

# Master – R-lequin

## Table of contents:

• Mismatch distribution: .....	1
• Heterozygosity: .....	2
• FST Matrix:.....	2
• Allelic size range: .....	3
• Relative FIS per locus:.....	4
• Haplotype distance matrix: (line by line Data input!) .....	5
• Expected/Observed Haplotype : (combination of 2 different tables) .....	6
• Haplotype distance matrix: (line by line Data input!) – improved.....	8
• Allelic size range with lines: .....	10
• Haplotype distance matrix between/within two Populations:.....	11
• Haplotype distance matrix between/within two Populations: (with mixed data from several populations) .....	13
• Expected/Observed Haplotype : save graphic as a .png file.....	15
• Read data between two XML tags and convert to numeric matrix (Exp. table): ....	19
• Move the legend in matrix plots with legend to the print area: .....	22
• Population assignment test .....	24
• Haplotype distance matrix between/within populations and groups.....	25
• Haplotype frequencies in population.....	29
• Molecular diversity indexes .....	30
• Divergence times allowing for unequal population sizes (tau) .....	31
• Population average pairwise difference .....	33
• Read data between two XML tags and convert to numeric matrix (improved): ....	37
○ Half matrix: .....	37
○ Full matrix:.....	38
○ Table with names: .....	39

# Master – R-lequin

## Week 1-2 (15.10.07 – 26.10.07)

- Read R-Tutorials
- Mismatch distribution:  
(lines\_Mismatch.r)

```
read.table("D:/Heidi/Master/R_Daten/Mismatch/Mismatch_mt4.txt") -> mismatch4
mismatch4

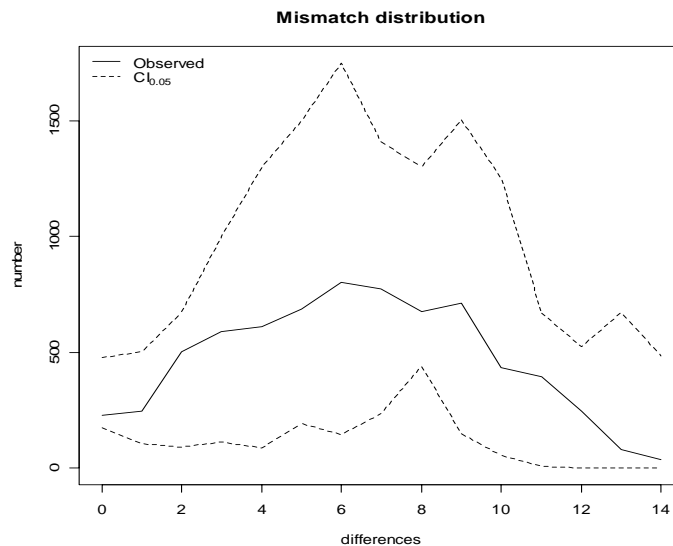
attach(mismatch4)

V1 -> Diff
V2 -> Observed
V3 -> Low.bound
V4 -> Up.bound
V5 -> ModelFreq.

max(Up.bound) -> max1

plot(Diff, Observed, type="l", ylim=c(0,max1), xlab="differences",
      ylab="number", main="Mismatch distribution")
lines(Diff, Low.bound, lty=2)
lines(Diff, Up.bound, lty=2)
legend("topleft", expression (Observed, CI[0.05]), lty=1:2, bty="n")

detach(mismatch4)
```



**Problems:**

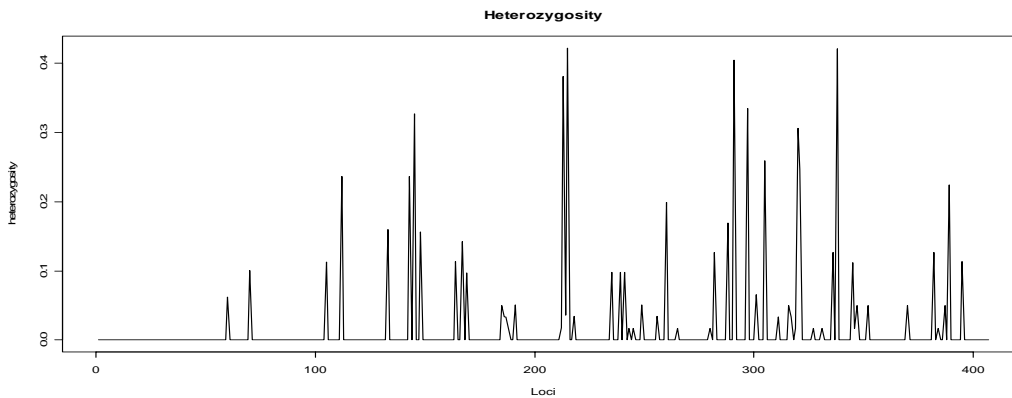
- **Heterozygosity:**  
(line\_Heterozygosity.r)

```
read.table("D:/Heidi/Master/R_Daten/Heterozygosity/heterocy_mt.txt") -> heterocy
attach(heterocy)
```

```
V1 -> Locus
V2 -> het
V3 -> mean
V4 -> sd
V5 -> total.het
```

```
plot(Locus, het, type="l", xlab="Loci", ylab="heterozygosity",
     main="Heterozygosity")
```

```
detach(heterocy)
```



**Problems:**

- **FST Matrix:**  
(matrix\_FstMatrix.r)

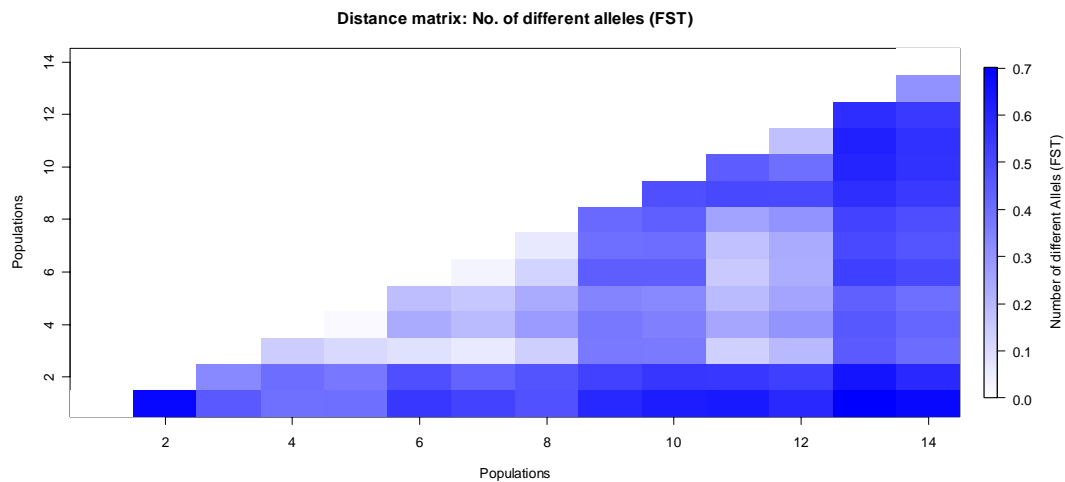
```
read.table("D:/Heidi/Master/R_Daten/FstMatrix/DistanceMatrix_mic.txt",
           header=TRUE, skip=1, row.names=1, fill=TRUE) -> Data
```

```
as.matrix.data.frame(Data) -> Matrix
```

```
Matrix
a <- ncol(Matrix)
b <- nrow(Matrix)
x <- c(1:a)
y <- c(1:b)
```

```
library(fields)
ColorRamp <- rgb( seq(1,0,length=256), # Red
                  seq(1,0,length=256), # Green
                  seq(1,1,length=256)) # Blue
```

```
image.plot(x,y,Matrix, col=ColorRamp, main="Distance matrix: No. of
different alleles (FST)", xlab="Populations",
           ylab="Populations", legend.lab="Number of different Allels
(FST) ")
```



**Problems:**

- Achse
- spiegeln

- Allelic size range:  
(bar\_AllelicSizeRang.r)

```
read.table("D:/Heidi/Master/R_Daten/AllelicSizeRange/AllelicSizeRange_mic.txt",
           skip=4, row.names=1, fill=TRUE )-> Data

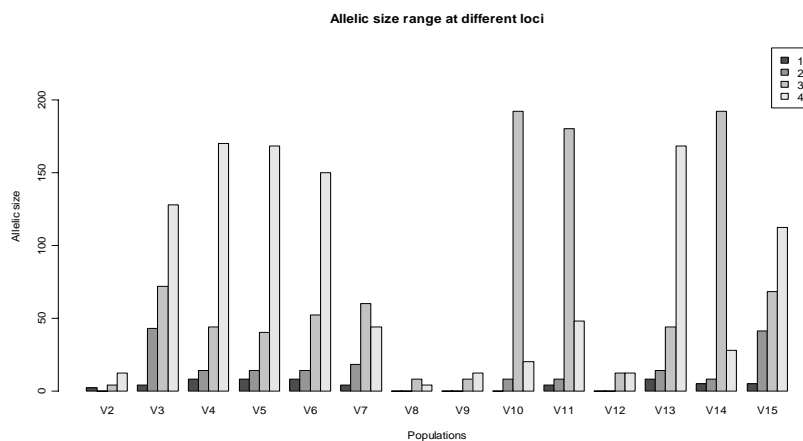
Data

nrow(Data) -> a
ncol(Data) -> b

Data2 <- as.matrix.data.frame(Data)

Data3 <- Data2[1:(a-2),1:(b-3)]
Data3

barplot(Data3, beside=TRUE, legend.text=TRUE, main="Allelic size range
at different loci", xlab="Populations", ylab="Allelic size",
        ylim=c(0, (max(Data3)+(max(Data3)/4))))
```



**Problems:**

- Legende beschriften

- Relative FIS per locus:  
(bar\_FisLocusRelative.r)

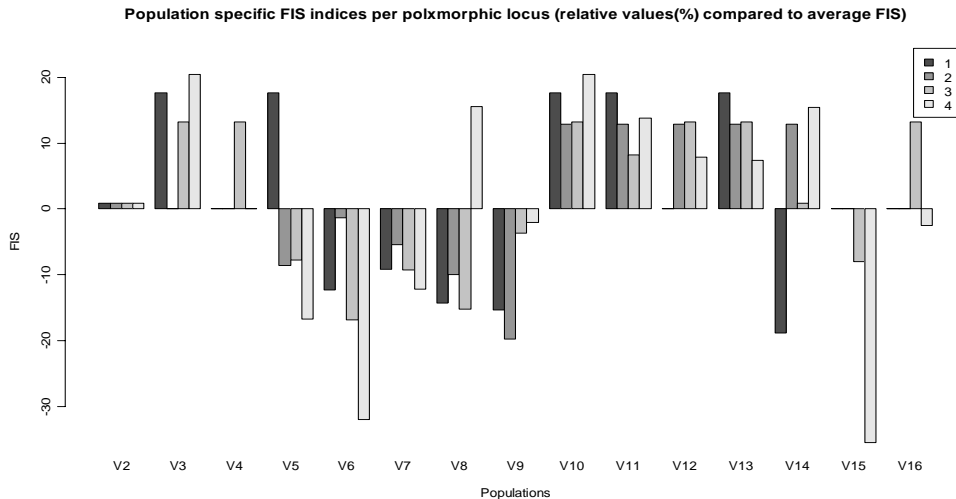
```
read.table("D:/Heidi/Master/R_Daten/Fis/FisLocus_mic.txt", skip=16,
           row.names=1, fill=TRUE, nrows=4, na.strings= "N.A.") -> Data
Data

#wandelt alle NA in 0 um:
for(i in 1:ncol(Data)) if(is.numeric(Data[,i])) Data[is.na(Data[,i]),
i] <- 0
Data

nrow(Data) -> a
ncol(Data) -> b

Data2 <- as.matrix.data.frame(Data)

barplot(Data2, beside=TRUE, legend.text=TRUE, main="Population specific
FIS indices per polymorphic locus (relative values(%) compared
to average FIS)", xlab="Populations", ylab="FIS",
ylim=c(min(Data2), (max(Data2) + (max(Data2)/4))))
```

