Software for GBS-based relationship calculations v0.8.3

Author: Ken Dodds
Address: Invermay Agricultural Centre, Puddle Alley, Private Bag 50034, Mosgiel 9053, New Zealand
Email: ken.dodds at agresearch.co.nz
Date: 29 June 2019

Contents

Background

R code is available for the analysis of genotyping-by-sequencing (GBS) data, primarily to construct a genomic relationship matrix ('G matrix') for the genotyped individuals. The code can be used on its own, or incorporated into other R programs. There are QC tools (primarily graphical output), relationship estimation tools, pedigree verification tools and pedigree 'mix and match' tools. The latter two operations require additional input information about the samples genotyped.

In this document, 'Individual' or 'sample' generally refers to the genotyping unit (possibly combined, if the same individual or sample is genotyped multiple times). Familial relationships are given the labels 'Father', 'Mother' and 'Offspring' (as appropriate).

The methods used are as described in Dodds *et al*. (2015), Bilton *et al*. (2019) and Dodds *et al*. (2019). Unless specified, relatedness estimates in this documentation refer to those using the 'G5' method of Dodds *et al*. (2015).

Program structure

There are two separate analysis program files, the first (GBS-Chip-Gmatrix.R) for genotype QC and relationship matrix construction and the second (GBSPedAssign.R) for pedigree verification and/or assignment, based on the related estimates. These programs can be invoked from another program file (using the *source* command), or users can insert all or parts of these programs into their own code. For the purposes of this documentation, it is assumed the first method is used, with calling program named GBSRun.R. If either of these programs are being called multiple times in an R session, care should be taken to make sure any default values still apply for the subsequent call(s), otherwise they should be removed (rm) or set specifically.

Calling program (GBSRun.R)

Variables that can be set or commands that can be run are shown in the following tables.

Variable / command	Type[1]	Description
genofile	V	Name (including path) of the genotype file. Default value is "HapMap.hmc.txt".
gform	V	Type of genotype file. Default is “uneak”; other options are “Tassel”, “TagDigger”, “ANGSDcounts” or “Chip”.
sampdepth.thresh	V	Minimum mean sample depth for retaining sample results. Default is 0.01.
snpdepth.thresh	V	Minimum mean SNP depth for retaining SNPs. Default is 0.01.
hirel.thresh	V	Lower threshold for reporting highly related individuals, and upper threshold for displaying positive control pairs which don’t seem sufficiently related. Default is 0.9.
triallelic.thresh	V	Upper threshold for the proportion of ignored reads for the third allele – SNPs with a higher proportion are removed (as a triallelic variant). Relevant only to the ANGSDcounts input format. Default is 0.005 (0.5%).
cex.pointsize	V	Relative value of pointsize used in output graphics. This has a default value of 1.
functions.only	V	Set to TRUE to source GBS-Chip-Gmatrix.R for setting up functions (not reading data etc). Default is FALSE.
alleles.keep	V	Set to TRUE to retain an updated version of alleles. This object is needed for some downstream uses, e.g. for writing VCF files or for the calculation of linkage disequilibrium (using GUS-LD). Default is FALSE.
outlevel	V	Integer (1-9) determining the level of output created – higher numbers give more output. At present only two levels are active; 5 to 9 give the full output while 1 to 4 gives less output. A value less than 8 will supress the sampled alleles setup and analysis (reducing time). Default is 9 (all available output)
use.Rcpp	V	Set to FALSE to prevent the C++ versions of functions being used. Default is TRUE.
nThreads	V	The number of OpenMP threads to be used by C++. The default is 4. Using 0 means all available threads would be used.
iemm.thresh	V	Identity excess mismatch rate threshold for displaying non-matching results putatively from the same individual. Currently only used in the posC-EMM plot.
source	C	Invoke GBS-Chip-Gmatrix.R code, to run QC procedures and define functions, e.g. the genomic relationship matrix function (*calcG*)
calcG	C	Calculate genomic relationship matrices. May be invoked several

	times with different options.

[1] Type is V for a variable to be set, or C for a command to be invoked or function to be run.

The following table shows variables and commands that are specific to the pedigree program.

Variable / command	Type[1]	Description
pedfile	V	Name of file containing pedigree and/or parent group information
groupsfile	V	Name of file containing which individuals are in which parent groups
GCheck	V	The name (as a string) of the G matrix to use for parent verification or assignment This must be set before calling GBSPedAssign.R.
indsubset	V	The subset of individuals used to calculate the matrix specified in *GCheck*.
rel.thresh	V	The relatedness threshold to use for parent verification or assignment, if the corresponding parent sex-specific threshold (*rel.threshF* or *rel.threshM*) has not been set. This has a default value of 0.4.
rel.threshF	V	The relatedness threshold to use for father verification or assignment. This has a default value of *rel.thresh*.
rel.threshF	V	The relatedness threshold to use for mother verification or assignment. This has a default value of *rel.thresh*.
mindepth.mm	V	Minimum depth to be used for calculating mismatch proportions in parent matching. Default is 1 (use all results).
snpsubset	V	The subset of SNPs to be used for calculating mismatch rates or for bootstrapping (usually the same set as used for calculating calculate the matrix specified in *GCheck)*. Default is all SNPs.
emm.thresh	V	The excess mismatch rate threshold to use for parent assignment. This has a default value of 0.01.
emm.thresh2	V	The excess mismatch rate threshold to use for parent-pair assignment. This has a default value 2 x emm.thresh.
emmdiff.thresh2	V	The excess mismatch rate difference (from that for the most related father and most related mother) threshold to use for suggesting an alternate parent-pair assignment. This has a default value of 0.
inb.thresh	V	The lower threshold for the difference between parent relatedness and twice the estimated inbreeding to exclude a parent-pair match with the inbreeding check. This has a default value of 0.2.
minr4inb	V	The lower threshold on parent relatedness to exclude a parent-pair match with the inbreeding check. This has a default value of NULL (no minimum).
boot.thresh	V	If the relatedness with the 2nd best parent is within *boot.thresh* of that for the best parent, a bootstrapping procedure will be invoked to further compare these possible matches. Default value of 0.05.
depth.min	V	Minimum mean depth of SNPs to be used for boostrapping. Default value is 0.
depth.max	V	Maximum mean depth of SNPs to be used for boostrapping. Default value is 0.
puse	V	Allele frequencies to be used bootstrapping. Default is to use *p*.
nboot	V	Number of bootstrap replicates. Default value is 1000.
boota.thresh	V	The upper threshold on bootstrap reliability for excluding a parent match with the bootstrapping check. This has a default value of 99.
matchmethod	V	The method used to find the best 2 matching parents (fathers

		and/or mothers). The default value is “rel” where the maximum relatedness is used. The alternative is “EMM” where minimum EMM is used. At this stage bootstrapping and alternate assignments were based on using “rel” so may not give sensible results with “EMM”.
source	C	Invoke GBS-PedAssign.R code to verify parents (if given) or assign parents (if parent groups are given)

[1] Type is V for a variable to be set, or C for a command to be invoked or function to be run.

Relatedness estimation program (GBS-Chip-Gmatrix.R)

This program performs some QC diagnostics, rudimentary data cleaning and defining a function (*calcG*) for relatedness estimation and reporting. A number of other functions are defined, such as those for checking and report on positive controls (negative control checks, based on a specified sample naming system, are yet to be included). Any procedures or output relating to depth are not implemented for chip data. The use of depth information to construct the GRM can be modified (see depth2K section).

Samples with very low depth are dropped from the analyses. The threshold is a mean depth of *sampdepth.thresh* (default of 0.01, but can be set in the calling program) or with a maximum depth of one (including those with no genotype calls). Samples that are dropped are reported in the program output, as is the remaining number of samples.

SNPs with no data or with a MAF (minor allele frequency) of zero are dropped. The remaining number of SNPs is reported.

Some basic statistics are reported: Proportion of missing genotypes is the number of SNP x individual combinations with no allele calls; Mean sample depth is the average depth (number of reads of either allele) for a sample.

