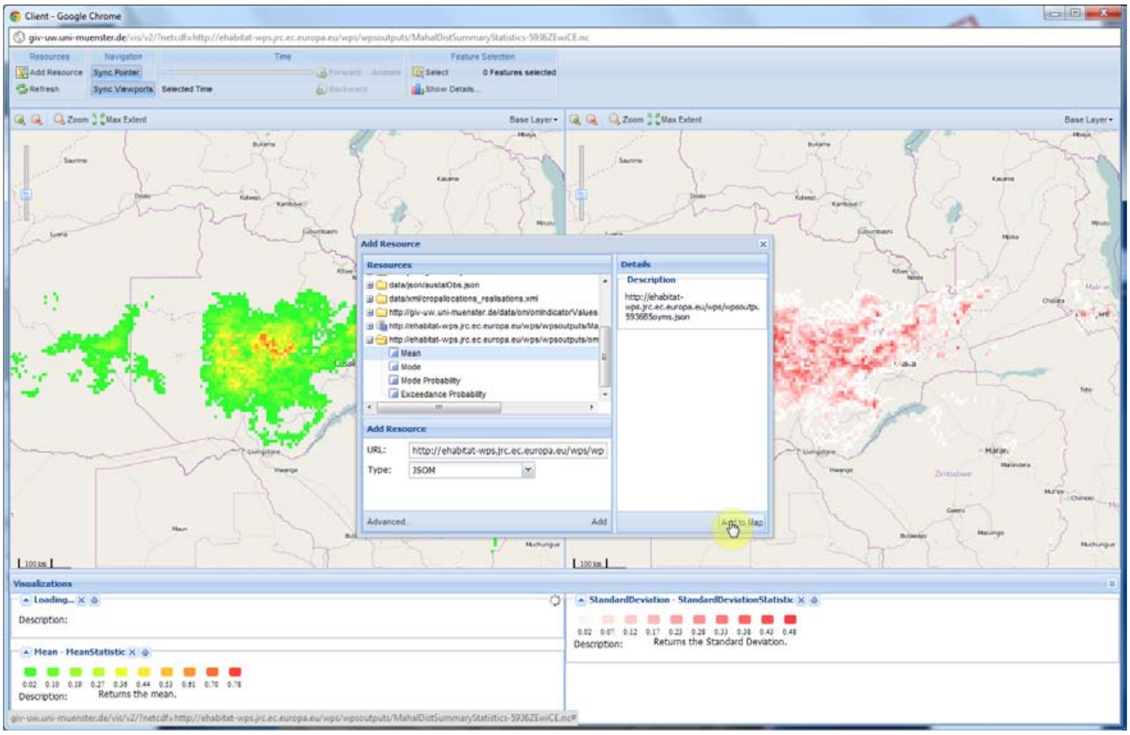

Figure 5. NetCDF-U of the same result, with Mean and StandardDeviation visualised in VISS client.

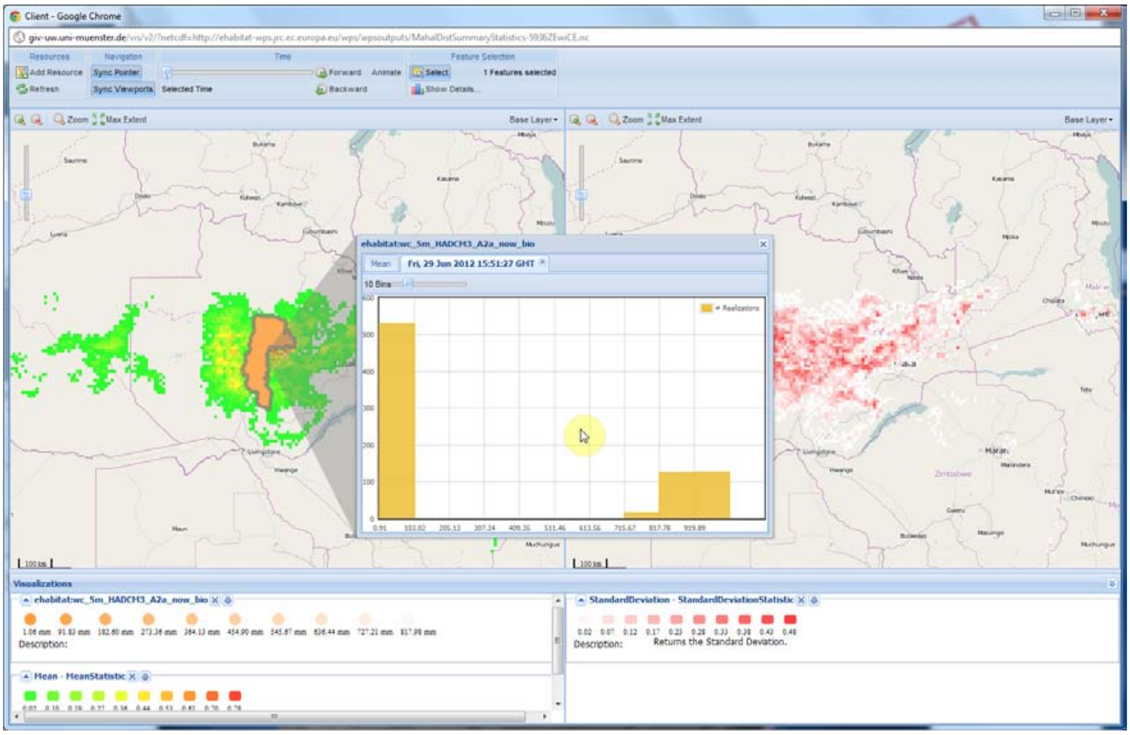

Figure 6. JSON O&M showing park boundary and distribution of biotemperature values inside the PA.

5 Use of eHabitat WPS

5.1 Default data

Among the typical parameters one would measure when modelling the ecological niche of a species, are climatic parameters as these are determinant factors in the species distributions. We have therefore made some of these available as Web Coverage Services, for testing the WPS, by using three climatic variables, namely the mean annual precipitation (*P*), the annual average of the Biotemperature (B) and the ratio of mean annual potential evapotranspiration (PET). These variables are as defined by Leslie Holdridge, who generated a bioclimatic scheme at the global level [*Holdridge*, 1947]. This simple scheme can be reused to identify the bioclimatic area a species would fall in, using today's climate. When the bioclimatic niche of the species (as defined from the mean and covariance function of the Mahalanobis distance) has been found from the current climate, the WPS can, using predicted climate from climate change models, be used for forecasting possible changes to a species' ecological niche. Figure 7 is a graphical representation of the use of the Holdridge variables for mapping areas with similar bioclimatic conditions.

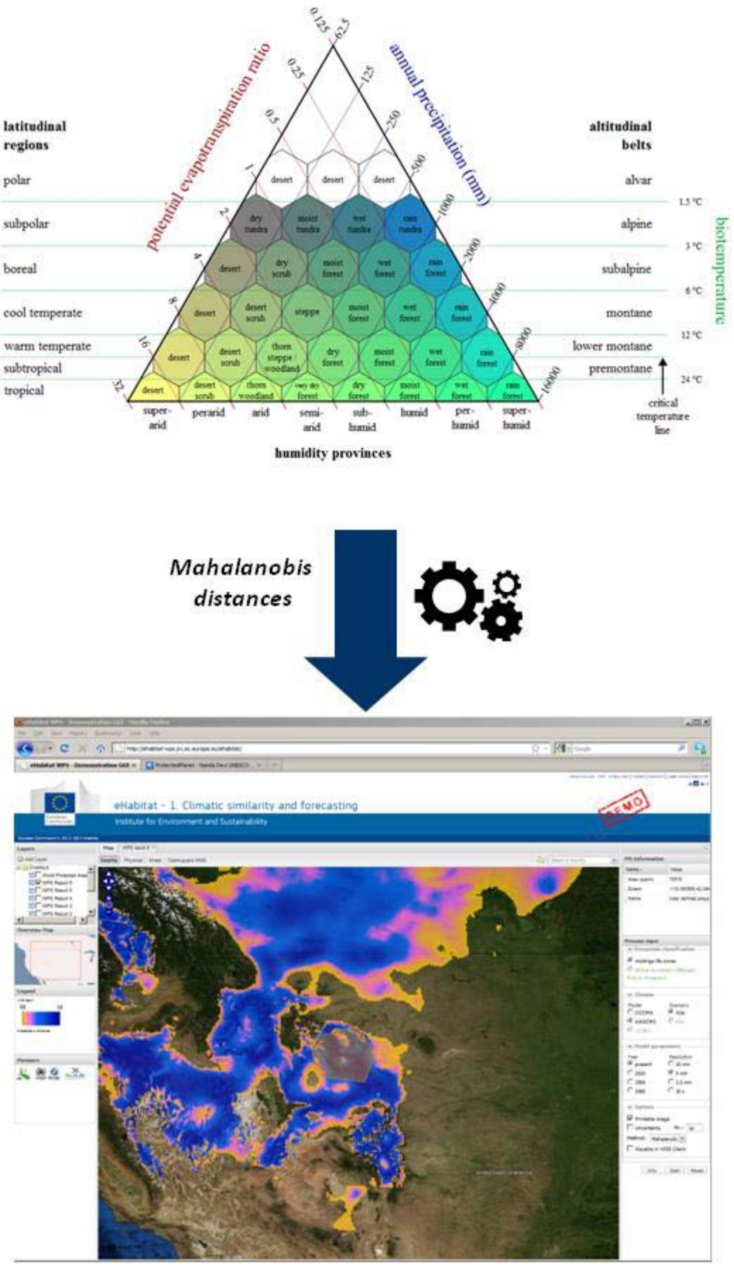

Figure 7. Holdridge's lifezones as a function of different climatic variables and their use for mapping bioclimatic similarities of a selected area (orange polygon in centre of web client)

5.2 Case study on climate change in the Nanda Devi UNESCO-MAB Biosphere Reserve, India.

The **Nanda Devi National Park**[9] is a national park situated around the peak of Nanda Devi, 7,817 m in the state of Uttarakhand in northern India. It was established as national park in 1982 and covers 630.33 km².

