# Basic Population Genetics Analyses in R

Laboratory Exercise Key

Rodney J. Dyer

## Exercises

Here is a key to the exercises with some rationalle behind the questions.

```
> require(gstudio)
> data(araptus_attenuatus)
> data <- araptus_attenuatus[ araptus_attenuatus$Species=="CladeC",]
> counts <- table(data$Pop)
> counts
```

```
  12  153  157  159  160  161  162  163  164  165  166  168  169
  10   10    2    9   10   10   10    7    8   10    8   10   10
 171  173  175  177   51   58   64   73   75   77   84   88   89
  10   10    7   10    7    9    5    2    1    9    9   10   10
   9   93   98  Aqu Const ESan  Mat  SFr
   9   10    1    4    3    2    1    9
```

If we look at this data, we can see we have a variable number of samples per population. In fact, for this Clade, there are several species with small sample sizes (as it turns out this is because what we thought was one species is actually two separate species in sympatry). So lets go through the data and remove those populations with fewer than 5 samples. If you look at the variable `counts` it is a numeric vector whose names are the population names. From this, we can find the population names whose counts are greater than 5

```
> keepers <- names(counts[ counts> 5 ])
> keepers
```

```
 [1] "12"  "153" "159" "160" "161" "162" "163" "164" "165" "166" "168" "169"
[13] "171" "173" "175" "177" "51"  "58"  "77"  "84"  "88"  "89"  "9"   "93"
[25] "SFr"
```

And then only use the data from those populations using the `%in%` operator.

```
> data <- data[ data$Pop %in% keepers, ]
> table(data$Pop)
```

```
 12 153 159 160 161 162 163 164 165 166 168 169 171 173 175 177  51  58  77  84
 10  10   9  10  10  10   7   8  10   8  10  10  10  10   7  10   7   9   9   9
 88  89   9  93 SFr
 10  10   9  10   9
```

This is pretty cool stuff because you can easily envision how easy it is to work with various subsets of your data set.

