

## **SELU WHITE PAPER**

### **John's guide to the GARP modelling system**

**Version 1.2**

**11/10/2004**

**John T. Lee**

Spatial Ecology & Landuse Unit (SELU), School of Biological & Molecular Sciences, Oxford Brookes University, Oxford, OX3 0BP. Tel.: (+44)(0)1865 483269 Fax.: (+44)(0)1865 483242 Email: jlee@brookes.ac.uk

#### **Abstract**

The Genetic Algorithm for Rule-set Production (GARP) algorithm was developed by David Stockwell of the San Diego Supercomputer Centre. It is at the core of software and web-based programs used for biodiversity and ecological research, which allows users to predict and analyse species' distributions. The software has a number of implementations, which are described to a greater or lesser extent. These implementations are: the GARP Modelling System, DesktopGarp and WhyWhere.

#### **Status**

The current document was originally intended to be a user guide for the GARP Modelling System, but it has become more of a guide to the various implementations of the GARP algorithm for species' prediction. The structure of the document reflects this original focus.

<b>1. INTRODUCTION.....</b>	<b>2</b>
<b>2. GARP MODELLING SYSTEM.....</b>	<b>3</b>
2.1 INSTALLATION.....	3
2.1.1 Assumptions.....	3
2.1.2 Notation used in the section .....	3
2.2 DATA PREPARATION AND INPUT.....	3
2.2.1 ArcInfo grids .....	3
2.2.2 Text files .....	4
2.3 MODELLING TOOLS.....	4
2.4 OUTPUT TOOLS .....	5
2.4.1 Image output.....	5
2.4.2 Rule output .....	5
<b>3. DESKTOP GARP .....</b>	<b>6</b>
3.1 RUNNING DESKTOPGARP .....	6
3.1.1 Installation .....	6
3.1.2 Species data.....	6
3.1.3 Optimisation parameters.....	6
3.1.4 Environmental data.....	7
3.1.5 Output parameters .....	7
<b>4. WHY WHERE .....</b>	<b>7</b>
4.1 TECHNICAL ASPECTS .....	7
4.1.1 Internet version .....	7
4.1.2 Workstation version .....	7
4.1.2.1 Installation - Windows .....	8
4.1.2.2 Usage – Windows .....	8
4.1.2.3 Installation - Linux .....	8
4.1.2.4 Usage – Linux .....	8
4.2 RUNNING WHYWHERE.....	8
4.2.1 Get point data.....	9
4.2.2 Get environmental data.....	9
4.2.3 Model output .....	9
4.2.4 Explain results.....	9
<b>5. CONCLUSION.....</b>	<b>9</b>
<b>REFERENCES.....</b>	<b>14</b>
<b>APPENDIX A .....</b>	<b>15</b>

## 1. INTRODUCTION

The Genetic Algorithm for Rule-set Production (GARP) suite of software and applications is designed primarily for predicting the potential distribution of biological entities (species). It is one such tool (others are cited in Anderson & Martinez-Meyer, 2004) that uses species distribution data collected in an unsystematic manner. GARP is a genetic algorithm that creates ecological niche models for species; the models describe the environmental conditions under which the species should be able to maintain populations. It uses point records of species distributions together with environmental parameter data

thought to limit the species' ability to survive. GARP searches iteratively for non-random correlations between species presence and absence and environmental parameter values using several different types of rules. Each rule type implements a different method for building species prediction models. Currently there are four types of rules implemented: atomic, logistic regression, bioclimatic envelope, and negated bioclimatic envelope rules. These are described in detail in Payne & Stockwell (1996). The software has developed over time and exists in (at least) three incarnations: the GARP Modelling System, DesktopGarp and WhyWhere. The remainder of this paper introduces and discusses each of these implementations

## 2. GARP MODELLING SYSTEM

This part of the paper is a user guide for the GARP Modelling System (GMS). GMS is the original implementation of GARP and runs under UNIX/Linux. Input and output is via ASCII text files and program control is through a command line interface. This renders the software difficult to use and it took six weeks to organise the data in the format required! Conversely, the software will interface easily with other applications as both input, output and control are in ASCII.