The park encompasses the Nanda Devi Sanctuary, a glacial basin surrounded by a ring of peaks between 6,000 metres and 7,500 m high, and drained by the Rishi Ganga through the Rishi Ganga Gorge, a steep, almost impassable defile. Together with the nearby Valley of Flowers National Park to the northwest, it is a designated World Heritage Site. Both parks are

[9] http://protectedplanet.net/sites/902492

encompassed in the Nanda Devi Biosphere Reserve (223,674 ha), which is further surrounded by a buffer zone (5,148.57 km²).

The hri result is returned from the WPS, but not from the web client. So whereas this part of the result is not visible in the figures below, the XML indicated that the mean hri was 1.28, with a standard deviation of 0.41.

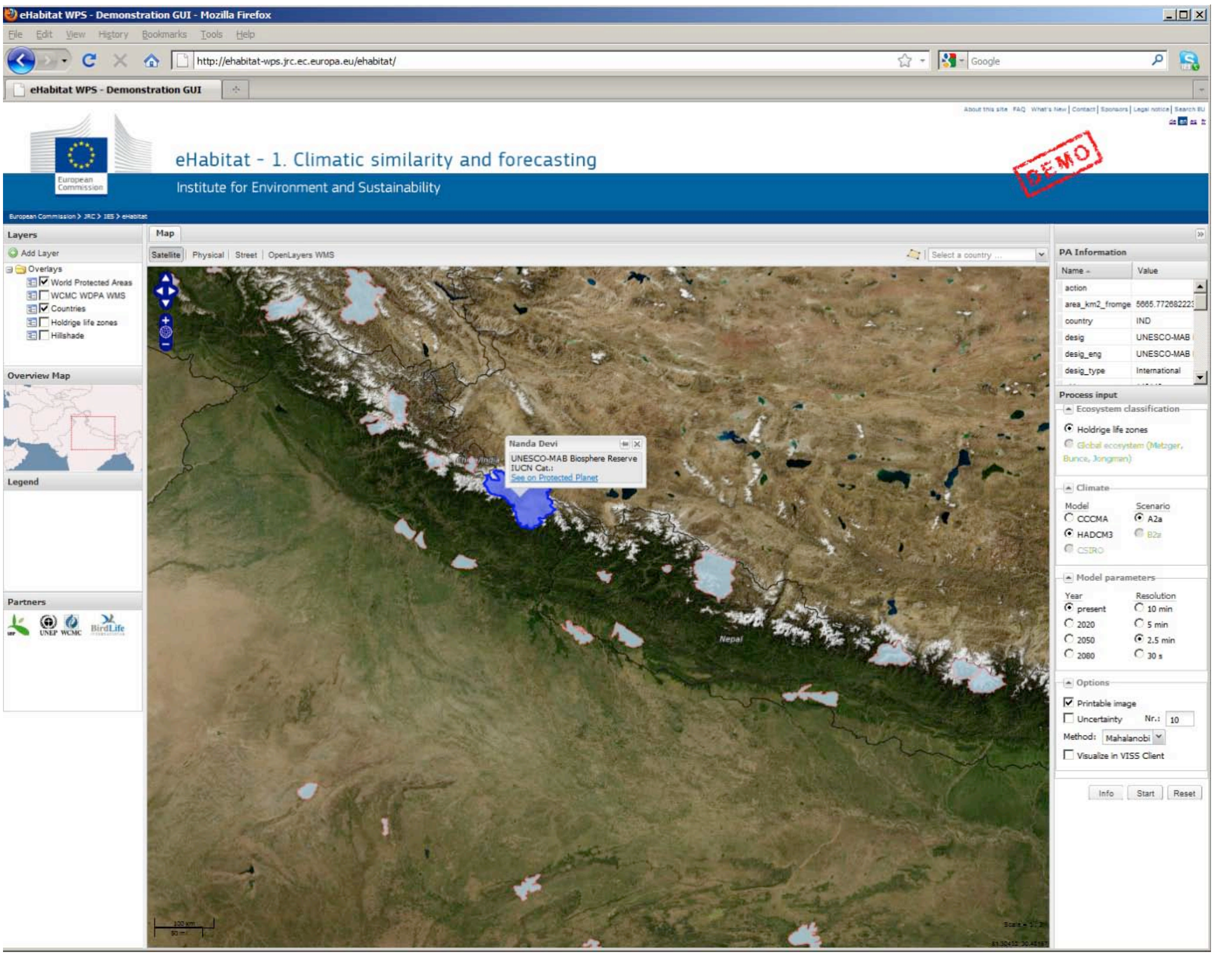

Figure 8. Selection of the Nanda Devi Biosphere Reserve in India in a web client of eHabitat

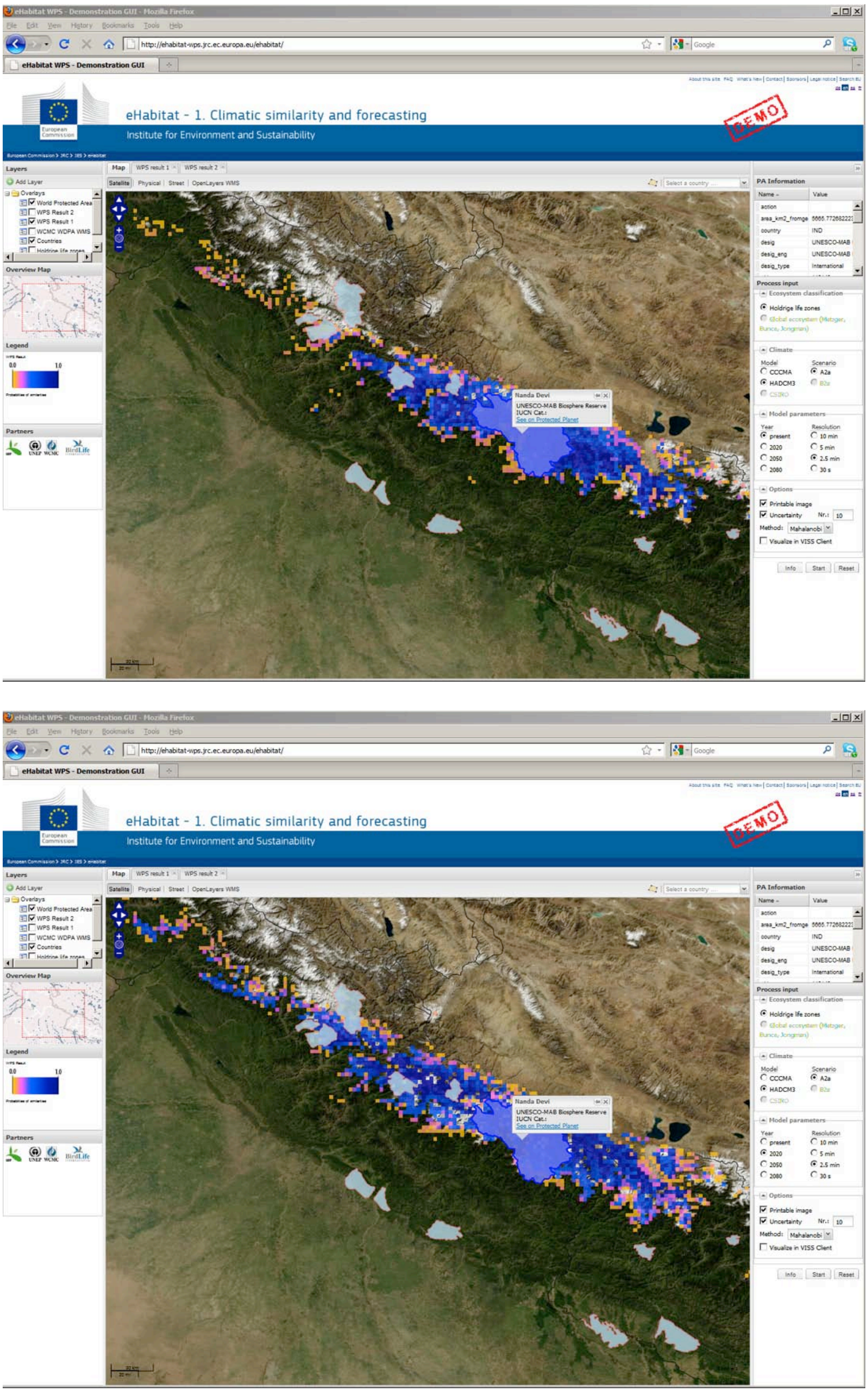

Figure 9. Holdridge's lifezones as a function of different climatic variables and their use for mapping bioclimatic similarities of a selected area, as illustrated here for the Nanda Devi Biosphere Reserve in India. The top figure shows the current situation, the bottom one shows the forecasted situation for 2020

