Inferring Population Structure in R

Kevin Keenan

22 April 2015

Introduction

Population structuring, the non-random distribution of genetic variation among individuals of a species, can be notoriously difficult to detect and characterise. In this practical we will attempt to characterise the genetic structuring among 444 brown trout, sampled from 25 seperate sites along three tributaries of a small river system. We will start with a standard PCA approach, making use only of genetic data, eventually building to sPCA and MEMGENE analysis, where we will use spatial data to refine any genetic structuring. Below is a map of our study system.

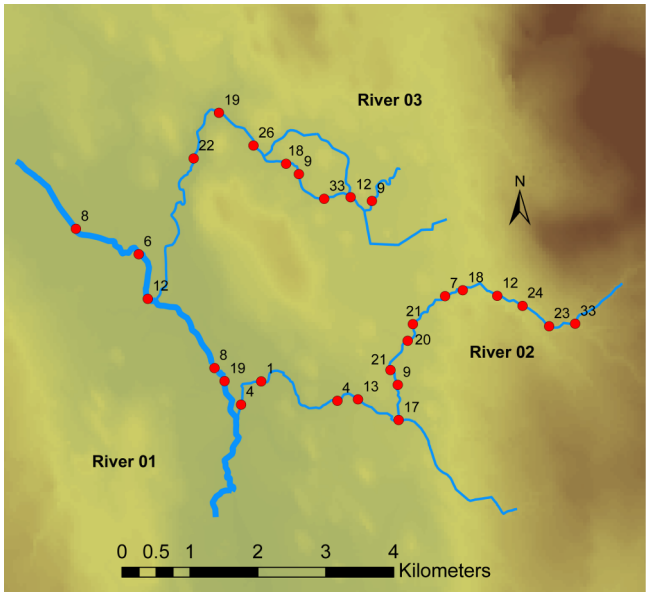


Figure 1: A map of our brown trout study system. Red circles show sampling locations, while adjescent numbers show sample sizes per site

You will notice that some sample sites have very small numbers, and overall our sampling is relatively high resolution. This is so because the brown trout, being a salmonid fish exhibits strong homing behaviour, meaning that population structure can arise over very small spatial scales. Therefore, to fully characterise the total genetic diversity and structuring of trout in this system, psudo-continuous sampling was employed.

This sampling design, while very good for understanding the total diversity within a given area, also presents a difficult analytical challenge. Clearly each site is not expected to represent a population sample, and in any case, some sample sizes per site would not be representative. As such, it would be meaningless to attempt to characterise populations based on the samples here (e.g. how relevent is allelic richness calculated per site sample?). The correct approach to analysing such data is to infer populations, and then characterise those populations (e.g. HWE, differentiation etc. . .).

The first attempt we will make to do this will be using principal component analysis (PCA), implemented in the `adegenet` package.