The remainder of this section is designed to complement the technical reference (Payne & Stockwell). As such, it contains no background on the programme or on the concepts of probability or genetic algorithms. It is designed to guide the user through the process of getting data into GARP and using each of the eight routines. There are four parts to the guide: (i) installation (ii) data preparation and input (iii) modelling tools and (iv) output tools.

### 2.1 Installation

GARP was installed and run on a machine with an AMD Athlon XP processor 2200+ with 526Mb RAM, running RedHat 9.0. GIS data were created using ESRI's ArcGIS ver. 8.3 running under WinXP. All commands used in the guide refer to these installations.

#### 2.1.1 Assumptions

This guide was written with the assumption that the reader has a reasonable knowledge of (i) the Unix/Linux OS and (ii) ESRI's ArcInfo/ArcGIS/Arcview suite and had read the GARP user's guide.

#### 2.1.2 Notation used in the section

Throughout the guide filenames are given double quotation marks (e.g. "file.txt"). GARP routines such as *rasteriz* appear in italics. Commands typed in at the terminal are preceded by a > indicating the machine prompt. Commands appear in the Square721 BT font. ArcInfo and Excel commands are types in UPPER CASE.

## 2.2 Data Preparation And Input

This section focuses on the *rasteriz* routine and preparation of the "paramete" file. Two alternative methods of data input were investigated, namely ArcInfo ASCII grids and text files.

### 2.2.1 ArcInfo grids

Data were export from ArcInfo as ASCII grids using the GRIDASCII command. These were passed through the *rasteriz* routine using the following command syntax, with output sent to "report.txt".

```
> cat altitude.asc | rasteriz -layer 2 > report.txt
```

This (-layer 2 option) reports on number of columns and rows, cell size, maximum and minimum data values and other parameters required in the “paramete” file. Following appropriate modifications to “paramete”, the actual conversion was undertaken (N.B. “report.txt” is overwritten).

```
> cat altitude.asc | rasteriz -layer 3 > report.txt
```

Unfortunately, this data type did not produce the expected layerXX files.

### 2.2.2 Text files

Data were either (i) exported from ArcInfo coverages using the EXPORT command to produce “\*.e00” files or (ii) exported from ArcInfo coverages using the EXPORT command in Info to produce text files. These were manipulated in Excel (using especially the IF and VLOOKUP commands) to produce tab delimited text files such that each row contains a location in X and Y and the corresponding value of an environmental variable, e.g. altitude in GB National Grid coordinates:

```
477575      189279      82
477625      189279      77
477675      189279      77
etc.....
```

The attributes of each of the variables were ascertained using the ArcInfo DESCRIBE command and “paramete” was edited accordingly. The layerXX files were created one at a time with appropriate changes to “paramete” each time.

```
> rasteriz -file altitude.txt -verbose > report.txt
```

This successfully created the layerXX files. The final stage of data preparation involved creating training and testing datasets with the *presampl* routine. The default resample number is 2500, but here 10000 were used to ‘stretch’ the software.

```
> presampl -data /home/john/garp_test/ -resamples 10000
```

**N.B.** This required creation of a ‘master’ “paramete” file with each of the environmental variables listed.

## 2.3 Modelling Tools

The remainder of the processes were relatively simple. The full “paramete” was used throughout and the location of the data files was defined with the appropriate flag after each command. A preliminary set of rules was created with the *initial* routine in the “prelim” file.

```
> initial -data /home/john/garp_test/
```

The *explain* routine uses a genetic algorithm (GA) to improve the first set of rules in “prelim”; output of the GA is directed to “output\_explain.txt”. There are numerous options for the *explain* routine (only –data and –display are used here) e.g. to direct and check output, to control level of significance, etc.

```
> explain -data /home/john/garp_test/ -display > output_explain.txt
```

These rules are then tested on the “test” dataset; the “paramete” file is required.

```
> verify -data /home/john/garp_test/ -verbose > output_verify.txt
```

The *predict* routine applies the rules to the layerXX files to produce a binary file.

```
> predict > output_predict.txt -data /home/john/garp_test/
```

## 2.4 Output Tools

Data output was in two forms, images and rules.

