Population Genetic Analysis III: Biallelic SNP Data
BZ/MIP 577 Fall 2015

Kelly Pierce

Part I: Detecting population structure at the genome level

Analysis of Next Generation Sequencing-derived Multilocus Genotypes

Last week you considered two methods for analyzing population structure with multilocus genotype data: detection of subpopulation clusters (STRUCTURE) and assignment tests (GeneClass2). This week we will expand our tool-kit for inferring population structure to include techniques that are suitable for the much larger multilocus datasets derived from Next Generation Sequencing (NGS) data. We will specifically focus on methods for **codominant, biallelic** SNP genotypes implemented in the R package ***adegenet***.

Objectives

1. Find population structure without an *a priori* hypothesis about group membership using clustering methods.
2. Find population structure with hypotheses about group membership using clustering methods, and additional evidence for those clusters with assignment tests.

Discriminant Analysis of Principal Components (DAPC)

Discriminant analysis of principal components combines (somewhat predictably) discriminant analysis (DA) and principal components analysis (PCA) to estimate the number of subpopulations present in a multilocus genotype dataset. The first step is to perform a PCA on the genotype data, reducing its dimensionality and capturing much of the variation in a set of principal components for further analysis. DA is then performed on the reduced data set to reveal clustering that might suggest population structure. *Figure 1* provides an overview of how PCA and DA both work to describe variation in a data set.

Background on principal components analysis (PCA) We observe genotypic variation in SNP datasets, and that variation may be the result of subpopulation differences. But with potentially thousands of SNPs to consider, looking for meaningful patterns in variation is a daunting task.

PCA is a technique used to describe key axes of variation in multi-dimensional data, including those from multilocus SNP studies. PCA transforms data so that it can be plotted on axes that describe the **most** variation in the data; these axes are the "principal components". This reduces the dimensionality of the data by capturing meaningful variation in a few axes and allowing others to be reasonably ignored (though it is important to retain enough principal components to capture as much of the variation as possible). For a really excellent visualization of how PCA transforms data to maximize variation along a single axis, check out this page: http://setosa.io/ev/principal-component-analysis/

Rarely does a single principal component capture a large fraction of variation in a genetic dataset. Typically many principal components must be retained. While we will not be able to visualize all of the principal component axes, performing the PCA will help us pull out meaningful variation.

Background on discriminant analysis (DA) DA is another type of data transformation that, like PCA, seeks to describe variation in a data set. However, DA differs from PCA in that is seeks to maximize not overall variation, but variation **between** groups. This makes DA particularly useful for detecting population structure. DA does not work well for high-dimensional data, so DA is actually performed on the principal components generated with the PCA.

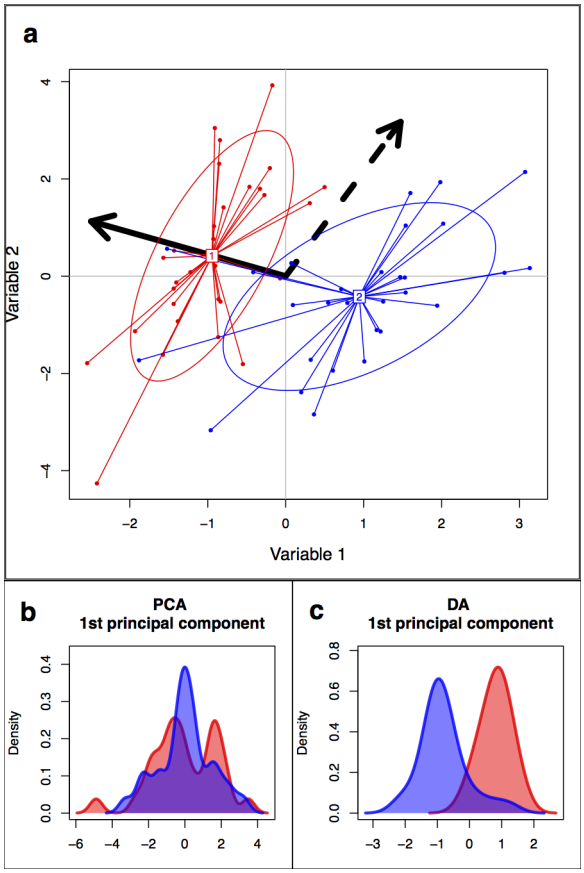


Figure 1. Comparison of PCA and DA for detection of clustering in a data set. (Jombart, Devillard, and Balloux 2010)