The default action when sourcing GBS-Chip-Gmatrix.R is to read the data file and run some QC procedures, as well as define various functions. If functions.only is set to TRUE, then only the function definition occurs. The default action can then be mimicked using the pair of commands:

```
readGBS()
GBSsummary()
```

These functions are not yet described in the documentation. Additional processing can be inserted between these statements, for example to manually remove samples or SNPs. The following objects need to be maintained correctly, before *GBSsummary* is run: *nsnps, SNP_Names, seqID, nind, alleles*. If *GBSsummary* has been run once, it could be re-run, e.g. after merging results from the same individual. In that case (detected by the presence of *depth*), processing that uses *alleles* (which is not recalculated in *mergeSamples* unless *keep.alleles* is set to TRUE) is omitted. This means that *depth* needs to correspond to actual depth (may need to replace with values in *depth.orig*). Alternatively, if *alleles* is present and corresponds to the current data, then *depth* could be removed so that it gets recalculated. This is mainly to obtain *genon* and *depth* which will be assumed to be present and correct, but it should be noted that *p* is not recalculated. *p* should remain unchanged when samples are merged, but could change, for example if the *sampdepth.thresh* is changed between calls to *GBSsummary*.

Some functions have been coded in C++ to improve speed. These will be used instead of the corresponding R functions if the libraries Rcpp and RcppArmadillo are installed, and use.Rcpp is TRUE (the default value).

Output - files

SampleStats.csv contains call rates for each sample, along with mean sample depths (for GBS data).

AlleleFreq.png is a plot of allele frequencies calculated using different methods (and as given, if the uneak format is used).

CallRate.png shows a histogram of sample call rates (proportion of SNPs with a result for a sample).

SampDepth.png plots mean sample depth against median sample depth.

SampDepth-scored.png plots mean sample depth, over SNPs that are scored for the individual, against mean sample depth over all SNPs for the individual.

SampDepthHist.png is a histogram of mean sample depths

SampDepthCR.png plots mean sample depth against call rate.

SNPDepthHist.png is a histogram of SNP depths (number of reads of either allele averaged over samples)

SNPCallRate.png is a histogram of SNP call rates (proportion of samples with a result for a SNP)

SNPDepth.png plots SNP depth against mean SNP depth (on a log scale). This may reveal SNPs that are called infrequently, but when they are called have good depth (these SNPs may be near the boundary of a size selection step in the laboratory).

finplot.png plots Hardy-Weinberg disequilibrium (HWD) against MAF, shaded by the SNP depth. HWD is the proportion of (reference allele) homozygotes minus the expected proportion (under Hardy-Weinberg equilibrium). HWD is the same whichever allele is used in the calculation. The 'fin plot' may reveal sets of SNPs that do not follow Mendelian inheritance, for example apparent SNPs in duplicated regions.

HWdisMAFsig.png is similar to the fin pot, but with shading by *l10pstar*, the $\log_{10}$ p-value corresponding to the depth-adjusted chi-squared test statistic of Hardy Weinberg equilibrium (versions prior to v0.702 used the likelihood ratio test statistic for HWD).

LRT-QQ.png is a QQ plot for the likelihood ratio test statistic for HWD.

LRT-hist.png is a histogram of the likelihood ratio test statistic for HWD.

X2star-QQ.png is a QQ plot for the depth-adjusted chi-square test statistic for HWD.

MAF.png is a histogram of the MAFs for each SNP (based on observed genotypes).

Variables defined

These include:

Variable	Description
nind	Number of samples analysed (after initial QC)
nsnps	Number of SNPs analysed (after initial QC)
seqID	Identifiers for each sample
SNP_Name	Identifiers for each SNP
chrom	chromosome label (character), if *gform* is Tassel
pos	chromosome position (numeric), if *gform* is Tassel
alleles	matrix (*nind* x 2**nsnps*) of read counts. The results for each SNP are in consecutive coloumns.
genon	matrix (*nind* x *nsnps*) of numeric genotype calls 0 (homozygous alternate allele), 1 (heterozygous), 2 (homozygous reference allele), NA for missing

depth.orig	matrix (nind x nsnps) of counts for each sample and SNP
sampdepth	mean depth for each sample
snpdepth	mean depth for each SNP
p	allele frequencies on the basis of allele counts
pg	allele frequencies on the basis of genotype calls
HWdis	Hardy-Weinberg disequilibrium (raw)
x2star	Depth-adjusted chi-squared test statistic of Hardy Weinberg equilibrium
l10pstar	$\log_{10}$ p-value corresponding to *x2star*

Function to read TagDigger format files (*readTD*)

This function is for reading TagDigger files. It can be used by the main program (if functions.only is FALSE, the default), but can also be used to read additional files (e.g. to compare results in two different files). The variables *nsnps, seqID, nind,* and *alleles* are defined. See the section on the TagDigger format for more information.

Usage: *readTD*(genofilefn0 = genofile, skipcols=0)

Arguments:

genofilefn0	the name of the file to read. Defaults to *genofile*.
skipcols	the number of columns of input to ignore. Defaults to 0.

Value: NULL

Function to remove samples from objects (*samp.remove*)

This function removes samples from the relevant objects (*alleles, depth, sampdepth, seqID nind*). It would normally be used between calls to *readGBS* and *GBSsummary*.

Usage: *samp.remove*(samppos = NULL, keep=FALSE)

Arguments:

samppos	the positions of the samples to remove. Defaults to NULL.
keep	If TRUE, the samples with positions samppos will be kept and other samples removed. Default value is FALSE

Function to remove SNPs from objects (*snp.remove*)

This function removes samples from the relevant objects (*p, nsnps, SNP_Names, alleles, depth,* and some others). It would normally be used between calls to *readGBS* and *GBSsummary*.

Usage: *snp.remove*(samppos = NULL, keep=FALSE)

Arguments:

snppos	the positions of the SNPs to remove. Defaults to NULL.
keep	If TRUE, the SNPs with positions snppos will be kept and other SNPs removed. Default value is FALSE

Depth functions (*depth2K, depth2Kbb, depth2Kmodp, depth2Kchoose*)

The GBS-Chip-Gmatrix.R program defines a default function for calculating "*K* values", as well as alternate functions (using alternate allele sampling models) and a function to reset the default to one of the alternatives. These functions are relevant for used both self-relatedness estimation and pedigree assignment diagnostics. If a different depth model is required for calculating the self-relatedness, this *depth2K* function should be re-defined before using the *calcG* function (defined below). *K* is the probability of observing an AA genotype, given that the true genotype is AB and the read depth is *k*. These models will be discussed in more detail elsewhere. The function is used within *calcG* for calculating the self-relatedness for G5, and in the pedigree assignment program, for calculating expected mismatch rates.

A function *depth2K* is defined. This function takes a vector of read depths and returns the corresponding set of *K* values. Initially the function is defined using a binomial sampling model (the number of A alleles is binomial with probability parameter 0.5 and sample size the read depth).

depth2Kbb is an alternate depth function which uses a beta-binomial model. This model has two parameters, α and β, but here these are set to be equal, so that P(AA|AB, *k*=1) = 0.5.
Usage: *depth2Kbb* (depthvals, alph=Inf)
Arguments:

depthvals	a vector of read depths
alph	the value of α (and also β) – the default is to use Inf, in which case the binomial model is used.

depth2Kmodp is an alternate depth function which uses a modified *p* value for 2nd and subsequent reads. The modified *p* can be thought of as the probability of seeing the same allele as in the previous read (for that SNP) for a true AB genotype, although because we are only interesting in the probability of all reads being the same allele, it is also the probability of seeing the same allele as *all* previous reads (for a true AB genotype).
Usage: *depth2Kmodp* (depthvals, modp=0.5)
Arguments:

depthvals	a vector of read depths
modp	the modified probability – the default is 0.5, which gives the binomial model. Normally a value ≥ 0.5 would be used to reflect an increased chance of seeing the same allele as in the previous read.

depth2Kchoose is function to re-define *depth2K* to one of the alternative models.
Usage: *depth2K* <- *depth2Kchoose* (dmodel="bb", param)
Arguments:

dmodel	the model to use, either "modp" (to use *depth2Kmodp*), or "bb" to use *depth2Kbb* – the default is "bb" (also used if any other string is used)
param	the parameter to use for the alternative function, used for alph for the bb model, and modp for the modp model.