### 2.4.1 Image output

The output of the *predict* routine is generally directed to the *image* visualisation routine. The result of the *image* routine is a series of files of a type controlled by the command options.

```
> predict > output_predict.txt -data /home/john/garp_test/ | image -grid
```

The `-grid` flag output text files, which can be imported to ArcInfo using the ASCII GRID command. However, the final images were not revealing!

### 2.4.2 Rule output

Of more use than the images, were the rules created with the *translat* routine.

```
> translat -data /home/john/garp_test/ > output_translat.txt
```

The rules were of five types (a – atomic; d – range; m – bioclim; r – logistic; ! – negated). Using test data for box (*Buxus sempervirens*) in the Chiltern Hills, the majority of rules were logistic....

TRANSLATE - model into natural language

No-rule number, Type, Prior-prior, Post-accuracy, Sig-Significance, Cov-coverage, Use-usage

No	Type	Prior	Post	Sig	Cov	Use
1	r	0.52	0.77	20.33	0.66	0.508

```
IF - alt*38 m + alt*127 m^2 + lu*21 c + lu*127 c^2
- soil*103 c + soil*127 c^2 + aspect*123 degree + aspect*127 degree^2
+ slope*30 degree + slope*6 degree^2 + aw*120 c - aw*0 c^2
THEN box=PRESENT
```

This output was confusing for a number of reasons:

- The values of the significance and the coefficients appeared to be 100 times larger than their likely values. This was modified in the regression equation (see below).
- In many cases there were two values of the coefficient for each independent variable, whereas only one would normally be expected.
- The repeat instance of the independent variable, resulted in units of the second instance raised to the power of 2 (e.g.  $\text{alt} * 127 \text{ m}^2$ ).

This rule was input to ArcInfo grid to output a predictive spatial model for box using the standard form of the logistic regression equation:  $\text{prediction} = 1 \text{ div } (1 + (\exp(- (a_0 + a_1x_1 + a_2x_2 + a_3x_3 + \dots))))$ .

&do

```
box_okr1 = 1 div (1 + (exp (- (0.233) + (garp_alt * -0.38) + (garp_alt * 0.127) + ~
```

```
(garp_lu * 0.21) + (garp_lu * 0.127) + (garp_soil * -0.103) + (garp_soil * 0.127) + ~
(garp_asp * 0.123) + (garp_asp * 0.127) + (garp_slo * 0.3) + (garp_slo * 0.6) + ~
(garp_aw * 0.12) + (garp_aw * -0)))
&end
```

Other rule types were easier to understand, namely bioclim/range rules rules and atomic rules.

```
No   Type  Prior Post  Sig   Cov   Use
20   m     0.50  0.86  17.76 0.24  0.041
IF   alt=[63,232]m AND lu=[-2,310]c AND soil=[-1,-1]c
     AND aspect=[-2,299]degree AND slope=[ 4,25]degree AND aw=[-0, 1]c
THEN box=PRESENT
```

```
No   Type  Prior Post  Sig   Cov   Use
25   a     0.49  0.82  15.38 0.22  0.000
IF   alt=199 AND lu=316 AND soil=57
     AND aspect=360 AND slope=59 AND aw= 1
THEN box=PRESENT
```

Using test data for box (*Buxus sempervirens*) in the Chiltern Hills, all the rules predicted the presence of the species, as the data were presence only.

### 3. DESKTOP GARP

DesktopGarp is a desktop version of the GARP algorithm, which runs on Intel architectures running Microsoft Windows 98 or newer. In addition to a revised WIMP interface, there is also facility for integration with ESRI's ArcView with Spatial Analyst extension for data preparation and analysis of resulting distribution maps. This part of the paper is aimed at complementing DesktopGarp users manual (Anonymous, 2004), which is comprehensive and needs little elaboration.

#### 3.1 Running DesktopGarp

There are five aspects to running DesktopGarp: installation, species data, optimisation parameters, environmental data and data output.

