Apêndices

APÊNDICE A – Códigos para extração dos dados de FST gerados pelo Arlequin

```
# parse arlequin to R
#  customize arlequin functions
require(XML)
doc = xmlTreeParse("projeto.xml", useInternal = TRUE)
root = xmlName(xmlRoot(doc))

# FST pairwise
PairFST<-getNodeSet(doc, "//PairFstMat")

tagData2<-as.character(sapply(PairFST[1],xmlValue))

tagData3<-strsplit(tagData2,"\n")
tagMatrix <- as.matrix(as.data.frame(tagData3))
tagMatrix <- subset(tagMatrix, tagMatrix[,1] != "")
tagMatrix <- subset(tagMatrix, tagMatrix[,1] != "") #trim empty lines
tagMatrix <- tagMatrix[3:nrow(tagMatrix)]
tagMatrix  <- gsub(" + ", " ", tagMatrix)   # trim white space
Data <- strsplit(tagMatrix, " ")

Row <- length(Data)
# to numeric matrix ----------
  Matrix <- as.matrix(as.data.frame(Data[1]))
  Matrix <- subset(Matrix, Matrix[,1] != "")
  Matrix <- rbind(Matrix, matrix(NA, ncol=1, nrow=(Row-1)))
  Matrix <- Matrix[2:nrow(Matrix),]
  numericList <- as.numeric(Matrix)
  numericMatrix <- t(as.matrix(numericList))

  if(Row >= 2){
    for(n in 2:(Row)){
      nextrow <- as.matrix(as.data.frame(Data[n]))
      nextrow <- subset(nextrow, nextrow[,1] != "")
```

```
      nextrow <- rbind(nextrow, matrix(NA, ncol=1, nrow=(Row-n)))
      nextrow <- nextrow[2:nrow(nextrow),]
      numericList <- as.numeric(nextrow)
      numericMatrix <- rbind(numericMatrix, t(as.matrix(numericList)))
    }
  }
fst<-numericMatrix



# Pvalue FST pairwise
PairFSTpval<-getNodeSet(doc, "//PairFstPvalMat")
tagData2<-as.character(sapply(PairFSTpval[1],xmlValue))
tagData3<-strsplit(tagData2,"\n")
tagMatrix <- as.matrix(as.data.frame(tagData3))
tagMatrix <- subset(tagMatrix, tagMatrix[,1] != "")
tagMatrix <- subset(tagMatrix, tagMatrix[,1] != "") #trim empty lines
tagMatrix <- tagMatrix[2:nrow(tagMatrix)]
tagMatrix  <- gsub(" * ", " ", tagMatrix)   # trim white space
tagMatrix <-  gsub("\\+-.{6}","",tagMatrix)
Data <- strsplit(tagMatrix, " ")

Row <- length(Data)
# to numeric matrix ----------
  Matrix <- as.matrix(as.data.frame(Data[1]))
  Matrix <- subset(Matrix, Matrix[,1] != "")
  Matrix <- rbind(Matrix, matrix(NA, ncol=1, nrow=lstlisting(Row-1)))
  Matrix <- Matrix[2:nrow(Matrix),]
  numericList <- as.numeric(Matrix)
  numericMatrix <- t(as.matrix(numericList))

  if(Row >= 2){
    for(n in 2:(Row)){
      nextrow <- as.matrix(as.data.frame(Data[n]))
      nextrow <- subset(nextrow, nextrow[,1] != "")
      nextrow <- rbind(nextrow, matrix(NA, ncol=1, nrow=(Row-n)))
      nextrow <- nextrow[2:nrow(nextrow),]
      numericList <- as.numeric(nextrow)
```

```
      numericMatrix <- rbind(numericMatrix, t(as.matrix(numericList)))
    }
  }
fstpval<-numericMatrix
bonferroni<-0.05/91


#Plot values
Pvalue<-na.exclude(as.vector(fstpval))
plot(Pvalue,cex=0.2)
abline(h=bonferroni)

# Pegando os nomes das pops no arquivo do Arlequin
labels<-getNodeSet(doc,"//pairDistPopLabels")
labels2<-unlist(xmlApply(labels,xmlValue))
labels2<-gsub(".[0-9]:\t","",labels2)
labesl2<-gsub("\n","",labels2)
labels3<-unlist(strsplit(labels2,"\n"))[-c(1:4)]
labels3<-gsub(" ","",labels3)
#transformando a matriz em um objeto "dist" no R
colnames(fst)<-labels3

fst.dist<-as.dist(fst)


require(stargazer)
mm<-as.matrix(fst.dist)

##nomes na ordem
nomesEmordem<-colnames(mm)
mm[upper.tri(mm,diag=T)]<-NA
stargazer(mm,title="Valores de FST pareado entre as 14
popula  es de P. vivipara.")

require(ape)
fst.tree<-nj(fst.dist)
fst.tree<-root(fst.tree,"BI",resolve.root=T)
write.tree(fst.tree,"fst.tree")
plot(fst.tree,use.edge.length=F)
```

APÊNDICE B – Código para rodar o Structure em modo paralelo.