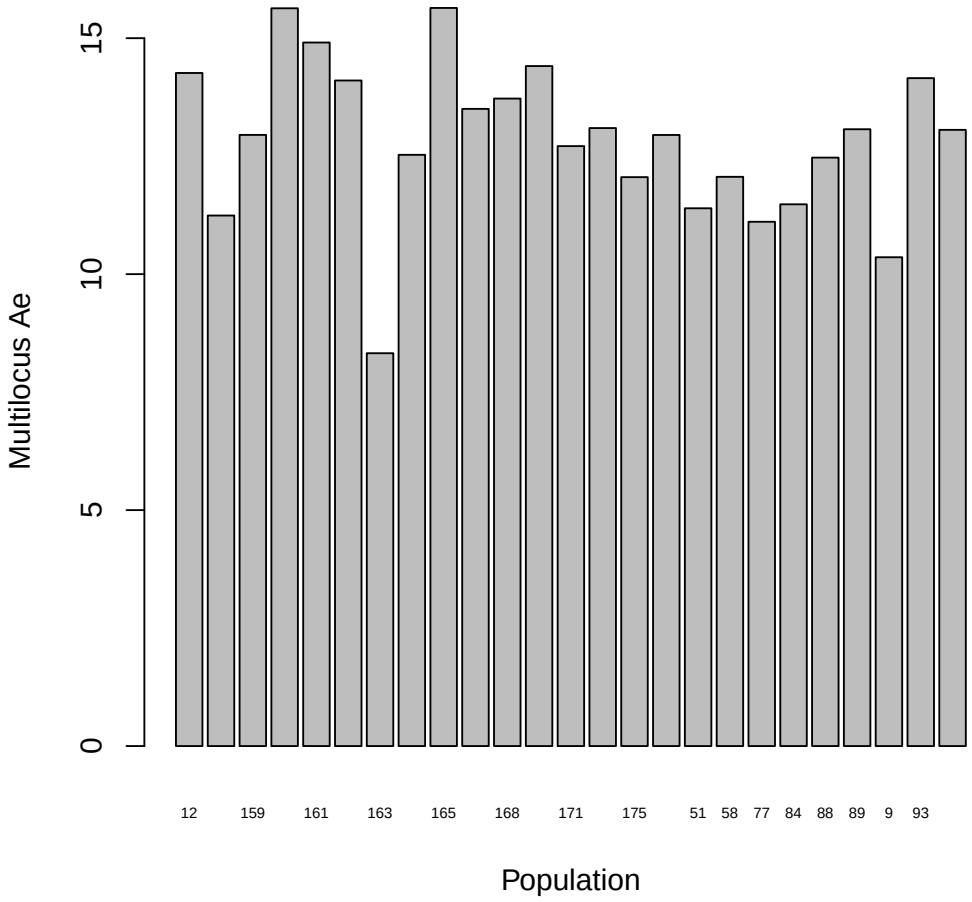
1. *Can you rank these populations in terms of genetic diversity? What metric did you choose and why?* The goal here is to introduce the concept that there are potentially several measures of 'diversity' that are commonly used. In the text and lecture we covered allelic diversity and heterozygosity. For allelic diversity, you can use the function `genetic.diversity` that will estimate these parameters, which by default takes all loci and estimates $A_e$. You can also set `num.perm`=0 to speed up the estimation but it should only take a minute or so on a reasonable computer. Here is an example. I do not show all the output, on the estimates of $Ae$ by population across loci (if you type `Ae` at the prompt, then you'll get a much longer output.)

```
> Ae <- genetic.diversity( data, stratum="Pop", mode="Ae", num.perm=0 )
> Ae$estimate
```

```
           12      153      159      160      161      162      163      164
LTRS 1.470588 1.342282 1.246154 1.470588 1.104972 1.104972 1.000000 1.753425
WNT  1.724138 1.923077 1.670103 1.652893 2.061856 2.000000 1.000000 2.461538
EN   1.000000 1.000000 1.800000 1.000000 1.000000 1.104972 1.000000 1.000000
EF   1.980198 1.000000 1.117241 1.834862 1.834862 2.000000 1.000000 1.753425
ZMP  1.470588 1.219512 1.000000 1.600000 1.000000 1.152941 1.000000 2.000000
AML  2.631579 2.061856 2.314286 3.636364 2.739726 2.409639 1.000000 1.280000
ATPS 1.000000 1.000000 1.384615 1.104972 1.000000 1.104972 1.000000 1.000000
MP20 2.985075 1.694915 2.417910 3.333333 4.166667 3.225806 1.324324 1.280000
          165      166      168      169      171      173      175      177
LTRS 1.724138 1.600000 1.600000 1.834862 1.219512 1.219512 1.000000 1.000000
WNT  2.531646 1.438202 1.680672 2.409639 2.000000 1.219512 1.507692 1.600000
EN   1.219512 1.000000 1.219512 1.000000 1.360544 1.219512 1.507692 1.470588
EF   1.724138 1.280000 1.000000 1.834862 1.000000 1.000000 1.000000 1.000000
ZMP  1.600000 1.438202 1.724138 1.600000 1.246154 1.724138 1.689655 2.000000
AML  2.666667 3.459459 2.597403 2.531646 2.469136 2.739726 2.390244 3.076923
ATPS 1.000000 1.000000 1.000000 1.000000 1.219512 1.503759 1.000000 1.000000
MP20 3.174603 2.285714 2.898551 2.197802 2.197802 2.469136 1.960000 1.801802
           51       58       77       84       88       89        9       93
LTRS 1.324324 1.528302 1.800000 1.000000 1.000000 1.219512 1.000000 1.724138
WNT  1.555556 1.528302 1.117241 1.280000 1.528302 1.104972 1.384615 1.600000
EN   1.000000 1.000000 1.000000 1.780220 2.061856 1.652893 1.800000 1.000000
EF   1.000000 1.000000 1.670103 1.117241 1.104972 1.000000 1.000000 2.000000
ZMP  1.960000 1.528302 1.000000 1.132743 1.117241 1.000000 1.000000 1.280000
AML  2.000000 2.571429 1.780220 1.855072 3.056604 2.857143 1.800000 2.564103
ATPS 1.000000 1.000000 1.000000 1.408696 1.000000 2.173913 1.255814 1.000000
MP20 1.555556 1.905882 1.741935 1.905882 1.600000 2.061856 1.117241 2.985075
          SFr
LTRS 1.528302
WNT  1.246154
EN   1.000000
EF   1.800000
ZMP  1.117241
AML  3.176471
ATPS 1.000000
MP20 2.189189
```

If you want to plot these values, you can do so with the following code.

```
> barplot( colSums(Ae$estimate), xlab="Population", ylab="Multilocus Ae", cex.names=0.5 )
```