##### 3.1.1 Installation

Installation was straightforward, being a download from the lifemapper website (<http://www.lifemapper.org/desktopgarp/Default.asp?Item=2&Lang=1> ).

##### 3.1.2 Species data

The interface allows full control and saving of the model run parameters. Species data are imported as comma delimited, MS Excel spreadsheets or ArcView point shapefiles. The list box displays all species loaded and the number of data points for each one and the user can control which species will be used in the experiment. Additionally, the user can specify the proportion and/or the minimum number of points used for training.

##### 3.1.3 Optimisation parameters

The user can control the model run parameters via the optimisation parameters panel. These include the runs per experiment (typically 10 per species), convergence limits (usually between 0.01 and 0.1), the maximum number of iterations (usually between 100 and 1000) and the rule types to be used. There is a running total of the total number of runs.

### **3.1.4 Environmental data**

The environmental layers panel controls the environmental data used in the experiment. Data are input as ASCII grids and the data can be used in a range of user-defined combinations. It is possible to define datasets or collections of data organised according to a common theme e.g. geographical area. These datasets are defined with the companion Dataset Manager software.

### **3.1.5 Output parameters**

The output panel defines the output prediction map format (bitmaps, ASCII grids or ARC/INFO grids) and the destination directory for the results.txt file (explained in detail in the users' manual).

## **4. WHY WHERE**

The WhyWhere implementation of GARP was originally web-based, although there now a desktop implementation. It evolved from the Biodiversity Species Workshop amid the realisation that integration and analysis can add to the research value of databases. However, the Biodiversity Species Workshop had certain limitations, principally the lack of associated environmental datasets and slow algorithm processing speed. WhyWhere has addressed this with the development of a set of biologically relevant data, albeit with a North American and/or global focus. These data are in raster format and include remote sensed scenes and their derivatives (e.g. NDVI), climate and topography. Currently there are over 400 datasets available.

As with other implementations of GARP, WhyWhere allows a user to enter the latitude and longitude of a set of locations (say species locations) and obtain a predictive map, a set of GIS-ready raster data potential explanations for the distribution of the points.

### **4.1 Technical Aspects**

The GARP algorithm used in WhyWhere uses image-processing technologies to increase the processing speed (further details in Stockwell, 2001). In addition, there is the option of using multiple (i.e. parallel) clustered processors to run the algorithm, with significant predicted improvements to run time and the potential of using grid computing. There are two versions of WhyWhere: Internet version and workstation version.

#### **4.1.2 Internet version**

The Internet version of WhyWhere uses a cgi-controlled client-server model, which is accessed at <http://landscape.sdsc.edu/cgi-bin/EIC/login.cgi>.

#### **4.1.2 Workstation version**

The workstation version of WhyWhere runs on a local machine, which is bundled with a simple web server. Software, data and manual are downloaded to the local machine from the Biodi group website at the San Diego Supercomputer Center ([http://biodi.sdsc.edu/ww\\_home.html](http://biodi.sdsc.edu/ww_home.html)). The original version was developed on Linux and used the GD library, but this no longer supports GIF images. Version 2.0 of the software supports .png format files and is portable to both Windows and Linux.

#### 4.1.2.1 Installation - Windows

There are a number of pre-requisites that be met prior to local installation of the WhyWhere software:

- A local server, which can be the default windows server IIS or a version of Apache; the software is bundled with a lightweight server, T.
- Perl for windows (e.g. active Perl for Windows at [www.activestate.com](http://www.activestate.com)). At the time of writing, version 5.6 is preferred as there is no GD module for version 5.8 (see below).
- The GD module, a graphics library, which allows programming code (e.g. PHP, Perl, Tcl or Pascal) to draw images with lines, arcs, text, multiple colours, cut and paste from other images, and flood fills, and export to PNG or JPEG files. To install GD, download GD.zip (<http://www.boutell.com/gd/>), unzip the zip file (GD.zip) into a directory, go to that directory, and type ppm install GD.ppd or install GD from a remote site, e.g. The University of Winnipeg (ppm install <http://theoryx5.uwinnipeg.ca/ppmpackages/GD.ppd>).
- A variety of additional libraries and utilities: *netpbm* - utilities for primitive manipulation of graphic images such as conversion, magnification and cropping (<http://sourceforge.net/projects/netpbm/>); *libpng* - a png reference library (<http://www.libpng.org/pub/png/libpng.html>) and *Gnuplot* - a portable, multi-platform, command-line driven interactive datafile (text or binary) and function plotting utility (<http://www.gnuplot.info/download.html>).

