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

Kelly Pierce

Exercises

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

```
## Warning: package 'scales' was built under R version 3.1.3
```

```
library(adegenet)
```

```
## Warning: package 'adegenet' was built under R version 3.1.3
```

```
## Loading required package: ade4
```

```
## Warning: package 'ade4' was built under R version 3.1.3
```

```
##
##    /// adegenet 2.0.0 is loaded ////////////
##
##    > overview: '?adegenet'
##    > tutorials/doc/questions: 'adegenetWeb()'
##    > bug reports/feature resquests: adegenetIssues()
```

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

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

```
# read in data
setwd('~/Dropbox/BZ 577/Week 5-6/BiallelicSNP_Analysis/')
both.sites<-read.snp('Ticks_from_two_sites.snp', parallel=F)
```

```
##
##  Reading biallelic SNP data file into a genlight object...
##
##
##  Reading comments...
##
##  Reading general information...
##
##  Reading 99 genotypes...
## .
##  Checking consistency...
##
##  Building final object...
##
## ...done.
```

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

```
##  /// GENLIGHT OBJECT /////////
##
##  // 99 genotypes,  2,387 binary SNPs, size: 290.9 Kb
##  22420 (0.09 %) missing data
##
##  // Basic content
##    @gen: list of 99 SNPbin
##    @ploidy: ploidy of each individual  (range: 2-2)
##
##  // Optional content
##    @ind.names:  99 individual labels
##    @pop: population of each individual (group size range: 16-83)
##    @other: a list containing: elements without names
```

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
```

```
##  [1] site1 site1 site1 site1 site1 site1 site1 site1 site1 site1 site1
## [12] site1 site1 site1 site1 site1 site1 site1 site1 site1 site1 site1
## [23] site1 site1 site1 site1 site1 site1 site1 site1 site1 site1 site1
## [34] site1 site1 site1 site1 site1 site1 site1 site1 site1 site1 site1
## [45] site1 site1 site1 site1 site1 site1 site1 site1 site1 site1 site1
## [56] site1 site1 site1 site1 site1 site1 site1 site1 site1 site1 site1
## [67] site1 site1 site1 site1 site1 site1 site1 site1 site1 site1 site1
## [78] site1 site1 site1 site1 site1 site1 site2 site2 site2 site2 site2
## [89] site2 site2 site2 site2 site2 site2 site2 site2 site2 site2 site2
## Levels: site1 site2
```

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

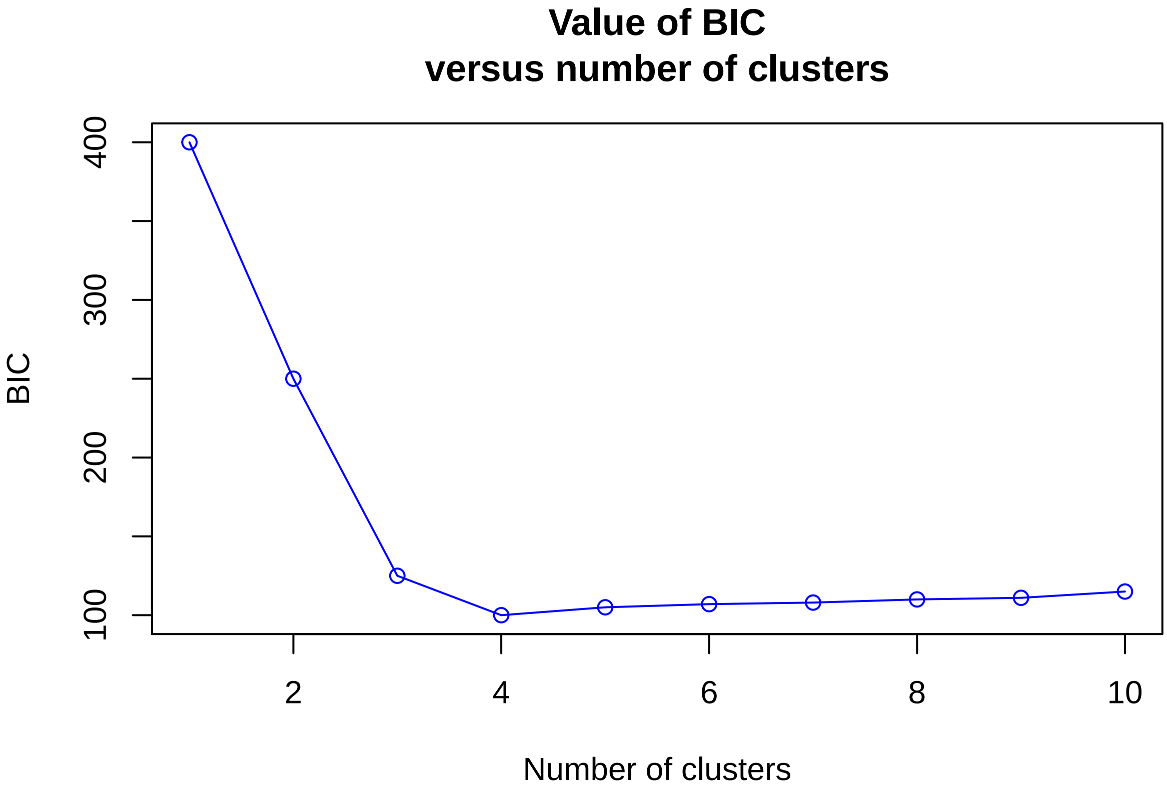
```
## [1] 2387
```

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)
```

```
## [1] 99
```

