Annotated R code for a deer parentage analysis

This uses the example data and methods described in:

Dodds, K G, McEwan, J C, Brauning, R, Van Stijn, T C, Rowe, S J, McEwan, K M and Clarke, S M (2019) Exclusion and genomic relatedness methods for assignment of parentage using genotyping-by-sequencing data. *G3: Genes, Genomes, Genetics* (in press). doi: 10.1534/g3.119.400501.

The data files are available from the paper's supplementary material, deposited in:
https://gsajournals.figshare.com/articles/Supplemental_Material_for_Dodds_et_al_2019/9243167
(doi: 10.25387/g3.9243167)

The code has been updated to work with version 1 of the KGD software.

Command	Description
genofile <- "HapMap.hmc.txt.gz"	Name of data file (example file can be downloaded from Supplementary Material)
pedfile <- "DeerPedGBS.csv"	"pedigree file" for parentage analysis (can be downloaded from Supplementary Material)
groupsfile <- "Ped-Groups.csv"	"groups file" for parentage analysis (can be downloaded from Supplementary Material)
sampdepth.thresh <- 0.3	Remove results with mean depth less than 0.3 (used in GBSsummary)
cex.pointsize <- 1.2	Increase default size of text in graphical output by 20%
functions.only <- TRUE	Do not run the KGD code when sourcing it in
sink("GBSParDeerOut.txt")	Redirect screen output to specified file
source("GBS-Chip-Gmatrix.R")	Load the main KGD functions (not run automatically because *functions.only* is TRUE). Obtained from https://github.com/AgResearch/KGD.
readGBS()	Read the input file of "reference" and "alternate" allele counts for each sample and SNP. The default format is used ("uneak").
outlevel <- 1	Reduce the amount of QC output
GBSsummary()	Run the main function for QC and setting up structures for further analyses. 12 samples are removed due to mean sample depth < 0.3. 2646 SNPs with mean depth <0.1 or with MAF=0 are removed.
breed <- read.table(text=seqID,sep="_", stringsAsFactors=FALSE)[,1]	*seqID* contains the sample identifiers from the input file. In this example the identifiers are of the form <breed>_<number>, where <breed> is one of R (Red Deer) or W (Wapiti). This instruction extracts the text before the "_" into a character vector called *breed*.
fcolo <- c("darkblue","darkred") [match(breed,c("W","R"))]	Set up a vector of colours (*fcolo*) to be used (darkblue for Wapiti, darkred for Red Deer)
snpsubset <- which(HWdis > -0.05)	Vector containing the positions of SNPs that pass the filter of Hardy Weinberg disequilibrium (*HWdis*) > -0.05. See finplot.png for a depiction of *HWdis*, MAF and mean SNP depth for the SNPs.

