Week 10 – Geographic diversity: analysis of fossil occurrences

Overview

In this assignment you will produce several habitat distribution models (aka, ecological niche models or species distribution models) for your taxa:

1. Model for modern world based on point occurrences
2. Model for modern world based on geographic range map
3. Model for last glacial maximum (LGM, 21 kya) using CCSM paleoclimate model data
4. Model for last glacial maximum using MIROC 3.2 paleoclimate model data

Data from Last Week’s Assignment

You should now have the following data in MySQL:

1. Individual occurrence points for your species and climate data for each occurrence point
2. Geographic range for your species resampled using 50 km grid points
3. Climate, elevation, and other data for the 50 km grid points (the “equidistantpoints” database)

New Data: Paleoclimate

You will need to download and install the paleoclimate data set for the Late Pleistocene from http://mypage.iu.edu/~pdpolly/Data.html. These data are in SQL format. In principle, you should be able to import them without creating a new database first. If the import fails, try creating a database called “paleoclimate” and import into it.

About the Data

All of the climate, terrain, and other environmental data in your new databases have been sampled using the same scheme of points spaced 50 km apart across the entire globe. The point IDs can be used to link these data together. Nearly all the tables in the equidistantpoints and paleoclimate databases have a field called “GlobalID”, which is the ID number of a 50 km grid point.
Latitude and Longitude

The latitude and longitude of all the 50 km grid points are stored in the table equidistantpoints.global50kmsquares. Note that this is a large data set, with 205,212 points in total. The latitudes and longitudes have been broken down by continent as well.

Modern Bioclimatic Variables

Modern bioclimatic variables, including temperature and precipitation, are available for each 50 km grid point in the table paleoclimate.modernworldclimbiovars. This table contains nineteen bioclimatic variables that are often used in habitat distribution modelling (Hijmans et al., 2005). These variables are defined in the table paleoclimate.BioClimVars.

Quaternary Bioclimatic Variables

Past bioclimatic variables are available for the 50 km grid points in the other tables in the paleoclimate database. These include the middle Holocene (about 6,000 years ago), the last glacial maximum (about 21,000 years ago), and the last interglacial (about 125,000 years ago). All of these data were generated by general circulation paleoclimate modelling, which means they are estimates of the Earth’s climate based on a series of simplifications and assumptions. They should not, therefore, but used uncritically. Details of the paleoclimate models can be found on the download page (http://mypage.iu.edu/~pdpolly/Data.html).

Elevation

Modern elevation in meters above sea level is available for each of the 50 km grid points. These data can be used to create a shaded relief map by plotting the points with the color set according to elevation. These data were resampled from the TerrainBase data set provided by the National Geophysical Data Center.

Macravegetation Cover

Modern historical macrovegetation cover is available for North American and Eurasian 50 km points. These data are from Matthews, 1984. "Prescription of Land-surface Boundary Conditions in GISS GCM II: A Simple Method Based on High-resolution Vegetation Data Sets".

Joining Data from Different Tables in SQL

You can join data in these tables using the GlobalID field. For example, you might want data on mean annual temperature (MAT) for Africa. The 50 km points for the African continent are stored in the table equidistantpoints.africa50kmpoints and the MAT data are stored in paleoclimatedata.modernworldclimbiovars. Each of these tables has a GlobalID field that contains the ID number of the associated points. You can link these data using a JOIN statement in SQL. The following is SQL to retrieve latitude, longitude, and mean annual temperature for Africa.
SELECT c.GlobalID, a.latitude, a.longitude, c.BIO1
FROM paleoclimatedata.modernworldclimbiovars c
JOIN equidistantpoints.africa50kmpoints a
ON c.GlobalID=a.GlobalID

There are three important new features of a JOIN statement:

First is that table names are specified more completely than we have done before. In this example
the two tables are in different databases, so the name of the table follows the name of the database
separated by a “.”. Each table is also given a short nickname because we need to specify a table
each time we mention a field by name. I used “c” as the nickname for the modernworldclimbiovars
table and “a” as the nickname for the africa50kmpoints table. I could have chosen any nickname.
Notice that these nicknames are used to specify the variables in the SELECT statement. GlobalID
and BIO1 come from the c table, and latitude and longitude come from the a table.

Second is that two tables are mentioned, separated by a JOIN statement. You can join more than
two tables, but the syntax is more complicated. The following page gives several examples of how
this can be done: http://stackoverflow.com/questions/3709560/mysql-join-three-tables (We’ll use
a triple join later in this exercise).

Third is that the variable used to join the tables is specified with an ON statement. Here we are
matching points in the two tables by their GlobalID number, so we join on c.GlobalID=a.GlobalID.