Function for calculating identity mismatch rates (*mismatch.ident*)

Mismatch rates for comparing two results from putatively the same individual. Currently under development. Used by *posCreport*.
Usage mismatch.ident(seqID1, seqID2, snpsubset=1:nsnps, puse=p, mindepth.mm=1)
Arguments:

seqID1	seqID for first result
seqID2	seqID for second result
snpsubset	a vector of integers (between 1 and *nsnps,* inclusive) of SNPs to be compared. The default is to use all SNPs.
puse	a length *nsnps* vector of allele frequencies to use in the calculations. The default is to use allele frequencies calculated on the basis of allele counts.
mindepth.mm	the minimum depth for a genotype to be used in the comparison

Value: a list containing mmrate (raw mismatch rate), ncompare (number of SNPs compared) and exp.mmrate, the expected mismatch rate.

Function for reporting on positive controls (*posCreport*)

A function, *posCreport*, for reporting on samples which are supposedly from the same individual. These will normally be one or more positive controls, but may also be repeat runs.
Usage posCreport(mergeIDs, Guse,indsubset, Gindsubset, snpsubset=1:nsnps, puse=p)
Arguments:

mergeIDs	a vector of identifiers, ordered as in *Guse*, where samples from the same individuals are given the same identifier
Guse	the G matrix for comparing samples
sfx	text to be included in output file names to allow output from multiple calls or runs to be identified
indsubset	a vector of integers (between 1 and *length(mergeIDs),* inclusive) of individuals in *mergeIDs* (and *Guse*) to be compared. The default is to use all individuals.

Gindsubset	a vector of integers (between 1 and *nind,* inclusive) of the individuals from the full data in *Guse* (normally the same as used for *indsubset* when calling *calcG* to obtain *Guse*)
snpsubset	a vector of integers (between 1 and *nsnps,* inclusive) of SNPs to be compared for mismatch rates. The default is to use all SNPs.
puse	a length *nsnps* vector of allele frequencies to use in the mismatch calculations. The default is to use allele frequencies calculated on the basis of allele counts.

Value: a data frame containing columns mergeID (the ID given in *mergeIDs*), nresults (the number of runs with this ID), selfrel (the average self-relatedness), meanrel (the mean relatedness between all pairs with the given value of mergeID), minrel (the minimum relatedness between all pairs with the given value of mergeID), meandepth (mean of *sampdepth*), mindepth (minimum *sampdepth*), meanCR (mean call rate). Only values of mergeID with nresults >1 are included.

Details: The function displays pairs of results where the estimated relatedness is less than 1 and below the selfrel by at least 1- *hirel.thresh*, and outputs the files:

posCchecks<sfx>.txt a copy of the results displayed on the default output (i.e. low relatedness pairs)

posCreport<sfx>.csv contains the data frame that was returned by the function

SelfRel<sfx>.png a plot of *meanrel* against *selfrel*. The line of equality is shown in red. A grey line gives the boundary where relatedness is lower than 1 and lower than selfrel by more than 1 - *hirel.thresh* (as a guide for results to check).

posC-MM<sfx>.png a plot of mean (over pairs of the same individual) raw mismatch rate against mean expected mismatch rate. The line of equality is shown in red, while a grey line denotes when the difference (identity EMM) is greater than the threshold *iemm.thresh*.

posC-EMM<sfx>.png a scatterplot of matrix mean of *selfrel, meanrel* and identity EMM.

Functions for merging results for the same individual (*mergeSamples*, *mergeSamples2*)

A function, *mergeSamples*, for merging samples from the same individual. The function *mergeSamples2* is similar, see below.

Usage *mergeSamples* (mergeIDs)

Arguments:

mergeIDs	a vector of identifiers for all *nind* samples, such that samples that have the same identifier are to be merged
indsubset	a vector of integers (between 1 and *nind,* inclusive) of individuals to be retained for merging

Value: a list of the following objects:

mergeIDs	a vector of identifiers, as per the input, but ordered as in the other output objects (and with unique values)
nind	the length of *mergeIDs*
seqID	normally one of the seqIDs that correspond to the *mergeIDs*. If the seqIDs can be broken into five parts, using an underscore (_) as a separator, then the second part will be replaced by “merged”, the third part by the number of results merged and the fourth part by “0”
genon	genotype (0/1/2) matrix after merging
depth.orig	depth matrix after merging
alleles	alleles matrix after merging, if *alleles.keep* is TRUE
sampdepth	sample mean read depths after merging
snpdepth	SNP mean read depths after merging
pg	allele frequencies based on genotype calls, after merging
nmerged	number of results merged (1, if not merged) for each individual.

Normally these objects would be used to replace their corresponding values before the merge, but this is not done automatically (it is up to the user). Note that some objects are not merged (e.g. the allele depth matrix, *alleles,* if *alleles.keep* is FALSE) and that the diagnostics produced when sourcing GBS-Chip-Gmatrix.R are not re-done by this function. *mergeSamples* will fail when the number of elements for a merged object (*genon*, and *alleles* if *alleles.keep* is TRUE) exceeds the maximum integer allowed (currently 2^{31}-1). *mergeSamples2* provides a strategy to allow larger merges when some of the records do not require merging, with the limit being 2^{31}-1 elements in the subset of *genon* containing *mergeIDs* with at least two observations in the input data.

Allele frequency function (*calcp*)

A function, *calcp*, for calculating allele frequencies (for all SNPs), is defined.
Usage: calcp(indsubset, pmethod="A")
Arguments:

indsubset	a vector of integers (between 1 and *nind,* inclusive) of the individuals for which are to be used for allele frequency estimation. The default is to use all individuals.
pmethod	a method for calculating the frequencies, being one of "A" (calculate on the basis of allele counts – the default method) or "G" (calculate on the basis of genotype calls)

Value: a vector of allele frequencies
Warning when using this after *mergeSamples*: pmethod A uses the object *alleles*, which is not recreated during the merge, so indsubset refers to sample positions prior to the merge. pmethod G uses *genon* whose positions are those following the merge.

Genomic relatedness function (*calcG*)

A function, *calcG*, for calculating the genomic relatedness, is defined.
Usage: calcG(snpsubset, sfx="", puse, indsubset, depth.min=0, depth.max=Inf, npc=0, calclevel=9, cocall.thresh=0, mdsplot=FALSE, mindepth.idr = 0.1, withPlotly=FALSE, plotly.group=NULL, plotly.group2=NULL, samp.info=NULL)
Arguments:

snpsubset	a vector of integers (between 1 and *nsnps,* inclusive) of the SNPs to use in the calculation. The default is to use all SNPs.
sfx	A suffix to use in output file names to identify which function call has produced that output.
puse	a vector of (reference) allele frequencies to use in the calculations. The default is to use allele frequencies calculated on the basis of allele counts. The values (for the snps in *snpsubset*) should be greater than 0 and less than 1. This is for the full set of snps (it is subsetted using *snpsubset*).
indsubset	a vector of integers (between 1 and *nind,* inclusive) of the individuals for which relatedness matrices will be calculated. The default is to calculate for all individuals.
depth.min	The minimum depth for a SNP result for an individual to be used.
depth.max	The maximum depth for a SNP result for an individual to be used.
npc	The number of principal components of the 'G5' relatedness matrix to display. If *npc*≤0, then the heatmap plot is omitted, but otherwise |*npc*| is used for *npc*. If *npc*=0 (the default) the principal component analysis is omitted.
calclevel	specifies the amount of calculation and output produced: 1 gives G5 (see below) and intermediate results only, 2 gives G5 and reports using G5, 3 gives all types of G available and 9 gives these and all reporting available.
cocall.thresh	Samples may be removed so that co-call rates (the proportion of SNPs with a call in both of a pair of samples) for heatmap and PCA analyses are above this value. Firstly, if *cocall.thresh* ≥0, samples with a maximum SNP depth of 1 are removed. The further samples are removed successively, with the sample appearing the most often in pairs not meeting the criterion removed at each step, until all pairs meet the

criterion. The removal of these samples under the default threshold allows the heatmap and PCA analyse to be performed (no NAs in the relationship matrix used).

mdsplot	if TRUE and the conditions for plotting the principal components is met, a two dimensional multidimensional scaling plot is also plotted. The default is FALSE.
mindepth.idr	minimum depth for including samples in the self-relatedness (or inbreeding) regression on log(sample depth), applied after any filtering specified in the other call parameters.
withPlotly	If TRUE, then plotly graphs are produced, else if FALSE the standard plots are produced The default is FALSE.
plotly.group	A character vector of length equal to the number of individuals (in *indsubset*). Gives grouping on the plotly graphs in terms of different coloured points. The default is NULL (no colouring).
plotly.group2	A character vector of length equal to the number of individuals (in *indsubset*).. Gives grouping on the plotly in terms of different points. The default is NULL (no grouping with symbols).
samp.info	A list where each element is a character vector of length equal to the number of individuals (in *indsubset*). Used to provide "hover" information for plotly graphs. The default is NULL, in which case the seqID is used.

