Microsatellite data analysis

version 2016-12-08 (T. Fér, F. Kolář)

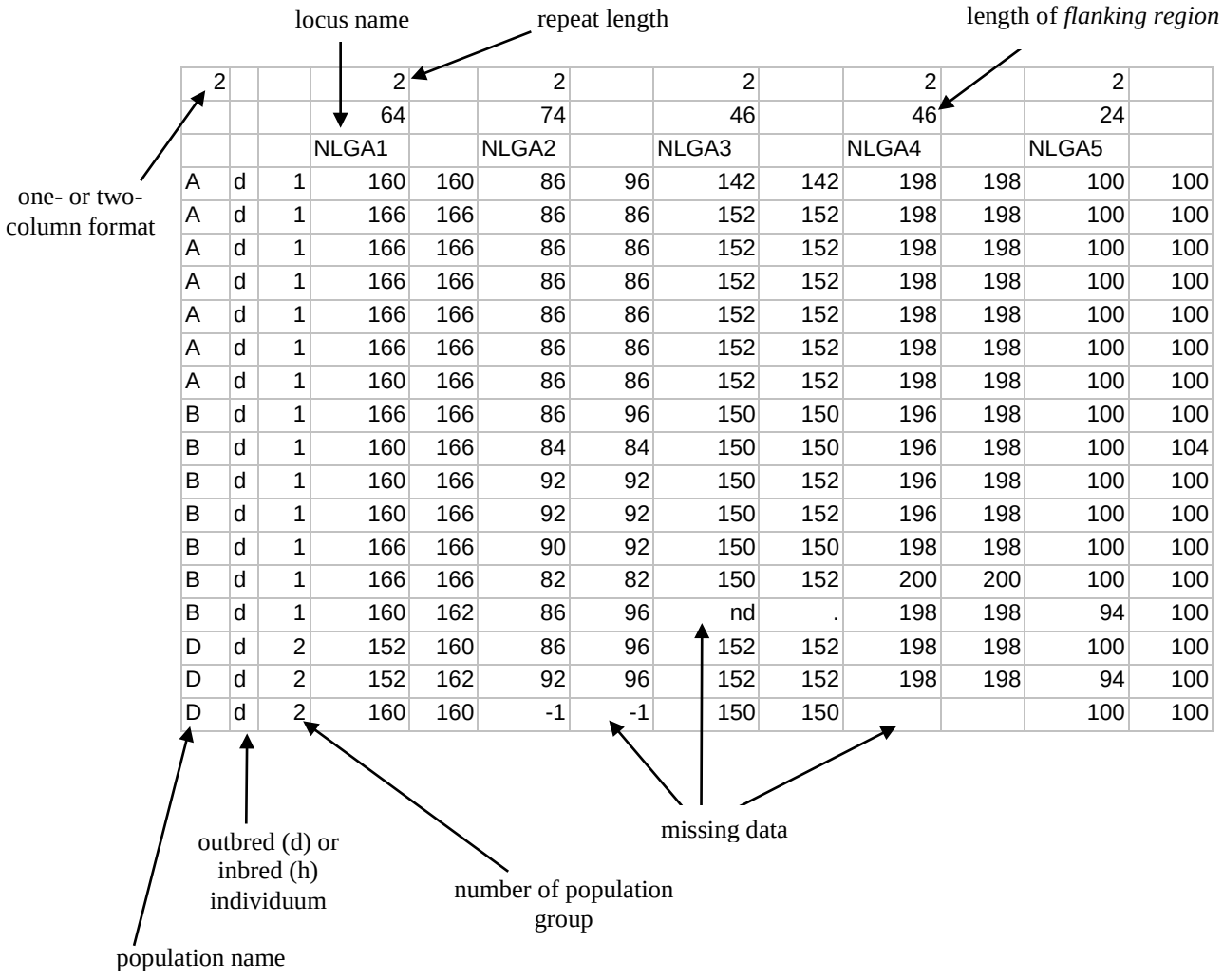
1. Basic analysis of microsatellite data using MSA

Using software MSA (http://i122server.vu-wien.ac.at/MSA/MSA_download.html, freeware) we can (for diploids!) calculate following population-genetic parameters:

- descriptive statistics for populations and loci (number of alleles, H_O, H_E, H_{Sh}, F_{IS}…)
- distance matrices among individuals and populations (D – standard genetic distance, $(\delta\mu)^2$, D_{ps} – proportion of shared alleles, D_{kf} – kinship coefficient…)
- *F*-statistics (F_{ST},F_{IS},F_{IT} – both global and pair-wise)

The software also warns us if there might be an error in the dataset (alleles do not correspond to the multiple of repeat unit, too big distance among alleles, distant alleles etc.).