Creating Climate Envelopes and Habitat Distribution Models

In this exercise, we will use the simplest kind of distribution modelling, the “Bioclim” method, which
is also known more generically as the “rectilinear envelope” method. Other methods such as GARP
and MaxEnt are available, but they are less appropriate for paleoclimate modelling because they
make use of correlations between climate variables in the modern world that may not exist in the
past.

First thing is to estimate the climate envelope for your species. To do this, you simply need to find
the values of the climate variables every place the species is found, from which you find the
maximum and minimum of each one. This envelope describes the range of climate found across the
distribution of the species. In principle, finding the envelope is easy because you simply select the
points from the species range and find the max and min of each variable.

Second thing is the find the geographic points whose climate falls between the max and min values
for the species. In principle, this is also pretty easy.

Simple habitat model with 1 climate variable

To start, let’s try a simple example where we only use mean annual temperature (MAT, BIO1) for
modelling. To find the envelope using the 50 km points you created from the range map, you would

1. Select all the points related to your species
2. Join them to the climate data using GlobalID
3. Find the max and min of Bio1
SELECT MIN(BIO1), MAX(BIO1)
FROM g563.50kmcoyotes s JOIN paleoclimatedata.modernworldclimbiovars c
ON s.GlobalID=c.GlobalID

You should get a minimum and maximum value for MAT for your species. If you do this in Mathematica, you can keep track of the results:

```
conn = OpenSQLConnection["MySQL"];
envelope = SQLExecute[conn, "SELECT MIN(BIO1), MAX(BIO1) FROM g563.50kmcoyotes s JOIN paleoclimatedata.modernworldclimbiovars c ON s.GlobalID=c.GlobalID;"];
CloseSQLConnection[conn];
```

The variable `envelope` should now contain the maximum and minimum for BIO1.

To select the geographic points with MATs within this range, you choose one of the continents (or the global data) and select with a WHERE clause that makes use of the two max min values from the previous selection:

```
SELECT e.Latitude, e.Longitude
FROM equidistantpoints.na50kmpoints e JOIN paleoclimatedata.modernworldclimbiovars c
ON e.GlobalID=c.GlobalID
WHERE c.BIO1>=-16.23 AND c.BIO1<=29.2
```

In Mathematica you can read the values directly from the `envelope` variable:

```
conn = OpenSQLConnection["mySQL"];
habitatmodel = SQLExecute[conn, "SELECT e.Longitude, e.Latitude FROM equidistantpoints.na50kmpoints e JOIN paleoclimatedata.modernworldclimbiovars c ON e.GlobalID=c.GlobalID
WHERE c.BIO1>" <> ToString[envelope[[1, 1]]] <> " AND c.BIO1<" <> 
  ToString[envelope[[1, 2]]] <> 
  "]"];
CloseSQLConnection[conn];
```

Latitude and Longitude can be plotted as x,y values to obtain a Mercator projection of the habitat distribution model:

```
ListPlot[habitatmodel, AspectRatio->Automatic]
```

![Map](image)
This map shows all the places in Eurasia where the MAT falls within the range of my species (*Sorex araneus*). You might want to show this map along with the places in Eurasia that don’t fall in the model. To do this, simply retrieve all the points for Eurasia and plot them with the model:

```mathematica
conn = OpenSQLConnection["mySQL"];
na = SQLExecute[conn, 
  "SELECT Longitude, Latitude  FROM equidistantpoints.na50kmpoints ;"];
CloseSQLConnection[conn];

ListPlot[{na, habitatmodel}, AspectRatio -> Automatic]
```

Habitat model with all 19 climate variables

You can do the same thing by adding all the climate variables to the SQL statements. Because there are 19 of them, you can use Mathematica to help. First get the names of the climate variable fields from the BioClimVars table in the `paleoclimatedata` database. The second column contains the names of the bioclim variable fields in the database, the other columns contain descriptions of each variable:

```mathematica
conn = OpenSQLConnection["mySQL"];
biovars = SQLExecute[conn, "SELECT * FROM paleoclimatedata.BioClimVars"];
CloseSQLConnection[conn];

Without going into the details, the following line takes the second column of `biovars` and converts it into SQL for taking the min and max for each variable. (NOTE: I used the nickname “c” for the table containing the climate variable. If you use a different nickname, then change this code.)