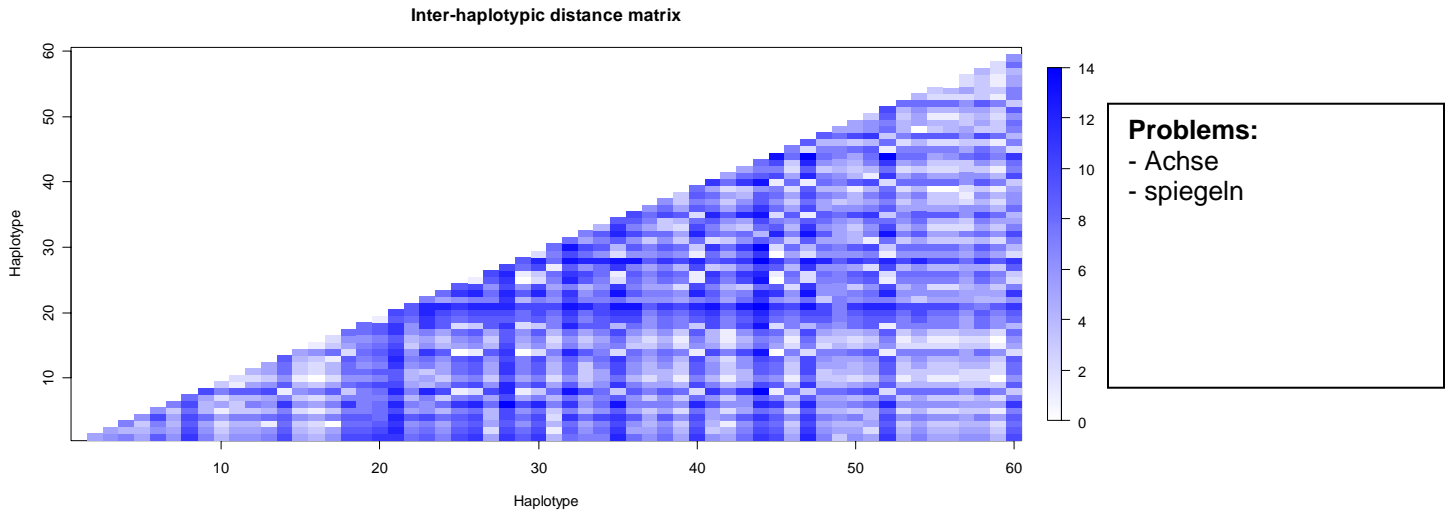


**Problems:**  
- Legende beschriften

## Week 2-3 (29.10.07 – 02.11.07)

- Haplotype distance matrix: (line by line Data input!)  
(matrix\_HapDistanceMatrix.r)

```
Data <-  
read.table("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatrix_mt.txt"  
           , skip=1)  
  
Columns <- ncol(Data) + 1  
Row <- nrow(Data)  
  
x <- 3  
n <- 1  
  
DistanceMatrix <-  
  as.matrix(scan("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatrix_mt.txt", what=double(0), skip=x, nlines=1, nmax=n), row.names=1)  
DistanceMatrix <- cbind(DistanceMatrix, matrix(NA, ncol=(Columns-n),  
                                                nrow=1))  
DistanceMatrix <- DistanceMatrix[,2:Columns]  
  
n <- n + 1  
x <- x + 1  
  
while(n<(Row+1)) {  
  nextrow <-  
    as.matrix(scan("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatrix_mt.txt", what=double(0), skip=x, nlines=1, nmax=n))  
  nextrow <- cbind(t(nextrow), matrix(NA, ncol=(Columns-n), nrow=1))  
  nextrow <- nextrow[,2:Columns]  
  
  DistanceMatrix <- rbind(DistanceMatrix, nextrow)  
  
  n <- n + 1  
  x <- x + 1  
}  
  
a <- ncol(DistanceMatrix)  
b <- nrow(DistanceMatrix)  
  
x <- c(1:a)  
y <- c(1:b)  
  
library(fields)  
ColorRamp <- rgb(seq(1,0,length=256), # Red  
                  seq(1,0,length=256), # Green  
                  seq(1,1,length=256)) # Blue  
  
image.plot(x,y,DistanceMatrix, col=ColorRamp, main="Inter-haplotypic  
distance matrix", xlab="Haplotype", ylab="Haplotype" )
```



- Expected/Observed Haplotype : (combination of 2 different tables)  
(bar2\_HaplotypeFreq.r)

```
read.table("D:/Heidi/Master/R_Daten/HaplotypeFrequency/ObsHaplotypeFreq_mt.txt"
           , skip=5, row.names=1, fill=TRUE )-> Observed

nrow(Observed) -> rObs
ncol(Observed) -> cObs

newObserved <- Observed[1:(rObs-1),1:(cObs-1)]

read.table("D:/Heidi/Master/R_Daten/HaplotypeFrequency/ExpHaplotypeFreq_mt.txt"
           , skip=6, row.names=1, fill=TRUE) -> Expected

nrow(Expected) -> rExp
ncol(Expected) -> cExp

newExpected <- Expected[,1:(cExp-3)]

HaplotypeFreq <- as.matrix(newExpected)
HaplotypeFreq <- cbind(HaplotypeFreq, as.matrix(newObserved))
tHaplotypeFreq <- t(HaplotypeFreq)

null <- matrix(0, ncol=1, nrow=rExp)
ExpSd <- cbind((Expected[,2]), as.matrix(null))
sdpos <- tHaplotypeFreq + t(ExpSd)
sdneg <- tHaplotypeFreq - t(ExpSd)

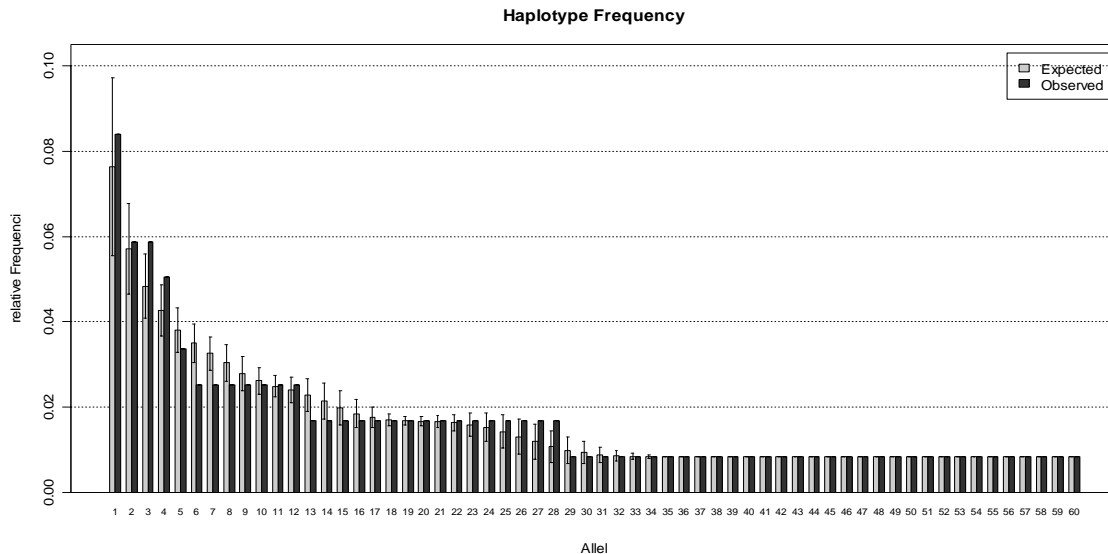
library(gplots)

mp <- barplot2(tHaplotypeFreq ,beside=TRUE, main="Haplotype Frequency",
               legend = c("Expected", "Observed"), xlab="Allel",
               ylab="relative Frequenci",
               ylim=c(0,(max(tHaplotypeFreq)+(max(HaplotypeFreq)/4))),
               plot.ci = TRUE, ci.l =sdneg, ci.u =sdpos,
               plot.grid = TRUE , col=c( gray(0.8), gray(0.2)))
```

```

mtext(side = 1, at = colMeans(mp), line = 0.5, text =
      c(1:rExp), cex=0.7)
box()

```



**Problems:**

(lines\_HaplotypeFreq.r)

```

read.table("D:/Heidi/Master/R_Daten/HaplotypeFrequency/ObsHaplotypeFreq_mt.txt",
           skip=5, row.names=1, fill=TRUE) -> Observed

nrow(Observed) -> rObs
ncol(Observed) -> cObs

newObserved <- Observed[1:(rObs-1), 1:(cObs-1)]

read.table("D:/Heidi/Master/R_Daten/HaplotypeFrequency/ExpHaplotypeFreq_mt.txt",
           skip=6, row.names=1, fill=TRUE) -> Expected

nrow(Expected) -> rExp
ncol(Expected) -> cExp

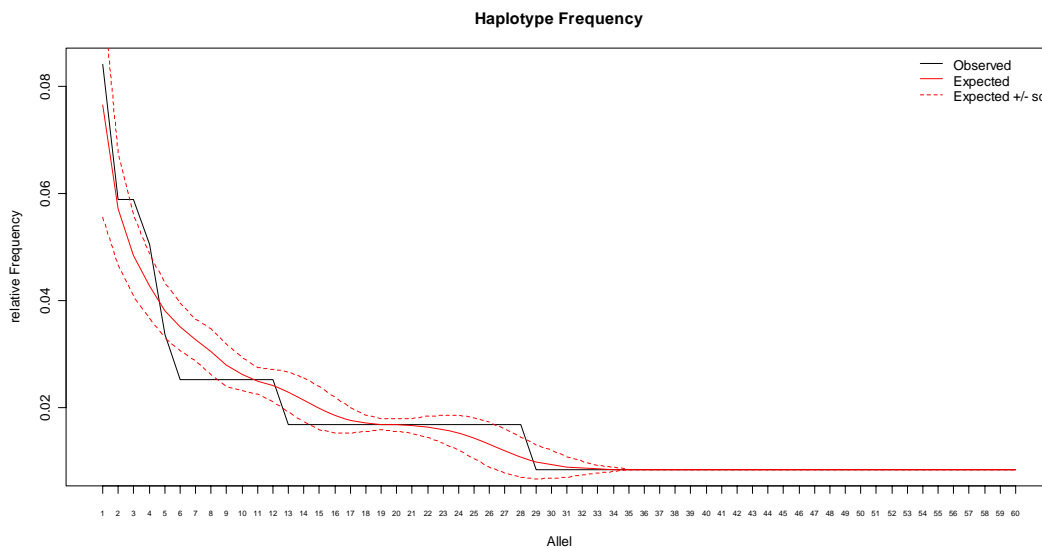
newExpected <- Expected[, 1:(cExp-3)]

sdExp <- Expected[, 2]
sdExpUp <- newExpected + sdExp
sdExpLo <- newExpected - sdExp

plot(newObserved, type="l", xlab="Allel", ylab="relative Frequency",
     main="Haplotype Frequency", xaxt="n")
axis(1, 1:rExp, cex.axis=0.6)
lines(newExpected, col="red")
lines(sdExpUp, lty=2, col="red")
lines(sdExpLo, lty=2, col="red")
legend("topright", c("Observed", "Expected", "Expected +/- sd"),
      lty=c(1,1,2), bty="n", col=c("black", "red", "red"))

```





**Problems:**

- Haplotype distance matrix: (line by line Data input!) – **improved**  
(matrix\_HapDistanceMatrix.r)

```
Data <-
read.table("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatrix_mt.txt"
, skip=1)

Columns <- ncol(Data) + 1
Row <- nrow(Data)

x <- 3
n <- 1

DistanceMatrix <-
  as.matrix(scan("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatr
ix_mt.txt", what=double(0), skip=x, nlines=1, nmax=n))
DistanceMatrix <- cbind(DistanceMatrix, matrix(NA, ncol=(Columns-n), nrow=1))
DistanceMatrix <- DistanceMatrix[,2:Columns]

n <- n + 1
x <- x + 1

while(n<(Row+1)) {
  nextrow <-
    as.matrix(scan("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatr
ix_mt.txt", what=double(0), skip=x, nlines=1, nmax=n))
  nextrow <- cbind(t(nextrow), matrix(NA, ncol=(Columns-n), nrow=1))
  nextrow <- nextrow[,2:Columns]

  DistanceMatrix <- rbind(DistanceMatrix, nextrow)

  n <- n + 1
  x <- x + 1
}
```

```

a <- ncol(DistanceMatrix)
b <- nrow(DistanceMatrix)

x <- c(1:a)
y <- c(1:b)

# Mirror matrix (left-right)
mirror.matrix <- function(x) {
  xx <- as.data.frame(x);
  xx <- rev(xx);
  xx <- as.matrix(xx);
  xx;
}

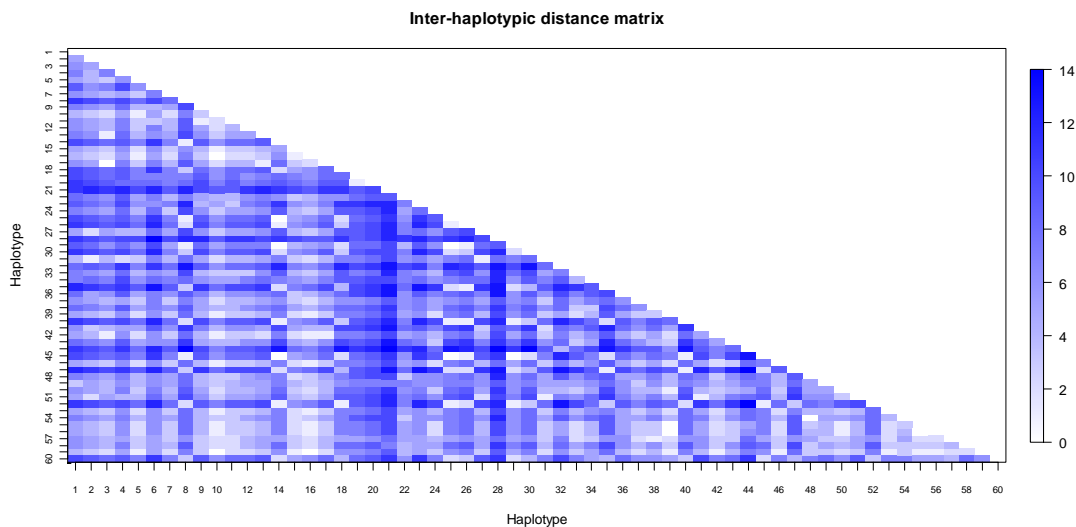
# Rotate matrix 270 clockworks
rotate270.matrix <- function(x) {
  mirror.matrix(t(x))
}

DistanceMatrix <- rotate270.matrix(DistanceMatrix)

library(fields)
ColorRamp <- rgb( seq(1,0,length=256), # Red
                  seq(1,0,length=256), # Green
                  seq(1,1,length=256)) # Blue

image.plot(x,y,DistanceMatrix, col=ColorRamp, main="Inter-haplotypic
distance matrix", xlab="Haplotype", ylab="Haplotype",
          axes = FALSE)
contour(DistanceMatrix, add = TRUE)
axis(1, at = c(1:a),cex.axis=0.7)
axis(2, at = c(1:b), labels=c(b:1),cex.axis=0.7)
box()