To ease installation all paths are relative. To install WhyWhere unzip WW\_source.zip and WW\_data.zip into the same directory, then modify wmmspath.conf file so that \$PGM path variable points to the directory the netpbm graph utilities is in (\$dir\bin, where \$dir is the installation directory of netpbm) and the \$Gnup path variable points to where GNU plot is installed. Finally, in WMS\winpgm.conf, modify \$PGM\_DIR variable to the netpbm installation directory.

#### 4.1.2.2 Usage – Windows

To execute the system, click on t.bat to launch 3 sessions of T, then open Start\_WW.html; the interface in the web browser appears exactly as at the remote site.

#### 4.1.2.3 Installation - Linux

As with the Windows version, there are a number of pre-requisites that be met prior to local installation of the WhyWhere software:

- A local server, which can be the default Linux Apache server; the software is bundled with a lightweight server, T.
- Install Perl for Linux, which is a standard component of Linux system.
- Install GD module, comprising both the GD library and Perl GD module. These need to be decompressed, then a ‘makefile’ constructed and run (requiring the following commands: “perl Makefile.PL”; “make”; “make test”; “make install”).
- Install netpbm, libpng and Gnuplot, which should be standard component of Linux. If missing, type “make install” or “rpm -i” to install rpm packages.

#### 4.1.2.4 Usage – Linux

As the Windows version, unzip the zip files and modify the corresponding path variables. Usage is the same as that as Windows.

## 4.2 Running WhyWhere

This section is intended to complement the WhyWhere tutorial (Stockwell, 2001), the on-line demonstration (<http://biodi.sdsc.edu/Doc/WhyWhere/demo.html>, although this refers to a older version



of the software) and the WhyWhere version 2.0 portal (<http://landscape.sdsc.edu/cgi-bin/EIC/login.cgi>). There are further details for system developers in Appendix A.

Analysis is started with creation of a temporary working directory accessed via guest logon; more options, such as access to supercomputing facilities, are available with suitable passwords. Version 2.0 of the software guides the user through the model run via buttons used to initiate each part: get point data, get environmental data, do prediction and explain results. Each stage is accompanied by explanations, the verbosity of which is user controlled.

#### **4.2.1 Get point data**

Running the software is straightforward with the potential of using the following: test data, existing species data (e.g. data returned from the SpeciesAnalyst via a search of museum databases) or pasting user data in the appropriate format.

#### **4.2.2 Get environmental data**

The environmental data are organised in subsets according to both theme and most frequently used. There are currently more than 400 data layers with additional datasets planned, as follows:

Currently available:

- Global ecosystems database – resolution of between 1deg and 1km, the data includes topographic (DEM), atmospheric, climatic, meteorological, hydrological, oceanographic, ecosystems, biogeochemical dynamics, geological and geophysical themes; 10GB in total;
- Continuous fields data – resolution of 1km and includes %cover of tree-cover of evergreen, deciduous and broadleaf species; 4GB in total;
- Marine – a range of resolutions including productivity, annual temperatures and deviations and salinity.

Planned:

- Satellite imagery – resolution of 250m to 1km including MODIS 12 level 3-4 land and 15 level 3-4 ocean products; 1TB to 4TB per annum;
- Digital elevation - 30m resolution showing minimum, maximum and median elevation, slope, aspect, rugosity and hydrological variables; 1TB.

Data with the same resolution and extent are extracted according to the extent of the species data or user-specified limits.

#### **4.2.3 Model output**

The system uses a dynamic surrogate algorithm for developing the models (further details in Stockwell, 2001). The user can modify some of the model parameters.