The input format for MSA is best to prepare in, e.g., Excel and save as TAB-delimited text (or better copy it to a simple text editor, e.g., Notepad++). Input file for diploid species:



2			2		2		2		2		2	
			64		74		46		46		24	
			NLGA1		NLGA2		NLGA3		NLGA4		NLGA5	
A	d	1	160	160	86	96	142	142	198	198	100	100
A	d	1	166	166	86	86	152	152	198	198	100	100
A	d	1	166	166	86	86	152	152	198	198	100	100
A	d	1	166	166	86	86	152	152	198	198	100	100
A	d	1	166	166	86	86	152	152	198	198	100	100
A	d	1	166	166	86	86	152	152	198	198	100	100
A	d	1	160	166	86	86	152	152	198	198	100	100
B	d	1	166	166	86	96	150	150	196	198	100	100
B	d	1	160	166	84	84	150	150	196	198	100	104
B	d	1	160	166	92	92	150	152	196	198	100	100
B	d	1	160	166	92	92	150	152	196	198	100	100
B	d	1	166	166	90	92	150	150	198	198	100	100
B	d	1	166	166	82	82	150	152	200	200	100	100
B	d	1	160	162	86	96	nd	.	198	198	94	100
D	d	2	152	160	86	96	152	152	198	198	100	100
D	d	2	152	162	92	96	152	152	198	198	94	100
D	d	2	160	160	-1	-1	150	150			100	100

Input file has to be saved to the same folder where we have `MSAnalyzerMr.exe`, which we then run. The window is opened and we can specify the input file and parameters of the analysis. Typing (i) we enter the input file (including its suffix!). Typing (d) and later (p) we can set diverse genetic distances to be calculated (1)-(9) and also set calculation of distance among populations (c) and individuals (i), and possibly also switch on bootstrapping (n). By typing (b) we go back to „distance menu". Typing (s) allows to set parameters of *F*-statistics, typing (c) switch on the calculation of *F*-statistics, typing (g) we select i fit will be calculated globally, pair-wise or both. Typing (m) we go back to „main menu", where by typing (c) we can select building input file for the software Arlequin and/or Structure (3). The analysis can be run by typing (!).

If no bootstrap was use the analysis is really fast. Depending on selected parameters MSA creates many Excel tables and text files and saves them to synoptic folder structure:

- Allelecount – allele numbers and frequencies for individual loci and populations
- Distance_data – text files with distance matrices among individuals and populations
- Formats&Data – input files for Arlequin, Structure and other software
- F-Statistic – *F*-statistics global and pair-wise
- Group_data – information about parameters according to predefined population groups
- Single_data – information about parameters for individual populations

For further informations about parameters, calculation methods and interpretation of output files see the MSA manual (http://i122server.vu-wien.ac.at/MSA/info.html/MSA_info.html).

2. PAST – principal coordinate analysis (PCoA) and distance-based trees

Comment: This tutorial is valid for the version 2.17, the newest version 3.1x looks slightly different.

 Click on the icon to run the program (downloaded from http://folk.uio.no/ohammer/past/). It allows performing principal coordinate analysis and construction of distance-based trees (neighbour-joining, UPGMA), but also many other analyses. This tutorial describes how to insert symmetrical distance matrix (e.g., as output from MSA). If we also want to insert sample labels it is necessary to tick *Edit labels*.

```
      01x1  01x2  01x3  01x4
01x1  1     0.225 0.75  0.44
01x2  0.225 1     0.33  0.98
01x3  0.75  0.33  1     0.55
01x4  0.44  0.98  0.55  1
```

1. Groups definition (groups will be coloured in all following outputs): Shift + select corresponding samples (rows) with mouse click → *Edit* → *Row color/symbol* → select colour/symbol (Comment: in this phase we can save the file and the group definition will be saved as well)
2. PCoA: Select all samples (click to the upper left corner) or desired selection (Shift + mouse click) → *Multivar* → *Principal coordinates* → select *User distance* → click to  copies % of explained variation to the clipboard → *View scatter*
3. Modification and saving of PCoA diagram: clicking to the graph allows to change symbol size, font etc. and also to *Save picture*. *View numbers* displays ordination scores of

individuals and again it is possible to save it via clipboard and make (nicer) picture elsewhere. Symbol colour and size is possible to change only in the source table via *Row color/symbol* (see above).

4. Distance tree (NJ): *Multivar* → *Neighbour joining* → select *User distance* → insert number of bootstrap replicates to the field *Boot N* and press *Enter*. UPGMA tree is made similarly under *Multivar* → *Cluster Analysis* (be sure that *Paired group* algorithm is selected).