Hypothesis 1: There is some structuring of the tick population, but we do not know how those subpopulations should be defined. Here is an example figure of **totally made up data** – your BIC curve WILL look different.

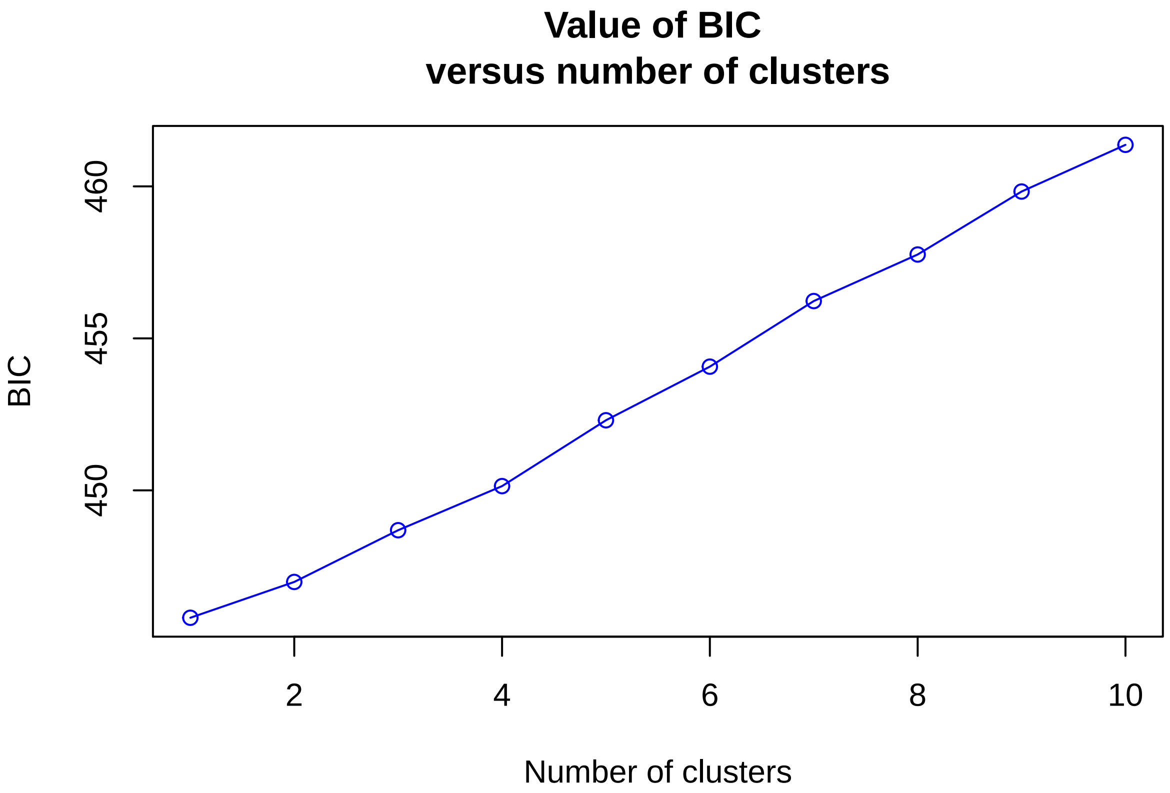


Question 1. Interpret the fake BIC curve above. If these were real data, how many subpopulations would you infer?