GHW <- calcG(npc=4, snpsubset=snpsubset, sfx="RWHW")	Main function for estimating relatedness. Here the filtered SNPs are used. A PCA and heatmap are requested along with the standard output. *npc*=4 principal components are output and plotted. The output object contains a genomic relatedness matrix using the KGD method (*G5*) as well as an object containing the PCA results (*PC*). Output graphics include "RWHW" as part of the name. Setting *npc*=-4 will omit the heatmap (which can use a lot of CPU time with larger datasets). Setting *npc*=0 will omit the PCA, although it is recommended to include the PCA for diagnostic purposes if possible. The plot GRWHWdiagdepth.png shows estimated self-relatedness as a function of mean sample depth. Within a set of samples sequenced using the same protocol, we would not expect to see a relationship. Sometimes a negative relationship is observed which may be due to non-optimised lab protocols.
G5 <- GHW$G5	Extract the KGD GRM from *GHW*.
GCheck <- "G5"	Specify the GRM to use in the parentage analysis.
set.seed(230985)	Specify a seed so that the same bootstrap results are generated if the code is rerun.
source("GBSPedAssign.R")	Load the main pedigree analysis functions (not run automatically because *functions.only* is TRUE). Obtained from https://github.com/AgResearch/KGD.
RWResults <- GBSPed()	Run the parentage analysis. The main results are in the data frame called BothMatches within the RWResults object, and are also written to the file BothMatches.csv. Various other diagnostics files are produced.
write.csv(RWResults$BothMatches,"BothMatchesRW.csv",row.names=FALSE,quote=FALSE) # Combined breeds	Rewrite the main results using a different filename (for future reference)
dir.create("W")	Create a subfolder for a Wapiti-only analysis
setwd("W")	Set the work directory to the Wapiti folder
indW <- which(breed=="W")	Create a vector of positions of Wapiti animals in the data.
pW <- calcp(indsubset=indW)	Calculate allele frequencies (based on allele counts) using only the Wapiti data.
snpsubset <- which(HWdis > -0.05 & pW > 0 & pW < 1)	Find the SNPs that pass the Hardy-Weinberg threshold (based on the full dataset) and that are not monomorphic in the Wapiti data.
GHWW <- calcG(snpsubset, indsubset=indW, sfx="W",puse=pW,calclevel=1 , npc=-2)	Calculate the GRM for the Wapiti only data, using the Wapiti allele frequencies. The first 2 principal components are found (to be used later). No heatmap is requested, and some other diagnostics are supressed (by calclevel=1).
G5W <- GHWW$G5	Extract the KGD GRM
seqIDW <- seqID[indW]; if(length(GHWW$samp.removed) > 0) seqIDW <- seqIDW[-GHWW$samp.removed]	Obtain the seqID vector for the Wapiti. Ensure that this list corresponds to the individuals in *G5W* (sometimes individuals are removed to ensure that all pairs have exceed the minimum specified co-call rate in *calcG* (in this case the default threshold of 0 was used). The reduced set of SNPs may have resulted in a few pairs with no SNPs in common, although this does not usually happen (especially if low depth samples are removed

	initially).
GCheck <- "G5W"	Specify the GRM to use in the parentage analysis.
puse <- pW	Specify the allele frequencies to use in the parentage analysis
indsubset <- indW	Specify the individuals to use in the parentage analysis (Wapiti animals)
rm(minr4inb)	Remove the *minr4inb* (minimum parent relatedness to use for checking the inbreeding threshold). This ensures the default (no minimum) is used. *minr4inb* is set to the minimum parent pair relatedness during the parentage analysis. Removing the variable allows it to be reset.
pedfile <- "../DeerPedGBS.csv"	Specify the location of the pedigree file relative to the current working directory
groupsfile <- "../Ped-Groups.csv"	Also for the groups file
WResults <- GBSPed()	Run the pedigree analysis for the Wapiti data
MatchesW <- WResults$BothMatches	Place the parentage results into another data frame
write.csv(MatchesW,"BothMatchesW.csv",row.names=FALSE,quote=FALSE)	Rewrite the results to a file with a name different to the standard output name (e.g. this allows these results and the combined breed results to be opened together in Excel)
bestparPCA(GHWW, sfx="W",keypos="bottomright", pedinfo=WResults$pedinfo, BothMatches=MatchesW)	Produce a PCA plot of all W animals, with lines connecting each offspring to it's best matching possible parent. The PCA is plotted in PC-BestParentsW.png.
# Alt models	
uY <- which(MatchesW$BothAssign=="Y")	Identify the offspring that were assigned both parents
bbopt <- optimize(ssbbmm,lower=0,upper=20, tol=0.001, uuse=uY, pedinfo=WResults$pedinfo, BothMatches=MatchesW)	Fit the beta-binomial model to the assigned trios. Note that the pedinfo and BothMatches arguments refer to data frames from the GBSPed output (this was not needed in version 0).
depth2K <- depth2Kchoose (dmodel="bb", bbopt$minimum)	Change the allele sampling model to the fitted beta-binomial model
mmstatsW.bb <- mismatch.2par(MatchesW$IndivID, MatchesW$BestFatherMatch, MatchesW$BestMotherMatch, pedinfo=WResults$pedinfo)	Calclulate mismatch rates using the fitted beta-binomial model
names(mmstatsW.bb) <- paste0(names(mmstatsW.bb),".bb")	Add ".bb" to the variable names of the beta-binomial mismatch rates
mpopt <- optimize(ssmpmm,lower=0.5,upper=0.9, tol=0.001, uuse=uY, pedinfo=WResults$pedinfo, BothMatches=MatchesW)	Fit the modified p model to the assigned trios
depth2K <- depth2Kchoose (dmodel="modp", mpopt$minimum)	Change the allele sampling model to the fitted modified p model
mmstatsW.mp <- mismatch.2par(MatchesW$IndivID, MatchesW$BestFatherMatch, MatchesW$BestMotherMatch, pedinfo=WResults$pedinfo)	Calclulate mismatch rates using the fitted modified p model
names(mmstatsW.mp) <- paste0(names(mmstatsW.mp),".mp")	Add ".mp" to the variable names of the modified p mismatch rates
MatchesW <- cbind(MatchesW, mmstatsW.bb, mmstatsW.mp)	Add the beta-binomial and modified p model mismatch rates to the Wapiti parentage results
write.csv(MatchesW,"BothMatchesW.csv", row.names=FALSE, quote=FALSE)	Write these results to a file (overwrites previous file)
depth2K <- depth2Kchoose (dmodel="modp") # back to default model	Reset the allele sampling model to the default model (can specify "modp" or "bb" with no parameter – in both