3. Descriptive analyses in R: principal component (PCA) and principal coordinate analyses (PCoA), distance trees, amova and in silico hybridization

We will use Structure input file for input into R. Just rename the output of MSA (**`Typha_US_MSA.txt.struct`**) by changing the extension of the file to „.stru“ (required by adegenet).

- We will sequentially create an R script file containing all the commands necessary for our analyses. If you save it at the end, you can rerun the entire set of analyses again whenever you want.

Open R Studio,

- *File -> New File -> R Script* - save this new file to your working directory (the same with input files)
- *Session -> Set Working Directory -> To source file location*
- Now sequentially copy-paste into R following commands to the new file and run each batch. To run the command, highlight the lines you just inserted by mouse and press small icon with green arrow and „Run“ in upper right corner.

- First load the required libraries – i.e. add the following text to your script and Run

```
library (adegenet)
library (adegraphics)
library (pegas)
library (ape)
```

- Import the data in Structure format (Typha_US_MSA.txt.stru) into genind object (native to adegenet package) and check the import. Note that in our case we have designated the three „species“ (i.e. *T. angustifolia* = A, *T. latifolia* = L. and T. ×*glauca* = G as separate „populations“.

```
# IMPORT from STRUCTURE format produced by MSA into adegenet
typ.genind <- read.structure (file="Typha_US_MSA.txt.stru", n.ind=114,
n.loc=9,  onerowperind=F, col.lab=1, col.pop=0, row.marknames=0, NA.char="-
9", ask=FALSE)  # set correct N individuals and loci

#add pop names - here it is first character of individual name
typ.genind@pop <- as.factor(substr(rownames(typ.genind@tab),1,1))

typ.genind
head(typ.genind@tab)  # see the data matrix
typ.genind@pop        # see the "population" assignmnet of individuals
summary(typ.genind)   # check the summary stats
```

- Now calculate Principal component analysis (on Euclidean distance)

```
### PCA = Euclidean distances on centered allele frqs (adegenet)
```

```
typ.pca <- scaleGen(typ.genind, NA.method="mean")        # NAs replaced by
mean allele freq, data are centered
pca.1 <- dudi.pca (typ.pca, cent=F, scale=F, scannf=F, nf=4)      # do PCA,
retain first four axes
pca.1$eig[1]/sum(pca.1$eig) # proportion of variation explained by 1st axis
pca.1$eig[2]/sum(pca.1$eig) # proportion of variation explained by 2nd axis
barplot(pca.1$eig[1:20],main="PCA eigenvalues")
```

- And visualize it

```
# plot
g1 <- s.class(pca.1$li, pop(typ.genind),  xax=1, yax=2,
col=transp((c("#FF0000", "#FFA500", "#008B00")),.6),
              ellipseSize=0, starSize=0, ppoints.cex=4, paxes.draw=T,
pgrid.draw =F, plot = FALSE)
g2 <- s.label (pca.1$li, xax=1, yax=2, ppoints.col = "red", plabels =
list(box = list(draw = FALSE),
              optim = TRUE), paxes.draw=T, pgrid.draw =F, plabels.cex=1,
plot = FALSE)
ADEgS(c(g1, g2), layout = c(1, 2))
```

- If you want to export the image just wrap the previous command into pdf function

```
pdf ("PCA_adegenet_species.pdf", width=14, height=7)
g1 <- s.class(pca.1$li, pop(typ.genind),  xax=1, yax=2,
col=transp((c("#FF0000", "#FFA500", "#008B00")),.6),
              ellipseSize=0, starSize=0, ppoints.cex=4, paxes.draw=T,
pgrid.draw =F, plot = FALSE)
g2 <- s.label (pca.1$li, xax=1, yax=2, ppoints.col = "red", plabels =
list(box = list(draw = FALSE),
              optim = TRUE), paxes.draw=T, pgrid.draw =F, plabels.cex=1,
plot = FALSE)
ADEgS(c(g1, g2), layout = c(1, 2))
dev.off()
```