#### **4.2.4 Explain results**

The model output is in the form of explanation and statistical figures (histograms of the frequency of presence and background points), Arc/Info grids, model images (the first variable as red, the second as blue and the third as green) or images of the prediction surface.

## **5. CONCLUSION**

The GARP algorithm offers considerable potential for modelling predicted species distributions, as it uses data typically available to the ecologist/GIS practitioner. These data are typically limited, are

generally present only and are collected in an unsystematic and *ad hoc* manner. There are several implementations of GARP (arguably reflecting the development of computing) characterised by different means of model input, control and output; these are summarised in Table 1.



Implementation	Input	Control	Output	Notes
GARP Modelling System	<ol style="list-style-type: none"> <li>1) ArcInfo grids exported to ASCII grids and passed through <i>rasteriz</i> routine to create layerXX files; unsuccessful in testing.</li> <li>2) ArcInfo coverages exported to ASCII and manipulated in Excel to produce Tab delimited ASCII files with 'X Y attribute' format</li> </ol>	Commands issued at the command line prompt or through a simple ASCII batch file.	<ol style="list-style-type: none"> <li>1) Output of images via the <i>image</i> routine in a variety of formats such as ASCII grid, .miff, .bmp or .pnm.</li> <li>2) Output of IF... THEN rules in ASCII with the <i>translat</i> routine.</li> </ol>	Run very quickly on Red Hat 9.0; no windows version.
DesktopGARP	<ol style="list-style-type: none"> <li>1) Species data input as comma delimited, Excel spreadsheet or ArcView point shapefiles.</li> <li>2) Environmental data input as ASCII grids or ArcView (with Spatial Analyst extension) point shapefiles. Data layers can be organised in datasets.</li> </ol>	Via form type controls (list boxes, check boxes, radio buttons etc.) on the software window panels.	<ol style="list-style-type: none"> <li>1) Model run statistics output as .txt file easily imported to Excel.</li> <li>2) One map for each run output as ASCII grid and bitmap.</li> </ol>	Run on Windows XP Pro – quite slow. No Mac or Linux/UNIX version. Final prediction map could be the result of overlaying maps of each model run.
WhyWhere	<ol style="list-style-type: none"> <li>1) Species data from the Species Analyst search of museum databases or copy and paste in 'decimal longitude decimal latitude' format – one record per row.</li> <li>2) Environmental data selected from pre-prepared datasets. May be possible to add customized datasets.</li> </ol>	<ol style="list-style-type: none"> <li>1) Control of model parameters via form type controls (list boxes, check boxes, radio buttons etc.) on web site.</li> <li>2) Control of model progress via live links to get options and buttons after selecting options to execute each step.</li> </ol>	Output as statistical values, ArcInfo grid format or bitmaps (.png)	<ol style="list-style-type: none"> <li>1) Internet version runs via web browser on standard platform (tested with IE 6 on WinXP Pro).</li> <li>2) Localhost version runs on Windows and UNIX/Linux (tested on WinXP Pro).</li> </ol>



## REFERENCES

Anderson, R. P. & Martinez-Meyer, E. (2004). Modeling species' geographic distributions for preliminary conservation assessments: an implementation with the spiny pocket mice (*Heteromys*) of Ecuador. *Biological Conservation* 116 167-179.

Anonymous (2004). DesktopGarp users manual.  
<http://www.lifemapper.org/desktopgarp/Default.asp?Item=3&Lang=1&Print>

Payne, K. & Stockwell, D.R.B. (1996?). GARP Modelling System User's Guide and Technical Reference. <http://biodi.sdsc.edu/Doc/GARP/Manual/manual.html>. Accessed 11th June 2004.

Stockwell, D. (2001). WhyWhere - answering the question "Where is it and why?" on a global scale DRAFT version. <http://biodi.sdsc.edu/Doc/WhyWhere/tutorial.html>. Accessed October 2004.