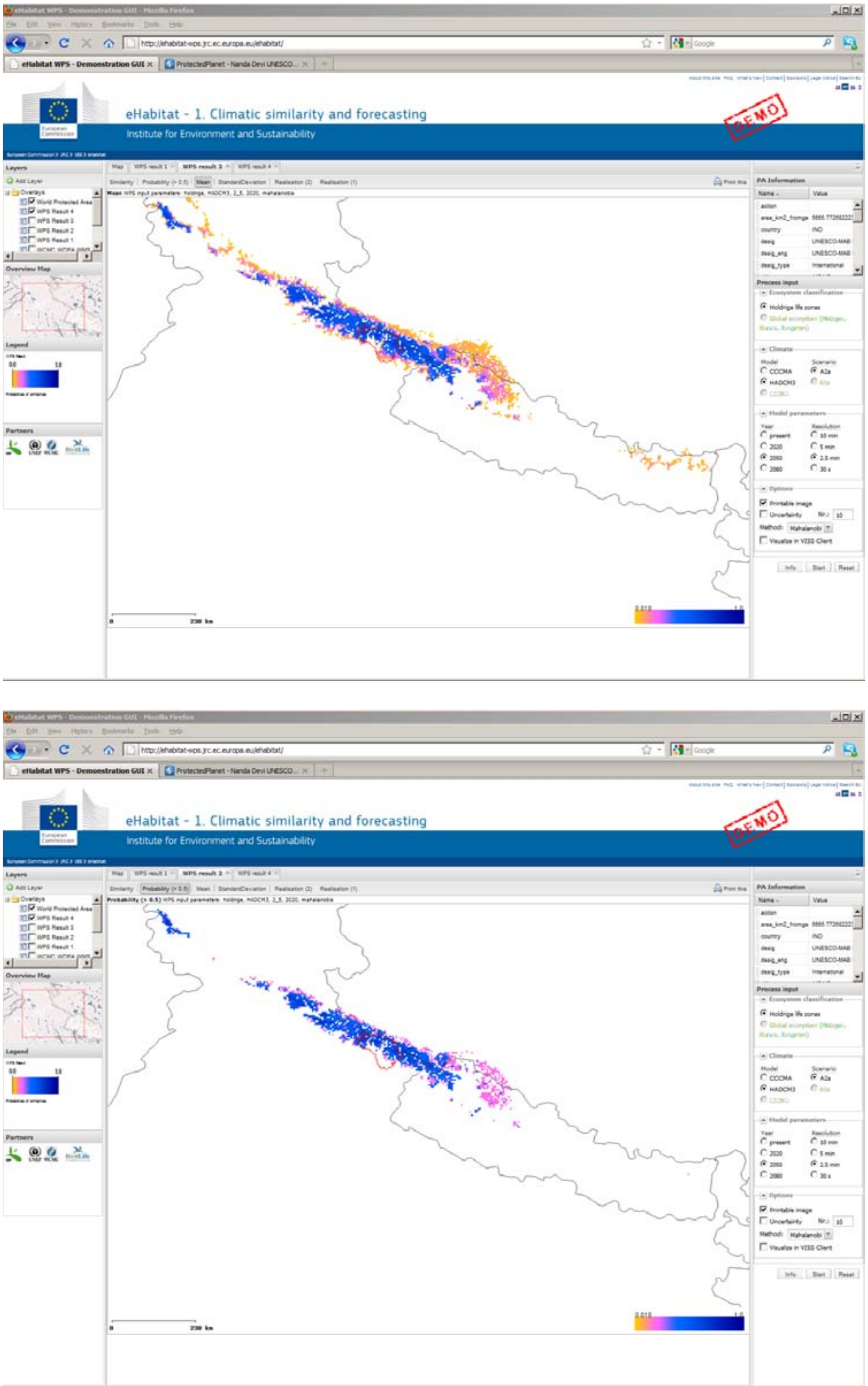

Figure 10. Visualisation of the use of uncertainties in eHabitat. Results of 10 simulations for the 2020 scenario: mean values (top) and probabilities > 0.5 (bottom).

Conclusions

Multi-disciplinary information integration is recognized by the scientific community as essential for the understanding of complex issues such as the response of biodiversity to global changes. This calls for the further development of flexible and scalable systems allowing integration with existing (and heterogeneous) services and data systems. The Digital Observatory for Protected Areas (DOPA), of which eHabitat is a component, is an example of such a platform where observations and models relating to trends in the world's ecosystems and species can be integrated. Relying on the dynamic model infrastructure envisioned in the Model Web, DOPA's many benefits include improved means to discover, access, reuse and chain models and datasets for multiple purposes.

The eHabitat WPS described here should illustrate these benefits: different web clients designed for different end-users and use-cases can be easily built on the top of a fundamental modeling service. The versatility of eHabitat allows it to be used within different contexts and workflows. The adoption of standard Web services to publish eHabitat WPS allows interoperability with other web services and should encourage its use by other communities; its reuse will largely depend on the development of new services allowing semantic interoperability.

The first steps for this have been taken, resulting in this OGC-WPS for ecological modelling and climatic forecasting, that is accessing datasets using OGC WCS, propagating results as OGC web services and enhancing them with uncertainty information encoded in XML or NetCDF-U using UncertML. The adoption of NetCDF-U to tightly couple the actual model results with correlated uncertainties eases propagation, enables reuse even in not UncertML aware clients. Communication of uncertainties is important not only for the final model outputs but also for input data and intermediate results. By coupling eHabitat results with the web-based thin client for visualization developed in WP 3, this can be easily realised for each processing step by providing the data reference through the client URL. This concept can also be applied when using eHabitat in a chain of web services.

We have in our implementation been limited by the lacking existence of relevant data sets with proper descriptions of the uncertainty. However, our system includes a simple method for generating uncertainties from the input data set, which will at least make it possible to assess the sensitivity of the results. When data sets with real uncertainties are available, these can be used directly.

Acknowledgements

This work is partly supported by the European Commission, under the 7th Framework Programme, by the UncertWeb project funded by the DG INFSO. The views expressed herein are those of the authors and are not necessarily those of the European Commission.

More information about eHabitat and UncertWeb can be found on the Internet, see http://ehabitat.jrc.ec.europa.eu/ and http://www.uncertweb.org/ , respectively.

References

Best, B. D., P. N. Halpin, E. Fujioka, A. J. Read, S. S. Qian, L. J. Hazen, and R. S. Schick (2007), Geospatial web services within a scientific workflow: Predicting marine mammal habitats in a dynamic environment, *Ecological Informatics*, *2*(3 SPEC. ISS.), 210-223.

Bigagli, L., and S. Nativi (2011), Managing uncertainty in data – the NetCDF uncertainty conventions, 79th OGC Technical Committee, Brussels Belgium, http://www.uncertweb.org/documents/presentations/managing-uncertainty-in-data-the-netcdf-uncertainty-conventions/download accessed on 2 March 2012.

Bivand, R. S., and C. Rundel (2012), rgeos: Interface to Geometry Engine - Open Source (GEOS). R package version 0.2-7. http://CRAN.R-project.org/package=rgeos, edited.

Bivand, R. S., E. J. Pebesma, and V. Gómez-Rubio (2008), *Applied spatial data analysis with R*, 378 pp., Springer.

Busby, J. R. (1991), BIOCLIM - a bioclimate analysis and prediction system, *Nature conservation: cost effective biological surveys and data analysis*, 64-68.

Carpenter, G., A. N. Gillison, and J. Winter (1993), DOMAIN: A flexible modelling procedure for mapping potential distributions of plants and animals, *Biodiversity and Conservation*, *2*(6), 667-680.

Čepický, J., and L. Becchi (2007), Geospatial processing via Internet on remote servers-PyWPS, *OSGeo Journal*, *1*, 39-42.

Clark, J. D. (2012), Personal communication.

Clark, J. D., J. E. Dunn, and K. G. Smith (1993), A multivariate model of female black bear habitat use for a geographical information system, *Journal of Wildlife Management 57*, 519-526.