The lowest BIC value will correspond to the most highly supported number of subpopulations (number of clusters). In this case, the lowest BIC value corresponds to 4 clusters/subpopulations. The BIC score for 4 clusters is more than 2 lower than the other closest BIC scores, which makes 4 clusters a highly supported inference.

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)
```



```
## Choose the number of clusters (>=2:
```

```
# you can choose any number of clusters >=2 that you would like to retain based on the BIC
# retaining two clusters is the most sensible
# 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?

From looking at the graph that opens up when you run the *find.clusters()* function, the lowest BIC is around 445 and corresponds to only a single cluster. The *guess.k* object saves all the BIC scores so you can look at their actual values:

```
guess.k$Kstat # this is the BIC score
```

```
##      K=1      K=2      K=3      K=4      K=5      K=6      K=7      K=8
## 445.8083 446.9861 448.6884 450.1391 452.3052 454.0697 456.2274 457.7592
##      K=9     K=10
## 459.8304 461.3661
```

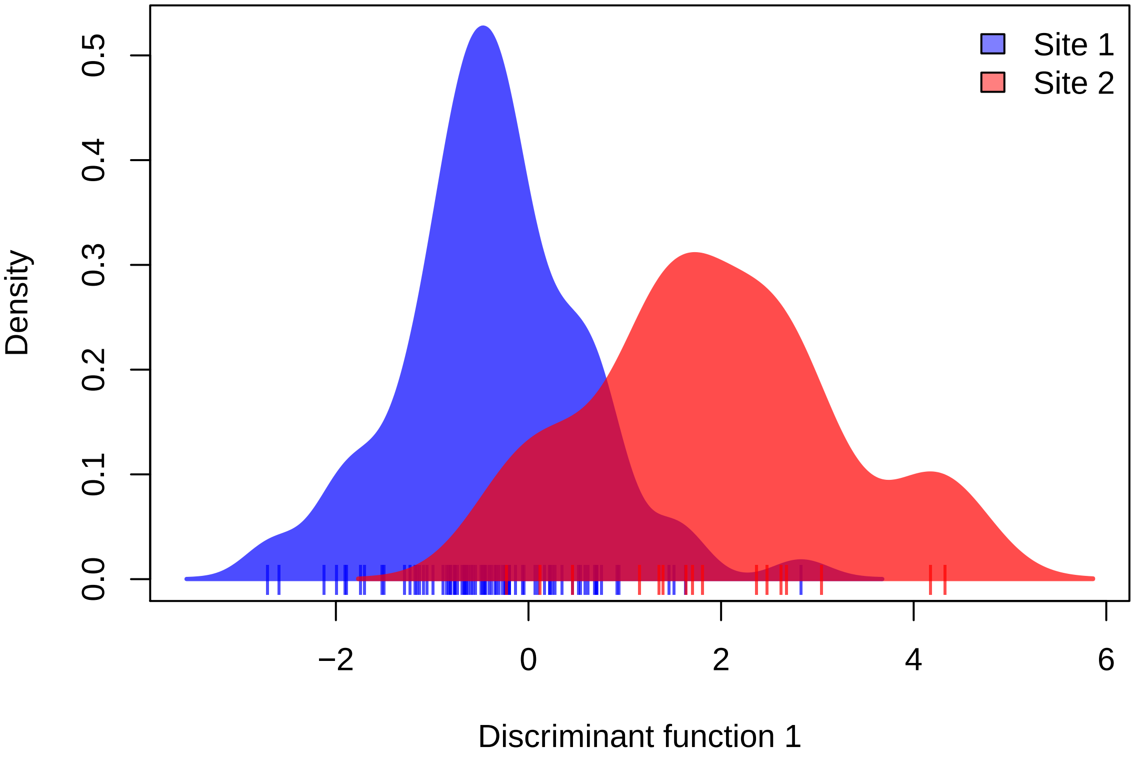
The lowest BIC score corresponds to a single cluster. However, the BIC score for 2 clusters is less than 2 points away. So there is not much compelling evidence for one cluster vs. two clusters. In fact, while the BIC scores steadily increase with the number of clusters (indicating lower support for higher numbers of clusters), each BIC score is within two points of its next highest or lowest score for cluster sizes <=5.