- Calculate Principal coordinate analysis based on MSA-derived distances among individual. For this you should have in the working dir a file „DAN_Ind.csv“.

```
# first import the matrix
matrix <- as.matrix(read.table("DAN_Ind.txt", skip=1, row.names=1, sep =
""))
matdist <- as.dist(matrix) # write "matrix" as distance matrix "matdist"
pop.identity <- as.factor(substr(rownames(matrix),1,1)) #extract population
identity

# calculate PCO
pcoa.1 <- dudi.pco (matdist, scannf=F, nf=4)      # do PCA, retain first
five axes
pcoa.1$eig[1]/sum(pcoa.1$eig) # proportion of variation explained by 1st
axis
pcoa.1$eig[2]/sum(pcoa.1$eig) # proportion of variation explained by 2nd
axis

# plot PCO
pdf ("PCO_MSAdistance_species.pdf", width=14, height=7)
g1 <- s.class(pcoa.1$li, pop.identity,  xax=1, yax=2,
col=transp((c("#FF0000", "#FFA500", "#008B00")),.6), ellipseSize=0,
starSize=0, ppoints.cex=4, paxes.draw=T, pgrid.draw =F, plot = FALSE)
g2 <- s.label (pcoa.1$li, xax=1, yax=2, ppoints.col = "red", plabels =
list(box = list(draw = FALSE),
              optim = TRUE), paxes.draw=T, pgrid.draw =F, plabels.cex=1,
plot = FALSE)
ADEgS(c(g1, g2), layout = c(1, 2))
dev.off()
```

- Create unrooted neighbor joining tree based on the same distance. The last part of the command export a newick tree which could be opened and edited, e.g., in FigTree.

```
#Make an unrooted tree from "matdist"
plot(nj(matdist), type = "u", show.tip.label=F)  # plot unrooted NJ tree
from "matdist" without tip labels
tiplabels(pch = c(1, 8, 19)[pop.identity])       # add symbols instead of
tip labels
legend("topright", pch=c(1,8,19), c("A", "G", "L")) # add legend to the top
right corner

#save tree and open e.g. in Figtree
tree <- nj(matdist)
sequence <- c(1:length(pop.identity))             # just annoying changing
of labels to avoid duplications
tree$tip.label <- paste0(pop.identity,sequence)   # just annoying changing
of labels to avoid duplications

write.tree(tree,file="NJ.tree.tre")
```

- Calculate AMOVA. For getting proportion of variance explained by „populations“ just divide the SSD for pops by SSD for Total

```
#based on Euclidean distances of adegenet
typ.dist <- scaleGen(typ.genind, center = F, scale=F, NA.method="mean")
# NAs replaced by mean allele freq
dm <- dist(typ.dist, method="euclidean")
pops <- typ.genind@pop
(res <- amova(dm ~ pops))  # default nperm=1000, pegas package must be
loaded

# based on MSA distance matrix
(res <- amova(matdist ~ pop.identity))
```

- Now subset the genind into list of geninds separate for each population and visualize PCA of one population (*T. angustifolia*)

```
# subsetting by population
typ.genind.sep <- seppop(typ.genind)
typ.genind.sep
typ.genind.sep$A

### separate PCA for T. angustifolia (pop A)
typ.pca.2 <- scaleGen(typ.genind.sep$A, NA.method="mean")        # NAs
replaced by mean allele freq, data are centered
pca.2 <- dudi.pca (typ.pca.2, cent=F, scale=F, scannf=F, nf=4)      # do
PCA, retain first four axes
pca.2$eig[1]/sum(pca.2$eig) # proportion of variation explained by 1st axis
pca.2$eig[2]/sum(pca.2$eig) # proportion of variation explained by 2nd axis
barplot(pca.2$eig[1:20],main="PCA eigenvalues")

# plot
g1 <- s.class(pca.2$li, pop(typ.genind.sep$A),  xax=1, yax=2,
col=transp((c("#FF0000", "#FFA500", "#008B00")),.6),
             ellipseSize=0, starSize=0, ppoints.cex=4, paxes.draw=T,
pgrid.draw =F, plot = FALSE)
g2 <- s.label (pca.2$li, xax=1, yax=2, ppoints.col = "red", plabels =
list(box = list(draw = FALSE),

optim = TRUE), paxes.draw=T, pgrid.draw =F, plabels.cex=1, plot = FALSE)
ADEgS(c(g1, g2), layout = c(1, 2))
```