Dubois, G., M. Clerici, S. Peedell, P. Mayaux, J.-M. Gregoire, and E. Bartholome (2010), A Digital Observatory for Protected Areas - DOPA, a GEO-BON contribution to the monitoring of African biodiversity, In: *Proceedings of Map Africa 2010*, Cape Town, South Africa.

Geller, G., and W. Turner (2007), The Model Web: A concept for ecological forecasting, in *IEEE International Geoscience and Remote Sensing Symposium*, edited, Barcelona, Spain, 23-27 July.

GEOSS AIP-3 Engineering Report (2011), eHabitat, http://www.ogcnetwork.net/system/files/CCBio-eHabitat-ER-v2.0-FINAL.pdf, accessed on.

Graham, C. H., et al. (2008), The influence of spatial errors in species occurrence data used in distribution models, *Journal of Applied Ecology*, *45*(1), 239-247.

Hiemstra, P. H., E. J. Pebesma, C. J. W. Twenhöfel, and G. B. M. Heuvelink (2008), Real-time automatic interpolation of ambient gamma dose rates from the Dutch Radioactivity Monitoring Network, *Computers & Geosciences*, *35*(8), 1711-1721.

Hijmans, R., and C. Graham (2006), The ability of climate envelope models to predict the effect of climate change on species distribution, *Global Change Biology*, *12*, 2272-2281.

Hijmans, R., and J. van Etten (2012), raster: Geographic analysis and modeling with raster data, edited.

Hijmans, R., S. Phillips, J. Leathwick, and J. Elith (2012), dismo: Species distribution modelling. R package version 0.7-17. http://CRAN.R-project.org/package=dismo.

Holdridge, L. R. (1947), Determination of world plant formations from simple climatic data, *Science*, *105*, 367-368.

Keitt, T. H., R. S. Bivand, E. Pebesma, and B. Rowlingson (2011), rgdal: Bindings for the Geospatial Data Abstraction Library., edited.

Knick, S. T., and D. L. Dyer (1997), Distribution of black-tailed jackrabbit habitat determined by GIS in southwestern Idaho, *Journal of Wildlife Management*, *61*(1), 75-85.

Mahalanobis, P. C. (1936), On the generalised distance in statistics, for the classification problem, *Proceedings of the National Institute of Sciences of India*, *2*(1), 49-55.

Muñoz, M. E. S., R. Giovanni, M. F. Siqueira, T. Sutton, P. Brewer, R. S. Pereira, D. A. L. Canhos, and V. P. Canhos (2011), openModeller: a generic approach to species' potential distribution modelling, *GeoInformatica*, *15*, 111-135.

Nativi, S., P. Mazzetti, and G. N. Geller (2012), Environmental model access and interoperability: The GEO Model Web initiative, *Environmental Modelling and Software*.

Nativi, S., P. Mazzetti, H. Saarenmaa, J. Kerr, and E. O. Tuama (2009), Biodiversity and climate change use scenarios framework for the GEOSS interoperability pilot process,, *Ecological Informatics*, *4*, 23-33.

Nix, H. A. (1986), A biogeographic analysis of Australian elapid snakes, in *Australian Flora and Fauna Series No. 7*, edited by R. Longmore, pp. 4-15, Australian Government Publishing Service, Canberra.

Pebesma, E. J. (2004), Multivariable geostatistics in S: the gstat package, *Computers & Geosciences*, *30*, 683-691.

Phillips, S. J., R. P. Anderson, and R. E. Schapire (2006), Maximum entropy modelling of species geographic distributions, *Ecological modelling*, *190*, 231-259.

R Development Core Team (2012), R: A language and environment for statistical computing*ISBN 3-900051-07-0*, R Foundation for Statistical Computing, Vienna, Austria.

Rocchini, D., J. Hortal, S. Lengyel, J. M. Lobo, A. Jiménez-Valverde, C. Ricotta, G. Bacaro, and A. Chiarucci (2011), Accounting for uncertainty when mapping species distributions: The need for maps of ignorance, *Progress in Physical Geography*, *35*(2), 211-226.

Rotenberry, J. T., K. L. Preston, and S. T. Knick (2006), GIS-based niche modelling for mapping species' habitat, *Ecology*, *87*, 1458-1464.

Schulz, M., J. O. Skøien, L. Gerharz, G. Dubois, and E. Pebesma (2012), Uncertainty propagation between web services – a case study using the eHabitat WPS to identify unique ecosystems, In: *Proceedings of 6th International Congress on Environmental Modelling and Software - iEMSs 2012*, Leipzig, Germany, July 1-5, 2012.

Searle, S. (1971), *Linear Models*, Wiley, New York.

Service, R. F. (2011), Coming soon to a lab near you: drag-and-drop virtual Worlds, *Science*, *331 (6018)*, 669-671.

Skøien, J. O., M. Schulz, G. Dubois, G. B. M. Heuvelink, and D. Cornford (2011), Uncertainty propagation in chained web based modeling services: the case of eHabitat, In: *Proceedings of EnviroInfo*, Shaker Verlag, Joint Research Centre, Ispra, Italy.

Tsoar, A., O. Allouche, O. Steinitz, D. Rotem, and R. Kadmon (2007), A comparative evaluation of presence-only methods for modelling species distribution, *Diversity and Distributions*, *13*, 397-405.

