**Appendix 2.**

**Calculation of partial inbreeding coefficients in R**

In this Appendix, we will show the code in R used to obtain the partial inbreeding coefficients for each of the animals, based on the example given in Appendix 1. The implementation of the program is based on the Mendelian decomposition (García-Cortés et al., 2010) , later used to compute the inbreeding load effects (Varona et al., 2019).

consang=function(animal,sire,dam){  
# The program reads the pedigree in numeric form.   
 if(any(is.na(sire)|is.na(dam))){  
 warning('NAs will be converted to 0')  
 sire[is.na(sire)]=0  
 dam[is.na(dam)]=0  
 }

if(any(is.na(animal))) stop('unknown animals in pedigree')  
 if(any(!is.numeric(animal)|!is.numeric(sire)|!is.numeric(dam))){  
 stop('non-numeric variables')  
 }   
  
# The program requires that the dam and sire are known, otherwise, the program generates # them.

s=length(animal)  
ic=c(); ic2=0  
for(i in 1:s){  
 if(sire[i]==0 & dam[i]!=0){ic2=ic2+1  
 }else if(sire[i]!=0 & dam[i]==0){ic2=ic2+1  
 }else{ic=ic}  
ic=c(ic,ic2)  
}  
ic=max(ic)  
# We create the new, complete pedigreegen=matrix(nrow=s+ic,ncol=3,0)  
it=c(); it2=0  
for(i in 1:s){  
 ia=animal[i]+ic  
 ip=im=0  
 ## case 1: sire unknown  
 if(sire[i]==0 & dam[i]!=0){  
 it2=it2+1  
 ip=it2  
 im=dam[i]+ic  
 }  
 ## caso 2: dam unknown  
 if(sire[i]!=0 &dam[i]==0){  
 it2=it2+1  
 ip=sire[i]+ic  
 im=it2  
 }  
 ## case 3: sire and dam unknown if(sire[i]==0 &dam[i]==0){  
 ip=im=0  
 it2=it2  
 }  
 ## case 4: sire and dam known  
 if(sire[i]!=0 & dam[i]!=0){  
 ip=sire[i]+ic  
 im=dam[i]+ic  
 it2=it2  
 }  
 it=c(it,it2)  
 gen[i+ic,1]=ip  
 gen[i+ic,2]=im  
}  
s=s+max(ic)  
  
# We calculate the coefficients for each of the animals in the pedigree## We create an empty list where we keep the results by animal  
CONSANG=list()  
  
for(anim in 1:s){  
# --- code to see the progress, by animalSys.sleep(0.1)message('Calculating for animal: ',anim)flush.console() # update GUI console#---------------------------   
 t=0; t2=0  
 gen[anim,3]=1  
 m=anim  
  
 ## Ancestors of the animal are given a 1 (as present in the genealogy)  
 for(j in m:1){  
 if(gen[j,3]==1){  
 if(gen[j,1]>0) gen[gen[j,1],3]=1  
 if(gen[j,2]>0) gen[gen[j,2],3]=1  
 }  
 }  
  
 ### Construction of a reduced pedigree based on the ancestors present icode=which(gen[,3]==1)  
 kk=length(icode)  
 ggg=gen[icode,]  
 icode[kk]=s  
 if(!is.null(nrow(ggg))){ggg[,3]=icode  
 }else{ggg[3]=icode}  
  
 #### Renumbering of the reduced pedigree, from 1 to n  
 if(!is.null(nrow(ggg))){  
 pedNUM=cbind(ggg[,3],1:length(ggg[,1]))  
 ggg=cbind(match(ggg[,1],pedNUM[,1]),  
 match(ggg[,2],pedNUM[,1]),pedNUM[,2])  
 ggg[is.na(ggg)]=0  
 }else{  
 pedNUM=cbind(ggg[3],1:length(ggg[1]))  
 ggg=cbind(match(ggg[1],pedNUM[,1]),  
 match(ggg[2],pedNUM[,1]),pedNUM[,2])  
 ggg[is.na(ggg)]=0  
 }  
  
 ##### Calculation of inbreeding for each individual  
 a=matrix(nrow=kk, ncol=kk, 0)  
 ff=rep(0,kk)  
 for(i in 1:kk){   
 for(j in i:kk){   
 if(i==j){   
 if(ggg[i,1]==0|ggg[i,2]==0){   
 a[i,i]=1.   
 }else{a[i,j]=1.+0.5\*a[ggg[i,1],ggg[i,2]]}   
 }else{   
 aux=0.   
 if(ggg[j,1]>0) aux=aux+0.5\*a[i,ggg[j,1]]   
 if(ggg[j,2]>0) aux=aux+0.5\*a[i,ggg[j,2]]   
 a[i,j]=a[j,i]=aux  
 aux=0.   
 }   
 }   
 ff[i]=a[i,i]   
 }   
 dd=rep(0,kk)  
  
 for(i in 1:kk){   
 dd[i]=1.   
 if(ggg[i,1]>0){   
 if(ggg[i,2]>0) dd[i]=0.5-0.25\*(ff[ggg[i,1]]+ff[ggg[i,2]]-2)   
 }   
 }  
 ###### Inbreeding decomposition for each individual in the pedigree  
 for (i in 1:kk){   
 a=matrix(nrow=kk, ncol=kk, 0)   
 # - Calculation of delta for individual "i"  
 if(ggg[i,2]==0){   
 delta=0.5   
 }else{delta=0.125\*(2.-ff[ggg[i,1]])+0.125\*(2.-ff[ggg[i,2]])}   
 # - Calculation of the upper triangular matrix   
 for(j in i:kk){   
 if(j==i){   
 a[j,j]=delta   
 }else if(ggg[j,2]>0){   
 a[j,j]=0.5\*(a[j,ggg[j,1]]+a[j,ggg[j,2]])   
 }   
 for(k in j+1:kk){   
 if(!k>kk){   
 if(ggg[k,2]>0) a[j,k]=a[k,j]=0.5\*(a[j,ggg[k,1]]+a[j,ggg[k,2]])   
 }  
 }   
 }  
  
 # - Saving results greater than 0  
 if(ggg[kk,1]>0){  
 if(a[ggg[kk,1],ggg[kk,2]]>0) t[i]=a[ggg[kk,1],ggg[kk,2]]   
 }  
 }  
  
 # We order the results into a dataframe and re-renumber with the ## original pedigree nomenclature  
 consang=data.frame(ancestor=(1:i)-ic)  
 length(t)=length(consang$ancestor)  
 t[is.na(t)]=0   
 consang$Fp=t; consang[is.na(consang)]=0  
 tmp=which(consang$Fp==0); consang=consang[-tmp,]  
 if**(**nrow(consang)!=0){  
 consang$ancestor=abs(consang$ancestor)   
 consang=aggregate(consang$Fp, by=list(ancestor=consang$ancestor), FUN=sum)  
 consang$animal=anim-ic  
 names(consang)[2]='Fp'  
 consang=consang[,c(3,1,2)]  
 }  
 CONSANG[[anim]]=consang  
}  
## We bind the results into one data frame   
CONSANG=do.call('rbind',CONSANG)  
CONSANG=CONSANG[order(CONSANG$animal),]  
return(CONSANG)  
}

Running the code given above in R, the results of the partial inbreeding coefficients are given when the following R code is run: consang(animal=iped$animal, sire=iped$sire, dam=iped$dam)