Value: a list of relatedness structures: G1, G4d (diagonal elements of G4), G5, samp.removed (positions of samples removed to ensure the cocall.thresh criterion) and PC, the output of the principal components analysis (if |*npc*|>0). The G*n* relatedness matrices are described in Dodds *et al*. (2015), except that a range of allele sampling models can be incorporated for the diagonal of G5 – see the depth2K section below. Some summary information is output.

Details: The function also produces a set of output files, as detailed below. If *withPlotly* is TRUE and both the plotly and heatmaply packages are available, interactive plotly plots are produced for some of the plots.

Co-call-<sfx>.png is a histogram of co-call rates (the proportion of SNPs with a call in both of a pair of samples) for all sample pairs.

MAF<sfx>.png is a histogram of the MAFs for the subset of SNPs used (if not all SNPs).

HighRelatedness<sfx>.csv contains pairs of samples, their G5 relatedness (G5rel) and self-relatednesses (SelfRel1 and SelfRel2), where the relatedness is > *hirel.thresh* (default value of *hirel.thresh* is 0.9).

Heatmap-G5<sfx>.png is a heatmap plot using G5 relatedness. This is not produced if *npc*≤0. If *fcolo* for the relevant individuals has more than one colour, colour bars are added to the plot.

HeatmapOrder<sfx>.csv contains a list of the samples in the order they are plotted on the heatmap. rowInd is the index values (written on the heatmap plot), seqIDInd is the position of the individual in seqID.csv; seqID is also included. For "standard" cases, where all seqID samples are included, the values of rowInd and seqIDind will be the same.

Heatmap-G5<sfx>.html is a plotly version of *Heatmap-G5<sfx>.png*. *plotly.group* and *plotly.group2* are not used.

Gcompare<sfx>.png is a plot comparing relatedness estimates for G1, G3 and G5.

G<sfx>-diag.png is a plot of diagonal elements (self-relatedness estimates) of G4 against those of G5 (illustrating the effect of correcting for depth).

G<sfx>-diag.html is a plotly plot of diagonal elements of G4 against those of G5. Produced if *withPlotly* is TRUE.

G<sfx>diagdepth.png is a plot of diagonal elements of G5 against the logged sample depth. We do not expect there to be a relationship between these variables (unless planned) so this serves as a diagnostic for e.g. non-Mendelian SNPs and/or the assumption of random sampling of alleles during sequencing.

G<sfx>diagdepth.html is a plotlly version of *G<sfx>diagdepth.png*.

PC1v2G5<sfx>.png (if *npc*>0) is a plot of 2nd versus the 1st principal components. Points are plotted with open (if 100 or more samples) or closed circles. If only one component was requested, a histogram of the 1st component is produced.

PC1v2G5<sfx>.html is the plotly version of *PC1v2G5<sfx>.png.*

PCG5<sfx>.pdf (if *npc*>2) is a scatterplot matrix of the first *npc* principal components.

MDS1v2G5<sfx>.png (if *npc*>0) is a plot of 2nd versus the 1st principal components. If only one component was requested, a histogram of the 1st component is produced.

MDS1v2G5<sfx>.html is the plotly version of *MDS1v2G5<sfx>.png.*

There is a vector *fcolo* (length *nind*) of colours to be used for the individuals in these plots. It defaults to all black, but can be reset after sourcing the program (and/or running GBSsummary) and before calling *calcG*.

Output genomic relationship matrix (*writeG*)

A function, *writeG*, for saving genomic relationship matrices, is defined.
Usage: writeG (Guse, outname, outtype=0, indsubset, IDuse, metadf=NULL)
Arguments:

Guse	the G matrix of relationships to output, should be a square matrix, or a list containing an element G5 (for *outtype*s 1 to 5) and/or PC (for *outtype* 6)
outname	text used in the naming of the output file(s)
outtype	constant or vector containing the type(s) of output required. If *outtype* contains any of the following values, the corresponding output is produced: 1 an R datasets file containing the G matrix and corresponding *seqID* 2 a .csv file containing the G matrix with row and column headings 3 a .csv file containing the G matrix in "long" format, i.e. one row for every (unique) relationship pair including selfs; columns are IDs of first and second individual, followed by the relatedness value 4 a .csv file containing inbreeding for each individual; first column contains IDs, second column contains inbreeding estimates 5 two tab delimited files (.tsv) for input into the t-SNE interactive browser at http://projector.tensorflow.org/ (allows exploration of dimension-reduced data from the PCA or t-SNE methods). 6 a .csv file containing the principal components (requires *Guse* to be a list with element PC)
indsubset	a vector of integers (between 1 and *nind,* inclusive) of the individuals in the G matrix. The default assumes all individuals.
IDuse	a vector of IDs to use in the output, corresponding to the order in *Guse*, the default is to use values of *seqID* as the identifiers (in which case seqID must exist)
metadf	a data frame with the same number of rows as the G matrix, containing sample information to pass to the t-SNE browser.

Details: One or more files are written to the default directory, according to *outtype*:

<outname>.RData an R data file containing the G matrix and corresponding *IDuse* values, produced when *outtype* contains a 1. The G matrix is named based on the object specified in *Guse*, removing text up to $ and from [, if either of these are present. As an example using *writeG(Gfull$G5[1:100,1:100],outtype=1)* will result in the G matrix being named G5.

<outname>.csv a csv file containing the G matrix, produced when *outtype* contains a 2. The first column is labelled with the name of the object passed to *IDuse* and contains the values of *IDuse*. The other columns are labelled with the values of *IDuse*.

<outname>-long.csv a csv file containing the unique relatedness values, one row for every pair of individuals (including selfs), produced when *outtype* contains a 3. The columns are labelled id1, id2 (lower case to avoid warning messages when opening with Excel) and rel. *IDuse* is used for the ID values.

<outname>-Inbreeding.csv a csv file containing inbreeding values (self-relatedness minus 1), produced when *outtype* contains a 4. The first column is labelled with the name of the object passed to *IDuse* and the second column as Inbreeding. *IDuse* is used for the ID values.

<outname>-pca_vectors.tsv a tsv file containing the G matrix in a format suitable for the t-SNE browser, produced when *outtype* contains a 5.

<outname>-pca_metadata.tsv a tsv file containing sample information (from *metadf,* or *IDuse* if *metadf* is NULL) in a format suitable for the t-SNE browser, produced when *outtype* contains a 5.

<outname>-PC.csv a csv file containing principal components, produced when *outtype* contains a 6 and *Guse* is a list containing PC (*Guse* is assumed to be the output from *calcG*). The first columns are from *metadf*, if given, or the object passed to *IDuse*. Subsequent columns are the principal components, labelled PC1, PC2 etc.

Output data in variant call format (*writeVCF*)

A function, *writeVCF*, for saving data in VCF format is defined.

Usage: writeVCF(indsubset, snpsubset, outname=NULL, ep=0.001, puse = p, IDuse, usePL = FALSE, contig.meta = FALSE)

Arguments:

indsubset	a vector of integers (between 1 and *nind,* inclusive) of the individuals in to be output. The default assumes all individuals.
snpsubset	a vector of integers (between 1 and *nsnps,* inclusive) of the SNPs to output. The default is to use all SNPs.
outname	base name of the output file which will have the extension ".vcf" appended. The default is "GBSdata".
ep	the probability of a sequencing error. Default to 0.001 (changed in v0.8.2).
puse	a vector of length *nsnps* of (reference) allele frequencies to use in the calculation of posterior genotype probabilities. The default is to use *p* (normally the allele frequencies calculated based on allele counts).
IDuse	a vector of IDs of length *nind* to use in the output. The default is to use values of *seqID* as the identifiers (in which case seqID must exist)
usePL	indicator which, if set to TRUE, will result in the output containing phred-scaled likelihoods instead of genotype likelihoods. The default is FALSE.
contig.meta	indicator which, if set to TRUE, will add contig meta info to the file (ID's only). Required for input into ANGSD. The default is FALSE.