arquivos de parametros gerados no Structure com interface gráfica

```
#!/bin/bash
foo(){
    local run=$1
    ./structure -m parms/mainparams.nov16.k1  -i
      parms/project_data -o results/k1+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k2  -i
      parms/project_data -o resultados2/k2+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k3  -i
      parms/project_data -o resultados2/k3+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k4  -i
      parms/project_data -o resultados2/k4+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k5  -i
      parms/project_data -o resultados2/k5+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k6  -i
      parms/project_data -o resultados2/k6+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k7  -i
      parms/project_data -o resultados2/k7+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k8  -i
      parms/project_data -o resultados2/k8+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k9  -i
      parms/project_data -o resultados2/k9+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k10  -i
      parms/project_data -o resultados2/k10+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k11  -i
      parms/project_data -o resultados2/k11+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k12  -i
      parms/project_data -o resultados2/k12+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k13  -i
      parms/project_data -o resultados2/k13+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k14  -i
      parms/project_data -o resultados2/k14+$(date +"%N")
    ./structure -m parms/mainparams.nov16.k15  -i
```

```
        parms/project_data -o resultados2/k15+$(date +"%N")

}

# Rodar 20 vezes cada configuracao
for run in `seq 1 20`; do foo "$run" & done
```

APÊNDICE C – Códigos para análise de diversidade em microssatélite

```
  # Carregamento de fun   o auxiliar para realizar HW

resort_microsat <- function(x){
  # Gather all alleles, convert to numeric, and reorder
  alls <- adegenet::alleles(x)
  alln <- lapply(alls, as.numeric)
  alln <- lapply(alln, order)

  # Loop over the names and paste together
  # the locus name and the sorted alleles.
  alls <- lapply(names(alls),
     function(i) paste(i, alls[[i]][ alln[[i]] ], sep = "."))

  # Convert to a vector an match the name to
  # the column names of the data matrix.
  cols <- unlist(alls, use.names = FALSE)
  cols <- match(cols, colnames(tab(x)))
  return(x[, cols])
}


require(pegas)
require(adegenet)
#Carregar os dados de localiza   o
gps<-read.table("gps.txt",sep=",")


#### Carregar o conjunto de dados de popula   o dados brutos.

Raw.sem.correcao<-read.table("BRASILreavalforR")

# Substituir os valores faltantes
Raw.sem.correcao[Raw.sem.correcao==0]=NA
```

```
colnames(Raw.sem.correcao)<-c("pop",",",paste0("pvm",1:16))

# Transformando em um objeto  da classe genind

geno.sem.correcao<-df2genind(Raw.sem.correcao[,-c(1,2)], ncode=3,pop=Raw.sem.



# Introduzindo dados de gps no obj genind
rownames(gps)<-gps$V1
perpop<-table(geno.sem.correcao@pop)
gps<-gps[names(perpop),]

xy=matrix(NA, ncol=2)
for(i in 1:14){

  xy<-rbind(xy,matrix(rep(c(gps[i,2],gps[i,3]),
                          each=perpop[i]),ncol=2))
}
geno.sem.correcao$other$latlong<-xy[-1,]

#Gerando obj genopop
geno.pop.sem.correcao<-genind2genpop(geno.sem.correcao)

# Sumario dos dados
sum.pop.sem.correcao<-summary(geno.sem.correcao)

### Carregar o conjunto de dados de popula  o dados corrigidos.
Raw.Br1.correcao<-read.table("BRASILbrok1forR")

# Substituir os valores faltantes
Raw.Br1.correcao[Raw.Br1.correcao==0]=NA
colnames(Raw.Br1.correcao)<-c("pop",",",paste0("pvm",1:16))

# Transforma  o em um obj genind
geno.Br1.correcao<-df2genind(Raw.Br1.correcao[,-c(1,2)], ncode=3,pop=Raw.Br1.

# Introduzindo dados de gps no obj genind
rownames(gps)<-gps$V1
perpop<-table(geno.Br1.correcao@pop)
```

```
gps<-gps[names(perpop),]

xy=matrix(NA, ncol=2)
for(i in 1:14){

  xy<-rbind(xy, matrix(rep(c(gps[i,2], gps[i,3]),
                           each=perpop[i]), ncol=2))
}
geno.Br1.correcao$other$latlong<-xy[-1,]

# Gerando obj genopop
geno.pop.Br1.correcao<-genind2genpop(geno.Br1.correcao)

#Sumario dos dados
sum.pop.Br1.correcao<-summary(geno.Br1.correcao)




## Teste HW

pops.sem.correcao<-levels(geno.sem.correcao@pop)
## Fazendo teste de HW
hw.table.sem.correcao<-matrix(NA, nrow=16)
for(i in 1:14){
  set.seed(42)
  tt<-hw.test(resort_microsat(
        geno.sem.correcao[,pop=pops.sem.correcao[i]]),B=1000)
  hw.table.sem.correcao<-cbind(hw.table.sem.correcao, tt[,4])
}

hw.table.sem.correcao<-hw.table.sem.correcao[,2:15]
colnames(hw.table.sem.correcao)<-pops.sem.correcao
#corre   o de bonferroni
pvalue<-0.05/(16*14)
#Numero de testes significativos
hw.Total.sem.correcao<-sum(hw.table.sem.correcao<pvalue)
hw.porloco<- rowSums(hw.table.sem.correcao<pvalue)
```

```
pops.Br1.correcao<-levels(geno.Br1.correcao@pop)

hw.table.Br1.correcao<-matrix(NA,nrow=16)
for(i in 1:14){
  set.seed(42)
  tt<-hw.test(resort_microsat(
        geno.Br1.correcao[,pop=pops.Br1.correcao[i]]),B=1000)
  hw.table.Br1.correcao<-cbind(hw.table.Br1.correcao,tt[,4])
}

hw.table.Br1.correcao<-hw.table.Br1.correcao[,2:15]
colnames(hw.table.Br1.correcao)<-pops.Br1.correcao
pvalue<-0.05/(16*14)
# N mero de testes significativos
hw.Total.Br1.correcao<-sum(hw.table.Br1.correcao<pvalue)
hw.porloco.Br1.correcao<-rowSums(hw.table.Br1.correcao<pvalue)
```