```mathematica
cmds = StringJoin[Riffle["Min(c." <> # <> ") , Max(c." <> # <> ")" & /@ biovars[[1 ;;, 2]], ", "]]
```

This long string can be inserted into the SQL statement we used to calculate the envelope above:

```mathematica
conn = OpenSQLConnection["MySQL"];
envelope = SQLExecute[conn, 
  "SELECT <\$cmds\$ FROM g563.50kmcoyotes s JOIN
```
paleoclimatedata.modernworldclimbiovars c ON s.GlobalID=c.GlobalID;
CloseSQLConnection[conn];

It is useful to partition the envelope into min/max pairs:

envelope = Partition[envelope[[1]], 2];

The variable envelope should now contain the max and min values for each of the nineteen climate variables for your species. If you’d like a nice table for your species, try the following TableForm[] function using the third column of the biovars as row labels and “min” and “max” as column labels:

TableForm[envelope, TableHeadings -> {biovars[[1 ;;, 3]], {"Min", "Max"}}]

<table>
<thead>
<tr>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Ann Temp</td>
<td>-26.4583</td>
</tr>
<tr>
<td>Mean Diurnal Range</td>
<td>4.1</td>
</tr>
<tr>
<td>Isothermality</td>
<td>9.4527</td>
</tr>
<tr>
<td>Seasonality (Temp)</td>
<td>37.9867</td>
</tr>
<tr>
<td>Max Temp Warm Month</td>
<td>-1.4</td>
</tr>
<tr>
<td>Min Temp Cold Month</td>
<td>-47.</td>
</tr>
<tr>
<td>Ann Temp Range</td>
<td>8.4</td>
</tr>
<tr>
<td>Mean Temp Wet Quart</td>
<td>-14.9667</td>
</tr>
<tr>
<td>Mean Temp Dry Quart</td>
<td>-43.4833</td>
</tr>
<tr>
<td>Mean Temp Warm Quart</td>
<td>-5.2667</td>
</tr>
<tr>
<td>Mean Temp Cold Quart</td>
<td>-43.4833</td>
</tr>
<tr>
<td>Ann Precip</td>
<td>54.</td>
</tr>
<tr>
<td>Precip Wet Month</td>
<td>8.</td>
</tr>
<tr>
<td>Precip Dry Month</td>
<td>0.</td>
</tr>
<tr>
<td>Seasonality (Precip)</td>
<td>6.6047</td>
</tr>
<tr>
<td>Precip Wet Quart</td>
<td>22.</td>
</tr>
<tr>
<td>Precip Dry Quart</td>
<td>0.</td>
</tr>
<tr>
<td>Precip Warm Quart</td>
<td>2.</td>
</tr>
<tr>
<td>Precip Cold Quart</td>
<td>4.</td>
</tr>
</tbody>
</table>

Mean Ann Temp 29.2042
Mean Diurnal Range 21.2667
Isothermality 86.8056
Seasonality (Temp) 1741.31
Max Temp Warm Month 43.2
Min Temp Cold Month 22.5
Ann Temp Range 56.7
Mean Temp Wet Quart 32.5333
Mean Temp Dry Quart 28.7167
Mean Temp Warm Quart 33.3333
Mean Temp Cold Quart 27.5333
Ann Precip 4860.
Precip Wet Month 865.
Precip Dry Month 184.
Seasonality (Precip) 137.349
Precip Wet Quart 2173.
Precip Dry Quart 661.
Precip Warm Quart 1304.
Precip Cold Quart 1567.

Without going into the details, the following code converts this envelope into a WHERE statement:

whereclause = StringJoin[Riffle[Table[biovars[[x, 2]] <> "=" <> ToString[envelope[[x, 1]]] <> " AND " <> biovars[[x, 2]] <> "<=" <> ToString[envelope[[x, 2]], {x, 19}], " AND "]]

Now you can insert the whereclause into the same code we used before:

conn = OpenSQLConnection["mySQL"];
habitatmodel = SQLExecute[conn,
  "SELECT e.Longitude, e.Latitude FROM equidistantpoints.eurasia50kmpoints
e JOIN paleoclimatedata.modernworldclimbiovars c ON e.GlobalID=c.GlobalID
WHERE "<<whereclause>>";
CloseSQLConnection[conn];

Now you have all the points for the habitat model based on all the bioclimatic variables. Plot these again using the same code as before:

ListPlot[{na, habitatmodel}, AspectRatio -> Automatic]
Assignment

For each of the following data sets, generate a climate envelope and habitat distribution model:

1. Modern Climate
2. Last Glacial Maximum CCSM3 data
3. Last Glacial Maximum MIROC 3.2 data
4. Last Interglacial CCSM3 data

Answer the following questions:

1. How different are the two models for the last glacial maximum?
2. Why? And do you think one might be a better estimate for your species than the other? How could you test if you had the appropriate data?
3. How different is the last interglacial from the modern distribution? To what extent do you think it is a good model and how would you test it?
4. To what extent has the geographic range of your species been altered by glacial/interglacial cycles?

References