Details: A VCF format (https://samtools.github.io/hts-specs/VCFv4.3.pdf) file of the requested data is written. The file contains four fields of information relating to a genotype:

GT: the inferred genotype (0/0, 0/1, 1/1, and ./. for homozygous for reference allele, heterozygous, homozygous for alternate allele and missing, respectively)

GP: the three posterior genotype probabilities with priors calculated from the allele frequencies and assuming Hardy-Weinberg equilibrium, ordered corresponding to genotypes 0/0, 0/1, 1/1 (in GT format),
GL: (if usePL=FALSE) three $\log_{10}$-scaled likelihoods, calculated as in Li (2011), in the same order as GP,
PL: (IF usePL=TRUE) phred-scaled likelihoods (-10 * (GL – max(GL)) rounded to the nearest integer,
AD: allelic depth (read depth for reference and alternate alleles).

The "chromosome" is specified as the SNP_Name and the position is numbered sequentially from 1. The variants are all denoted as C (REF allele) and G (ALT allele). Currently there is no facility for incorporating other genomic information passed either in the input file (e.g. if Tassel format) or as additional information.

<outname>.vcf a vcf formatted file of the data specified.

Output GBS data (*writeGBS*)

A function, *writeGBS*, for saving data is defined. Currently the only supported format is the UNEAK format (see the section "GBS via UNEAK").

Usage: writeGBS(indsubset, snpsubset, outname= "HapMap.hmc.txt", outformat=gform, seqIDuse=seqID)

Arguments:

indsubset	a vector of integers (between 1 and *nind,* inclusive) of the individuals in to be output. The default assumes all individuals.
snpsubset	a vector of integers (between 1 and *nsnps,* inclusive) of the SNPs to output. The default is to use all SNPs.
outname	name of the output file. The default is "HapMap.hmc.txt".
outformat	the format of the output file. The default value is *gform* (the format of the input file). Currently only the "uneak" format will produce an output file. Any other value of *outformat* will produce a warning message.
seqIDuse	a vector of IDs of length *nind* to use in the output (and which will be read as *seqID* if the file is read back in). The default is to use values of *seqID.*

Details: A data file with the specified format is written. The function requires that the object *alleles* exists and that it corresponds to the genotype matrix (*genon*). It may be necessary to set *alleles.keep* to TRUE before data manipulation to ensure this is the case.

<outname> a file of the data specified.

Relatedness comparison function (*GCompare*)

This is a function to help make comparisons between different estimates of relatedness on the same set (or overlapping subsets) of individuals. These different estimates may come from different genotyping technologies (e.g. SNP chip vs GBS), different protocols (e.g. GBS with different restriction enzymes, different levels of multiplexing samples, different SNP callers) or using different SNP filters.

The program inputs a set of (genomic) relationship matrices (GRMs) and a corresponding set of individual IDs. The output is a set of scatterplots (possibly as scatterplot matrices) and corresponding regression output. The relatedness estimates between each pair of (different) individuals for each pair of GRMs are compared, as are those for the self-relatedness estimates for each individual. For any pair of GRM, all individuals common to the GRM are used. If there are duplicated IDs within any set a warning is printed and only the first observation for the individual is used.

Additionally, if the *MethComp* (Carstensen, 2015) R package is installed, there can be corresponding sets of plots using this package, with scatterplots of relatedness estimates below the diagonal, and 'Bland-Altman' (BA) plots (Altman and Bland, 1983) above the diagonal. The Bland-Altman plots have the differences on the vertical and the means on the horizontal access,

for the two relatedness estimates. These plots take a lot more CPU time than the regression plots.

Usage: GCompare (Glist, IDlist, Gnames = paste0("G.",1:length(Glist)), plotname = "", whichplot="both", doBA=FALSE, ...)

Arguments:

Glist	a list of G matrices
IDlist	a list of ID variables, paired to the G matrices and in the same order as the data in the corresponding G matrix
Gnames	a set of labels to use for the G matrices (defaults to G1, G2, …)
plotname	text to use in the naming of output files
whichplot	variable to choose which plot types are produced, can be one of: "diag": compare diagonals (self-relatedness) "off": compare off-diagonals (relatedness between individuals) "both": compare both diagonals and off-diagonals. This is the default.
doBA	Additionally produce Bland-Altman plots. The default is FALSE.
…	Arguments to be passed to the plotting functions (e.g. col= for coloring). These need to be relevant to the plot types being produced (e.g. if a vector of colours, then it should not be used with *whichplot*="both").

Details: One or more plots are produced, depending on the options used. A set of ignorable warnings is issued.

Gcompare- <plotname>-diag.png a plot of the diagonal comparison(s). If more than 2 G matrices, this will be a scatterplot matrix with regression results in the upper matrix panels.

Gcompare- <plotname>-offdiag.png a plot of the diagonal comparison(s). If more than 2 G matrices, this will be a scatterplot matrix with regression results in the upper matrix panels.

GcompareBA- <plotname>-diag.png a scatterplot matrix BA plot of the diagonal comparison(s). The regression plots are in the lower diagonal and the BA plots in the upper diagonal. A grey line indicates equality (y=x for lower plots, y=0 for upper plots). The BA plots have 3 additional horizontal lines being the mean & mean ± 1.96sd ('95% limits of agreement).

GcompareBA- <plotname>-offdiag.png a scatterplot matrix BA plot of the off-diagonal comparison(s). See description of the BA plot for the diagonals for more details.

Gender prediction (*genderassign*)

The function *genderassign* can be used to predict gender using the methods described in Bilton *et al*. (2019). The assignment boundaries are specified by two functions *upperbounday*(x) and *lowerboundary*(x) where x represents the proportion of heterozygotes on the homogametic sex chromosome. These functions can be modified before using *genderassign*, but only *upperboundary* can be non-linear (not checked). Their initial values are:

```
upperboundary <- function(x){ 20*pmax(rep(0,length(x)),x)^2+0.2}
lowerboundary <- function(x){ 0.1 + x}
```

Usage: genderassign (ped.df, index_Y_SNPs, index_X_SNPs, sfx="", hetgamsex = "M", homgamsex = "F", hetchrom = "Y", homchrom = "X")

Arguments:

ped.df	a dataframe of individuals for gender prediction, as if read from a pedigree file (see Input formats section). This optionally contains variables Sex (with values M, F or U for male, female, unknown) and Relationship (character, e.g. "progeny", "sire" or "dam")
index_Y_SNPs	a vector of positions of SNPs on the homogametic sex chromosome (Y chromosome for X/Y systems) to use.
index_X_SNPs	a vector of positions of SNPs on the heterogametic sex chromosome (X chromosome for X/Y systems) to use.

sfx	text to be included in output file names
plotname	text to use in the naming of output files
hetgamsex	gender label for the heterogametic sex. The default is “M” (for males, assumes X/Y system). Use “F” for the Z/W system.
homgamsex	gender label for the homogametic sex. The default is “F” (for females, assumes X/Y system). Use “M” for the Z/W system.
hetchrom	chromosome label for the heterogametic sex chromosome. The default is “Y” (assumes X/Y system). Use “W” for the Z/W system.
homchrom	chromosome label for the homogametic sex chromosome. The default is “X” (assumes X/Y system). Use “Z” for the Z/W system.

Details: Outputs a dataframe containing the input data frame (*ped.df*), the predicted gender, new_prop_X (the ratio of heterozygosity proportion of SNPs in *index_X_SNPs* compared to their expected proportions given the depth), proportion_SNPs_Y (proportion of SNPs in *index_Y_SNPs* with a result) and sampdepth (the mean sample depth for the individual). An output file and plot are also produced.

gender_prediction<sfx>.csv a .csv file containing the same information as the output dataframe.

GenderPlot<sfx>.png a plot of the results similar to Figure 1 of Bilton *et al*. (2019), where females are plotted in red, males in blue, unknowns in grey. The light blue shaded region indicates individuals predicted to be male, while the light red shaded region indicates individuals predicted to be female.