```



**Problems:**

## Week 4-5 (05.11.07 – 16.11.07)

- Allelic size range with lines:  
(lines\_AllelicSizeRang.r)

```
read.table("D:/Heidi/Master/R_Daten/AllelicSizeRange/AllelicSizeRange_mic.txt",
           skip=4, row.names=1, fill=TRUE )-> Data

a <- nrow(Data)
b <- ncol(Data)

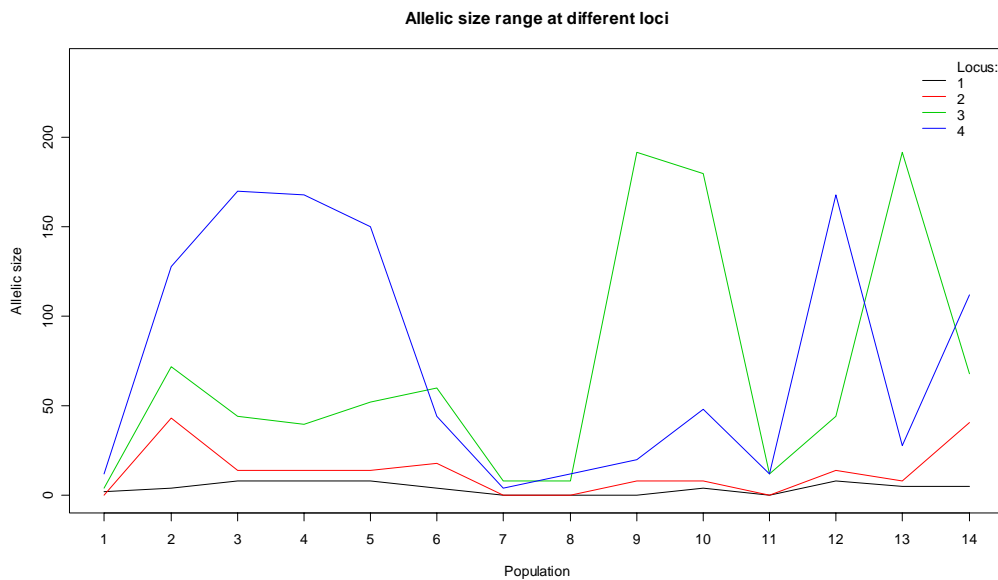
Data <- as.matrix.data.frame(Data)

Data <- Data[1:(a-2),1:(b-3)]
nRow <- nrow(Data)
nCol <- ncol(Data)

x <- 1

plot(Data[x,], type="l", xlab="Population", ylab="Allelic size",
     main="Allelic size range at different loci",
     ylim=c(0,(max(Data)+(max(Data)/4))), xaxt="n", col=x, lwd=1.5)

x <- x + 1
while(x <= nRow){
  lines(Data[x,], col=x, lwd=1.5)
  x <- x + 1
}
axis(1, 1:nCol)
legend("topright", legend=c("Locus:", (1:(x-1))), lty=1, bty="n",
      col=c(0:x), lwd=1.5)
```



**Problems:**

- Haplotype distance matrix between/within two Populations:  
(matrix\_HapDistance\_between-within.r)

```
#----read haplotype list----
Data <-
read.table("D:/Heidi/Master/R_Daten/HaplotypeDistance/ListHaplotype_betweenBsp.
           txt",skip=1)

Row <- nrow(Data)
Columns <- Row

#----read data row by row:----
x <- 0
n <- 1

DistanceMatrix <-
  as.matrix(scan("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatr
                ix_betweenBsp.txt", what=double(0), skip=x, nlines=1, nmax=n))
DistanceMatrix <- cbind(DistanceMatrix, matrix(NA, ncol=Columns, nrow=1))
DistanceMatrix <- DistanceMatrix[,1:Columns]

n <- n + 1
x <- x + 1

while(n<(Row+1)){
  nextrow <-
  as.matrix(scan("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatrix_bet
                weenBsp.txt", what=double(0), skip=x, nlines=1, nmax=n))
  nextrow <- cbind(t(nextrow), matrix(NA, ncol=Columns, nrow=1))
  nextrow <- nextrow[,1:Columns]

  DistanceMatrix <- rbind(DistanceMatrix, nextrow)

  n <- n + 1
  x <- x + 1
}

#----Mirror matrix (left-right)----
mirror.matrix <- function(x) {
  xx <- as.data.frame(x);
  xx <- rev(xx);
  xx <- as.matrix(xx);
  xx;
}

#----Rotate matrix 270 clockworks----
rotate270.matrix <- function(x) {
  mirror.matrix(t(x))
}

DistanceMatrix <- rotate270.matrix(DistanceMatrix)

#----draw matrix----
library(fields)
ColorRamp <- rgb( seq(1,0,length=256), # Red
                  seq(1,0,length=256), # Green
                  seq(1,1,length=256)) # Blue
```

```

a <- ncol(DistanceMatrix)
b <- nrow(DistanceMatrix)

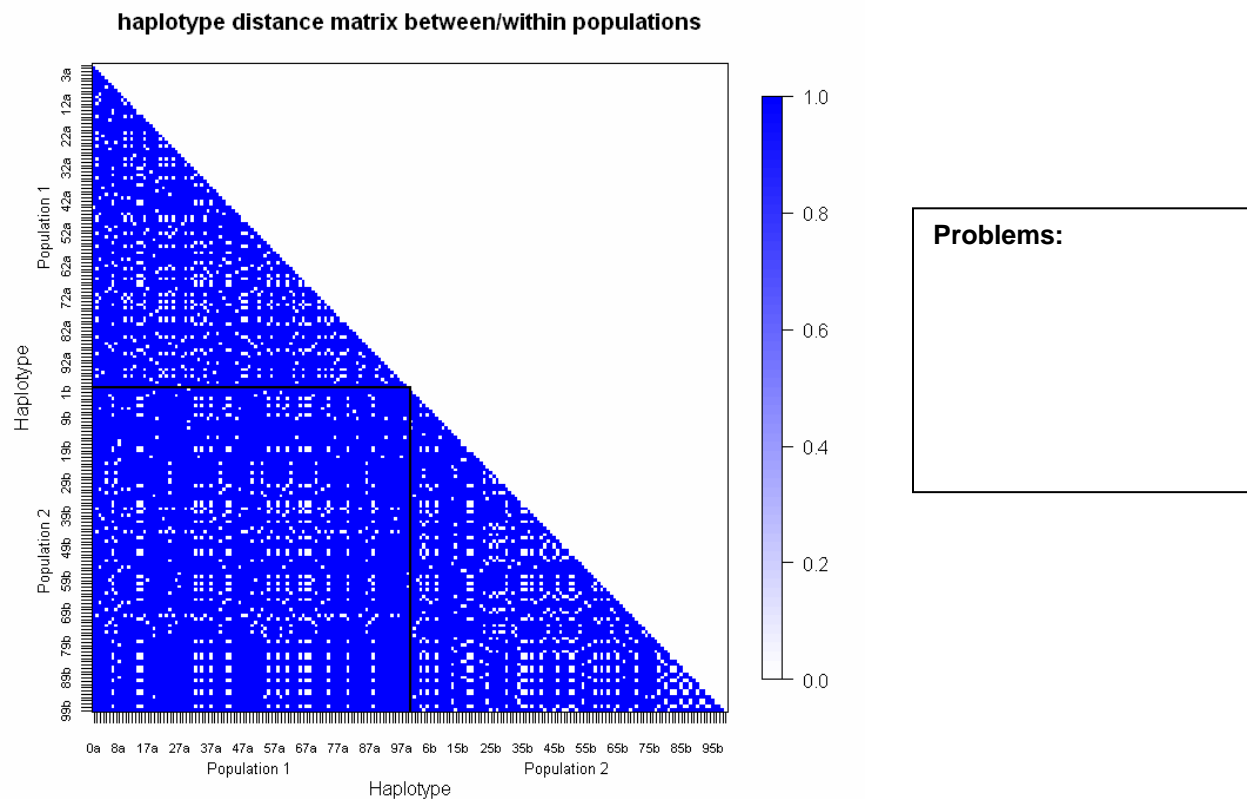
x <- c(1:a)
y <- c(1:b)

image.plot(x,y,DistanceMatrix, col=ColorRamp, main="haplotype distance matrix
          between/within populations", xlab="Haplotype", ylab="Haplotype",
          axes = FALSE)
contour(DistanceMatrix, add = TRUE)
axis(1, at = c(1:a), labels=Data[,1], cex.axis=0.7)
axis(2, at = c(1:b), labels=Data[(Row:1),1], cex.axis=0.7)
box()

half <- (Row/2) + 0.5
lines(c(0,half),c(half,half), lwd=2)
lines(c(half,half),c(0,half), lwd=2)

mtext(side = 1, at =(Row/4), line = 2, text = "Population 1", cex=0.8)
mtext(side = 1, at =(3*Row/4), line = 2, text = "Population 2",
      cex=0.8)
mtext(side = 2, at =(Row/4), line = 2, text = "Population 2", cex=0.8)
mtext(side = 2, at =(3*Row/4), line = 2, text = "Population 1",
      cex=0.8)

```



- Haplotype distance matrix between/within two Populations: (with mixed data from several populations)