The main conclusion to take away from this is that determining the number of subpopulations with no *a priori* hypothesis often leads to no clear answer. In this case take the most conservative approach – by far the lowest BIC value (even though it is close to the others) is the BIC value for 1 cluster indicating no structure within the population.

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 hypothesees 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))
sites.colors<-c('blue', 'red')
scatter(both.sites.dapc, col=sites.colors)
legend("topright", legend=c('Site 1', 'Site 2'),
fill=alpha(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?

Each tick-mark on the x-axis corresponds to the discriminant function value of a single individual in the dataset. The colors correspond to the collection site for each individual. The height of the curve (density on the y-axis) gives information about the number of individuals with a certain discriminant function value. Qualitative evidence comes from judging the extent to which the two density curves overlap. Because the curves do not fully overlap there is some evidence for population structure by collection site. These populations are not fully distinct though; you can see along the x-axis that not all the blue and red points are grouped with each other. They are mixed a bit in the middle, indicating that genotype clusters do not fully correspond to 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.

```r
par(xpd=T, mar=c(5,4,2,8))
compoplot(both.sites.dapc, names.arg=rep(NA, 99), col=sites.colors, legend=F)
```

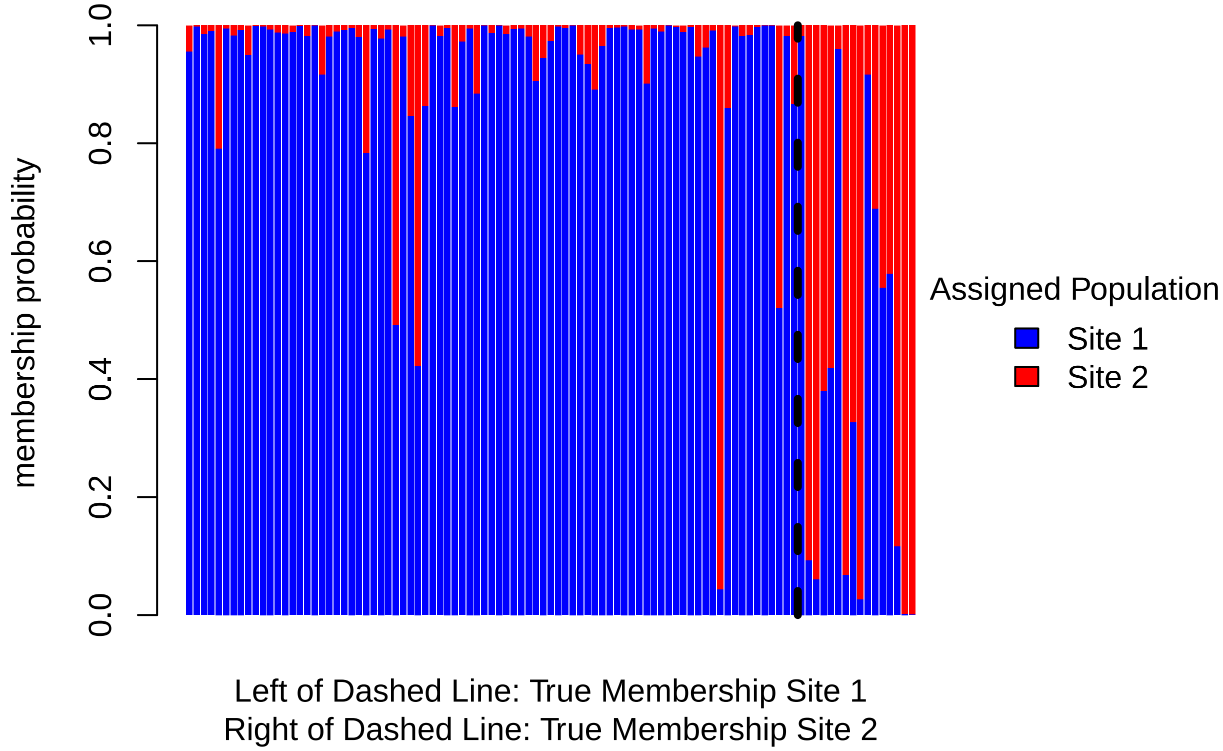
```
## Warning in plot.window(xlim, ylim, log = log, ...): "names" is not a
## graphical parameter
```

```
## Warning in axis(if (horiz) 2 else 1, at = at.l, labels = names.arg, lty =
## axis.lty, : "names" is not a graphical parameter
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "names" is not a graphical parameter