Pedigree program (GBSPedAssign.R)

This program uses a relatedness matrix and excess mismatch rate (EMM) results to verify given pedigrees and/or to find the best matching parents from groups of potential parents using the methods described in Dodds *et al.* (2019). Both these tasks require a pedigree file (with name given in *pedfile*). For parent matching a groups file (with name given in *groupsfile*) is also required. See below for the formats for these files. Father (Mother) verification is undertaken if the pedigree file contains a FatherID (MotherID) variable. Father (Mother) matching is undertaken if a groups file is given and the pedigree file contains a FatherGroup (MotherGroup) variable. The Group fields are read as text fields; if they are not present this causes a warning which can be ignored.

For parent matching, mismatch statistics are calculate for reporting and using, in addition to relatedness values, for assigning parentage. The ‘raw’ mismatch rate is the proportion of apparent (i.e. using observed genotypes) mismatches (i.e., genotypes inconsistent with parentage). ‘Excess’ rates are the differences between raw rates and rates that are expected given the genotype uncertainty due to the GBS process (manuscript in prep). A number of variables (see below) control how the mismatch rates are calculated and used. Mismatch rates are calculated for offspring-parent pairs and for offspring-parent trios (if matching to both parents). If both parents are being matched, the apparent parent-pair mismatch rates (offspring and parent genotypes incompatible) are given for each combination of the best two matching parents.

Before calling the program, the variable *GCheck* must be set to the name (as a string) of the G matrix to use. If this is for a subset of individuals, *indsubset* must be set to the indices of those individuals (as used in *calcG*). In addition, *rel.thresh* (and/or *rel.threshF* and/or *rel.threshM* for fathers and mothers, respectively) may be set to override the default relatedness value of 0.4 for declaring a parentage verification (or to allow parent assignment). A number of other variables control calculated results and reporting for parent matching. *mindepth.mm* may be set to override the default minimum depth (1) for a SNP for the individuals being compared when calculating (excess) mismatch rates for parentage matching. The default value is recommended for calculating excess rates, but raw rates are likely to be more useful when using a higher threshold. *snpsubset* may be set to indices of SNPs to be considered for use in calculating

mismatch rates and for bootstrapping (see below, this will usually be the same subset as used for calculating the G matrix being used). The excess mismatch rate thresholds for declaring parentage are set by *emm.thresh* (parent-offspring pair; default value of 0.01) and *emm.thresh2* (parent-offspring trio; default value of twice *emm.thresh*). An alternative parentage is suggested when a possible pair (mother and father) have an excess mismatch rate that is lower than that for the best (i.e., most highly related) father and best mother by more than *emmdiff.thresh2* (default value of 0).

For parent pair matching, the estimated relatedness between the parent pairs (all four combinations of best and 2nd best matching fathers and mothers) are calculated. The relatedness for the best matching pair of parents is compared with the estimated inbreeding for the individual. High values of parent relatedness (compared with the inbreeding of the individual) may indicate that one of the parents has been incorrectly assigned to a relative of the other parent. A parent-pair match will be excluded as a match if the parent relatedness exceeds offspring inbreeding by at least *inb.thresh* (default value 0.2).

A bootstrapping procedure is available to provide a metric on the closeness of parent-offspring match compared to that with the 2nd best parent. The procedure resamples SNPs (with replacement), recalculates the relatedness values (for the offspring and each of the two best parents) and reports the percentage of times that the best parent is still the better of the two among the bootstrap replicates. This should not be taken as a significance level test, as the resampled SNPS are not independent. As bootstrapping is quite time-consuming, it is invoked only when there are 2 possible parents with similar (within *boot.thresh*) parent-offspring relatedness values, and if the best parent exceeds the relatedness and excess mismatch thresholds. The number of bootstrap replicates is set by *nboot* (default value 1000). Three other variables (*depth.min, depth.max, puse)* mirror those used in calcG to allow the bootstrapping to calculate relatedness in the same way as was used for the G matrix being used in parentage assignment. These variables should be set to the same values as those used for calculating the G matrix. An assignment is flagged (see below) if the best parent is the better one in the bootstrap samples in less than *boota.thresh* percent (default value 99) of the replicates.

The output files contain variables to indicate whether the parentage should be accepted. These variables are called *FatherAssign* and *MotherAssign* for single parent matching of fathers and mothers, respectively. The codes used as values for these variables are:

Assign code	Description
N	Relatedness estimate for best matching parent is below *rel.threshF* or *rel.threshM* (for fathers and mothers, respectively).
E	Excess mismatch rate for best matching parent exceeds *emm.thresh.*
A	Alternate assignment: the 2nd best parent appears acceptable. This parent has relatedness exceeding *rel.threshF* or *rel.threshM* (for fathers and mothers, respectively) and excess mismatch rate that is lower than *emm.thresh* when the best parent had excess mismatch rate exceeding this threshold.
B	Best matching parent is the better one in less than *boota.thresh* % of the bootstrap replicates.
Y	Best matching parent passes all assignment criteria

The variable for indicating whether a parent-pair match should be accepted is *BothAssign* and takes values as shown:

Assign code	Description
N	Relatedness estimate for best matching parent is below *rel.threshF* or *rel.threshM* (for fathers and mothers, respectively).
M	Mother assigned, father not assigned.
F	Father assigned, mother not assigned.

E	Excess mismatch rate for best matching parent-pair exceeds *emm.thresh2*, except when one parent assigned and the other has an E code, then the parent assignment is made.
A	An alternate parent-pair appears acceptable. This pair has excess mismatch rate less than *emm.thresh2* and lower than that for the best parent-pair by more than *emmdiff.thresh2*. If the alternate pair also passes the other checks, the pair is indicated by the value of *Alternate*, e.g. a value of F1M2 indicates that the alternate pair is the best father and 2nd best mother.
B	At least one of the parents has a B code. (It may still be possible to assign the other parent).
I	The best parent-pair relatedness exceeds twice the offspring inbreeding by at least *inb.thresh*, and is above *minr4inb* (if that threshold has been set). An alternate pair may be indicated by the value of *Alternate*, similarly to the A code offspring.
Y	Best matching parent passes all assignment criteria

Where more than one of the assign codes is possible, the one that ranks the highest (in the order given in the above tables) is used.

This program outputs summary statistics and a number of files. The %s of verified fathers and mothers are given, as well as the mean relatedness estimates for matching and non-matching fathers and mothers. The files, where relevant, are as follows:

PedVerify.csv returns the pedigree file with additional columns, as shown below:

Variable name	Description
FatherRel	Relatedness estimate between individual and it's specified father
FatherEMM	The specified father-offspring EMM
FatherMatch	TRUE if *FatherRel* > *rel.threshF* and *FatherEMM* < *emm.thresh*
MotherRel	Relatedness estimate between individual and it's specified mother
MotherEMM	The specified mother-offspring EMM
MotherMatch	TRUE if *MotherRel* > *rel.threshM* and *MotherEMM* < *emm.thresh*
FandMEMM	The specified parent pair – offspring EMM
FandMmatch	TRUE if *FatherMatch* and *MotherMatch* are both TRUE and *FandMEMM* < *emm.thresh2*

FatherVerify.png is a scatterplot matrix showing *FatherRel, FatherEMM* (see above), the position of the individual in the pedigree file and the position of the recorded father in the pedigree file. This is useful for seeing the distribution of relatedness values, and possibly for detecting sample tracking issues (if the order in the pedigree file relates to the order samples are processed at a particular stage).

MotherVerify.png is a scatterplot matrix like FatherVerify.png but for mother verification.

FatherMatches.csv shows the results of the father matching. It returns the first two columns of the pedigree file with additional columns, as shown below:

Variable name	Description
BestFatherMatch	IndivID of the father from the *FatherGroup* having the highest estimated relatedness to the individual (or lowest EMM, if *matchmethod* is "EMM").
FatherMatch2nd	IndivID of the father from the *FatherGroup* having the 2nd highest estimated relatedness to the individual (or 2nd lowest EMM, if *matchmethod* is "EMM")

Fatherrel	The estimated relatedness for *BestFatherMatch*
Fatherrel2nd	The estimated relatedness for *FatherMatch2nd*
Father12rel	The estimated relatedness between *BestFatherMatch* and *FatherMatch2nd.*
mmrateFather	The (raw) mismatch rate for *BestFatherMatch*
mmnumFather	The number of snps used to calculate *mmrateFather*
exp.mmrateFather	The expected mismatch rate for *BestFatherMatch*
mmrateFather2	The (raw) mismatch rate for *FatherMatch2nd*
exp.mmrateFather2	The expected mismatch rate for *FatherMatch2nd*
Fathersd	The bootstrap sd of *Fatherrel* values (for bootstrapped cases, the variable is present only if there are bootstrapped caess)
FatherReliability	The % of bootstrap results where *Fatherrel* > *Fatherrel2nds* (for bootstrapped cases, the variable is present only if there are bootstrapped caess)
FatherAssign	The code for father assignment.