## APPENDIX A

From C:\Program Files\WhyWhere\_GARP\WW\Doc

### WhyWhere system developer's overview

=====

by Haowei Liu, [hwhliu@cs.ucsd.edu](mailto:hwhliu@cs.ucsd.edu)

#### Contents

=====

I. System Introduction

II. Directory files overview.

#### I. System Introduction

=====

The WhyWhere system is developed by Dr. David Stockwell at San Diego Super Computer Center. This system aims at providing global prediction and explanation of biodiversity data. With WhyWhere, a user can specify the locations on the earth's surface or marine environment and obtain a predictive map and possible explanation for the distribution of the species. For more information, check [http://biodi.sdsc.edu/ww\\_home.html](http://biodi.sdsc.edu/ww_home.html)

#### II. Directory files overview

=====

The main purpose of this document is to provide an overview of /var/www/cgi-bin directory which contains all the source code of WhyWhere program. Below I will briefly describe each directory under /var/www/cgi-bin. I will talk about what they are and the functionalities they have. Hopefully, future developers can have a clear view about the program after reading this document. The .pl and .cgi files are under /var/www/cgi-bin directory, while the official web pages are under /var/www/html.

##### ***EIY/EIC:***

This directory contains all the source code of the old version of WhyWhere. There are six major steps of the system, and each corresponds to a .cgi file under this directory.(login.cgi, start.cgi, GetPoints.cgi, GetGIS.cgi, Predict.cgi, and Explain.cgi) The functionality of each file can be easily seen by its name.

##### *login.cgi:*

Display a login interface to authenticate users.

##### *start.cgi:*

Perform user authentication. If a user successfully login, create a temporary file folder for this session to store the query result and display the query interface. However, at this stage, no password matching is performed. That is, anyone can log onto the system.

##### *Get\_points.cgi:*

Provide an interface for the users to select and submit the species location. The system supports four options. The user can select a default location, a prepared data set, or upload their location file or enter the point manually.

*Get\_GIS.cgi:*

Provide an interface for the users to select the source of environmental data. The user can select from the prepared data set and specify the border of data points and the resolution.

*Predict.cgi:*

Perform the main algorithm for prediction of the species distribution.

*Explain.cgi:*

Generate the distribution map. All the temporary query results generated at each step will all be stored in the temporary folder created in the first stage. The previous version of WhyWhere is a standalone system, meaning that the whole program is responsible for both interface design and computation. The following versions of WhyWhere are trying to separate presentation and logic part of the system. In this case, the system is divided into two components, the presentation module and the logic module. The presentation module, which is a client of logic module, is responsible for getting user input, delivering the data to the logic module, and displaying the result. On the other hand, the logic module is only responsible for computing and predicting the possible distributions according to the user queries. The separation makes it possible to execute the two modules on two different machines, and thus increase the flexibility of developing.

***EIC , WMMS & wmms.cgi***

As EIY, EIC contains all the CGI files EIY has. However, the computation has been removed and all these files do is just provide a UI to get user request, deliver the request to the server, that is the logic module, and display the results returned by the server. So, how does a client connect to the server? Well, after getting the user request at each step, the client will issue a POST command to `wmms.cgi`, telling which stage the user is in, then, according to the information, `wmms.cgi` will invoke the proper function calls under WMMS directory to perform the computation. You can see something like `whywhere.cgi.Explain` under WMMS, these are functions the server may invoke.

***WMS***

This is a simple client conforms to OGC standard (Open GIS Consortium). It is used to evaluate the speed of visualizing large images. `index.cgi` and `wms.cgi` contain the code to implement the test. You can try the following link to see how the program works:

<http://landscape.sdsc.edu/cgi-bin/WMS/index.cgi>

***SOAP-WMMS***

This one is still under development. It tries to make the web service SOAP compatible. SOAP, Simple Object Access Protocol, is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is XML based. For more information, check the link <http://www.w3.org/TR/soap/>.

***Others***

***IVO***

This directory contains all the source code, which renders the distribution map. `queryhome.cgi` provides an interface that the user can use to specify the map and the resolution, while `list.cgi` lists all the maps available on the server. They all need to invoke `render.cgi` to render the images.

***MARINE***

This directory contains code to demonstrate a simple example of prediction for Global distribution for Deepwater Scorpion Fish.