- Now see the PCA of only two species without the hybrid. This could be done easily by merging (function repool) the two geninds of *T. angustifolia* and *T. latifolia*

```
# merge two populations i.e. remove hybrids
typ.genind.nohybr <- repool(typ.genind.sep$A, typ.genind.sep$L)
typ.genind.nohybr


### separate PCA for T. angustifolia and T. latifolia without hybrids
typ.pca.3 <- scaleGen(typ.genind.nohybr, NA.method="mean")        # NAs
replaced by mean allele freq, data are centered
pca.3 <- dudi.pca (typ.pca.3, cent=F, scale=F, scannf=F, nf=4)      # do
PCA, retain first four axes
pca.3$eig[1]/sum(pca.3$eig) # proportion of variation explained by 1st axis
pca.3$eig[2]/sum(pca.3$eig) # proportion of variation explained by 2nd axis
barplot(pca.3$eig[1:20],main="PCA eigenvalues")

# plot
g1 <- s.class(pca.3$li, pop(typ.genind.nohybr),  xax=1, yax=2,
col=transp((c("#FF0000", "#FFA500", "#008B00")),.6),
              ellipseSize=0, starSize=0, ppoints.cex=4, paxes.draw=T,
pgrid.draw =F, plot = FALSE)
g2 <- s.label (pca.3$li, xax=1, yax=2, ppoints.col = "red", plabels =
list(box = list(draw = FALSE),

optim = TRUE), paxes.draw=T, pgrid.draw =F, plabels.cex=1, plot = FALSE)
ADEgS(c(g1, g2), layout = c(1, 2))
```

- Finally, create in silico hybrid and do PCA with all samples and these hybrids

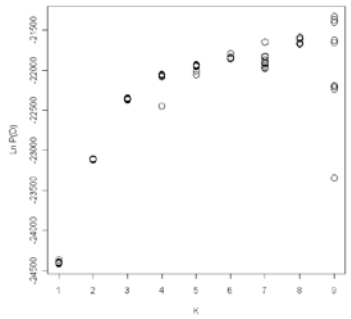
```
# create in silico hybrids and view all in one PCA
typ.hybrids <- hybridize (typ.genind.sep$A, typ.genind.sep$L, n=40,
pop="hybrid")    # create hybrids
typ.genind.hybrids <- repool(typ.genind, typ.hybrids)
# merge with all real samples

### separate PCA for T. angustifolia (pop A)
typ.pca.4 <- scaleGen(typ.genind.hybrids, NA.method="mean")        # NAs
replaced by mean allele freq, data are centered
pca.4 <- dudi.pca (typ.pca.4, cent=F, scale=F, scannf=F, nf=4)      # do
PCA, retain first four axes
pca.4$eig[1]/sum(pca.4$eig) # proportion of variation explained by 1st axis
pca.4$eig[2]/sum(pca.4$eig) # proportion of variation explained by 2nd axis
barplot(pca.4$eig[1:20],main="PCA eigenvalues")

# plot, hybrids will be in black
pdf ("PCA_adegenet_insilicohybrids.pdf", width=14, height=7)
g1 <- s.class(pca.4$li, pop(typ.genind.hybrids),  xax=1, yax=2,
col=transp((c("#FF0000", "#FFA500", "#008B00", "#000000")),.6),
              ellipseSize=0, starSize=0, ppoints.cex=4, paxes.draw=T,
pgrid.draw =F, plot = FALSE)
g2 <- s.label (pca.4$li, xax=1, yax=2, ppoints.col = "red", plabels =
list(box = list(draw = FALSE),

optim = TRUE), paxes.draw=T, pgrid.draw =F, plabels.cex=1, plot = FALSE)
ADEgS(c(g1, g2), layout = c(1, 2))
dev.off()
```

!!! For further info and ispiration read adegenet tutorials at https://github.com/thibautjombart/adegenet/wiki/Tutorials and further materials at http://adegenet.r-forge.r-project.org/documentation.html

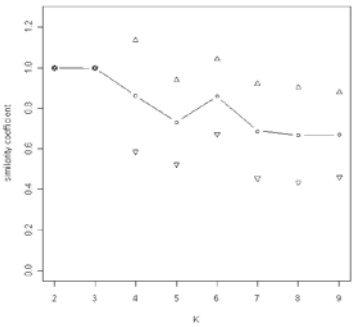
4. STRUCTURE – genetic structure using Bayesian model-based approach

The software STRUCTURE looks for a such partitioning of individuals to K groups (clusters), which is best based on the molecular information (the most probable, with the

highest *likelihood*) and at the same time is this partitioning found in repeated runs of the program. *Markov chains* (MC) that are gradually converging to optimal solution are used to find the best model. Because at the beginning the Marcov chains are outside the optimum solution it is necessary a priori to set number of steps that are before the stable phase (so-called *burn-in*). During a typical run we successively look for the best model for K=1 up to, e.g., K=10 and we repeat the run for each K, e.g., 10 times. One of the crucial questions is: which K is optimal, because models with higher K could also have higher likelihood. Following picture shows the relationship between model likelihood [LnP(D)] and increasing K. We could see that after steep likelihood increase from K=1 up to K=4 the curve starts flattening. Hence, optimal K is the K where the curve flattens. The methods for detection this point sis called deltaK.