```

```
## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...): "names" is
## not a graphical parameter
```

```r
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=sites.colors,
title='Assigned Population', bty='n')
```



```r
# 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?

The assignment probabilities indicate that individuals from collection site 2 really are more likely to be assigned to a different subpopulation. There is still evidence of mixture between the two subpopulations – only a small number of individuals have 100% probability of assignment to a single subpopulation, population membership cannot be definitively inferred from genotype for most individuals. This presents more compelling evidence however that there is population structure separating the two collection sites.

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?

If the genotypic diversity of collection site 2 has not been fully sampled then sample bias may influence the results. Sampling more genotypes from collection site 2 may reveal more genotypes shared with collection site 1, and reduce the evidence for population structure. Alternatively, the differences between the populations may be made more pronounced by the inclusion of additional data.

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:

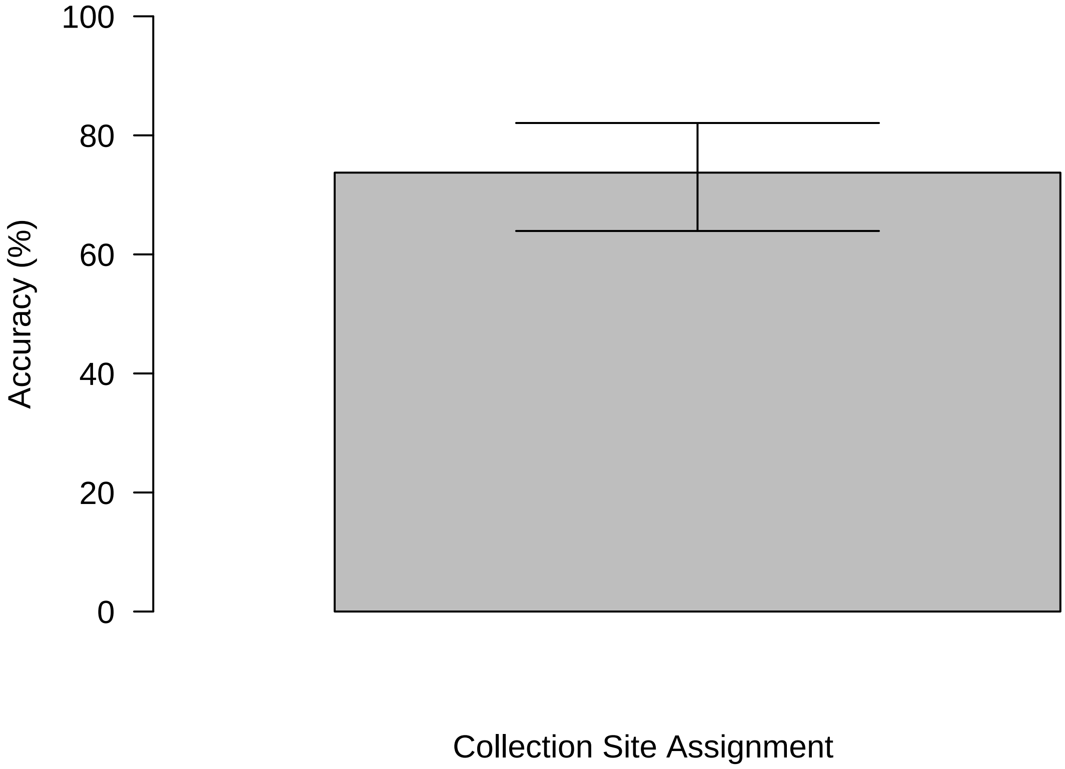
```r
n.minus.one<-function(genind.data){
correct=0
for(i in 1:nInd(genind.data)){
  x.rm<-genind.data[i] # remove individual i
  x.kp<-genind.data[-i] # keep all but individual i
  x.kp.dapc<-dapc(x.kp, n.pca=32, n.da=1, parallel=F)
  predict.x.rm<-predict.dapc(x.kp.dapc, newdata=x.rm)
  if(as.character(predict.x.rm$assign)==as.character(pop(x.rm))){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