MotherMatches.csv shows the results of the mother matching (with columns as for FatherMatches.csv but for mothers instead of fathers).

BothMatches.csv shows the results of both father and mother matching (for individuals with both *FatherGroup* and *MotherGroup*). It contains the columns of FatherMatches.csv and MotherMatches.csv with additional columns, as shown below:

Variable name	Description
mmrateF<fatherrank>M<motherrank>	The (raw) mismatch rate for possible parent matches, where <*fatherrank*> is 1 to indicate *BestFatherMatch* and 2 to indicate *FatherMatch2nd*, and similarly for <*motherrank*>*.*
mmnumF<fatherrank>M<motherrank>	The number of SNPs used to calculate *mmrateF<fatherrank>M<motherrank>*
exp.mmrateF<fatherrank>M<motherrank>	The expected mismatch rate corresponding to *mmrateF<fatherrank>M<motherrank>*
relF<fatherrank>M<motherrank>	The estimated relatedness between the pair of possible parents
Inb	The estimated inbreeding of the offspring
BothAssign	The code for the parent-pair assignment
Alternate	An alternative (to F1M1) parent pair

GroupsParentCounts.csv returns the groups file with additional columns, as shown below:

Variable name	Description
FatherFreq	Number of offspring where this father is the *BestFatherMatch* in this group
MotherFreq	Number of offspring where this mother is the *BestMotherMatch* in this group

BestFatherMatches.png is a plot of the raw mismatch rate for *BestFatherMatch* against the estimated relatedness (*Fatherrel*). Points are coloured using *fcolo* and a grey vertical line indicates the value of *rel.thresh* used.

BestFatherMatchesE.png is the same BestFatherMatches.png except that the excess mismatch rate is plotted. A grey horizontal line indicates the value of *emm.thresh* used.

Best2FatherMatches.png is a plot of the estimated relatedness for *FatherMatch2nd* (*Fatherrel2nd*) against that for *BestFatherMatch* (*Fatherrel*). Points are coloured using a scale based on the excess mismatch rate (*mmrateFather- exp.mmrateFather*) for father-offspring and the line of equality is drawn (by definition all points fall below the line). Vertical and horizontal

grey lines indicate the value of *rel.threshF* or *rel.threshM* (for fathers and mothers, respectively) used.

ExpMM-Father.png is a plot of the raw mismatch rate against the expected mismatch rate for *BestFatherMatch*. A red line shows where these are equal and a grey line shows the boundary for an E assign code. Points are coloured using *fcolo* and the symbols indicate *FatherAssign*.

BestMotherMatches.png, *BestMotherMatchesE.png*, *Best2MotherMatches.png* and *ExpMM-Mother.png* are the corresponding plots to BestFatherMatches.png, BestFatherMatchesE.png and Best2FatherMatches.png and ExpMM-Father.png, respectively, for mothers.

ParRel-Inb.png is a plot of estimated parent-pair relatedness against offspring estimated inbreeding. Points are coloured according to the mean depth in the offspring (as depth is more critical for inbreeding than relatedness estimation), and with a symbol corresponding to *BothAssign* (see ExpMM-Both.png for a key).

MMrateBoth.png is a scatterplot matrix plot of the four combinations of parent-pair raw mismatch rates that were saved in BothMatches.csv. Points are coloured using *fcolo* and the lines of equality are drawn (in red).

MMrateBothE.png is a scatterplot matrix plot of the four combinations of parent-pair excess mismatch rates. Points are coloured using *fcolo*, the lines of equality are drawn (in red), a grey line shows the boundary for an E assign code and the symbols for the points denote *BothAssign*. The key for the symbols can be found in ExpMM-Both.png.

ExpMM-BothE.png is a plot of raw versus expected parent-pair mismatch rates. Points are coloured using *fcolo* and the symbols for the points denote *BothAssign*.

EMM sum of squared for beta-binomial model (*ssbbmm*)

A function to use for determining the fit of a beta-binomial model in terms of trio matches. Should only be run after a trio parentage assignment. *Under development and likely to change in future updates.*

Usage: ssbbmm(bbpar, ,uuse=uY)
Arguments:

bbpar	the parameter for the beta-binomial model
uuse	the offpring set to use (positions in BothMatches) as true trios

Details: Returns the sum of squared trio EMM values for the uuse offspring and their assigned parents. The function can be used to estimate the beta-binomial parameter, e.g. with optimize(ssbbmm,lower=0,upper=20, tol=0.001)

EMM sum of squared for modified p model (*ssmpmm*)

A function to use for determining the fit of a modified p model in terms of trio matches. Should only be run after a trio parentage assignment. *Under development and likely to change in future updates.*

Usage: ssmpmm(mppar, ,uuse=uY)
Arguments:

mppar	the parameter for the modified p model
uuse	the offpring set to use (positions in BothMatches) as true trios

Details: Returns the sum of squared trio EMM values for the uuse offspring and their assigned parents. The function can be used to estimate the modified p parameter, e.g. with optimize(ssmpmm, lower=0.5,upper=0.8, tol=0.001)

Add tagID function (*addtagIDs*)

This is a function will add alternative IDs for offspring, father and mother.

Usage: addtagIDs(sampinfo, indvar, matchtype ="both")
Arguments:

sampinfo	a dataframe containing the relevant IDs for all individuals
indvar	quoted text giving the name of the variable in sampinfo that corresponds to IndivID in the pedigree file
tagvar	quoted text giving the name of the variable in sampinfo that contains the tag (identifier) to be returned
matchtype	one of "both", "father" or "mother" (not case-sensitive) specifying which data frame to update

Details: Returns a data frame with the additional IDs added. It is not output to a file.

Parentage PC plot function (*bestparPCA*)

This is a function generates a PC2 vs PC1 plot with lines joining each progeny with its best father and mother.

Usage: bestparPCA(Gobj, sfx="", keypos=NULL)
Arguments:

Gobj	an object produced from calcG (usually the same one used to obtain the GCheck matrix for parentage), with npc ≥ 2
sfx	text to be included in output file name to allow output from multiple calls or runs to be identified
keypos	the location (if given) on the plot of the legend of assign codes, using a value as accepted by the legend command (e.g. "topleft")

Details: Generates a PC plot.

PC-BestParents<sfx>.png a plot of PC2 vcs PC1 relating to the parentage analysis. Points are coloured according to fcolo. Parents are shown as dots and offspring have symbols representing their assignment codes. Blue and pink lines are drawn from best father and best mother, respectively, to offspring.

Population genetics analysis (GBS-PopGen.R)

This R code makes available some functions for population genetics analyses. These are currently under development and only a brief description is provided here. The methods and syntax of this code is likely to change in the future.

Heterozygosity measures (*heterozygosity*)

This function gives various measures of observed and expected heterozygosity. keep.alleles should be set to TRUE to use this function.

Usage: heterozygosity(indsubsetgf=1:nind,snpsubsetgf=1:nsnps,maxiter=100,convtol=0.001)
Arguments:

indsubsetgf	a vector of integers (between 1 and *nind,* inclusive) of individuals to use for the heterozygosity measures. The default is to use all individuals.
snpsubsetgf	a vector of integers (between 1 and *nsnps,* inclusive) of SNPs to use for the heterozygosity measures. The default is to use all SNPs.
maxiter	maximum number of iterations to use in the estimation process. The default is 100.
convtol	convergence tolerance – the difference between genotype frequency estimates in successive iterations that is sufficiently small to assume convergence. The default value is 0.001.