STRUCTURE vs DAPC for detecting population structure

	STRUCTURE	DAPC in *adegenet*
model	likelihood based allele frequency model (MCMC)	deterministic allele frequency model
customization	can set priors; see manual	can look for hierarchical clusters; no priors
evaluation	negative log likelihood	Bayesian Information Criterion (BIC)
approx. run time	somewhat slow, esp. with large datasets	very quick ID of clusters
ploidy	haploid to polyploid	haploid to polyploid
linkage	none or weak linkage	none, but robust to violation
marker type	codominant or dominant/recessive; see manual	codominant

Data

You will be working with a reformatted version of the American dog tick (*Dermacentor variabilis*) SNP data. These ticks carry wildlife pathogens and feed on a large diversity of wildlife host species. American dog ticks have an interesting life history. While they spend most of their lives in the environment either digesting blood or looking for new hosts, they mate on their host animals. This makes host animals islands of sorts – the possible available mates for a tick are only those other ticks on the same animal at the same time. If some American dog ticks prefer one host species, and another segment of the tick population prefers a different host species, that reproductive isolation may lead to the development of population structure. We hypothesized that this structure would exist and could tell us something about the feeding behavior of ticks.

We collected ticks directly from raccoons and opossums at two different locations in Texas to address questions about how genotype correlates with collection site and host species (*Figure 2*). Consequently there are four possible subpopulations: site 1 - raccoon, site 1 - opossum, site 2 - raccoon, and site 2 - opossum. Approximately 3% of the tick genome was sequenced in a reduced-representation library. We sequenced 99 ticks and obtained 2387 SNP loci.

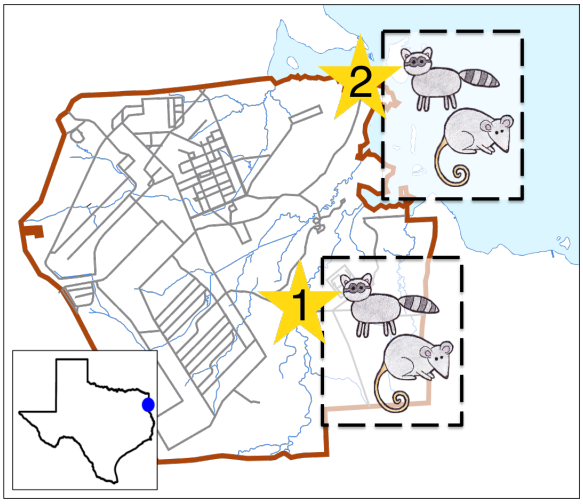


Figure 2: Map of tick collection sites.

You will evaluate a series of hypotheses regarding spatial and by-host species population structure in this dataset.

SNP Data Format

You will use DAPC in *adegenet* to test these hypotheses. *Adegenet* requires input data to be in *genind* format (not the VCF files you have experience working with). You will find if you continue to work with NGS data that each analysis package requires a different data format. Sometimes you can use software like CONVERT to change data formats; other times you will need to do more tedious by-hand reformatting work. CONVERT does not output to *genind* format, but the proper format can be achieved manually. We have spared you the hassle of manually reformatting the VCF file and provided you with the SNP data in *genind* format.

The *genind* format has the following structure:

```
>>>> begin comments - do not remove this line <<<<
Any comments you would like to add about your data go here.
```

```
>>>> end comments - do not remove this line <<<<
>> population
1 2
>> ploidy
2
> sample-1
0000000---10101100011222
> sample-2
00000012000200022220-10
```

The ">> population" line contains a population identifier for each individual in the data set in one row. You should have one entry for each individual in your population.

The ">> ploidy" line indicates whether you are providing diploid or haploid genotypes.

The "> sample" lines contain the genotypes for each indivial. Our data are biallelic codominant markers from a diploid organisms, so the genotypes are coded in 0/1/2 format. The value represents the number of copies of the reference allele (typically the major allele for that locus) an individual has. There will be one number for each SNP locus in the data set. Missing data are coded as "-".