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

```
## Warning: package 'gplots' was built under R version 3.1.3

##
## Attaching package: 'gplots'
##
## The following object is masked from 'package:stats':
##
##     lowess
```

```r
# 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?

```
site.correct
```

```
## [1] 73
```

The correct population assignment is made 7300% of the time. The results of the binomial test are statistically significant:

```
CI$p.value
```

```
## [1] 2.484126e-06
```

This is consistent with the results from the full model, which were visualized using the DAPC scatter plot and the STRUCTURE-like plot. Both indicated that there was some weak level of population structure.

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?

The results we obtained with a specific hypothesis in mind, that each collection site would have a different subpopulation of ticks, were more conclusive than when we tried to analyze the data without a hypothesis. Evaluating a specific hypothesis often provides you with more statistical power. In this case, evaluating the hypothesis let us use prior information about population membership (collection site) to look for structure that increased our power to detect said structure.

It is best in this case to use the results obtained by assessing our specific hypothesis. We should conclude that collection sites 1 and 2 have genetically distinct populations of ticks though they are not fully diverged populations. Any subsequent analysis we perform will need to control for this differentiation.

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:

```
setwd('~/Dropbox/BZ 577/Week 5-6/BiallelicSNP_Analysis/')
site1.only<-read.snp('Ticks_from_site_1.snp', parallel=F)
```

```
##
##  Reading biallelic SNP data file into a genlight object...
##
##
##  Reading comments...
##
##  Reading general information...
##
##  Reading 83 genotypes...
## .
##  Checking consistency...
##
##  Building final object...
##
## ...done.
```

```
site1.only@pop
```

```
##  [1] O O O O O O O O O O O O O O O O O O O O O O O O O O O O O R R R R R R
## [36] R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R R
## [71] R R R R R R R R R R R R R
## Levels: O R
```

```
# the first population is opossums ("O")
# the second population is raccoons ("R")

# how many opossum ticks?
length(site1.only@pop[site1.only@pop == 'O'])
```

```
## [1] 29
```

```
# how many raccoon ticks?
length(site1.only@pop[site1.only@pop == 'R'])
```

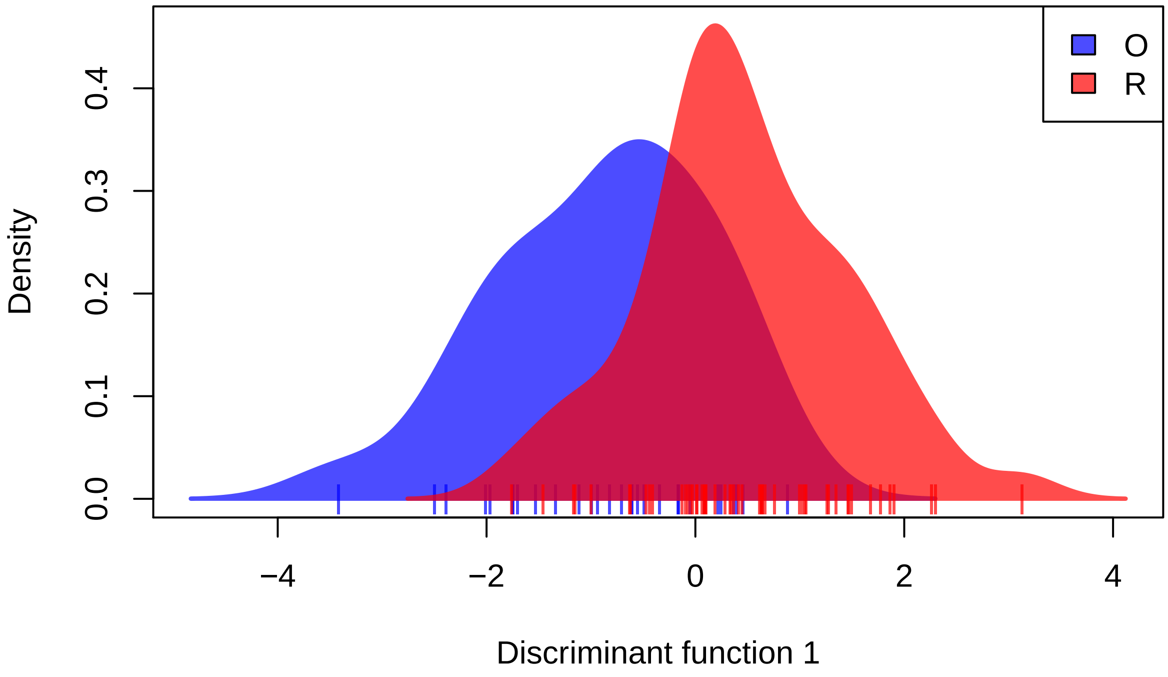
```
## [1] 54
```

Since you are given a specific hypothesis to assess, you do not need to start with the *find.clusters()* function. Instead, perform the discriminant analysis using the hypothesized populations.