Annex

A. eHabitat WPS DescribeProcess.xml


```
 <?xml version="1.0" encoding="utf-8"?>
 <wps:ProcessDescriptions xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd"
  service="WPS" version="1.0.0" xml:lang="en-CA">
  <ProcessDescription wps:processVersion="0.2" storeSupported="true"
statusSupported="true">
    <ows:Identifier>ehabitatprocess_bbox_multi_uncert</ows:Identifier>
    <ows:Title>ehabitatprocess_bbox_multi_uncert</ows:Title>
    <ows:Abstract>The eHabitat process calculates the mahalanobis distance of
various input variables (to be provided as WCS/CSW URLs) wrt. to a certain sample
area (e.g. a protected area or arbitrary polygon) within a desired bounding
box.</ows:Abstract>
    <DataInputs>
     <Input minOccurs="0" maxOccurs="1">
      <ows:Identifier>scale</ows:Identifier>
      <ows:Title>The png image will contain a scale bar on the lower right
corner</ows:Title>
      <LiteralData>
       <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#boolean">boolean</ows:DataType>
       <DefaultValue>true</DefaultValue>
       <ows:AnyValue />
      </LiteralData>
     </Input>
     <Input minOccurs="0" maxOccurs="1">
      <ows:Identifier>numRealisations</ows:Identifier>
      <ows:Title>Number of realisations</ows:Title>
      <ows:Abstract>Number of realisations done during the computation to calculate
uncertainty.</ows:Abstract>
      <LiteralData>
       <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#integer">integer</ows:DataType>
       <ows:AllowedValues>
         <ows:Range ows:rangeClosure="None">
          <ows:MinimumValue>3</ows:MinimumValue>
          <ows:MaximumValue>50</ows:MaximumValue>
         </ows:Range>
       </ows:AllowedValues>
      </LiteralData>
     </Input>
     <Input minOccurs="1" maxOccurs="1">
      <ows:Identifier>aoi</ows:Identifier>
      <ows:Title>Bounding box describing the area of interest</ows:Title>
      <BoundingBoxData>
       <Default>
        <CRS>EPSG:4326</CRS>
       </Default>
       <Supported>
        <CRS>EPSG:4326</CRS>
       </Supported>
      </BoundingBoxData>
     </Input>
     <Input minOccurs="0" maxOccurs="1">
      <ows:Identifier>forecast</ows:Identifier>
      <ows:Title>Enable forecasting, defaults to false</ows:Title>
      <ows:Abstract>Enable forecasting, defaults to false. If set to true, the
indicators must be two sets of equally ordered coverages.</ows:Abstract>
```

```
      &lt;LiteralData&gt;
        <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#boolean">boolean&lt;/ows:DataType&gt;
        &lt;DefaultValue&gt;false&lt;/DefaultValue&gt;
        &lt;ows:AnyValue /&gt;
      &lt;/LiteralData&gt;
    &lt;/Input&gt;
    &lt;Input minOccurs="0" maxOccurs="1"&gt;
      &lt;ows:Identifier&gt;sitePolygon&lt;/ows:Identifier&gt;
      &lt;ows:Title&gt;Polygon to be used if SiteID is not used, polygon should be
described in WKT format. In PostGIS the ST_asWKT() or ST_AsEWKT() outputs the
proper format. Example:'POLYGON((-8.7986005103092 37.9237094675553,-
8.79821062608261 37.9233896392756,...-8.7986005103092
37.9237094675553))&lt;/ows:Title&gt;
      &lt;LiteralData&gt;
        <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#string">string&lt;/ows:DataType&gt;
        &lt;ows:AnyValue /&gt;
      &lt;/LiteralData&gt;
    &lt;/Input&gt;
    &lt;Input minOccurs="0" maxOccurs="1"&gt;
      &lt;ows:Identifier&gt;siteID&lt;/ows:Identifier&gt;
      &lt;ows:Title&gt;WDPA site identification number. The WDPA will have precedence
over sitePolygon input&lt;/ows:Title&gt;
      &lt;LiteralData&gt;
        <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#integer">integer&lt;/ows:DataType&gt;
        &lt;ows:AnyValue /&gt;
      &lt;/LiteralData&gt;
    &lt;/Input&gt;
    &lt;Input minOccurs="1" maxOccurs="9"&gt;
      &lt;ows:Identifier&gt;indicators&lt;/ows:Identifier&gt;
      &lt;ows:Title&gt;Multiple WCS/CSW URLs pointing to indicator coverages&lt;/ows:Title&gt;
      &lt;ows:Abstract&gt;Multiple WCS/CSW URLs pointing to indicator coverages using
valid WCS DescribeCoverage URLs (supported WCS version 1.0.0 - 1.1.1) or a CSW
GetRecordByID URL (as discovered thru a catalog service), for forecasting same
indicator coverages in the same order have to be specified. Instead of direct WCS
URLs, CSW GetRecordById requests that point to WCS are also
supported.&lt;/ows:Abstract&gt;
      &lt;LiteralData&gt;
        <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#anyURI">anyURI&lt;/ows:DataType&gt;
        &lt;ows:AnyValue /&gt;
      &lt;/LiteralData&gt;
    &lt;/Input&gt;
    &lt;Input minOccurs="0" maxOccurs="1"&gt;
      &lt;ows:Identifier&gt;method&lt;/ows:Identifier&gt;
      &lt;ows:Title&gt;Choose the statistical backend method.&lt;/ows:Title&gt;
      &lt;LiteralData&gt;
        <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#string">string&lt;/ows:DataType&gt;
        &lt;DefaultValue&gt;mahalanobis&lt;/DefaultValue&gt;
        &lt;ows:AllowedValues&gt;
          &lt;ows:Value&gt;mahalanobis&lt;/ows:Value&gt;
          &lt;ows:Value&gt;maxent&lt;/ows:Value&gt;
          &lt;ows:Value&gt;bioclim&lt;/ows:Value&gt;
          &lt;ows:Value&gt;domain&lt;/ows:Value&gt;
        &lt;/ows:AllowedValues&gt;
      &lt;/LiteralData&gt;
    &lt;/Input&gt;
    &lt;Input minOccurs="0" maxOccurs="1"&gt;
      &lt;ows:Identifier&gt;png&lt;/ows:Identifier&gt;
      &lt;ows:Title&gt;The image will be dumped in a 255bit image in PNG format together
with GeoTiff&lt;/ows:Title&gt;
      &lt;LiteralData&gt;
        <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#boolean">boolean&lt;/ows:DataType&gt;
        &lt;DefaultValue&gt;true&lt;/DefaultValue&gt;
        &lt;ows:AnyValue /&gt;
```

```xml
      </LiteralData>
     </Input>
    </DataInputs>
    <ProcessOutputs>
     <Output>
      <ows:Identifier>processReport</ows:Identifier>
      <ows:Title>Process report</ows:Title>
      <LiteralOutput>
       <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#string">string</ows:DataType>
      </LiteralOutput>
     </Output>
     <Output>
      <ows:Identifier>PNGoutput</ows:Identifier>
      <ows:Title>Mahalanobis distance</ows:Title>
      <ows:Abstract>Return the Mahalanobis Distance result as PNG image along with
country borders and the PA outline</ows:Abstract>
      <ComplexOutput>
       <Default>
        <Format>
         <MimeType>image/png</MimeType>
        </Format>
       </Default>
       <Supported>
        <Format>
         <MimeType>image/png</MimeType>
        </Format>
       </Supported>
      </ComplexOutput>
     </Output>
     <Output>
      <ows:Identifier>MahalDist</ows:Identifier>
      <ows:Title>Mahalanobis distance</ows:Title>
      <ows:Abstract>Return the Mahalanobis Distance result as GeoTIFF
image</ows:Abstract>
      <ComplexOutput>
       <Default>
        <Format>
         <MimeType>image/tiff</MimeType>
        </Format>
       </Default>
       <Supported>
        <Format>
         <MimeType>image/tiff</MimeType>
        </Format>
        <Format>
         <MimeType>image/netCDF</MimeType>
        </Format>
       </Supported>
      </ComplexOutput>
     </Output>
     <Output>
      <ows:Identifier>MahalDistSummaryStatistics</ows:Identifier>
      <ows:Title>Mahalanobis summary statistics (mean/stddev)
information</ows:Title>
      <ows:Abstract>Return the summary statistics (mean/stddev) for then
Mahalanobis Distance result as uncertML or netCDF-U</ows:Abstract>
      <ComplexOutput>
       <Default>
        <Format>
         <MimeType>text/xml</MimeType>
         <Encoding>utf-8</Encoding>
         <Schema>http://www.uncertml.org/uncertml.xsd</Schema>
        </Format>
       </Default>
       <Supported>
        <Format>
         <MimeType>text/xml</MimeType>
         <Encoding>utf-8</Encoding>
```

```
          <Schema>http://www.uncertml.org/uncertml.xsd</Schema>
        </Format>
        <Format>
          <MimeType>image/netCDF-U</MimeType>
        </Format>
      </Supported>
    </ComplexOutput>
  </Output>
  <Output>
    <ows:Identifier>omIndicatorValues</ows:Identifier>
    <ows:Title>Indicator layer values in OM</ows:Title>
    <ows:Abstract>Return the raw values of the indicator layer values inside the
PA OM encoded, to be visualised thru the VISS client.</ows:Abstract>
    <ComplexOutput>
      <Default>
        <Format>
          <MimeType>text/xml</MimeType>
          <Encoding>utf-8</Encoding>
          <Schema>http://www.opengis.net/om/2.0</Schema>
        </Format>
      </Default>
      <Supported>
        <Format>
          <MimeType>text/xml</MimeType>
          <Encoding>utf-8</Encoding>
          <Schema>http://www.opengis.net/om/2.0</Schema>
        </Format>
      </Supported>
    </ComplexOutput>
  </Output>
  <Output>
    <ows:Identifier>layerMahalDist</ows:Identifier>
    <ows:Title>Mahalanobis distance</ows:Title>
    <ows:Abstract>Return the Mahalanobis Distance result as an OGC-WMS getMap
URL</ows:Abstract>
    <ComplexOutput>
      <Default>
        <Format>
          <MimeType>image/png</MimeType>
        </Format>
      </Default>
      <Supported>
        <Format>
          <MimeType>image/png</MimeType>
        </Format>
      </Supported>
    </ComplexOutput>
  </Output>
  <Output>
    <ows:Identifier>hri</ows:Identifier>
    <ows:Title>HRI raw value</ows:Title>
    <LiteralOutput>
      <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#float">float</ows:DataType>
    </LiteralOutput>
  </Output>

 </ProcessOutputs>
</ProcessDescription>
</wps:ProcessDescriptions>
```

B. eHabitat WPS Execute.xml


```xml
<wps:Execute service="WPS" version="1.0.0"
  xmlns:wps="http://www.opengis.net/wps/1.0.0"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
  <ows:Identifier>ehabitatprocess_bbox_multi_uncert</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>indicators</ows:Identifier>
      <wps:Data>
        <wps:ComplexData>
          <WCS_GDAL>
            <ServiceURL>http://localhost/mapserver/index.php?version=1.0.0</ServiceURL>
            <CoverageName>wc_5m_HADCM3_A2a_now_bio</CoverageName>
          </WCS_GDAL>
        </wps:ComplexData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>indicators</ows:Identifier>
      <wps:Data>
        <wps:ComplexData>
          <WCS_GDAL>
            <ServiceURL>http://localhost/mapserver/index.php?version=1.0.0</ServiceURL>
            <CoverageName>wc_5m_HADCM3_A2a_now_epratio</CoverageName>
          </WCS_GDAL>
        </wps:ComplexData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>indicators</ows:Identifier>
      <wps:Data>
        <wps:ComplexData>
          <WCS_GDAL>
            <ServiceURL>http://localhost/mapserver/index.php?version=1.0.0</ServiceURL>
            <CoverageName>wc_5m_HADCM3_A2a_now_prec</CoverageName>
          </WCS_GDAL>
        </wps:ComplexData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>aoi</ows:Identifier>
      <wps:Data>
        <wps:BoundingBoxData ows:dimensions="2" ows:crs="epsg:4326">
          <ows:LowerCorner>-9.443359375 35.64453125</ows:LowerCorner>
          <ows:UpperCorner>6.70654296875 52.05810546875</ows:UpperCorner>
        </wps:BoundingBoxData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>log</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>false</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>forecast</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>false</wps:LiteralData>
      </wps:Data>
    </wps:Input>
    <wps:Input>
      <ows:Identifier>sitePolygon</ows:Identifier>
      <wps:Data>
```

```
      <wps:LiteralData>POLYGON((-4.027099609375 42.999877929688,-4.1259765625
    42.461547851563,-3.44482421875 42.417602539063,-3.35693359375 42.933959960938,-
    4.027099609375 42.999877929688))</wps:LiteralData>
    </wps:Data>
   </wps:Input>
   <wps:Input>
    <ows:Identifier>scale</ows:Identifier>
    <wps:Data>
     <wps:LiteralData>false</wps:LiteralData>
    </wps:Data>
   </wps:Input>
   <wps:Input>
    <ows:Identifier>png</ows:Identifier>
    <wps:Data>
     <wps:LiteralData>false</wps:LiteralData>
    </wps:Data>
   </wps:Input>
   <wps:Input>
    <ows:Identifier>numRealisations</ows:Identifier>
    <wps:Data>
     <wps:LiteralData>3</wps:LiteralData>
    </wps:Data>
   </wps:Input>
   <wps:Input>
    <ows:Identifier>method</ows:Identifier>
    <wps:Data>
     <wps:LiteralData>mahalanobis</wps:LiteralData>
    </wps:Data>
   </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
   <wps:ResponseDocument lineage="false" storeExecuteResponse="true" status="false">
    <wps:Output asReference="true">
     <ows:Identifier>layerMahalDist</ows:Identifier>
    </wps:Output>
    <wps:Output asReference="false">
     <ows:Identifier>processReport</ows:Identifier>
    </wps:Output>
    <wps:Output asReference="true" mimeType="image/netCDF-U">
     <ows:Identifier>MahalDistSummaryStatistics</ows:Identifier>
    </wps:Output>
    <wps:Output asReference="true">
     <ows:Identifier>omIndicatorValues</ows:Identifier>
    </wps:Output>
    <wps:Output asReference="false">
     <ows:Identifier>hri</ows:Identifier>
    </wps:Output>
   </wps:ResponseDocument>
  </wps:ResponseForm>
</wps:Execute>
```