	cases reverts to the standard binomial model)
setwd("..")	Set the working directory to the original analysis folder.
dir.create("R")	Repeat the Wapiti analysis workflow for the Red Deer data
setwd("R")	
indR <- which(breed=="R")	
pR <- calcp(indsubset=indR)	
snpsubset <- which(HWdis > -0.05 & pR > 0 & pR < 1)	
GHWR <- calcG(snpsubset, indsubset=indR, sfx="R",puse=pR,calclevel=1 , npc=-2)	
G5R <- GHWR$G5	
seqIDR <- seqID[indR]; if(length(GHWR$samp.removed) > 0) seqIDR <- seqIDR[-GHWR$samp.removed]	
GCheck <- "G5R"	
puse <- pR	
indsubset <- indR	
rm(minr4inb)	
RResults <- GBSPed()	
MatchesR <- RResults$BothMatches	
write.csv(MatchesR, "BothMatchesR.csv", row.names=FALSE, quote=FALSE)	
bestparPCA(GHWR, sfx="R", keypos="bottomright", pedinfo=RResults$pedinfo, BothMatches=MatchesR)	
# Alt models	
uY <- which(MatchesR$BothAssign=="Y")	
bbopt <- optimize(ssbbmm,lower=0,upper=20, tol=0.001, uuse=uY, pedinfo=RResults$pedinfo, BothMatches=MatchesR)	
depth2K <- depth2Kchoose (dmodel="bb", bbopt$minimum) # 3.956	
mmstatsR.bb <- mismatch.2par(MatchesR$IndivID, MatchesR$BestFatherMatch, MatchesR$BestMotherMatch, pedinfo=RResults$pedinfo)	
names(mmstatsR.bb) <- paste0(names(mmstatsR.bb),".bb")	
mpopt <- optimize(ssmpmm,lower=0.5,upper=0.8, tol=0.001, uuse=uY, pedinfo=RResults$pedinfo, BothMatches=MatchesR)	
depth2K <- depth2Kchoose (dmodel="modp", mpopt$minimum)	
mmstatsR.mp <- mismatch.2par(MatchesR$IndivID, MatchesR$BestFatherMatch, MatchesR$BestMotherMatch, pedinfo=RResults$pedinfo)	
names(mmstatsR.mp) <- paste0(names(mmstatsR.mp),".mp")	
MatchesR <- cbind(MatchesR, mmstatsR.bb, mmstatsR.mp)	
write.csv(MatchesR, "BothMatchesR.csv", row.names=FALSE, quote=FALSE)	

depth2K <- depth2Kchoose (dmodel="modp")	
setwd("..")	
sink()	Stop writing screen output to a file