```
# retain 27 principal components
# retain 1 discriminant axis
host.dapc<-dapc(site1.only, n.pca=27, n.da=1)
```

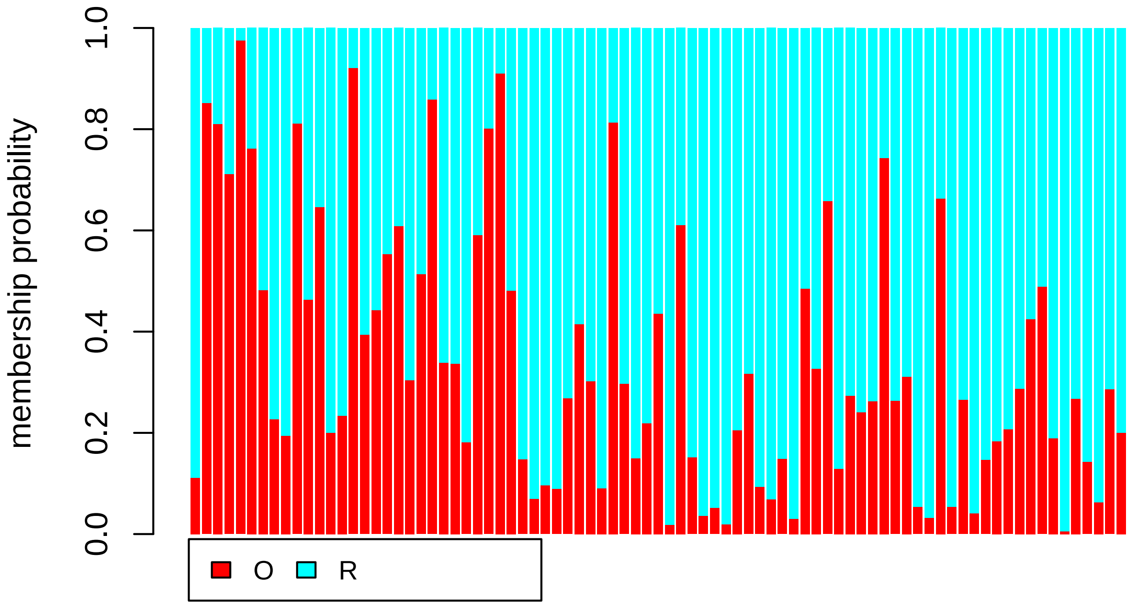
```
## Loading required package: parallel
```

```
scatter(host.dapc, legend=T)
```



```
compoplot(host.dapc, names.arg=rep(NA, 83))
```

```
## Warning in plot.window(xlim, ylim, log = log, ...): "names" is not a
## graphical parameter

## Warning in axis(if (horiz) 2 else 1, at = at.l, labels = names.arg, lty =
## axis.lty, : "names" is not a graphical parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "names" is not a graphical parameter
## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...): "names" is
## not a graphical parameter
```