Next parameter that should be explored is whether the repeated runs for the same K converge to the similar solution or not. We could calculate Nordborg's coefficient of similarity for all pairs of runs for the same K. Usable K should have these values high and approaching 1. Following picture shows means and variances of this similarity coefficient for individual Ks. For K=2 and K=3 all repeated runs converge to the same solution (similarity coefficient is 1), all higher Ks have repeated solutions different.



Microsatellite data matrix for STRUCTURE should look like this:

```
vz1   01    88    148   -9    56
vz1   01    86    148   -9    59
vz2   01    88    146   23    56
vz2   01    88    148   25    59
vz3   02    88    148   -9    56
vz3   02    88    148   -9    56
vz4   02    88    146   23    56
vz4   02    88    146   25    59
```

In rows there are number encoding individual alleles in all loci; each locus is in a separate column (in our case the first locus has alleles 86 and 88, second locus alleles 146 and 148 etc.). The first two columns determine the sample and population. Caution: population code may not include text just numbers! Each sample is at two separate rows (in case of diploids). Mind that in case of heterozygous individuals/loci the alleles are different.

Running STRUCTURE locally with graphical user interface

1.  Start the program by clicking to this icon and first start a new project *File* → *New project*. Set the folder, select input file with the data. In the next window fill in number of individuals (not number of rows!), number of loci and write -9 as a value for missing data. In the next window leave everything unselected (in case of microsatellites). Next select that our input file includes both *Individual ID for each individual* and *Putative population origin*. Continue further and check whether the matrix was imported correctly.
2. Under *Parameter set* → *New* set *Length of Burnin Period* (e.g., 10 000 for clearly structured dataset) and *Number of MCMC Reps after Burnin* (e.g., 30 000); in next windows control that *Admixture ancestry model* and *Correlated Allele frequencies* are selected. Name this parameter set and save it.
3. The set of analyse sis run through *Project* → *Start a Job*. Select our above prepared and save parameter set and under *Set K from ... to* select the range of Ks we want to analyse (e.g., 1 and 10) and *Number of Iterations*, i.e., number of independent runs for each K (e.g., also 10). Start!
4. We can easily display analyses results after individual run finishes: Select desired run and go to *Bar plot* → *Show*. You can also display results of any project by *File* → *Open Project* to open the file with a project (*.spj) and in the folder „Parameter Sets/Name OfTheProject/Results“ left in the directory tree select desired run. By clicking the run is displayed in the right window. Similarly do *Bar plot* → *Show*. However, the software STRUCTURE does not allow easy comparison of more runs at the same time.

5. Processing of STRUCTURE results with the software Structure-sum

STRUCTURE-sum is a collection of functions for R. It allows processing outputs of all STRUCTURE runs and summarize the results as tables and graphs. The software also calculates deltaK for inferring the optimal number of clusters (populations) – see above.

First we need to prepare a text file describing which STRUCTURE output file run for which K. It looks as follows:

```
1     output_f.1
1     output_f.2
2     output_f.3
2     output_f.4
3     output_f.5
3     output_f.6
…
```

Save the file as `list.txt` to the same folder where we have STRUCTURE results (the folder was selected when we established the new STRUCTURE project; now there is a new subfolder with parameter set name and a subfolder „Results“ in it. Here are the results from all the runs which ends with a number and the character „f“. Now we can start  . Find and select the file with functions `Structure-sum-2009.r` using *File* → *Source R code…* With *File* → *Change dir…* select the folder „Results“ – see above). Continue with writing following commands at the command line:

1. `Structure.table ("list.txt", x)`
   - *x* is a number of populations in the input file

- generates a graph K vs. logarithm of the model likelihood [LnP(D)] – see above the comments to STRUCTURE

2. `Structure.simil ("list.txt", x)`

- generates a graph K vs. similarity coefficient among runs for the same K – see above the comments to STRUCTURE

3. `Structure.deltaK ("list.txt", x)`

- generates four graphs, bottom right graph shows K vs. deltaK, i.e., inferring of the optimal number of clusters (K)

6. Processing of STRUCTURE results using Distruct

The software Distruct provides graphical representation of probabilistic membership of samples to individual groups (clusters). It outputs *.ps file (postscript), which can be converted to PDF and shows the well-known colourfull column diagram (*bar plot*) where a priori defined populations/species are separated and a legend is added.