(matrix\_HapDistance\_between-within\_complex.r)

```
#----read haplotype list----
Data <-
read.table("D:/Heidi/Master/R_Daten/HaplotypeDistance/ListHaplotype_betweenPop.
          txt", skip=1)

Row <- nrow(Data)
Columns <- Row

#----read data row by row:----
x <- 0
n <- 1

DistanceMatrix <-
as.matrix(scan("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatrix_bet
              weenPop.txt", what=double(0), skip=x, nlines=1, nmax=n))
DistanceMatrix <- cbind(DistanceMatrix, matrix(NA, ncol=Columns, nrow=1))
DistanceMatrix <- DistanceMatrix[,1:Columns]

n <- n + 1
x <- x + 1

while(n<(Row+1)){
  nextrow <-
  as.matrix(scan("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatr
                ix_betweenPop.txt", what=double(0), skip=x, nlines=1, nmax=n))
  nextrow <- cbind(t(nextrow), matrix(NA, ncol=Columns, nrow=1))
  nextrow <- nextrow[,1:Columns]

  DistanceMatrix <- rbind(DistanceMatrix, nextrow)

  n <- n + 1
  x <- x + 1
}

#----select the data of DistanceMatrix----
dimnames(DistanceMatrix) <- list(Data[,1], Data[,1])

#--population1:
Pop1 <-
read.table("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatrix_withinP
          op1.txt", skip=5)
Pop1 <- as.character(Pop1[,1])
DistanceMatrixPop1 <- DistanceMatrix[Pop1, Pop1]

#--population2:
Pop2 <-
read.table("D:/Heidi/Master/R_Daten/HaplotypeDistance/HapDistanceMatrix_withinP
          op2.txt", skip=5)
Pop2 <- as.character(Pop2[,1])
DistanceMatrixPop2 <- DistanceMatrix[Pop2, Pop2]
```

```

#--between population1/2:
#whole DistanceMatrix
wholeDistanceMatrix <- DistanceMatrix
x <- 1
while(x <= ncol(wholeDistanceMatrix)){

  twholeDistanceMatrix <- t(wholeDistanceMatrix)
  wholeDistanceMatrix[x,(x:ncol(wholeDistanceMatrix))] <-
    twholeDistanceMatrix[x,(x:ncol(wholeDistanceMatrix))]

  x <- x + 1
}

wholeDistanceMatrixBetween <- wholeDistanceMatrix[Pop2, Pop1]

#----together:-----
DistanceMatrixUp <- cbind(DistanceMatrixPop1, matrix(NA,
  ncol(DistanceMatrixPop1), nrow(DistanceMatrixPop2)))
DistanceMatrixLo <- cbind(wholeDistanceMatrixBetween, DistanceMatrixPop2)
DistanceMatrixTogether <- rbind(DistanceMatrixUp, DistanceMatrixLo)

#----Mirror matrix (left-right)----
mirror.matrix <- function(x) {
  xx <- as.data.frame(x);
  xx <- rev(xx);
  xx <- as.matrix(xx);
  xx;
}

#----Rotate matrix 270 clockworks----
rotate270.matrix <- function(x) {
  mirror.matrix(t(x))
}

DistanceMatrixTogether <- rotate270.matrix(DistanceMatrixTogether)

#----draw matrix----
library(fields)
ColorRamp <- rgb( seq(1,0,length=256), # Red
  seq(1,0,length=256), # Green
  seq(1,1,length=256)) # Blue

a <- ncol(DistanceMatrixTogether)
b <- nrow(DistanceMatrixTogether)

x <- c(1:a)
y <- c(1:b)

image.plot(x,y,DistanceMatrixTogether, col=ColorRamp, main="haplotype distance
  matrix between/within populations", xlab="Haplotype",
  ylab="Haplotype", axes = FALSE)
contour(DistanceMatrixTogether, add = TRUE)
axis(1, at = c(1:a), labels=c(Pop1,Pop2), cex.axis=0.6)
axis(2, at = c(1:b), labels=c(Pop2[NROW(Pop2):1], Pop1[NROW(Pop1):1]),
  cex.axis=0.6)
box()

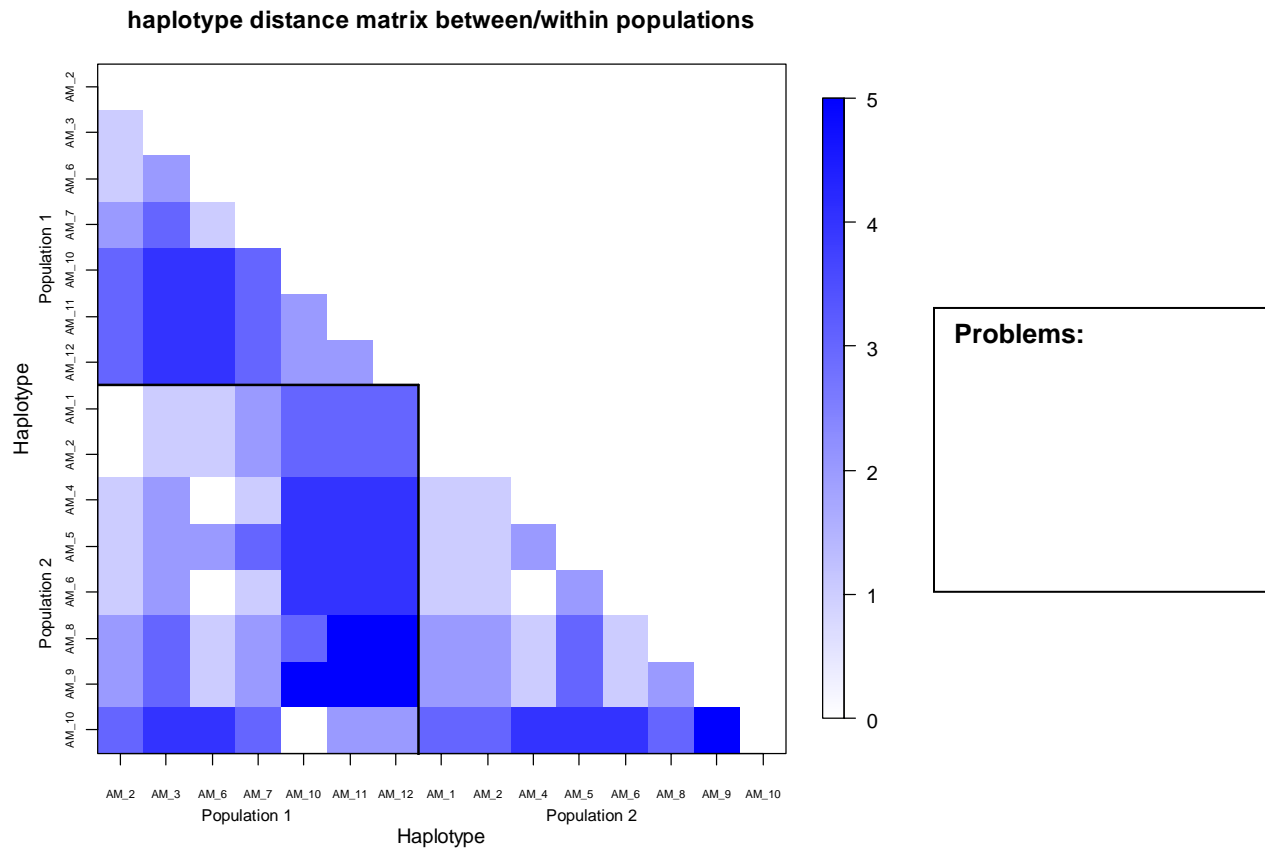
```

```

lines(c(0,NROW(Pop1)+0.5),c(NROW(Pop2)+0.5,NROW(Pop2)+0.5), lwd=2)
lines(c(NROW(Pop1)+0.5,NROW(Pop1)+0.5),c(0,NROW(Pop2)+0.5), lwd=2)

mtext(side = 1, at =(Row/4), line = 2, text = "Population 1", cex=0.8)
mtext(side = 1, at =(3*Row/4), line = 2, text = "Population 2", cex=0.8)
mtext(side = 2, at =(Row/4), line = 2, text = "Population 2", cex=0.8)
mtext(side = 2, at =(3*Row/4), line = 2, text = "Population 1", cex=0.8)

```



- Expected/Observed Haplotype : **save graphic as a .png file**  
(lines\_HaplotypeFreq.r)

```

read.table("D:/Heidi/Master/R_Daten/HaplotypeFrequency/ObsHaplotypeFreq_mt.txt"
, skip=5 ,row.names=1, fill=TRUE )-> Observed

nrow(Observed) -> rObs
ncol(Observed) -> cObs

newObserved <- Observed[1:(rObs-1),1:(cObs-1)]

read.table("D:/Heidi/Master/R_Daten/HaplotypeFrequency/ExpHaplotypeFreq_mt.txt"
, skip=6, row.names=1, fill=TRUE) -> Expected

```



```

nrow(Expected) -> rExp
ncol(Expected) -> cExp

newExpected <- Expected[,1:(cExp-3)]

sdExp <- Expected[,2]
sdExpUp <- newExpected + sdExp
sdExpLo <- newExpected - sdExp

png("D:/Heidi/Master/R_Graphiken/lines_HaplotypeFreq.png", width=550,
     height=550)

plot(newObserved, type="l", xlab="Allel", ylab="relative Frequency",
     main="Haplotype Frequency", xaxt="n")
axis(1, 1:rExp, cex.axis=0.6)
lines(newExpected, col="red")
lines(sdExpUp, lty=2, col="red")
lines(sdExpLo, lty=2, col="red")
legend("topright", c("Observed", "Expected", "Expected +/- sd"),
     lty=c(1,1,2), bty="n", col=c("black", "red", "red"))

dev.off()

```

- **Fst Matirx: read data from a HTML (XML) file**  
(matrix\_PairwiseFst\_HTML.r)

```

#----read HTML/XML file----
whole <-
  readLines("D:/Heidi/Master/R_Daten/XML/PairwiseFst_HTML_geändert.html")
  whole

x <- 1
while(x <= length(whole)){
  if(whole[x] == "<FstDistanceMatrix>"){
    Begin <- x
  }
  if(whole[x] == "</FstDistanceMatrix>"){
    End <- x
  }
  x <- x + 1
}

#----how much rows to read----
Data <- scan("D:/Heidi/Master/R_Daten/XML/PairwiseFst_HTML_geändert.html",
            skip=End-2, nlines=1)

Columns <- (length(Data))
Row <- Columns

#----read data row by row:----
x <- (Begin+2)
n <- 1

DistanceMatrix <-
as.matrix(scan("D:/Heidi/Master/R_Daten/XML/PairwiseFst_HTML_geändert.html",
              what=double(0), skip=x, nlines=1, nmax=n))

```

```

DistanceMatrix <- cbind(DistanceMatrix, matrix(NA, ncol=(Columns), nrow=1))
DistanceMatrix <- DistanceMatrix[,2:Columns]

n <- n + 1
x <- x + 1

while(n<(Row)){
  nextrow <-
    as.matrix(scan("D:/Heidi/Master/R Daten/XML/PairwiseFst_HTML_geändert.html",
                  what=double(0), skip=x, nlines=1, nmax=n))
  nextrow <- cbind(t(nextrow), matrix(NA, ncol=(Columns), nrow=1))
  nextrow <- nextrow[,2:Columns]

  DistanceMatrix <- rbind(DistanceMatrix, nextrow)

  n <- n + 1
  x <- x + 1
}

#----Mirror matrix (left-right)----
mirror.matrix <- function(x) {
  xx <- as.data.frame(x);
  xx <- rev(xx);
  xx <- as.matrix(xx);
  xx;
}

#----Rotate matrix 270 clockworks----
rotate270.matrix <- function(x) {
  mirror.matrix(t(x))
}

DistanceMatrix <- rotate270.matrix(DistanceMatrix)

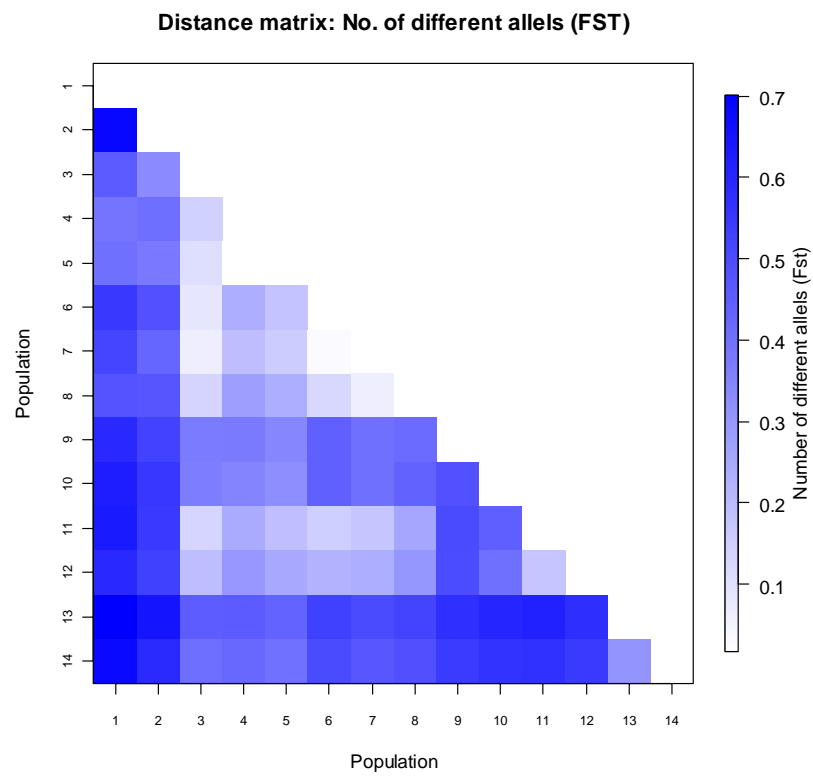
#----draw matrix plot----
library(fields)

a <- ncol(DistanceMatrix)
b <- nrow(DistanceMatrix)
x <- c(1:a)
y <- c(1:b)

ColorRamp <- rgb( seq(1,0,length=256), # Red
                  seq(1,0,length=256), # Green
                  seq(1,1,length=256)) # Blue

image.plot(x,y,DistanceMatrix, col=ColorRamp, main="Distance matrix: No. of
different alleles (FST)", xlab="Population",
ylab="Population", axes = FALSE, legend.mar=4.3,
legend.width=0.8, legend.lab="Number of different alleles (Fst)")
contour(DistanceMatrix, add = TRUE)
axis(1, at = c(1:a),cex.axis=0.7)
axis(2, at = c(1:b), labels=c(b:1),cex.axis=0.7)
box()