```
graphics.off()
```

There is substantially more overlap between the curves here. The Structure-like plot shows no clear pattern in grouping of subpopulations (the bars are ordered along the x-axis in the same order as they show up in the data file, with all the opossums ticks first, then the raccoon ticks).

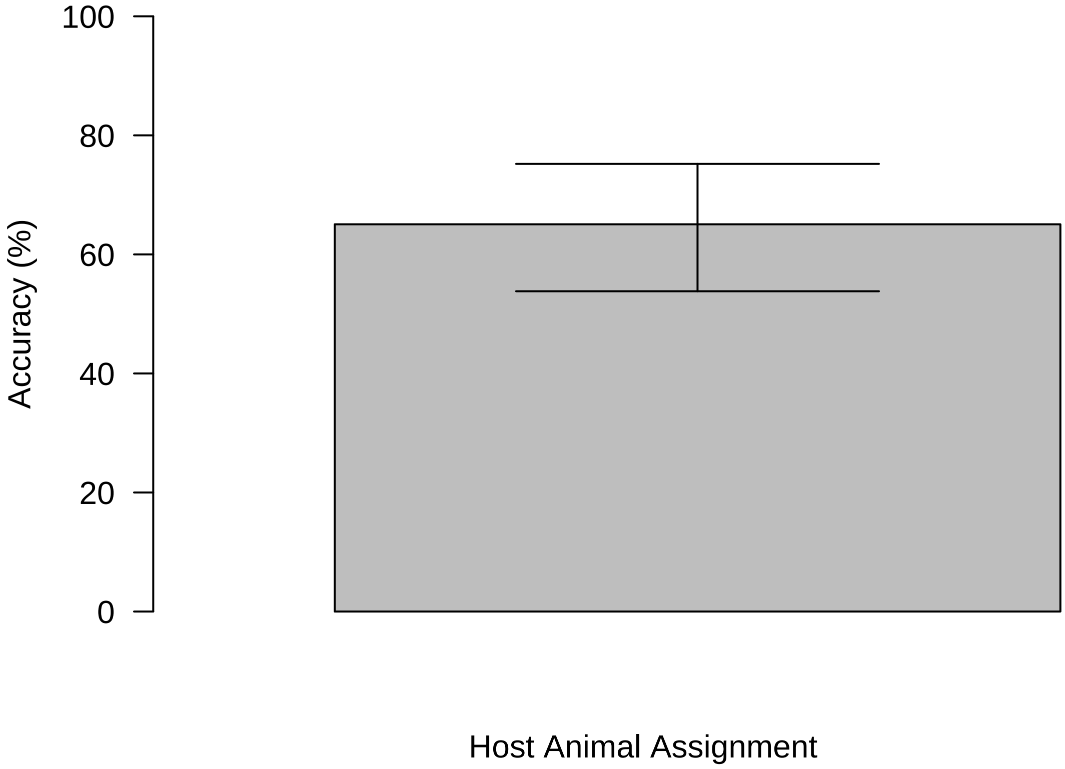
The last thing we might want to check is the ability to accurately assign novel genotypes to the correct population.

If you ran the n.minus.one function as defined above, you should have seen a series of errors regarding the number of principle components retained. You can ignore those warnings if you choose; it does not change the end result dramatically. I've used the *suppressWarnings()* function, alternatively you could edit the function to retain a lower number of principal components.

```
host.correct<-suppressWarnings(n.minus.one(site1.only))

# binomial test to calculate 95% confidence interval
CI<-binom.test(host.correct, nInd(site1.only), p=0.5, alternative="two.sided")

CI$p.value
```

```
## [1] 0.008036706
```

```
par(mar=c(5,5,2,2))

# gplots::barplot2 plots confidence intervals (other arguments purely for aesthetics)
barplot2(height=host.correct/nInd(site1.only)*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='Host Animal Assignment', ylab='Accuracy (%)')
```



The p-value here is 0.0080367, which is low enough to infer statistical significance. But the lower confidence limit, 0.5380705, almost includes 50% (the null hypothesis of equal probability of assignment to both groups). Given the unbalanced sample sizes we should be a little skeptical of this result – while it may technically be statistically significant, any structure here is weak at best.