The program needs several input files for successful run:

- `*.indivq` (individual probabilities – from STRUCTURE *output file* for particular K)

```
1        1    (0)    2 :  0.083 0.917
2        2    (0)    2 :  0.218 0.782
3        3    (0)    2 :  0.236 0.764
4        4    (0)    2 :  0.152 0.848
```

- `*.popq` (population probabilities – from STRUCTURE *output file* for particular K)

```
 2:     0.119  0.881              62
 3:     0.824  0.176              79
 5:     0.155  0.845               5
18:     0.564  0.436               3
```

- `*.names` (numbers and corresponding population names)

```
2     pop1
3     pop2
5     pop3
18    pop4
```

- `*.perm` (colour names for bar plot)

```
1 blue
2 yellow
```

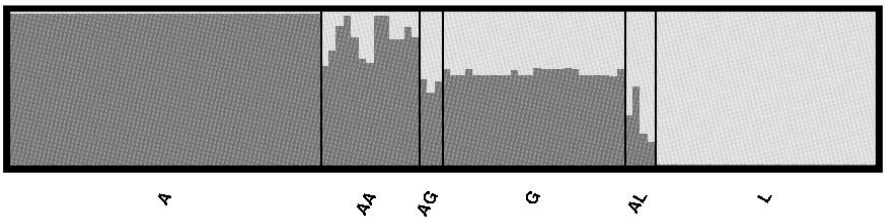
- `drawparams` (parameters for graph drawing – self-explanatory… – bold and grey-shaded parts should be edited). Comment: Do not change the name of this file nor do add a sufix to it!

```
"(int)" means that this takes an integer value.
"(B)"   means that this variable is Boolean
        (1 for True, and 0 for False)
"(str)" means that this is a string (but not enclosed in quotes)
"(d)"   means that this is a double (a real number).

Data settings

#define INFILE_POPQ        data.popq      // (str) input file of population q's
#define INFILE_INDIVQ      data.indivq    // (str) input file of individual q's
#define INFILE_LABEL_BELOW data.names     // (str) input file of labels for below figure
#define INFILE_CLUST_PERM  data.perm      // (str) input file of permutation of clusters to print
#define OUTFILE            data.ps        // (str) name of output file

#define K        2                        // (int) number of clusters
#define NUMPOPS  4                        // (int) number of pre-defined populations
#define NUMINDS  149                      // (int) number of individuals

Main usage options

#define PRINT_INDIVS      1  // (B) 1 if indiv q's are to be printed, 0 if only population q's
#define PRINT_SEP         1  // (B) print lines to separate populations

Figure appearance
```

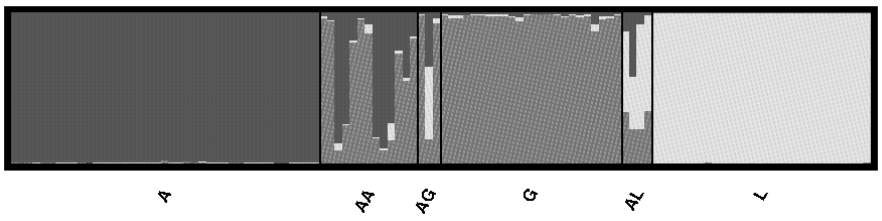
```
#define FONTHEIGHT 6      // (d) size of font
#define DIST_ABOVE 5      // (d) distance above plot to place text
#define DIST_BELOW -7     // (d) distance below plot to place text
#define BOXHEIGHT  36     // (d) height of the figure
#define INDIVWIDTH 1.5    // (d) width of an individual
```

All above mentioned files have to be in the same folder together with `distructWindows1.1.exe`. After cklicking on the exe file seemingly nothing happens. However, if everything went correct there is a new and non-zero sized file with a sufix *.ps in the same folder. This file should be converted to PDF format, e.g., using Ghostscript+GSView (http://pages.cs.wisc.edu/~ghost), or using an on-line servis (http://view.samurajdata.se/). The result could looks like:

K=2



K=3



7. PGDSpider – conversion among different data formats

The software could convert among ca. 30 different data formats used for coding population genetic/molecular data (e.g., Arlequin, BAPS, FSTAT, GENELAND, IM, MSA, NewHybrids, NEXUS, PHYLIP, Structure etc.). Software is possible to download from http://www.cmpg.unibe.ch/software/PGDSpider/ and requires Java (http://www.oracle.com/technetwork/java/javase/downloads/index.html; download JRE, not JDK!). Click on `PGDSpider2.exe` to run graphical version of the program. Select *Data Input File* | *File format:* (e.g., MSA) and with *Select input file* select a file of your choice. Similarly set *Data Output File* | *File format:* (e.g., NEWHYBRIDS). Then click on *Convert* and the program ask for additional information about data type or format of input/output file. Click on *Apply* starts the conversion and output file is generated. (We use the software for conversion from the MSA format to the NewHybrids and FSTAT format.)