Code	Genotype
0	AA
1	AB
2	BB
-	missing

Exercises

Instructions

Complete the following steps in R. Your end result should be a script that loads the requisite packages and walks through the analyses described. Write answers to specific questions in the comments of your code.

Install and load the required packages. Install *adegenet* and *scales* (for fine-tuning the figures you will make):

```r
# for OSX and Windows users, download directly from CRAN
install.packages("adegenet")
# for Ubuntu users, download source code and install manually
install.packages("~/your/path/adegenet", repos = NULL, type = "source")

# packages
library(scales)
library(adegenet)
```

Download and load the data. Data for this exercise is in

```
/Dropbox/BZ 577/Week 5-6/BiallelicSNP_Analysis/
```

Execute the following code to load the data and inspect its structure. Remember to set your working directory appropriately!

```
# read in data
both.sites <- read.snp("Ticks_from_two_sites.snp", parallel = F)

# look at the data structure
both.sites
```

The '@' symbol denotes a data slot in the *genind* object. You can access data by referencing its slot:

```
# how many populations are listed?
both.sites@pop

# how many SNPs are there?
both.sites@n.loc
```

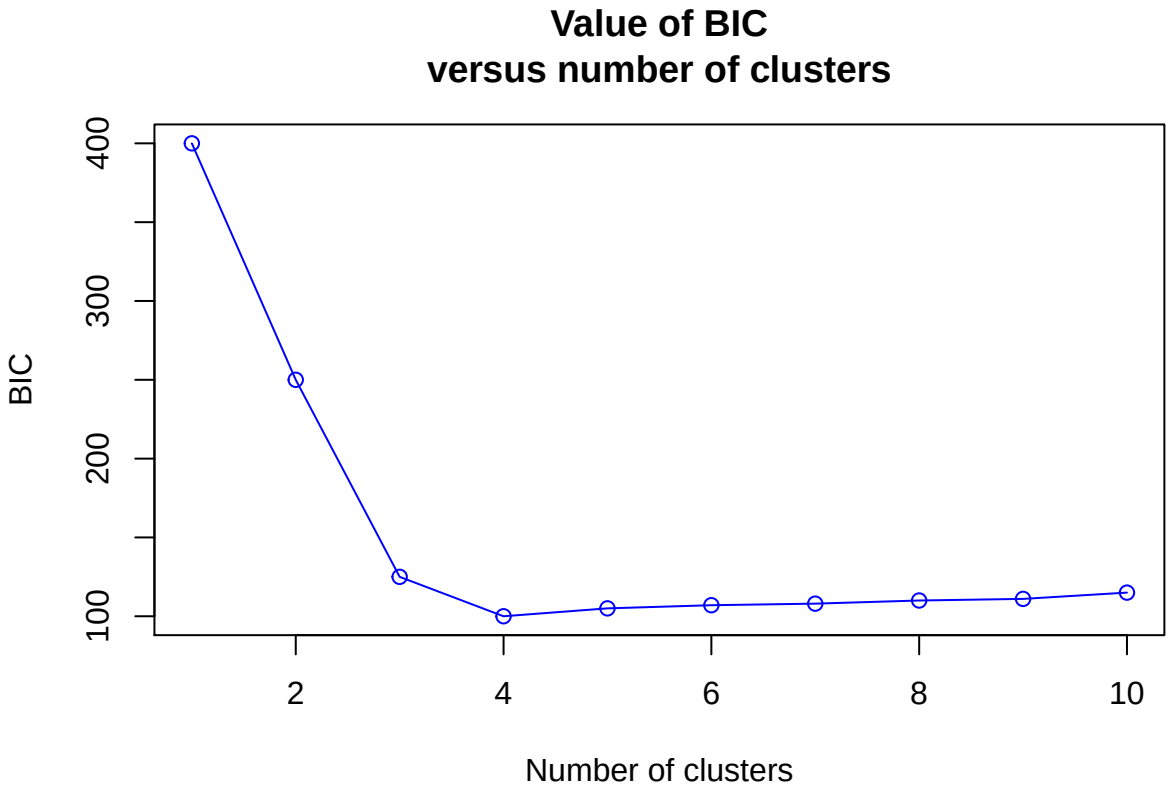
There are also dedicated functions for accessing data. A handy one tells you the number of individuals in your dataset:

```
# how many individuals?
nInd(both.sites)
```