```



**Problems:**

## Week 6 (19.11.07 – 23.11.07)

- Read data between two XML tags and convert to numeric matrix (Exp. table):  
(Node auslesen\_BspTabelle.r)

```
#----open XML package-----
library(XML)

#----read data between an XML tag-----
filename = "D:/Heidi/Master/R_Daten/XML/Beispiel.xml"
tag = "//Fst"
doc = xmlTreeParse(filename, useInternal = TRUE)
ch = getNodeSet(doc, tag)

subDoc = xmlDoc(ch[[1]])
tagData <- xpathApply(subDoc, tag, xmlValue)
free(subDoc)
#print(tagData, indent=FALSE)

#----convert string data to a numeric matrix-----
#----split string----
tagData2 <- as.character(tagData)

tagData3 <- strsplit(tagData2, "\n")
tagMatrix <- as.matrix(as.data.frame(tagData3))
tagMatrix <- tagMatrix[2:nrow(tagMatrix)]

Data <- strsplit(tagMatrix, " ")
Matrix <- as.matrix(as.data.frame(Data))

#----to numeric matrix----
numericList <- as.numeric(Matrix[,1])
numericMatrix <- t(as.matrix(numericList))

for(n in 2:nrow(Matrix)){
  numericList <- as.numeric(Matrix[,n])
  numericMatrix <- rbind(numericMatrix, t(as.matrix(numericList)))
}
numericMatrix

# numericTable <- as.table(numericMatrix)
# numericTable
```

### XML file (Beispiel.xml):

```
<?xml version="1.0" encoding="iso-8859-1"?>
<uebung>
  <beispiel_1>
    <titel>pairwise_Fst</titel>
    <Fst>
1 2 3 4
1 2 3 4
5 6 7 8
5 6 7 8
    </Fst>
  </beispiel_1>
  <beispiel_2>
    <titel>mismatch</titel>
    <mismatch>
9 8 7 6
5 4 3 2
    </mismatch>
  </beispiel_2>
</uebung>
```

extracted data (with R):

```
> numericMatrix
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    1    2    3    4
[3,]    5    6    7    8
[4,]    5    6    7    8
```

- Read data between two **XML** tags and convert to numeric matrix (Fst matrix):  
(Node auslesen\_matrix\_PairwiseFst.r)

```
#----open XML package-----
library(XML)

#----read data between an XML tag-----
filename = "D:/Heidi/Master/R_Daten/XML/PairwiseFst_XML.xml"
tag = "//Fst"
doc = xmlTreeParse(filename, useInternal = TRUE)
ch = getNodeSet(doc, tag)

subDoc = xmlDoc(ch[[1]])
tagData <- xpathApply(subDoc, tag, xmlValue)
free(subDoc)
#print(tagData, indent=FALSE)

#----convert string data to a numeric matrix-----
#----split string----
tagData2 <- as.character(tagData)

tagData3 <- strsplit(tagData2, "\n")
tagMatrix <- as.matrix(as.data.frame(tagData3))
tagMatrix <- tagMatrix[4:nrow(tagMatrix)]
tagMatrix <- gsub(" +", " ", tagMatrix) # trim white space
Data <- strsplit(tagMatrix, " ")

#----to string matrix----
Row <- length(Data)

Matrix <- as.matrix(as.data.frame(Data[1]))
Matrix <- rbind(Matrix, matrix(NA, ncol=1, nrow=(Row-1)))

for(n in 2:(Row)){
  nextrow <- as.matrix(as.data.frame(Data[n]))
  nextrow <- rbind(nextrow, matrix(NA, ncol=1, nrow=(Row-n)))
  Matrix <- cbind(Matrix, nextrow)
}
Matrix <- Matrix[3:nrow(Matrix),]

#----to numeric matrix----
numericList <- as.numeric(Matrix[,1])
numericMatrix <- t(as.matrix(numericList))

for(n in 2:(Row)){
  numericList <- as.numeric(Matrix[,n])
  numericMatrix <- rbind(numericMatrix, t(as.matrix(numericList)))
}

# numericTable <- as.table(numericMatrix)
# numericTable

DistanceMatrix <- numericMatrix

#----Mirror matrix (left-right)----
mirror.matrix <- function(x) {
  xx <- as.data.frame(x);
  xx <- rev(xx);
  xx <- as.matrix(xx);
  xx;
}
```

```
#----Rotate matrix 270 clockworks----
rotate270.matrix <- function(x) {
  mirror.matrix(t(x))
}

DistanceMatrix <- rotate270.matrix(DistanceMatrix)

#----draw matrix plot----
library(fields)

a <- ncol(DistanceMatrix)
b <- nrow(DistanceMatrix)
x <- c(1:a)
y <- c(1:b)

ColorRamp <- rgb( seq(1,0,length=256), # Red
                  seq(1,0,length=256), # Green
                  seq(1,1,length=256)) # Blue

image.plot(x,y,DistanceMatrix, col=ColorRamp, main="Distance matrix: No. of
different alleles (FST)", xlab="Population", ylab="Population",
axes = FALSE, legend.lab="Number of different alleles (Fst)",
legend.mar=4.3, legend.width=0.8)
contour(DistanceMatrix, add = TRUE)
axis(1, at = c(1:a),cex.axis=0.7)
axis(2, at = c(1:b), labels=c(b:1),cex.axis=0.7)
box()
```

graphik: see page 18

extracted data (with R):

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]
[1,]	0.00000	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[2,]	0.68035	0.00000	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[3,]	0.45650	0.32854	0.00000	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[4,]	0.39038	0.40234	0.14075	0.00000	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
[5,]	0.39891	0.37543	0.10492	0.01652	0.00000	NA	NA	NA	NA	NA	NA	NA	NA	NA
[6,]	0.54999	0.48260	0.08569	0.23355	0.17917	0.00000	NA	NA	NA	NA	NA	NA	NA	NA
[7,]	0.51610	0.42821	0.06202	0.19169	0.15525	0.03104	0.00000	NA	NA	NA	NA	NA	NA	NA
[8,]	0.47851	0.46968	0.13281	0.27648	0.23387	0.12209	0.06420	0.00000	NA	NA	NA	NA	NA	NA
[9,]	0.59330	0.52194	0.36959	0.37137	0.34153	0.44308	0.39951	0.41309	0.00000	NA	NA	NA	NA	NA
[10,]	0.62345	0.55361	0.36238	0.34845	0.32370	0.44222	0.40016	0.43896	0.48323	0.00000	NA	NA	NA	NA
[11,]	0.63122	0.54883	0.13147	0.24649	0.18974	0.14969	0.17140	0.25724	0.50326	0.44678	0.00000	NA	NA	NA
[12,]	0.58956	0.52853	0.19458	0.29505	0.25074	0.22508	0.23474	0.29846	0.49881	0.39891	0.17418	0.00000	NA	NA
[13,]	0.70162	0.64672	0.45231	0.45789	0.43639	0.52847	0.50150	0.51927	0.57398	0.60126	0.61293	0.57697	0.00000	NA
[14,]	0.67711	0.58836	0.40512	0.42183	0.39560	0.50269	0.46657	0.48734	0.54140	0.56386	0.56939	0.54419	0.30345	0

part of the XML file (PairwiseFst\_XML.xml):

## 21.12.07

- Move the legend in matrix plots with legend to the print area:

(matrix\_PairwiseFst.r)

```
#-----read data-----
read.table("D:/Heidi/Master/R_Daten/FstMatrix/PairwiseFst_mic.txt",
           header=TRUE, skip=5 ,row.names=1, fill=TRUE )-> Data
as.matrix.data.frame(Data) -> Matrix

a <- ncol(Matrix)
b <- nrow(Matrix)

x <- c(1:a)
y <- c(1:b)

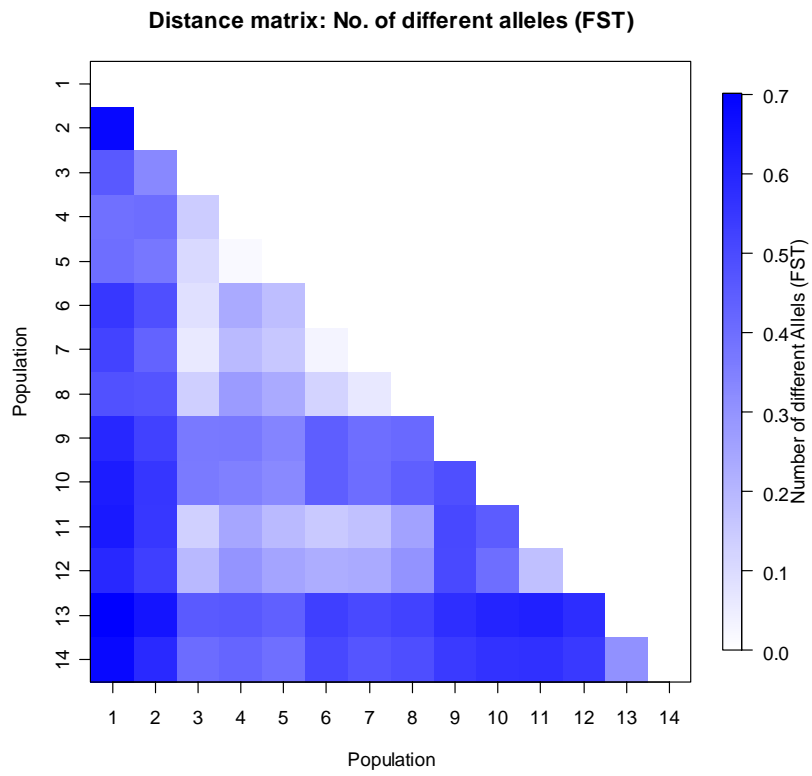
#-----draw plot-----
library(fields)
ColorRamp <- rgb( seq(1,0,length=256), # Red
                  seq(1,0,length=256), # Green
                  seq(1,1,length=256)) # Blue

#----Mirror matrix (left-right)----
mirror.matrix <- function(x) {
  xx <- as.data.frame(x);
  xx <- rev(xx);
  xx <- as.matrix(xx);
  xx;
}

#----Rotate matrix 270 clockworks----
rotate270.matrix <- function(x) {
  mirror.matrix(t(x))
}

Matrix <- rotate270.matrix(Matrix)

#----draw matrix plot----
image.plot(x,y,Matrix, col=ColorRamp, main="Distance matrix: No. of
different alleles (FST)", xlab="Population", ylab="Population",
          legend.args=list( text="Number of different Allels (FST)",
                             cex=1.0, side=4, line=2), axes = FALSE)
contour(Matrix, add = TRUE)
axis(1, at = c(1:a))
axis(2, at = c(1:b), labels=c(b:1))
box()
```



**Problems:**



## 18.01.08

- Population assignment test

(points\_assignmentTest.r)

```
#----read data-----
sample1 <- read.table("D:/Heidi/Master/R_Daten/assignmentTest/sample1.txt",
                      skip=3 )
sample2 <- read.table("D:/Heidi/Master/R_Daten/assignmentTest/sample2.txt",
                      skip=3 )

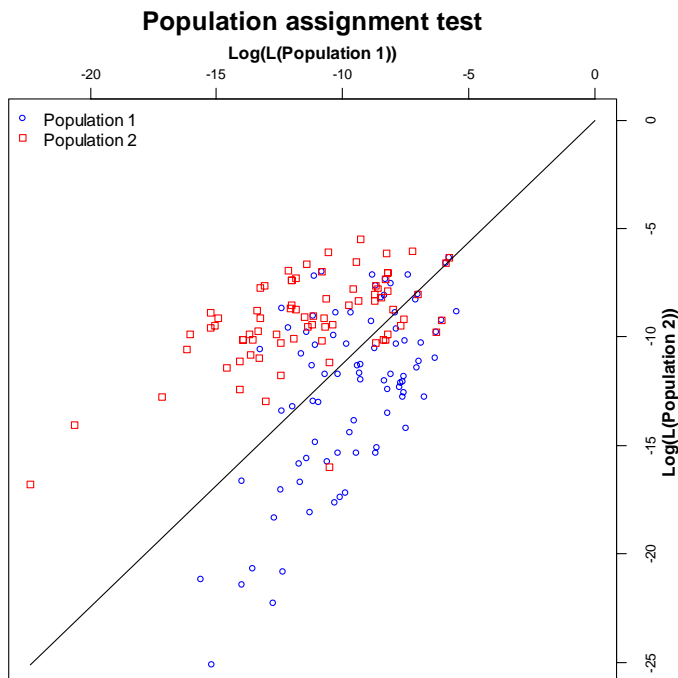
sample1 <- as.matrix(sample1[2:3])
sample2 <- as.matrix(sample2[2:3])

min_x <- min(sample1[,1], sample2[,1])
min_y <- min(sample1[,2], sample2[,2])

#----draw plot-----
op <- par(mar=c(2,2,6,5))
plot(sample1, col="blue", pch=21, xlim=c(min_x, 0), ylim=c(min_y, 0),
      xlab="", ylab="", axes=FALSE)
points(sample2, col="red", pch=22)
lines(c(min_x, 0), c(min_y, 0))

legend("topleft", c("Population 1", "Population 2"), bty="n",
      col=c("blue", "red"), pch=c(21,22))
axis(side=3, cex.axis=0.8)
axis(side=4, cex.axis=0.8)
mtext("Population assignment test", side=3, line=3.5, font=2, cex=1.5)
mtext("Log(L(Population 1))", side=3, line=2, font=2)
mtext("Log(L(Population 2))", side=4, line=2, font=2)
box()

#---- At end of plotting, reset to previous par settings:----
par(op)
```