Practical 3: PCA of brown trout data

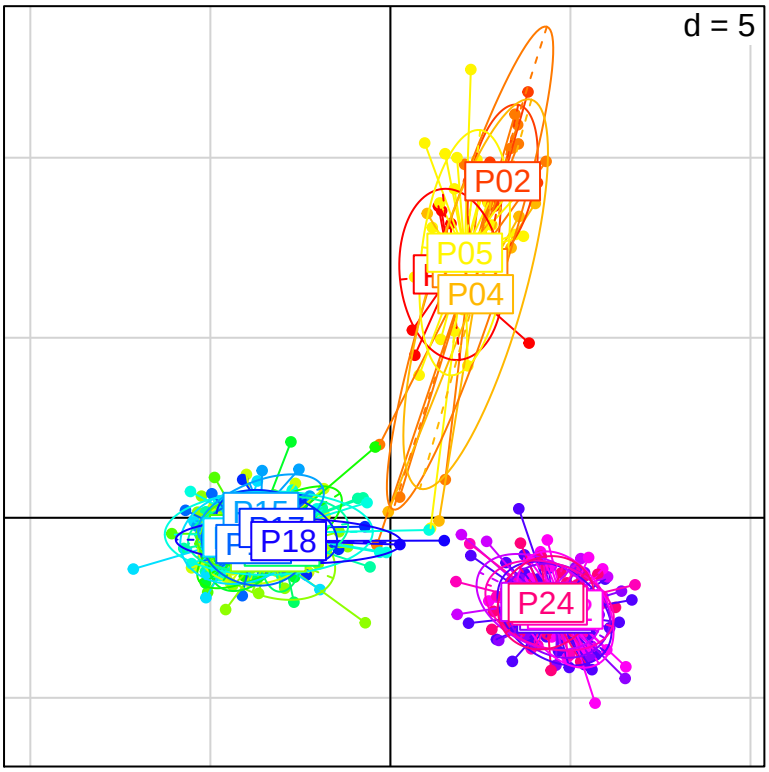
1. Installing `adegenet`
   a. Using the function `install.packages` or the handy interface provided by RStudio, install the `adegenet` package.
   b. Run the following command and save the resulting file somewhere in your project folder.

```
library("adegenet")
adegenetTutorial("basics")
```

   The second command here will open a pdf manual on the web. There is no need to read this in detail now, but if you aim to use `adegenet` more extensively in the future, this document will prove very useful.
   c. Visit http://adegenet.r-forge.r-project.org/ and have a look around the site to get an idea of the capabilities of the `adegenet` package.
2. Loading our trout data
   a. All of the data files neccessary for the analysis of the trout data can be downloaded from http://goo.gl/RBirQp. Extract the contents of this archive and save them in your **data** folder. The contents should be *trout.gen*, *trout_coords.txt*, and *trout_river.asc*. These files are the genotype data, the xy coordinates of each individual, and a raster of the study river, respectively. For this practical we will only use the *trout.gen* file.
   b. Once you have extracted and saved the results, making sure that `adegenet` is loaded, read the *trout.gen* file into R. Assign the genind object to the variable name `trout_data`.
   HINT: you can use relative paths rather that telling the `read.genepop` function the entire directory path.

```
C:/users/kevin/popgen-workshop/data/trout.gen == ./data/trout.gen
```

   c. How many population samples, loci and individuals are in the *trout.gen* file.
3. Running a PCA
   a. Now that we have our data loaded into R, the first thing to do is deal with missing data and scale our allele frequencies. Using the function `scaleGen`, replace missing data with allele means and scale frequencies. Save the scaled data to the variable `scaled_trout`.
   b. Using the function `dudi.pca`, carry out a PCA on your scaled data. Choose how many PC axes to retain from the resulting barplot. HINT: the "zoom" button in RStudio might help you to decide (make sure you keep at least 2 PCs for plotting purposes).
   c. Using the function `s.label`, can you see any evidence of genetic structuring?
   d. How about with the `s.class` plot? HINT: the function `rainbow` lets you choose colours based on an integer (e.g. `rainbow(20)` gives 20 unique colours).
   e. Your `s.class` plot should have looked something like this



This is quite a mess. There are no real indications of patterns of genetic structuring at the sample site level. Let's try one more trick. Create a copy of your *trout.gen* file and name it *trout_riv.gen*. Open the new file and remove all of the "POP" indicators so that each river (i.e. River 01, River 02 and River 03) is represented as three population samples. The rivers that samples belong to are indicated by the first three characters of an individuals name (i.e. any sample with a name beginning with "R01" was sampled from River 01).

f. When you have completed this, rerun your analysis and plotting. Is there any better evidence of genetic structuring?
g. What percentage of the total veriation does PC1 explain?

Conclusions

Clearly there is some indication of genetic structuring on a River level, but it is not very clear how individuals should be grouped. In the next lecture we will see how the PCA method can be combined with clustering techniques to classify individuals into discrete populations.