Hypothesis 1: There is some structuring of the tick population, but we do not know how those subpopulations should be defined. As with STRUCTURE, DAPC requires that you indicate how many subpopulations you think are in your dataset. While we have an idea what the subpopulations *should* be for the tick dataset, it is valuable to understand how you would approach this analysis without such a hypothesis. The typical approach when you do not know how many subpopulations there might be is to try a range of different subpopulation numbers and use a metric to determine the best fitting model. STRUCTURE uses negative log likelihoods to determine the most supported number of subpopulations. The DAPC clustering process, on the other hand, uses the Bayesian Information Criterion (BIC). BIC is derived from negative log likelihoods, but favors models with fewer parameters to avoid overfitting the model to the data.

The adegenet function *find.clusters()* will perform the DAPC clustering and compute the BIC for each number of subpopulations. The function is interactive; it will present a graph of number of clusters vs BIC from which you can choose the number of clusters to retain. You will want to choose the number of clusters corresponding the the **lowest BIC**. The shape of the curve is also informative. For example, BIC values that are close to each other suggest that there is insufficient information to determine the number of clusters. If two BIC scores are within two points you cannot definitively favor one over the other. Larger differences are more meaningful and can be used to rule out numbers of subpopulations that are not supported by the data.

Here is an example figure of **totally made up data** – your BIC curve WILL look different.



Question 1. Interpret the fake BIC curve above. If these were real data, how many subpopulations would you infer?

Question 2. Find the number of population clusters in the tick data using the *find.clusters()* function. You have to instruct the *find.clusters()* function to save a certain number of principal components (argument *n.pca*) to use in the DA. If you do not, you will be prompted to choose a number. We want to save as many principal components as possible in order to retain as much variation as we can. The largest number of principal components we can retain is 1/N, or 33 principal components from a sample size of 99 ticks. You will see a warning if you attempt to save more principal components.

```
guess.k<-find.clusters(both.sites, n.pca=33, parallel=F)
# parallel=F for Windows compatibility
```

Question 3. What is the lowest BIC for the tick dataset? Does this support the hypothesis that the population is structured?

Hypothesis 2: Tick populations are clustered spatially (each collection site has a distinct subpopulation). Because we actually do have some *a priori* idea of what the tick subpopulations might be, so we can leverage this information to improve our structure inference. The following code addresses more specific hypotheeses about structure. While we have up to four possible subpopulations, we can break our analysis into two parts and first look for evidence of structure by collection site.

We specifically test for population structure corresponding to collection site by providing the DAPC function with the collection site of each individual. DAPC then identifies the genotypes that best differentiate your pre-defined populations.

```
# discriminant analysis of principal components to detect clusters
# the "pop" argument gives the known collection sites
# again, save 33 principal components
```

```
both.sites.dapc<-dapc(both.sites, pop=both.sites@pop, n.pca=33, n.da=1, parallel=F)
par(mar=c(5,5,2,2))
both.sites.colors<-c('blue', 'red')
scatter(both.sites.dapc, col=both.sites.colors)
legend("topright", legend=c('Site 1', 'Site 2'),
       fill=alpha(both.sites.colors, 0.5), bty='n')
```

The x-axis for your plot shows the value for each individual in the sample along a single discriminant function axis. Because there are only two populations, we need only a single axis to describe the variation between them.

Question 4. Do you believe there is evidence for distinct subpopulations at each collection site?

We can generate STRUCTURE-like plots for the DAPC output. The typical structure plot is simply a stacked barplot where the bar height indicates the probability of membership in a given subpopulation.

```
par(xpd=T, mar=c(5,4,2,8))
compoplot(both.sites.dapc, names.arg=rep(NA, 99), col=site.colors, legend=F)
lines(y=c(0,1), x=c(99.7, 99.7), lwd=4, lty=2)
mtext(side=1, line=1.3, text="Left of Dashed Line: True Membership Site 1")
mtext(side=1, line=2.3, text="Right of Dashed Line: True Membership Site 2")
legend(120, 0.6, legend=c('Site 1', 'Site 2'), fill=both.sites.colors,
       title='Assigned Population', bty='n')
# you can ignore the "names" is not a graphical parameter errors
# but if it's annoying you, shut off the graphics device
# beware, this will clear all of your plots from the plotting pane of R Studio
graphics.off()
```

Question 5. Now that you have considered the assignment probabilities for each individual to the respective populations, do you still believe your conclusions in Question 4?

