Larger mtDNA than Y-chromosome differences between matrilocal and patrilocal groups from Sumatra

Ellen Dröfn Gunnarsdóttir, Madhusudan R. Nandineni, Mingkun Li, Sean Myles, David Gil, Brigitte Pakendorf and Mark Stoneking.

Supplementary Software:

R scripts used to calculate diversity indices:

1) Haplotype diversity for mtDNA sequences and YSTRs:

```
## script to randomly sample and calculate haplotype diversity for 2
groups

## Input = a fasta file with mtDNA sequences.

source("H_sampler.R")

fas = read.table("inputfile.fas", header=FALSE)

## read in the aligned fasta file, the fasta file should not contain
white spaces at the end of each line.

## H.Sampler <- function(fas.data, pop1.size, pop2.size,
replicate_number)

outfile = H.Sampler(fas, 36, 36, 1000)

## the third column of the outfile table now contains the difference of
the mean number of pairwise difference for the two populations, sampled
1000 times. This difference was plotted as a histogram.
```

the H.Sampler:

H.Sampler <- function(fas.data, pop1.size, pop2.size, replicate_number) {

```
   data=matrix(NA, length(fas.data[,1])/2, 1)

## create a data matrix where each row is a sequence
    r=seq(2, length(fas.data[,1]), by=2)

## creates a vector of even numbers to pull the sequences out of the
fasta file.

    for (i in 1:length(fas.data[,1])/2) {
            data[i,1]=as.character(fas.data[r[i],1])
        }

    Data.table = matrix(NA, replicate_number, 3, dimnames =
list(c(1:replicate_number), c("pop1.H", "pop2.H", "H.diff")))

## The outputting data table with haplotype diversity for the two pops
and the difference in H.
```

```
    "square" = function(x) x^2

## a new function for squaring a value

    number.of.samples = (pop1.size + pop2.size)

##the total number of samples
    samples = c(1:number.of.samples)

## a list of sample number, it will randomly draw from this list to
sample the sequence data

    for (i in 1:replicate_number) {
        pop1.sam.num = sample(c(1:number.of.samples), pop1.size)

##population 1 sampling number

        pop2.sam.num = samples[!samples %in% pop1.sam.num]

## population 2 sampling numbers; it will sample the numbers from
"samples" that are NOT found in pop1.sam.num

        pop1 = data[pop1.sam.num,1]

## vecor containing the sequence data for pop1

    pop2 = data[pop2.sam.num,1]

## vector containing the sequence data for pop2

        pop1.table = table(pop1)

## "table" the data, which creates a table of counts for identical
scalor values.

        pop2.table = table(pop2)

        freq.squared.1 = sum(apply(pop1.table/pop1.size,1,square))
        freq.squared.2 = sum(apply(pop2.table/pop2.size,1,square))
        Data.table[i,1] = (pop1.size/(pop1.size-1))*(1-freq.squared.1)
        Data.table[i,2] = (pop2.size/(pop2.size-1))*(1-freq.squared.2)
        Data.table[i,3] = abs(Data.table[i,1] - Data.table[i,2])
    }

   return(Data.table)
}
```

The YSTRs were treated as haplotypes, i.e. the number of repeats and then the same script was applied.

2. Mean number of pairwise difference:

a) mtDNA sequences:

```
## script to simulate and calculate 1000 times the mean number of
pairwise differences

## read the mtDNA sequences from fasta file

library(ape)

xdata <- read.dna("inputfile.fasta", format="fasta")
xdatam <- as.matrix(xdata)
nnucl <- ncol(as.matrix(xdata))

##the number of nucleotides or the total length of the sequence

res_mnd <- matrix(NA, ncol=3, nrow=1000)
colnames(res_mnd) <- c("pop1", "pop2", "diff")
Nsize = dim(xdatam)[1]

## the total number of individuals

ndiff.mat <- dist.dna(xdatam,model = "K80", variance = FALSE, gamma =
FALSE, pairwise.deletion = FALSE, base.freq = NULL, as.matrix = TRUE)

for (i in 1:1000) {
  id1 <- sample(1:Nsize, Nsize/2)
  id2 <- c(1:Nsize) [-id1]
  res_mnd[i,1] <- mean(ndiff.mat[id1,id1]*nnucl)
  res_mnd[i,2] <- mean(ndiff.mat[id2,id2]*nnucl)
  print(i)
}
res_mnd[,3] <- abs(res_mnd[,1]-res_mnd[,2])

## the third column of the res_mnd table now contains the difference of
the mean  number of pairwise difference for the two populations,
sampled 1000 times. This difference was plotted as a histogram.
```

b) YSTRs:

```
## calculate the mean number of pairwise for STR

ydata <- read.table("ystr_input.txt", header=T, sep="\t", as.is=T)
strs <- c("str1","str2","str3","str4","str5","str6", "str7", "str8",
"str9", "str10", "str11", "str12")

## the tab separated file ystr_input.txt should look as follow:
#id   str1  str2  str3
#ind1 10    11    13
#ind2 10    10    13
#ind3 11    10    12

dist.str <- function(x,y) { # formula to calculate the number of STR
binary differences (e.g 0 => equal, 1 => different independently of the
number of repeats)  between individuals
```

```
    ## output will be a matrix (size = n x n individuals)

      x1 <- x[,y]
      mt <-  matrix(0, ncol=nrow(x), nrow=nrow(x))
      for (i in 1:nrow(x1)) {
        for (j in 1:nrow(x1)) {
          if (i > j) {
            a <- abs(x1[i,y]-x1[j,y])
            mt[i,j]<- sum(a !=0)
            mt[j,i]<- sum(a !=0)
          }
        }
      }
      mt
    }
    mnd <- function(x) {

    ##calculate the mean number of pairwise differences
    ##x is a matrix calculated by means of the function "dist.str"

      mean(x[lower.tri(x,diag=F)])
    }


    ## calculate the matrix of differences between all individuals in the
    dataset

    dist_mt <- dist.str(ydata,strs)

    ## make permutations test by assigning randomly individuals to one (out
    of two) populations

    res_mnd <- matrix(NA, ncol=3, nrow=1000, dimnames=list(
    paste("sim",1:1000,sep=""), c("pop1","pop2", "diff")))

    ## contain the results of 1000 permutations

    for (i in 1:1000) {
      id <- sample(1:nrow(ydata),nrow(ydata)/2 ) # without replacement
      res_mnd[i,c(1,2)] <- c(mnd(dist_mt[id,id]), mnd(dist_mt[-id,-id]))
      res_mnd[i,3] <- abs(res_mnd[i,1]-res_mnd[i,2])
}

    ## the third column of the res_mnd table now contains the difference of
    the mean  number of pairwise difference for the two populations,
    sampled 1000 times. This difference was plotted as a histogram.
```