If someone is interested in Heterozygosity instead, we provided an example of estimating population-level heterozygosity in the text when we plotted it spatially. Here is a short snippet for how to do it for a single locus.

```
> subpops <- partition( data, stratum="Pop")
> he <- unlist(lapply( subpops, function(x) he(allele.frequencies(x,"AML")[[1]] ) ))
> he
```

```
    12.he    153.he    159.he    160.he    161.he    162.he    163.he    164.he
0.6200000 0.5150000 0.5679012 0.7250000 0.6350000 0.5850000 0.0000000 0.2187500
   165.he    166.he    168.he    169.he    171.he    173.he    175.he    177.he
0.6250000 0.7109375 0.6150000 0.6050000 0.5950000 0.6350000 0.5816327 0.6750000
    51.he     58.he     77.he     84.he     88.he     89.he      9.he     93.he
0.5000000 0.6111111 0.4382716 0.4609375 0.6728395 0.6500000 0.4444444 0.6100000
   SFr.he
0.6851852
```

2. *In the Clade C data, is there any indication of changes in expected heterozygosity as a function of either latitude or longitude? You can use the `cor.test()` function to test for significance.* This is also pretty much straight from the text. In the mapping example, we grabbed the heterozygosity and the coordinates. First, we can find the lat & lon and then go through the loci and look for correlations. Here is an example using the "AML" locus.

```
> lat <- unique( data$Lat)
> lon <- unique( data$Long)
> subpops <- partition( data, stratum="Pop")
```

```
> he <- unlist(lapply( subpops, function(x) he(allele.frequencies(x,"AML")[[1]] ) ))
> cor.test(he,lat)

        Pearson's product-moment correlation

data:  he and lat
t = -0.6805, df = 23, p-value = 0.503
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5074548  0.2696029
sample estimates:
       cor
-0.1404947

> cor.test(he,lon)

        Pearson's product-moment correlation

data:  he and lon
t = 0.5937, df = 23, p-value = 0.5585
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2861668  0.4940056
sample estimates:
      cor
0.1228558
```

3. *In addition to strata-level genetic distances, there are also several individual-level genetic distance measures available. How correlated are the individual genetic distances from methods such as "AMOVA" and "Jaccard"? You may want to use the `mantel` function from the `ecodist` library as we did for population-level distances. Also, since the Jaccard distance is a single-locus estimates, you can either combine them across loci for a multilocus estimate or look at the loci individually.* Here is how you would do this for a single locus.

```
> require(ecodist)
> dist.amova <- genetic.distance( data, loci="AML", mode="AMOVA")[[1]]
> dist.jaccard <- genetic.distance( data, loci="AML", mode="Jaccard")[[1]]
> mantel( as.dist( dist.amova) ~ as.dist(dist.jaccard) )

   mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
 0.8818005  0.0010000  1.0000000  0.0010000  0.8764694  0.8916971
```

4. *I didn't use Bray-Curtis in the previous question because there are some missing data. Can you think of a way to handle missing data using this metric so that a comparison can be made?* This is an open ended question. I could imagine the following responses:

   (a) Remove individuals with missing loci

   (b) Set all missing "NA" BrayCurtis values to the mean value or to 0.

5. *Of the single-locus measures of genetic structure, which one would you use to estimate among-population structure and why? Is there a lot of structure in these data or a little?* You can find the number of alleles like we did in the text as:

```
> freqs <- allele.frequencies( data )
> for( locus in names(freqs) )
+         cat(locus, length( freqs[[locus]]), "\n")

LTRS 2
WNT 4
EN 5
EF 2
ZMP 2
AML 10
```

```
ATPS 5
MP20 8
```

and see that we have some loci with few alleles and some with many. Perhaps the most prudent method would be to use one of the corrected methods for all loci. Here is a simple output using $D_{est}$:

```
> genetic.structure( data, stratum="Pop", mode="Dest", num.perm=0)

Geneic Structure Analysis:
  Estimator: Dest
  Stratum: Pop
  Loci: { LTRS, WNT, EN, EF, ZMP, AML, ATPS, MP20 }
   - LTRS ;  Dest = 0.193264016072309
   - WNT ;  Dest = 0.207821757545231
   - EN ;  Dest = 0.0288268957862242
   - EF ;  Dest = 0.338829188801251
   - ZMP ;  Dest = 0.217639958670906
   - AML ;  Dest = 0.286730577709045
   - ATPS ;  Dest = 0.467073477107683
   - MP20 ;  Dest = 0.274255494670032
  MV: 0.251805170795335
```