C. eHabitat WPS ExecuteResponse.xml


```
<wps:ExecuteResponse
 xmlns:wps=http://www.opengis.net/wps/1.0.0
 xmlns:un="http://www.uncertml.org/2.0"
 xmlns:om="http://www.opengis.net/om/2.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:sams="http://www.opengis.net/samplingSpatial/2.0"
 xmlns:sf="http://www.opengis.net/sampling/2.0"
 xmlns:gml="http://www.opengis.net/gml/3.2"
 xmlns:ows="http://www.opengis.net/ows/1.1"
 xmlns:xlink="http://www.w3.org/1999/xlink"
 xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  http://schemas.opengis.net/wps/1.0.0/wpsExecute_response.xsd h
  ttp://www.opengis.net/om/2.0 ../UncertWeb_OM.xsd"
 service="WPS" version="1.0.0" xml:lang="en-CA"
 serviceInstance="http://ehabitat-wps.jrc.ec.europa.eu/cgi-
  bin/pywps.cgi?service=WPS&request=GetCapabilities&version=1.0.0"
 statusLocation="http://ehabitat-wps.jrc.ec.europa.eu/wps/wpsoutputs/pywps-
  134805508222.xml">
 <wps:Process wps:processVersion="0.2">
  <ows:Identifier>ehabitatprocess_bbox_multi_uncert</ows:Identifier>
  <ows:Title>ehabitatprocess_bbox_multi_uncert</ows:Title>
  <ows:Abstract>
   The eHabitat process calculates the mahalanobis distance of
   various input variables (to be provided as WCS/CSW URLs) wrt. to a certain
   sample area (e.g. a protected area or arbitrary polygon) within a desired
   bounding box.
  </ows:Abstract>
 </wps:Process>
 <wps:Status creationTime="2012-09-19T13:45:06Z">
  <wps:ProcessSucceeded>
   PyWPS Process ehabitatprocess_bbox_multi_uncert successfully
   Calculated
  </wps:ProcessSucceeded>
 </wps:Status>
 <wps:ProcessOutputs>
  <wps:Output>
   <ows:Identifier>layerMahalDist</ows:Identifier>
   <ows:Title>Mahalanobis distance</ows:Title>
   <ows:Abstract>
    Return the Mahalanobis Distance result as an OGC-WMS getMap URL
   </ows:Abstract>
   <wps:Reference href="http%3A//ehabitat-wps.jrc.ec.europa.eu/cgi-
    bin/mapserv%3Fmap%3D/srv/www/htdocs/wps/wpsoutputs/wps22754-
    tmpoFsZK6.map%26SERVICE%3DWMS%26REQUEST%3DGetMap%26VERSION%3D1.1.1%26LAYERS
    %3DlayerMahalDist%26SRS%3Depsg%3A4326%26BBOX%3D-
    9.44332%2C35.6445328%2C6.706501%2C52.0581%26HEIGHT%3D197%26WIDTH%3D194%26FO
    RMAT%3Dimage/png" mimeType="image/png">
   </wps:Reference>
  </wps:Output>
  <wps:Output>
   <ows:Identifier>MahalDistSummaryStatistics</ows:Identifier>
   <ows:Title>Mahalanobis summary statistics (mean/stddev)
    Information
   </ows:Title>
   <ows:Abstract>
    Return the summary statistics (mean/stddev) for then
    Mahalanobis Distance result as uncertML or netCDF-U
   </ows:Abstract>
   <wps:Reference href="http://ehabitat-
    wps.jrc.ec.europa.eu/wps/wpsoutputs/MahalDistSummaryStatistics-
    22754dn9TXF.nc" mimeType="image/netCDF-U">
   </wps:Reference>
  </wps:Output>
  <wps:Output>
 <ows:Identifier>omIndicatorValues</ows:Identifier>
```

```
    &lt;ows:Title&gt;Indicator layer values in OM&lt;/ows:Title&gt;
    &lt;ows:Abstract&gt;
      Return the raw values of the indicator layer values inside the PA OM encoded,
      to be visualised thru the VISS client.
    &lt;/ows:Abstract&gt;
    <wps:Reference href="http://ehabitat-
      wps.jrc.ec.europa.eu/wps/wpsoutputs/omIndicatorValues-22754euOfpB.json"
      mimeType="text/xml" encoding="utf-8" schema="http://www.opengis.net/om/2.0">
    &lt;/wps:Reference&gt;
  &lt;/wps:Output&gt;
  &lt;wps:Output&gt;
    &lt;ows:Identifier&gt;hri&lt;/ows:Identifier&gt;
    &lt;ows:Title&gt;HRI raw value&lt;/ows:Title&gt;
    &lt;wps:Data&gt;
      &lt;wps:LiteralData dataType="float" uom="None"&gt;32.8269230769&lt;/wps:LiteralData&gt;
    &lt;/wps:Data&gt;
  &lt;/wps:Output&gt;
 &lt;/wps:ProcessOutputs&gt;
&lt;/wps:ExecuteResponse&gt;
```

D. Help files of the eHab R package

The following pages are from a pdf of all help files in the eHab R package. The package is also used for other purposes, and not all functions are relevant for the work within the UncertWeb project. The functions are listed alphabetically.

Package 'eHab'

October 15, 2012

Type Package

Title Computational back-end of eHabitat

Version 0.5-19

Date 2012-10-12

Author Jon Olav Skoien

Maintainer Jon Olav Skoien <jon.skoien@jrc.ec.europa.eu>

Description Computational backend of eHabitat

Depends MASS, dismo, rgdal, automap, rgeos

License GPL (>=2)

LazyLoad yes

R topics documented:

`eHab-package` *Backend for e-Habitat*

Description

The package eHab will work as the computational backend for the environmental Web Processing Service e-Habitat.

Details

Package:	eHab
Type:	Package
Version:	0.1-1
Date:	2010-06-04
License:	GPL >= 2?
LazyLoad:	yes

Author(s)

Jon Olav Skoien <jon.skoien@jrc.ec.europa.eu>

See Also

pHabitat, hri

`ecohri` *Computation of hri for multiple parks*

Description

Function for computing the habitat replacement index for a set of protected areas.

Usage

```
ecohri(ecoregions, parks, indicators, ecoID = names(ecoregions)[1],
  pvals = seq(0.05, 1, 0.05), tiffdir = "tiffs", pngdir = "nopng",
  hriRes = NULL, hriRes2 = NULL, hriInRes = NULL, hriInRes2 = NULL,
  minVar = NULL, istart = 1, tparks = 0, ecoBuffer = 1,
  wdpaid = "wdpaid", tmpfile = ".grd")
```

Arguments

`ecoregions`	`SpatialPolygonsDataFrame` with one or more subregions (typically ecoregions) within which to compute the hri.
`parks`	`SpatialPolygonsDataFrame` with one or more protected areas for which to compute the hri
`indicators`	A `stack` of the raster-package with the indicators used to estimate the similarity of the habitat.
`ecoID`	the column name of the id of the ecoregions. This is necessary as some ecoregions can be multiple polygons
`pvals`	The probability limit for the Mahalanobis distance used to decide if a pixel could be a suitable replacement for the protectedArea or not
`tiffdir`	the directory where tiff-files of the similarity will be saved
`pngdir`	the directory where png-files of the similarity will be saved
`hriRes`	a data frame with existing results for the same parks and ecoregions, to be used for a possible restart, see details below

hriRes2	a data frame with existing results for the same parks and ecoregions, to be used for a possible restart, see details below
hriInRes	a data frame with existing results for the same parks and ecoregions, to be used for a possible restart, see details below
hriInRes2	a data frame with existing results for the same parks and ecoregions, to be used for a possible restart, see details below
minVar	a function defining the minimum variance for a set of indicators
istart	which number in the list of ecoregions to start with - to be used for restarts
tparks	total number of parks so far - to be used for restarts
ecoBuffer	The buffer to use around ecoregion, in the metric of ecoregions
wdpaid	the column name of the id of the parks
tmpfile	in which format to store temporary rasters (".grd" or ".tif"

Details

The `ecohri` function is a method for computing the hri for one or more parks within one or more ecoregions. The side effect is that a similarity map is printed to a tiff-file.