**Problems:**

## 22.01.08

- Haplotype distance matrix between/within populations and groups  
(matrix\_multiplePlots.r)

```
#----read data-----
#----read haplotype list----
Data <- read.table("D:/Heidi/Master/R_Daten/HaplotypeDistance/
                  ListHaplotype_betweenPop.txt", skip=1)

Row <- nrow(Data)
Columns <- Row

#----read data row by row:----
x <- 0
n <- 1

DistanceMatrix <- as.matrix(scan("D:/Heidi/Master/R_Daten/
                                HaplotypeDistance/HapDistanceMatrix_betweenPop.txt",
                                what=double(0), skip=x, nlines=1, nmax=n))
DistanceMatrix <- cbind(DistanceMatrix, matrix(NA, ncol=Columns, nrow=1))
DistanceMatrix <- DistanceMatrix[,1:Columns]

n <- n + 1
x <- x + 1

while(n<(Row+1)){
  nextrow <- as.matrix(scan("D:/Heidi/Master/R_Daten/
                            HaplotypeDistance/HapDistanceMatrix_betweenPop.txt",
                            what=double(0), skip=x, nlines=1, nmax=n))
  nextrow <- cbind(t(nextrow), matrix(NA, ncol=Columns, nrow=1))
  nextrow <- nextrow[,1:Columns]

  DistanceMatrix <- rbind(DistanceMatrix, nextrow)

  n <- n + 1
  x <- x + 1
}

#----select the data of DistanceMatrix-----
dimnames(DistanceMatrix) <- list(Data[,1], Data[,1])

#--population1:
Pop1 <- read.table("D:/Heidi/Master/R_Daten/HaplotypeDistance/
                  HapDistanceMatrix_withinPop1.txt", skip=5)
Pop1 <- as.character(Pop1[,1])
DistanceMatrixPop1 <- DistanceMatrix[Pop1, Pop1]

#--population2:
Pop2 <- read.table("D:/Heidi/Master/R_Daten/HaplotypeDistance/
                  HapDistanceMatrix_withinPop2.txt", skip=5)
Pop2 <- as.character(Pop2[,1])
DistanceMatrixPop2 <- DistanceMatrix[Pop2, Pop2]

#--between population1/2:
#whole DistanceMatrix
wholeDistanceMatrix <- DistanceMatrix
x <- 1
```

```

while(x <= ncol(wholeDistanceMatrix)){

  twholeDistanceMatrix <- t(wholeDistanceMatrix)
  wholeDistanceMatrix[x,(x:ncol(wholeDistanceMatrix))] <-
    twholeDistanceMatrix[x,(x:ncol(wholeDistanceMatrix))]

  x <- x + 1
}
wholeDistanceMatrixBetween <- wholeDistanceMatrix[Pop2, Pop1]

#----together:-----
DistanceMatrixUp <- cbind(DistanceMatrixPop1, matrix(NA,
  ncol(DistanceMatrixPop1), nrow(DistanceMatrixPop2)))
DistanceMatrixLo <- cbind(wholeDistanceMatrixBetween, DistanceMatrixPop2)
DistanceMatrixTogether <- rbind(DistanceMatrixUp, DistanceMatrixLo)

#----whole DistanceMatrix both Populations together:----
Pop_together <- c(Pop1, Pop2)

wholeDistanceMatrixTogether <- wholeDistanceMatrix[Pop_together,
  Pop_together[c(length(Pop_together)):1]]

#----Mirror matrix (left-right)----
mirror.matrix <- function(x) {
  xx <- as.data.frame(x);
  xx <- rev(xx);
  xx <- as.matrix(xx);
  xx;
}

#----Rotate matrix 270 clockworks----
rotate270.matrix <- function(x) {
  mirror.matrix(t(x))
}

DistanceMatrixTogether <- rotate270.matrix(DistanceMatrixTogether)

#----draw matrix-----
windows() #new graphic window
library(fields)
ColorRamp <- rgb( seq(1,0,length=256), # Red
  seq(1,0,length=256), # Green
  seq(1,1,length=256)) # Blue

a <- ncol(DistanceMatrixTogether)
b <- nrow(DistanceMatrixTogether)

x <- c(1:a)
y <- c(1:b)

png("D:/Heidi/Master/R_Graphiken/matrix_HapDistance_between-within Pop
  and Groups.png", width = 680, height = 660)
op <- par(mfrow=c(2,2), oma=c(0, 0 ,3, 0))

#----matrix one----
op1 <- par(mfg=c(1,1),mar=c(0.2, 5.2, 4.2, 0.2))
  image(x,y,DistanceMatrixTogether, col=ColorRamp, ylab="Group 1",
    font.lab=2, axes = FALSE)
  contour(DistanceMatrixTogether, add = TRUE)

```

```

axis(2, at = c(1:b), labels=c(Pop2[NROW(Pop2):1], Pop1[NROW(Pop1):1]),
     cex.axis=0.6, las=2)
box(bty="l")

lines(c(0,NROW(Pop1)+0.5),c(NROW(Pop2)+0.5,NROW(Pop2)+0.5), lwd=2)
lines(c(NROW(Pop1)+0.5,NROW(Pop1)+0.5),c(0,NROW(Pop2)+0.5), lwd=2)

mtext(side = 2, at =(3.2*Row/4), line = 2.7, text = "Population 1",
      cex=0.7)
mtext(side = 2, at =(Row/4), line = 2.7, text = "Population 2",
      cex=0.7)
par(op1)

#----matrix two----
op2 <- par(mfg=c(2,1), mar=c(4.2, 5.2, 0.2, 0.2))
image(x,y, wholeDistanceMatrixTogether, col=ColorRamp,
      xlab="Group 1", ylab="Group 2", font.lab=2, axes = FALSE)
contour(DistanceMatrixTogether, add = TRUE)
axis(1, at = c(1:a), labels=c(Pop_together), cex.axis=0.6, las=2)
axis(2, at = c(1:b), labels=c(Pop2[NROW(Pop2):1], Pop1[NROW(Pop1):1]),
     cex.axis=0.6, las=2)
box()

lines(c(0,NROW(Pop1)+ NROW(Pop2)+0.5), c(NROW(Pop2)+ 0.5,NROW(Pop2)
+0.5), lwd=2)
lines(c(NROW(Pop1)+0.5,NROW(Pop1)+0.5),c(0,NROW(Pop1)+ NROW(Pop2)
+0.5), lwd=2)

mtext(side = 1, at =(Row/4), line = 2.5, text = "Population 1",
      cex=0.7)
mtext(side = 1, at =(3.2*Row/4), line = 2.5, text = "Population 2",
      cex=0.7)
mtext(side = 2, at =(3.2*Row/4), line = 2.7, text = "Population 3",
      cex=0.7)
mtext(side = 2, at =(Row/4), line = 2.7, text = "Population 4",
      cex=0.7)
par(op2)

#----matrix three----
op3 <- par(mfg=c(2,2), mar=c(4.2, 0.2, 0.2, 6.2))
image(x,y,DistanceMatrixTogether, col=ColorRamp, xlab="Group 2",
      font.lab=2, axes = FALSE)
contour(DistanceMatrixTogether, add = TRUE)
axis(1, at = c(1:a), labels=c(Pop1,Pop2), cex.axis=0.6, las=2)
box(bty="l")

lines(c(0,NROW(Pop1)+0.5),c(NROW(Pop2)+0.5,NROW(Pop2)+0.5), lwd=2)
lines(c(NROW(Pop1)+0.5,NROW(Pop1)+0.5),c(0,NROW(Pop2)+0.5), lwd=2)

mtext(side = 1, at =(Row/4), line = 2.5, text = "Population 3",
      cex=0.7)
mtext(side = 1, at =(3.2*Row/4), line = 2.5, text = "Population 4",
      cex=0.7)
par(op3)

#----legend----
maximum <- max(DistanceMatrixTogether, wholeDistanceMatrixTogether,
               DistanceMatrixTogether , na.rm = TRUE)

Legend <- seq(from=0, to=maximum, length=100)
Legend <- as.matrix(Legend)
op4 <- par(mfg=c(1,2), mar=c(0, 15.6, 3.2, 6.2) )
image(y=Legend, t(Legend), col=ColorRamp, axes=FALSE)

```

```

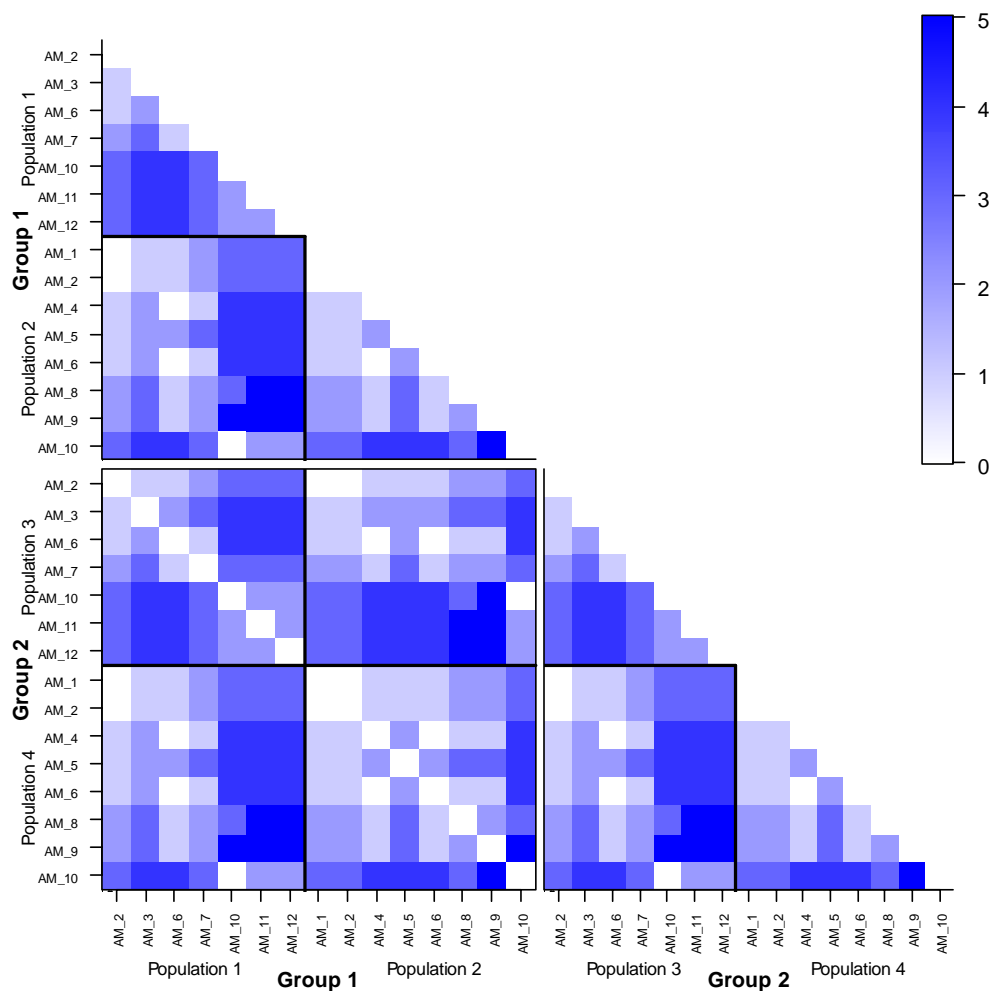
axis(side=4, las=2)
box()

title("haplotype distance matrix between/within populations and
      groups", line=0, outer=TRUE)
par(op4)

#----reset par parameter----
par(op)
dev.off()

```

haplotype distance matrix between/within populations and groups



28.01.08

- Haplotype frequencies in population

(lines\_HaplotypeFreqMultiple.r)