Question 6. The sample sizes are very unbalanced – 83 ticks from Site 1 and 16 ticks from Site 2. What consequence does this have for the analysis?

We can improve our confidence in the results by conduting assignment tests on novel genotypes. Because you tell DAPC exactly which population each indiviual comes from, it is not terribly surprising that you recover evidence in support of your hypothesis. If we truly observe strong population structure, novel genotypes should be correctly assigned to the appropriate subpopulaton. Poor model fits will not produce consistent assignment of genotypes despite showing strong evidence of clustering when the full dataset is considered.

But wait – where do we get *novel* genotypes if we included all of our data in the DAPC clustring? We can redo the DAPC iteratively, each time withholding the genotype of a different individual. We can then use the *predict.dapc()* function to apply the DAPC model to the withheld genotype. The *predict.dapc()* function will assign the withheld genotype to a population. Since we know which population (that is, which collection site) each genotype is from, we can see how many times the correct prediction is made and calculate an overall accuracy of prediction for the DAPC model.

Use the following function to perform the prediction for withheld genotypes:

```
n.minus.one<-function(genind.data){
  correct=0
  for(i in 1:nInd(genind.data)){
    x.rm<-genind.data[i] # remove individual i
    x.kp<-genind.data[-i] # keep all but individual i
    x.kp.dapc<-dapc(x.kp, n.pca=32, n.da=1, parallel=F)
```

```
    predict.x.rm<-predict.dapc(x.kp.dapc, newdata=x.rm)
    if(as.character(predict.x.rm$assign)==as.character(pop(x.rm))){
      correct=correct+1
    }
  }
  return(correct)
}

site.correct<-n.minus.one(both.sites)
```

We can also calculate a 95% confidence interval for our accuracy estimate. If assignment is no better than random, we expect 50% accuracy. If the 95% confidence interval does not include 50% accuracy, that is evidence that genotype has some ability to predict subpopulation membership.

```
library(gplots) # for plotting confidence intervals on barplots

# binomial test to calculate 95% confidence interval
CI<-binom.test(site.correct, nInd(both.sites), p=0.5, alternative="two.sided")
par(mar=c(5,5,2,2))

# gplots::barplot2 plots confidence intervals (other arguments purely for aesthetics)
barplot2(height=site.correct/nInd(both.sites)*100, width=1, xlim=c(0, 1.25),
         ylim=c(0,100), plot.ci=T, ci.l=CI$conf.int[1]*100, ci.u=CI$conf.int[2]*100,
         las=1, xlab='Collection Site Assignment', ylab='Accuracy (%)')
```

Question 7. What fraction if individuals does DAPC accurately assign to the correct subpopulation? Is this significantly different from 50% accuracy (random assignment)? Is this consistent with the results from the full model including all genotypes?

Question 8. Compare your results from the specific hypothesis test to the results from the naive approach using the *find.clusters()* function. Does using a specific hypothesis about subpopulation membership change your conclusion about population structuring? Which result should you believe?

Question 9. We can assess populations structure by host species using the same basic approach as we used for population structure by collection site. Looking just within site 1, do you believe that tick populations are structured by host animal (raccoon or opossum)?

Site 1 only data are found in Dropbox:

```
/Dropbox/BZ 577/Week 5-6/BiallelicSNP_Analysis
```

Load the new dataset and analyze as you did for the structure by collection site questions:

```
site1.only<-read.snp('Ticks_from_site_1.snp', parallel=F)
```

References

Jombart, Thibaut, Sébastien Devillard, and François Balloux. 2010. "Discriminant Analysis of Principal Components: a New Method for the Analysis of Genetically Structured Populations." *BMC Genetics* 11 (1): 94. doi:10.1186/1471-2156-11-94. http://www.biomedcentral.com/1471-2156/11/94.