Value

The function will return two or four (if `minVar` is different from `NULL`) `data.frames` in a list. `hriRes` give the estimated habitat replaceability index (hri) from the function `hri` for the protected area within the ecoregion of interest, `hriRes2` the similar value computed with a minimum variability. `hriInRes` and `hriInRes2` are the similar hri-values inside the boundaries of the park. This is mainly for explaining purposes, not so much for analysis.

Author(s)

Jon Olav Skoien

See Also

`hri`

Examples

```
## Not run:
library(eHab)
options(error = recover)
data(ecohri)
system.time(hriImage<-ecohri(ecoregions=ecoregions, parks=parks, indicators=indicators,
    tiffdir="e:/tmp/", pngdir="nopng", minVar=minVar, ecoBuffer=10, pval = 0.5))

## End(Not run)
```

`eHab-help` *eHab help functions*

Description

Some useful functions in the eHab package

Usage

```
recenterPoints(spdf, action = NA)
makePolygon(xcor = c(0,1), ycor = c(0,1), coor = NULL, pstring = NA, id = "1",
              resol = NULL, data = NULL )
scatterplot3dm(x, y=NULL, z=NULL, color=par("col"), pch=NULL,
    main=NULL, sub=NULL, xlim=NULL, ylim=NULL, zlim=NULL,
    xlab=NULL, ylab=NULL, zlab=NULL, scale.y=1, angle=40,
    axis=TRUE, tick.marks=TRUE, label.tick.marks=TRUE,
    x.ticklabs=NULL, y.ticklabs=NULL, z.ticklabs=NULL,
    y.margin.add=0, grid=TRUE, box=TRUE, lab=par("lab"),
    lab.z=mean(lab[1:2]), type="p", highlight.3d=FALSE,
    mar=c(5,3,4,3)+0.1, col.axis=par("col.axis"),
    col.grid="grey", col.lab=par("col.lab"),
    cex.symbols=par("cex"), cex.axis=0.8 * par("cex.axis"),
    cex.lab=par("cex.lab"), font.axis=par("font.axis"),
    font.lab=par("font.lab"), lty.axis=par("lty"),
    lty.grid=par("lty"), lty.hide=NULL, lty.hplot=par("lty"),
    log="", ...)
size(envir = .GlobalEnv)
```

Arguments

`spdf`	SpatialPointsDataFrame with geographical coordinates
`action`	"sub" or "add", for change from Pacific to Atlantic view, or the opposite, will be guessed from the data if not given.
`xcor`	limits of x-coordinates of polygon to be created or x-coordinate of centre of gravity
`ycor`	limits of y-coordinates of polygon to be created or y-coordinate of centre of gravity
`coor`	alternatively a matrix/data.frame of the limits, or xy-coordinates of centre of gravity
`pstring`	project4string
`id`	id for polygon
`resol`	size of polygon if centre of gravity is given instead of boundaries
`data`	data for the spatial polygons
`envir`	environment
`x`	the coordinates of points in the plot.
`y`	the y coordinates of points in the plot, optional if `x` is an appropriate structure.
`z`	the z coordinates of points in the plot, optional if `x` is an appropriate structure.

`color` colors of points in the plot, optional if `x` is an appropriate structure. Will be ignored if `highlight.3d = TRUE`.

`pch` plotting "character", i.e. symbol to use.

`main` an overall title for the plot.

`sub` sub-title.

`xlim, ylim, zlim`
the x, y and z limits (min, max) of the plot. Note that setting enlarged limits may not work as exactly as expected (a known but unfixed bug).

`xlab, ylab, zlab`
titles for the x, y and z axis.

`scale.y` scale of y axis related to x- and z axis.

`angle` angle between x and y axis (Attention: result depends on scaling).

`axis` a logical value indicating whether axes should be drawn on the plot.

`tick.marks` a logical value indicating whether tick marks should be drawn on the plot (only if `axis = TRUE`).

`label.tick.marks`
a logical value indicating whether tick marks should be labeled on the plot (only if `axis = TRUE` and `tick.marks = TRUE`).

`x.ticklabs, y.ticklabs, z.ticklabs`
vector of tick mark labels.

`y.margin.add` add additional space between tick mark labels and axis label of the y axis

`grid` a logical value indicating whether a grid should be drawn on the plot.

`box` a logical value indicating whether a box should be drawn around the plot.

`lab` a numerical vector of the form c(x, y, len). The values of x and y give the (approximate) number of tickmarks on the x and y axes.

`lab.z` the same as `lab`, but for z axis.

`type` character indicating the type of plot: "p" for points, "l" for lines, "h" for vertical lines to x-y-plane, etc.

`highlight.3d` points will be drawn in different colors related to y coordinates (only if `type = "p"` or `type = "h"`, else `color` will be used).
On some devices not all colors can be displayed. In this case try the postscript device or use `highlight.3d = FALSE`.

`mar` A numerical vector of the form c(bottom, left, top, right) which gives the lines of margin to be specified on the four sides of the plot.

`col.axis, col.grid, col.lab`
the color to be used for axis / grid / axis labels.

`cex.symbols, cex.axis, cex.lab`
the magnification to be used for point symbols, axis annotation, labels relative to the current.

`font.axis, font.lab`
the font to be used for axis annotation / labels.

`lty.axis, lty.grid`
the line type to be used for axis / grid.

`lty.hide` line style used to plot 'non-visible' edges (defaults of the `lty.axis` style)

`lty.hplot` the line type to be used for vertical segments with `type = "h"`.

`log` Not yet implemented! A character string which contains "x" (if the x axis is to be logarithmic), "y", "z", "xy", "xz", "yz", "xyz".

`...` more graphical parameters can be given as arguments, `pch = 16` or `pch = 20` may be nice.

Note

scatterplot3dm is modified from the scatterplot3d function of the package with same name, using bpy.colors instead of rgb for colors, and fixing the y-colors for repeated calls to the function with different data.

size() gives the size of all objects in the global environment

recenterPoints can recenter a SpatialPointsDataFrame from Pacific to Atlantic centred, or the opposite

makePolygon makes a square polygon based on the x and y coordinates of the extremes

Author(s)

Jon Olav Skoien

`findPatches` *Identify continuous patches*

Description

Function that identifies continuous patches of possible replacement areas for a habitat

Usage

```
findPatches(hrep, plot = TRUE)
```

Arguments

`hrep`	an `SpatialGridDataFrame` object from the function `hri` with pixels that could possible replace the pixels in the original habitat
`plot`	logical; defines whether the function should plot the patches

Value

The function returns the submitted `SpatialGridDataFrame` including a new column with IDs of patches

Author(s)

Jon Olav Skoien

hri *Habitat replacement index*

Description

Functions that calculates similarities between habitat of interest and surroundings and/or future.