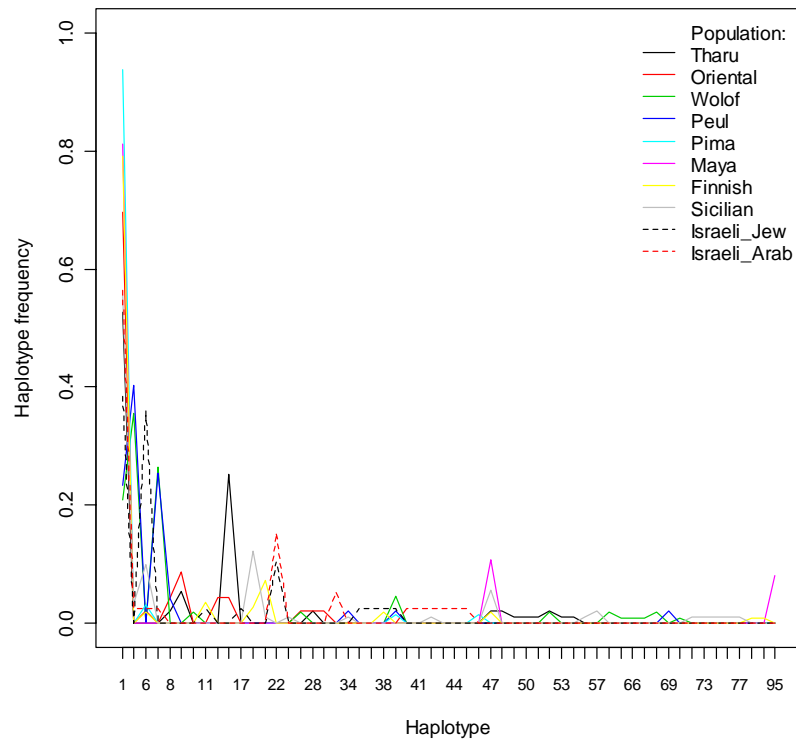
```
#-----read data-----
Names <- scan("D:/Heidi/Master/R_Daten/SummaryStatistics/
              haplotype_frequ.txt", what="list", skip=5, nlines=1)
Data <- read.table("D:/Heidi/Master/R_Daten/SummaryStatistics/
                  haplotype_frequ.txt", skip=8)

nRow <- nrow(Data)
nCol <- ncol(Data)

#-----draw plot-----
x <- 1
plot(Data[, (x+1)], type="l", xlab="Haplotype", ylab="Haplotype
frequency", main="Haplotype frequencies in populations",
      ylim=c(0,1), xaxt="n", col=(x), lwd=1 )

x <- x + 1
while(x <= nCol){
  lines(Data[, (x+1)], col=(x), lwd=1, lty=(x/9)+1 )
  x <- x + 1
}
axis(1, at=c(1:nRow), labels=Data[,1], cex.axis=0.8)
legend("topright", legend=c("Population:", Names[2:length(Names)]),
      lty=c(0,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2), bty="n",
      col=c(0:x), lwd=1.5)
```

**Haplotype frequencies in populations**



- Molecular diversity indexes

(lines\_MolecDiversityIndexes.r)

```
#----read data-----
Names <- scan("D:/Heidi/Master/R_Daten/SummaryStatistics/
             molec_diversity_indexes.txt", what="list", skip=6, nlines=1)

Data <- read.table("D:/Heidi/Master/R_Daten/SummaryStatistics/
                  molec_diversity_indexes.txt", skip=8, row.names=1)
Data <- as.matrix(Data[1:(length(Data)-2)])

#----draw plot-----
plot(Data[1,], type="l", xlab="Population", ylab=" ", main="molecular
      diversity indexes", ylim=c(0, max(Data)), lwd=2, axes=FALSE)
axis(side=2, at=c(0:max(Data)),ylim=c(0:max(Data)),
     labels=c(0:max(Data)))
axis(side=1, at=c(0:length(Data[1,])),
     labels=Names[1:(length(Names)-2)], las=2, cex.axis=0.7)
box()

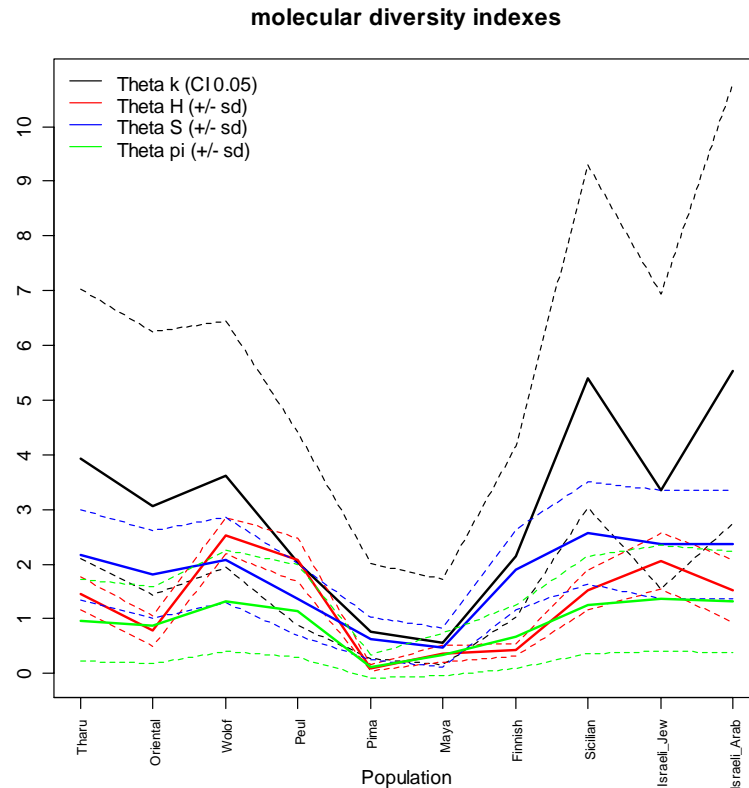
lines(Data[2,], lty=2)
lines(Data[3,], lty=2)

lines(Data[4,], col="red", lwd=2)
lines((Data[4,]+ Data[5,]), lty=2, col="red")
lines((Data[4,]- Data[5,]), lty=2, col="red")

lines(Data[6,], col="blue", lwd=2)
lines((Data[6,]+ Data[7,]), lty=2, col="blue")
lines((Data[6,]- Data[7,]), lty=2, col="blue")

lines(Data[8,], col="green", lwd=2)
lines((Data[8,]+ Data[9,]), lty=2, col="green")
lines((Data[8,]- Data[9,]), lty=2, col="green")

legend("topleft", c("Theta k (CI 0.05)" , "Theta H (+/- sd)",
                  "Theta S (+/- sd)", "Theta pi (+/- sd)"),
      lty=1, bty="n", col=c("black", "red", "blue", "green"))
```



- Divergence times allowing for unequal population sizes (tau)  
(matrix\_tau.r)

```
#----read data-----
Data <- read.table("D:/Heidi/Master/R_Daten/SummaryStatistics/tau.txt",
                  skip=5, fill=TRUE)

Columns <- ncol(Data)+1
Row <- nrow(Data)

x <- 6
n <- 2

#----read data line by line-----
tauMatrix <- as.matrix(scan("D:/Heidi/Master/R_Daten/SummaryStatistics/
                           tau.txt", what=double(0), skip=x, nlines=1, nmax=n))
tauMatrix <- cbind(t(tauMatrix), matrix(NA, ncol=(Columns-n), nrow=1))
tauMatrix <- tauMatrix[,2:Columns]

n <- n + 1
x <- x + 1

while(n<(Row+1)){
  nextrow <- as.matrix(scan("D:/Heidi/Master/R_Daten/SummaryStatistics/
                           tau.txt", what=double(0), skip=x, nlines=1, nmax=n))
```



```

nextrow <- cbind(t(nextrow), matrix(NA, ncol=(Columns-n), nrow=1))
nextrow <- nextrow[,2:Columns]

tauMatrix <- rbind(tauMatrix, nextrow)

n <- n + 1
x <- x + 1
}

a <- ncol(tauMatrix)
b <- nrow(tauMatrix)

x <- c(1:a)
y <- c(1:b)

#----draw plot-----
library(fields)
ColorRamp <- rgb( seq(1,0,length=256), # Red
                  seq(1,0,length=256), # Green
                  seq(1,1,length=256)) # Blue

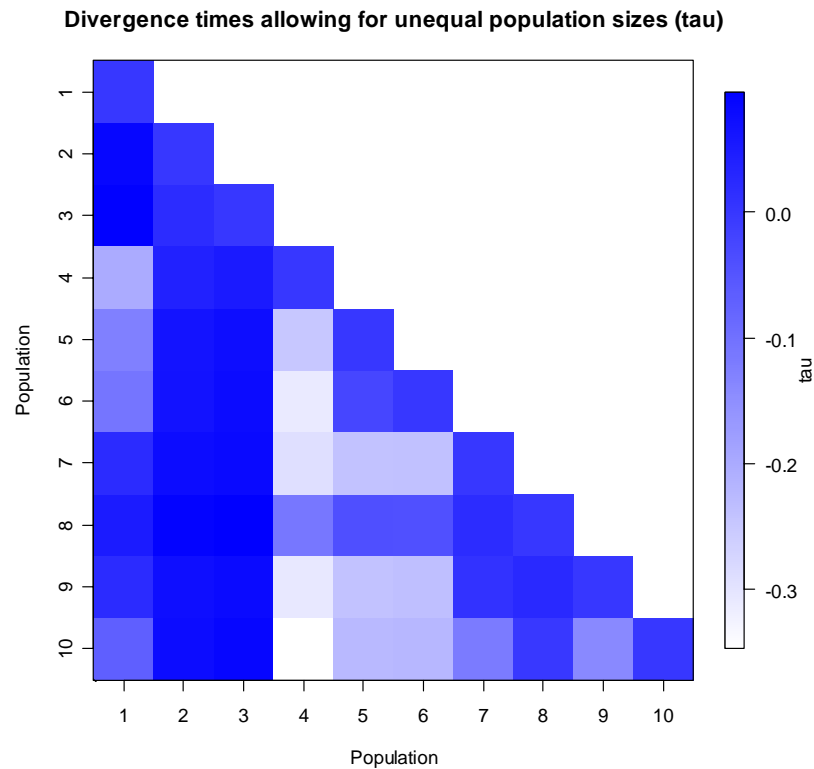
#----Mirror matrix (left-right)----
mirror.matrix <- function(x) {
  xx <- as.data.frame(x);
  xx <- rev(xx);
  xx <- as.matrix(xx);
  xx;
}

#----Rotate matrix 270 clockworks----
rotate270.matrix <- function(x) {
  mirror.matrix(t(x))
}

tauMatrix <- rotate270.matrix(tauMatrix)

#----draw matrix plot----
image.plot(x,y, tauMatrix, col=ColorRamp, main="Divergence times allowing
for unequal population sizes (tau)", xlab="Population",
ylab="Population" , legend.args=list(text="tau", cex=1.0,
side=4, line=2.2), axes = FALSE)
contour(tauMatrix, add = TRUE)
axis(1, at = c(1:a))
axis(2, at = c(1:b), labels=c(b:1))
box()

```



- Population average pairwise difference  
(matrix\_PairwiseDifferences.r)

```
#----read data-----
Data <- read.table("D:/Heidi/Master/R_Daten/SummaryStatistics/
                  pairwise_differences.txt", skip=10, row.names=1)
DataMatrix <- as.matrix(Data)

#----UnderMatrix----
n <- 2
x <- 1

UnderMatrix <- matrix(NA, ncol=ncol(DataMatrix), nrow=1)

while(n<=nrow(DataMatrix)){
  nextrow <- DataMatrix[n,1:x]
  nextrow <- cbind(t(nextrow), matrix(NA, ncol=ncol(DataMatrix)-x,
                                     nrow=1))

  UnderMatrix <- rbind(UnderMatrix, nextrow)

  n <- n+1
  x <- x+1
}
```

```

#----UpperMatrix----
n <- 2
x <- 1

UpperMatrix <- matrix(NA, ncol=x, nrow=1)
UpperMatrix <- cbind(UpperMatrix, t(DataMatrix[x,n:ncol(DataMatrix)]))

n <- n+1
x <- x+1
while(n<=nrow(DataMatrix)){
  nextrow <- matrix(NA, ncol=x, nrow=1)
  nextrow <- cbind(nextrow, t(DataMatrix[x,n:ncol(DataMatrix)]))

  UpperMatrix <- rbind(UpperMatrix, nextrow)

  n <- n+1
  x <- x+1
}
UpperMatrix <- rbind(UpperMatrix, matrix(NA, ncol=ncol(DataMatrix),
                                         nrow=1))

#----DiagonalMatrix----
n <- 1

DiagonalMatrix <- DataMatrix[n,n]
DiagonalMatrix <- cbind(DiagonalMatrix, matrix(NA,
                                                ncol=ncol(DataMatrix)-n, nrow=1))

n <- n+1
x <- 1
while(n<=nrow(DataMatrix)){
  nextrow <- matrix(NA, ncol=x, nrow=1)
  nextrow <- cbind(nextrow, t(DataMatrix[n,n]))
  nextrow <- cbind(nextrow, matrix(NA, ncol=ncol(DataMatrix)-n, nrow=1))

  DiagonalMatrix <- rbind(DiagonalMatrix, nextrow)

  n <- n+1
  x <- x+1
}

#----plot-----
a <- ncol(DataMatrix)
b <- nrow(DataMatrix)

x <- c(1:a)
y <- c(1:b)

ColorRamp <- rgb( seq(1,0,length=256), # Red
                  seq(1,0,length=256), # Green
                  seq(1,1,length=256)) # Blue

ColorRamp2 <- rgb( seq(1,0,length=256), # Red
                   seq(1,0.6,length=256), # Green
                   seq(1,0,length=256)) # Blue

ColorRamp3 <- rgb( seq(1,1,length=256), # Red
                   seq(1,0.5,length=256), # Green
                   seq(1,0,length=256)) # Blue