8. FSTAT – basic population-genetic parameters for codominant data

The software FSTAT (http://www2.unil.ch/popgen/softwares/fstat.html) is intended for work with allelic (codominant) data and can for individual populations and loci calculate allelic frequencies, Nei´s gene diversity, inbreeding coefficient, F-statistics and also their estimates according to Weir & Cockerham (1984; F, θ, f). It also uses randomisation to calculate significant deviation from Hardy-Weinberg equilibrium.

The software FSTAT has its own data format but can possibly also import data in GENPOP format (this forma tis possible to export from MSA and in FSTAT use *Utilities → File Conversion → Genepop->Fstat*). Data in FSTAT format look as follows:

```
6 9 146 3
Loc_1
Loc_2
…
1   088105 139143 095098 143143 089096 088088 053000 046050 139140
1   105105 143143 091098 134143 089096 088088 053000 046050 139140
1   105105 144146 098098 136143 096096 088088 053000 045049 139140
2   089105 139143 091098 134143 089096 088088 053000 046051 139140
2   088105 138143 095098 134143 089096 088088 053000 046051 139140
2   090105 139143 095098 134143 089096 088088 053000 046050 139140
…
```

Numbers in the first row: number of populations, number of loci, highest allele number, 3=allele has 3 characters. Names of the loci and rows with data (population number first then allelic data) follow. Comment: FSTAT calls population as '*sample*'.
We import data using *File → Open*. Then select (by clicking) what should be calculated and click on *Run*. Software writes all the results to the file with suffix *.out. In the results we find following:

- number and frequencies of individual alleles in particular locus – for each locus the allelic frequency in each population and averaged allelic frequency (over populations) is shown
- gene diversities, i.e., expected heterozygosity (H_e) for individual loci and populations
- inbreeding coefficient per locus and population (F_{IS})
- heterozygosity estimate according Nei (H_o, H_S, H_T, G_{ST} etc.)
- estimate of F_{IT} (CapF – F), F_{ST} (theta – θ) and F_{IS} (smallF – f) per each locus and allele across all populations, also jackknife estimate across populations (mean and SE) and bootstrapping across populations with 99% and 95% confidence interval

9. NewHybrids – Bayesian analysis of hybrid individuals and their parents

This software (http://ib.berkeley.edu/labs/slatkin/eriq/software/software.htm) is intended for hybrid identification between two species. For each individual the probability of belonging to each of the pre-defined hybrid classes (F1, F2, diverse types of back-crosses) and/or clear parental species is computed using Bayesian clustering with MCMC. Individuals could be assigned to more classes. I tis not necessary to a priori define what are the clear parental species. Input file for NewHybrids could be prepared, e.g., using PGDSpider. Data file for microsatellites looks as follows:

```
NumIndivs 114
NumLoci 9
Digits 3
Format Lumped
1 088105 139143 095098 143143 089096 088088 053000 046050 139140
2 105105 143143 091098 134143 089096 088088 053000 046050 139140
3 105105 144146 098098 136143 096096 088088 053000 045049 139140
4 105105 143143 091098 143143 096096 088088 053000 050051 139140
5 105105 143143 095098 134143 096096 088088 053000 050051 139140
…
```

In case of AFLP data the coding is +/– . File with data matrix has to be saved to the same folder where we have the software and then run the file `NewHybrids_PC_1_1.exe`. Enter the input file name and answer '0' when asked for '*genotype frequency classes*' and '*prior allele frequency information*'. Enter two numbers for random number generator (which dos not influence the analysis), press *Enter* and a window is opened (*Info Window*), where after pressing *Space* the analysis starts. By pressing '1' many other windows are opened showing the progress of MCMC simulations and means for diverse parameters. To top left windows show the probabilistic assignment of individuals to particular classes – *Category Probabilities* (for current MCMC step and a mean). Bottom window (*Data LogL Trace*) shows graph of logarithm of MCMC probability – it is necessary to scale it by clicking on it and by pressing '*v*', and then we see how after chain stabilization the value oscillates around a particular value. Burn-in values can be removed from the mean after the chain is stabilized by clicking to 'Info Window' and pressing '*e*'. It resets the mean and the mean is calculated only from values after pressing '*e*'. Legend in each window can be called by pressing '*L*'. Many other settings are possible using pop-up menu which is displayed by right mouse click in each window. Program saves results to several output files; the most important is `aa-PofZ.txt`, which includes for each sample its posterior probabilities to each of the pre-defined hybrid classes.