Usage

```
hri(indicators = NULL, habitat = NULL, geometry = NULL, populationDensity,
  pval = seq(0,0.95,0.05), forecast = FALSE, ...)
mhri(indicators, habitat, indicators2 = NULL, forecast = FALSE,
    nsim = 1, range = NULL, exc = 0.5, nugget = 0, lmc, debug.level = 0,
    pval = seq(0,0.95,0.05), ...)
whri(habitat, geometry, indicators, forecast = FALSE, indicators2 = NULL, ...)
```

Arguments

`indicators`	A `SpatialGridDataFrame` with the indicators used to estimate the similarity of the habitat. The `data.frame` can include local variances when calling `mhri` if `forecast = FALSE`.
`habitat`	A `SpatialPolygons`* object with the boundaries of the protected area or habitat of interest, or a `SpatialPoints` object with precence locations
`geometry`	Either a `SpatialPolygons`* object with the ranges of all species of interest from a protected area (PA) or important bird area (IBA), or a `SpatialPointsDataFrame` with observations of all the species of interest from the PA or IBA
`populationDensity`	A `SpatialGridDataFrame` or a `data.frame` with the population density of the same locations as used for the indicators. If not available, the population density is set equal to 1 within the protectedArea and 0 outside.
`pval`	The probability limit for the Mahalanobis distance used to decide if a pixel could be a suitable replacement for the protectedArea or not
`indicators2`	A second `SpatialGridDataFrame` with forecasted indicators. The `data.frame` can also contain local variances of the forecast data when calling `mhri`
`forecast`	Logical; if TRUE and neither `indicators2` nor `covPoly` and `meanPoly` are not submitted, the data of the indicators will be split in two columnwise, assuming that the first part of the data frame represents the current state and the second part the forecast. If covPoly and meanPoly are submitted. If `forecast` is TRUE, also the area within the habitat will be included for computation of hri.
`nsim`	Number of simulations to use when doing Monte Carlo simulations of uncertain input. Function will call `hri` directly when `nsim = 1`
`range`	Correlation length to be used in conditional simulations, `range = NULL` gives automatic fitting of range, see details below.
`exc`	The similarity level for computation of exceedance probabilities
`nugget`	Nugget effect to be used in conditional simulations

lmc	Linear model of coregionalization, to be used in conditional simulations, see also fit.lmc for more details. The model should have sill = 1, as the local variance is given separately.
debug.level	Passed to predict.gstat when creating realizations of uncertain input
...	Variables to be passed to hri from mhri or whri, or from whri to mhri

Details

hri is called by the two other functions after some preprocessing of their input. The base of all these functions are pHabitat, which computes similarities to a region of interest from the surroundings and/or future, based on the Mahalanobis distance to the mean and variance of a set of indicators at the region of interest. See the reference below for more details.

If the function is used for forecasting, the indicators for the future can either be given in variable indicators2, or as additional columns of indicators with forecast = TRUE.

mhri is a function for estimating habitat similarity, assuming uncertainty of the indicators. The correlation of the uncertainty can either be supplied as a linear model of coregionalization (see also fit.lmc), or through nugget and range. If none are supplied, the function will estimate range from the average of ranges fitted to the sample variograms of the individual indicators with autofitVariogram.

The local variability can be given as standard deviation, using the same column names with an additional u first, so that original column names of bio, epratio and prec would give standard deviation column names of ubio, uepratio and uprec. These columns can be added to indicators2, or to indicators if indicators2 is not given and forecast = FALSE.

The function will for simplicity assume no uncertainty for the indicators used to estimate the mean and covariance at the region of interest. Instead, the Linear model of coregionalization is used to create sets of uncoditional simulations of errors, that are multiplied with the local standard deviations and added to the original set of indicators to be used for estimation of similarity.

whri is called when the mean and variance to be used for computation of habitat similarity should be based on a weighted set of indicators. The variable geometry will usually give a set of range maps or occurence observations of trigger species observed in a protected area or important bird area. Mean and variance for the region of interest is then computed as a weighted mean of the values of the indicators for these ranges/occurence locations, rather than for the area of the PA/IBA itself.

Value

hri calls pHabitat, which adds the columns mDist (Mahalanobis distance) and pHab (similarity between the indicators of pixels and their values at the locations of interest) to the indicators object. In addition, there is a column replace with values 0/1, with 1 if the pixel has a value larger than pval.

The object will also have three attributes:

attr(spdf, "hri" A data frame with the total size of the area with pHab > pval, compared to the original protected area, with one row for each value of pval. The hri is split between the hri inside and outside the park boundaries. For mhri, this attribute gives the hri for each individual simulation, in addition to the mean over simulations and the standard deviation.

attr(spdf,"mean") = meanPoly The mean values of indicators used in computation of similarity.

attr(spdf,"cov") = covPoly The covariance matrix of indicators used in computation of similarity.

mhri will also return the results of the individual realisations and some summary statistics.

whri wirll return the same results as hri if called with nsim = 1 and as whri if called with nsim > 1.

Author(s)

Jon Olav Skoien

References

Clark, J.D., Dunn, J.E. and Smith, K.G. (1993) A multivariate model of female black bear habitat use for a geographic information system. Journal of Wildlife Management, 57, 519-526.

See Also

pHabitat

Examples

```
library(eHab)
options(error = recover)
data(eHab)
hrep = hri(indicators,protectedArea, pval = 0.5)
spplot(hrep,"pHab",col.regions = bpy.colors(),
      panel = function(x,y, ...){
        panel.gridplot(x,y, ...)
        sp.polygons(protectedArea, col=2,fill=0,lw = 2)
      }
)

hrep = mhri(indicators,protectedArea, pval = 0.5, nsim = 4)
```

---

indicators *Test data sets for the eHab package*

Description

Data sets for resting and exampled in the eHab package

Usage

```
data(eHab)
```

Details

The indicators are some of the heavy metal concentrations of the Meuse data set, interpolated to a grid. protectedArea is a random square that is used to define the reference geometry.

pHabitat — *Estimate habitat replaceability*

Description

Function for estimating similarities between the indicator values at different pixels and a region of interest (protected area, important bird area or presence only observations), using the Mahalanobis distance.

Usage

```
pHabitat(indicators = NULL, habitat = NULL, meanPoly = NULL, covPoly = NULL,
  polyData = NULL, meanCovPoly = NULL, na.rm = FALSE, indicators2, forecast = FALSE,
  nonNum = NULL, method = "mahalanobis")
```

Arguments

`indicators`	A `SpatialGridDataFrame` with the indicators used to estimate the similarity to the habitat
`habitat`	A `SpatialPolygons`*-object with the boundaries of the protected area or habitat of interest, or a `SpatialPoints` object with precence locations. See also details below
`meanPoly`	Mean of the indicators at the locations of `habitat`, should only be given if already computed or if it can not be found from habitat and indicators
`covPoly`	Covariance matrix of indicators at the locations of `habitat` should only be given if already computed or if it can not be found from habitat and indicators. See details below for usage.
`polyData`	matrix or data.frame with the values of the indicators valid for the habitat. See details below for usage.
`meanCovPoly`	list of covariance matrix, mean, polyData etc from a call to the internal function eHab:::habMeanCovPoly. See details below for usage.
`na.rm`	Logical; defining whether NA-values in the indicator data set should be removed or not, or whether they should be interpolated (`na.rm = "int"`).
`indicators2`	A second `SpatialGridDataFrame` with forecasted indicators
`forecast`	Logical; if TRUE and neither `indicators2` nor `covPoly` and `meanPoly` are not submitted, the data of the indicators will be split in two columnwise, assuming that the first part of the data frame represents the current state and the second part the forecast. If covPoly and meanPoly are submitted. If `forecast` is TRUE, also the area within the habitat will be included for computation of the similarity.
`nonNum`	list of non-numerical inputs, usually from a call to ehab:::habMeanCovPoly.
`method`	Which method to use for computing similarites. The possibilities are at the moment "mahalanobis", "maxent", "bioclim" and "domain". The three last methods are mainly available through the `dismo`-package.

Details

The function computes similarities to a region of interest from the surroundings and/or future. Different methods are available, such as: "mahalanobis", "maxent", "bioclim" and "domain". The last three are available through the package `dismo`, maxent requires also installation of the java package maxent.jar in the java directory of dismo.

The habitat will normally refer to the habitat or the presence only observations of one species, or the `SpatialPolygons`*-object defining a region of interest, typically a protected area.

There are four different ways of submitting the habitat characteristics.

- indicators Let the function find the habitat characteristics from the set of indicators. A habitat must be submitted.
- meanCovPoly The result of an earlier call to `pHabitat`, stored in found as `attr(pHabImg, "meanCovPoly")` if `pHabImg` is the result of a call to `link{pHabitat}` or `link{hri}`.
- meanPoly and covPoly The means and the covariance matrix that describes the characteristics of the habitat. This is only useful if the method is "mahalanobis".
- polyData A matrix or a data.frame of the habitat characteristics

If any of the last three options are chosen, the indicators are not used. It is only possible to choose one of the last three options, combinations will cause an error.

Value

The function returns a `SpatialGridDataFrame` equal to `indicators` but extended with two columns:

`mDist`	Mahalanobis distance between the pixel and the average of the habitat/observation locations
`pHab`	The probability of a pixel being a suitable habitat

The object will also have a range of attributes:

`attr(spdf,"mean") = meanPoly` The mean values of indicators used in computation of similarity.

`attr(spdf,"cov") = covPoly` The covariance matrix of indicators used in computation of similarity.

`attr(spdf,"inHA") = inHA` An array indicating which rows from the `data.frame` of `indicators` are within the habitat

`attr(spdf,"polyData") = polyData` A data.frame with the values of the indicators for the selected habitat.

`attr(spdf, "hist") = hist` A histogram of the values of the indicators for the selected habitat

`attr(spdf, "meanCovPoly")` A list with information about the habitat characteristics, and, if necessary, information about categorical data, constant values etc.

Author(s)

Jon Olav Skoien

References

Clark, J.D., Dunn, J.E. and Smith, K.G. (1993) A multivariate model of female black bear habitat use for a geographic information system. Journal of Wildlife Management, 57, 519-526.

See Also

hri

Examples

```
data(eHab)
pp = pHabitat(indicators,protectedArea)
```

Index