```

```

#----Mirror matrix (left-right)----
mirror.matrix <- function(x) {
  xx <- as.data.frame(x);
  xx <- rev(xx);
  xx <- as.matrix(xx);
  xx;
}

#----Rotate matrix 270 clockworks----
rotate270.matrix <- function(x) {
  mirror.matrix(t(x))
}

UnderMatrix <- rotate270.matrix(UnderMatrix)
UpperMatrix <- rotate270.matrix(UpperMatrix)
DiagonalMatrix <- rotate270.matrix(DiagonalMatrix)

#-----draw matrix-----
#png("D:/Heidi/Master/R_Graphiken/matrix_PairwiseDifferences.png",
     width = 680, height = 660)

#----devide plot region in 4 parts----
def.par <- par(no.readonly = TRUE) # save default, for resetting...
layout(rbind(c(1,2), c(1,3), c(1,4)),
       heights=rbind(c(2,1), c(2,1), c(2,1)),
       respect=rbind(c(0,1), c(0,0), c(0,0)))

#-----draw matrixe plot----
op <- par(mar=c(7.1, 5.1, 9.1, 2.1))
image(x,y, UnderMatrix, col=ColorRamp, xlab="", ylab="", axes=FALSE)
image(x,y, UpperMatrix, col=ColorRamp2, xlab="", ylab="", axes=FALSE,
      add=TRUE)
image(x,y, DiagonalMatrix, col=ColorRamp3, xlab="", ylab="",
      axes=FALSE, add=TRUE)
axis(1, at = c(1:a))
axis(2, at = c(1:b), labels=c(b:1))
box()
mtext(text="Population", side=1, line=2.5)
mtext(text="Population", side=2, line=2.5)
mtext(text="Population average pairwise difference", line=4.5,
      cex=1.2, font=2)
par(op)

#----draw legends----
#--upper legend--
op2 <- par(mar=c(0.6, 0, 3.6, 7.6))
UpperLegend <- seq(from=0, to=max(UpperMatrix, na.rm=TRUE),
                  length=100)
UpperLegend <- as.matrix(UpperLegend)

image(y=UpperLegend, t(UpperLegend), col=ColorRamp2, axes=FALSE)
axis(side=4, las=2)
mtext(text="Average number of pairwise differences
         between populations (PiXY)", side=4, line=4.5, cex=0.75)
box()
par(op2)

#--diagonal legend--
op3 <- par(mar=c(2.1, 0, 2.1, 7.6))
DiagonalLegend <- seq(from=0, to=max(DiagonalMatrix, na.rm=TRUE),

```

```

length=100)
DiagonalLegend <- as.matrix(DiagonalLegend)

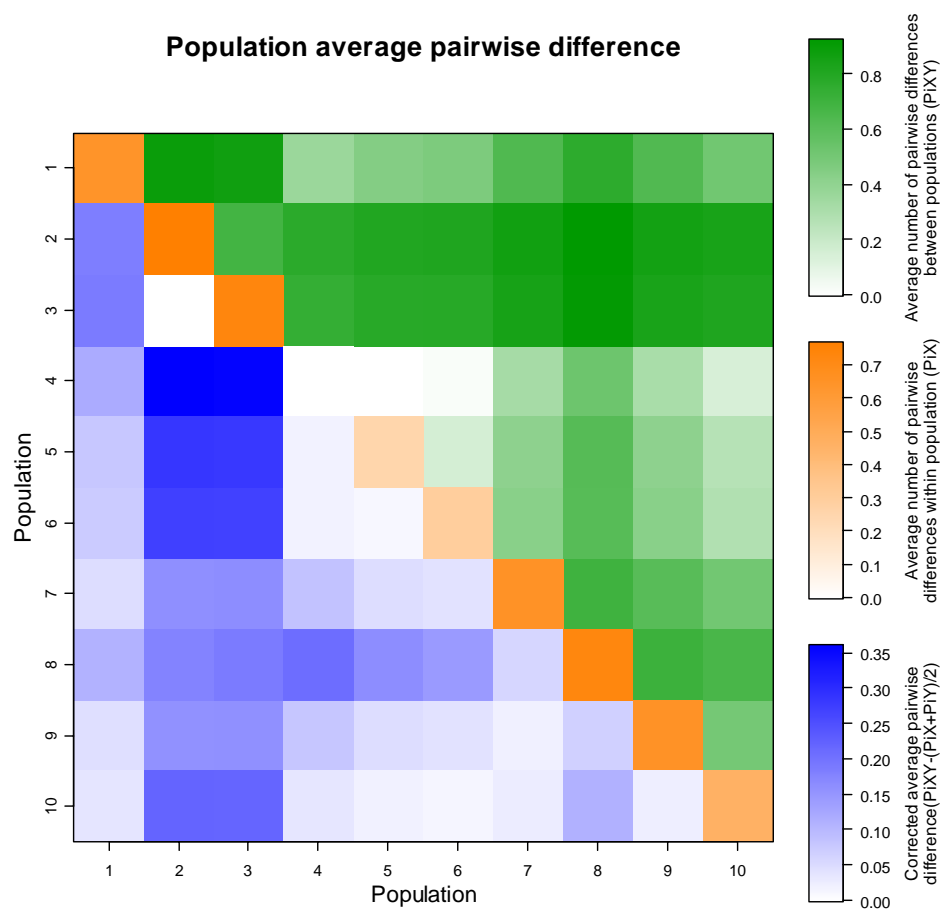
image(y=DiagonalLegend, t(DiagonalLegend), col=ColorRamp3,
      axes=FALSE)
axis(side=4, las=2)
mtext(text="Average number of pairwise differences within
population (PiX)", side=4, line=4.5, cex=0.75)
box()
par(op3)

#--under legend--
op4 <- par(mar=c(3.6, 0, 0.6, 7.6))
UnderLegend <- seq(from=0, to=max(UnderMatrix, na.rm=TRUE),
                    length=100)
UnderLegend <- as.matrix(UnderLegend)

image(y=UnderLegend, t(UnderLegend), col=ColorRamp, axes=FALSE)
axis(side=4, las=2)
mtext(text="Corrected average pairwise difference
(PiXY-(PiX+PiY)/2)", side=4, line=4.5, cex=0.75)
box()
par(op4)

#dev.off()
par(def.par) #reset to default

```



## 29.01.08

- Read data between two XML tags and convert to numeric matrix (improved):

- Half matrix:

(read\_tag-halfMatrix.r)

```
#----open XML package-----
library(XML)

#----read data between an XML tag-----
filename = "D:/Heidi/Master/R_Daten/XML/PairwiseFst_XML2.xml"
tag = "//Fst"
doc = xmlTreeParse(filename, useInternal = TRUE)
ch = getNodeSet(doc, tag)

subDoc = xmlDoc(ch[[1]])
tagData <- xpathApply(subDoc, tag, xmlValue)
free(subDoc)

#----convert string data (half matrix) to a numeric matrix-----
#----split string----
tagData2 <- as.character(tagData)

tagData3 <- strsplit(tagData2, "\n")
tagMatrix <- as.matrix(as.data.frame(tagData3))
tagMatrix <- tagMatrix[4:nrow(tagMatrix)]
tagMatrix <- gsub(" +", " ", tagMatrix) # trim white space
Data <- strsplit(tagMatrix, " ")

#----to numeric matrix----
Row <- length(Data)

Matrix <- as.matrix(as.data.frame(Data[1]))
Matrix <- rbind(Matrix, matrix(NA, ncol=1, nrow=(Row-1)))
Matrix <- Matrix[3:nrow(Matrix),]
numericList <- as.numeric(Matrix)
numericMatrix <- t(as.matrix(numericList))

for(n in 2:(Row)){
  nextrow <- as.matrix(as.data.frame(Data[n]))
  nextrow <- rbind(nextrow, matrix(NA, ncol=1, nrow=(Row-n)))
  nextrow <- nextrow[3:nrow(nextrow),]
  numericList <- as.numeric(nextrow)
  numericMatrix <- rbind(numericMatrix, t(as.matrix(numericList)))
}
numericMatrix
```

- Full matrix:

(read\_tag-fullMatrix.r)

```
#----open XML package-----
library(XML)

#----read data between an XML tag-----
filename = "D:/Heidi/Master/R_Daten/XML/XML_with_inserted_data.xml"
tag = "//pairwise_differences"
doc = xmlTreeParse(filename, useInternal = TRUE)
ch = getNodeSet(doc, tag)

subDoc = xmlDoc(ch[[1]])
tagData <- xpathApply(subDoc, tag, xmlValue)
free(subDoc)
#print(tagData, indent=FALSE)

#----convert string data (full matrix) to a numeric matrix-----
#----split string----
tagData2 <- as.character(tagData)

tagData3 <- strsplit(tagData2, "\n")
tagMatrix <- as.matrix(as.data.frame(tagData3))
tagMatrix <- gsub(" + ", " ", tagMatrix) # trim white space

tagMatrix <- tagMatrix[12:nrow(tagMatrix)]
Data <- strsplit(tagMatrix, " ")

#----to numeric matrix----
Row <- length(Data)

Matrix <- as.matrix(as.data.frame(Data[1]))
Matrix <- Matrix[3:(nrow(Matrix))]
Matrix <- as.numeric(Matrix)
numericMatrix <- t(as.matrix(Matrix))

for(n in 2:(Row)){
  nextrow <- as.matrix(as.data.frame(Data[n]))
  nextrow <- nextrow[3:(nrow(nextrow))]
  nextrow <- as.numeric(nextrow)
  numericMatrix <- rbind(numericMatrix, t(as.matrix(nextrow)))
}

numericMatrix
```

- o Table with names:  
(read\_tag-table(with names).r)

```
#----open XML package-----
library(XML)

#----read data between an XML tag-----
filename = "D:/Heidi/Master/R_Daten/XML/XML_with_inserted_data.xml"
tag = "//theta"
doc = xmlTreeParse(filename, useInternal = TRUE)
ch = getNodeSet(doc, tag)

subDoc = xmlDoc(ch[[1]])
tagData <- xpathApply(subDoc, tag, xmlValue)
free(subDoc)
#print(tagData, indent=FALSE)

#----convert string data (table with names) to a numeric matrix---
#----split string----
tagData2 <- as.character(tagData)

tagData3 <- strsplit(tagData2, "\n")
tagMatrix <- as.matrix(as.data.frame(tagData3))
tagMatrix <- gsub(" + ", " ", tagMatrix) # trim white space

#-----names-----
Names <- tagMatrix[9]
Names <- strsplit(Names, " ")
Names <- as.matrix(as.data.frame(Names))
Names <- Names[3:(nrow(Names)-2)]
Names

#-----data-----
tagMatrix <- tagMatrix[11:nrow(tagMatrix)]
Data <- strsplit(tagMatrix, " ")

#----to numeric matrix----
Row <- length(Data)

Matrix <- as.matrix(as.data.frame(Data[1]))
Matrix <- Matrix[3:(nrow(Matrix)-2)]
Matrix <- as.numeric(Matrix)
numericMatrix <- t(as.matrix(Matrix))

for(n in 2:(Row)){
  nextrow <- as.matrix(as.data.frame(Data[n]))
  nextrow <- nextrow[3:(nrow(nextrow)-2)]
  nextrow <- as.numeric(nextrow)
  numericMatrix <- rbind(numericMatrix, t(as.matrix(nextrow)))
}

numericMatrix

# numericTable <- as.table(numericMatrix)
# numericTable
```



- zum drucken immer PDF machen (Drucker verstellt Seite!!!)