Details: A data frame is returned with a (unlabelled) row for each SNP and the columns:

Variable name	Description

ohetstar	Observed heterozygosity on the raw scale
ehetstar	Observed heterozygosity on the raw scale (the proportion of genotype results expected to contain reads from both alleles)
ohet	Observed heterozygosity (estimated) on the true genotype scale
ohet2	An alternative measure of *ohet*
ehet	Expected heterozygosity (estimated) on the true genotype scale

F_{ST} calculations (*Fst.GBS* and *Fst.GBS.pairwise*)

These functions calculate approximate F_{ST} (estimates) accounting for read depth.

Usage: Fst.GBS(snpsubset, indsubset, populations, varadj=0) and/or Fst.GBS.pairwise (snpsubset, indsubset, populations,sortlevels=TRUE, ...)

Arguments:

snpsubset	a vector of integers (between 1 and *nsnps,* inclusive) of SNPs calculate F_{ST} for. The default is to use all SNPs.
indsubset	a vector of integers (between 1 and *nind,* inclusive) of individuals to use for the calculations. The default is to use all individuals.
populations	a vector of length *nind* containing population labels
varadj	use varadj=1 to get Fst as Weir p166, varadj=0 for usual Fst.
sortlevels	determines whether populations are listed as encountered in *populations* (sortlevels=FALSE) or sorted (sortlevels=TRUE)
…	arguments passed to Fst.GBS (currently only varadj)

Details: F_{ST} with depth adjustment. The adjustment is "approximate" and may result in estimates outside [0,1]. The .pairwise version calculates the statistics for each pair of populations. The (pairwise) means and medians are displayed. The values for each SNP are returned in a vector (Fst.GBS) or three-dimensional array (Fst.GBS.pairwise) with first two dimensions being the population and the third dimension being the SNP.

MAF plots by population (*popmaf*)

Plot minor allele frequency distributions by population.

Usage: popmaf(snpsubset, indsubset, populations=NULL, subpopulations=NULL, indcol, colobj, minsamps=10, mafmin=0, sortlevels=TRUE, unif=FALSE)

Arguments:

snpsubset	a vector of integers (between 1 and *nsnps,* inclusive) of SNPs calculate F_{ST} for. The default is to use all SNPs.
indsubset	a vector of integers (between 1 and *nind,* inclusive) of individuals to use for the calculations. The default is to use all individuals.
populations	a vector of length *nind* containing population labels. The default is NULL in which case a MAF plot for all indsubset individuals is given.
subpopulations	a vector of length *nind* containing subpopulation labels. These are treated as being nested within *populations*.
indcol	A colour assignment for each individual. The plot is coloured by *indcol* if all individuals in that population have that colour (otherwise black).
colobj	an object created by *colorby* (undocumented). If present it will be used for populations and their colours.
minsamps	Minimum number of samples in a (sub)population to invoke plotting. The default is 10.
mafmin	Minimum (sub)population MAF to include a SNP in the plot. The default is 0.
sortlevels	determines whether data are processed by populations as encountered (sortlevels=FALSE) or sorted (sortlevels=TRUE)
unif	determines whether the plots are drawn with the same vertical axis range. The default is FALSE (population-specific range).

Details: A MAF distribution is plotted, possibly with a different colour for each population, and different shading for each subpopulation. Some summary statistics are also displayed.

Manhattan plots (*manhatplot*)

Simple plotting of results as a Manhattan plot.

Usage: manhatplot(value, chrom, pos, plotname, qdistn=qunif, ...)

value	vector of statistic values to be plotted (for each SNP).
chrom	chromosome name (numeric or character) for each SNP.
pos	position (numeric) on chromosome for each SNP.
plotname	text used as prefix for names of output plots
keyrot	rotation angle for chromosome key. The default is 0, but 90 is a better choice for longer chromosome labels.
qdistn	function name of null distribution. The default is the uniform distribution (qunif) which could be used e.g., with p-values. A more common example would be the chi-squared distribution (qchisq).
…	further arguments passed to qdistn, for example the df (degreeso of freedom) parameter (if using qdistn=qchisq).

Details: Two plots are produced.

<plotname>-Manhat.png a Manhattan plot of *value.* Points are sorted by *chrom* and *pos* with a different colour for each value of *chrom*.

<plotname>-QQ.png a QQ plot *value* using the distribution specified in *qdistn*.

Input formats

The genotype input format is set with *gform*, one of “uneak” (the default), “Tassel”, “TagDigger” or “Chip”.

GBS via UNEAK

The default input format (‘uneak’) is a ‘hapmap count’ formatted file as produced by the UNEAK pipeline (Lu *et al*. 2013). This is a tab-separated flat text file with the first column being the SNP identifier, then a column for each genotyped individual (or sample, or other genotyping unit), followed by 5 columns of summary information (HetCount_allele1, HetCount_allele2, Count_allele1, Count_allele2, Frequency). Only the last of these 5 is used. Each row is for a different SNP. The column for each individual contains the genotype information as the allele depth (number of reads of that allele) for the ‘reference’ and ‘alternate’ alleles, respectively. The designation of reference and alternate is arbitrary for this software. The numbers of reads are separated by a pipe symbol (“|”). There is a header line, which, for the genotype columns, is taken as the identifiers of the individuals.

GBS via Tassel

An additional format (‘Tassel’) is available that may be easier to use for GBS data that has been manipulated in Tassel. It is similar to the uneak format, but allele depths in a genotype are separated by a comma (“,”), has two columns before genotype data (, and no columns following the genotype data. The first two columns are the chromosome and position (which together, separated by an underscore, serve as the SNP identifier), respectively. As with the “uneak” format, this is a tab- separated flat text file with a header row.

GBS via TagDigger

TagDigger (https://github.com/lvclark/tagdigger, Clarke and Sacks, 2016) is a tool for SNP calling from a given set of tags (sequences). It is likely to be used in a production environment, where

the set of SNPs being called is unlikely to change much with additional samples being added. The 'TagDigger' format requires a comma delimited file with sample results in rows and SNP results in pairs of columns (count of reference allele, count of alternate allele). The first column contains the sample identifier. The header row, apart from the first value, contains SNP/allele identifiers. It is assumed that these identifiers have a SNP identifier followed by an underscore, followed by the allele identifier. The text preceding the underscore is taken as the SNP name (the other text is ignored).

TagDigger files will be read with the fread function from the data.table package, if that package is installed. This is faster than the method used when the package is not available. Files compressed with the gzip (.gz) format can be read by both methods on linux platforms, but not for fread on other platforms.

GBS via ANGSD

ANGSD (http://www.popgen.dk/angsd, Korneliussen *et al*., 2014) is a program for analysing sequencing data, and can output SNP information. The 'ANGSDcount' format reads files created by the –dumpCounts 4 option of ANGSD. This file has a header row, followed by a row for each SNP. There is a column for each of the 4 possible alleles (A, C, G, T) for each SNP and sample. The columns for a sample are together. The header contains an identifier for each column consisting of the sample identifier followed by underscore and the allele (e.g. ind0_A). After reading this file, SNPs are checked for which alleles are most common. The two most common alleles are taken as the variant of interest, and other alleles are ignored, except that a SNP is discarded if the proportion of reads for the third most common alleles exceeds the threshold triallelic.thresh. SNPs are named as 'SNP' followed by the zero-padded position. SNPs that have been dropped by the triallelic threshold can be identified by finding gaps in the SNP_Name sequence.

vcf files

A python helper script vcf2ra_ro_ao.py is available to convert .vcf files to the 'Tassel' format. The .vcf file must have either the AD (allelic depth) field, or both the AO (alternate allele observation count) and RO (reference allele observation count) fields.

Chip

Fully recorded genotypes can be entered via the "Chip" format. This comma-separated format has results for each individual in the rows and SNP results in a column. There is a header row (SNP identifiers) and the first column contains individual identifiers. Subsequent columns contain the SNP results. Genotype data is given in 0/1/2 format, representing first homozygote, heterozygote and second homozygote, respectively. Designation of which allele is the 'first' is arbitrary.

Pedigree file

An optional pedigree file can be given, and will be used to verify or find parent matches. This is a comma separated file (csv). All individuals to be considered as offspring or parents need to have a row in this file. The columns of this file are specified below. The names must be exactly as specified. Additional columns may be present in the file.

Variable name	Required?	Description
IndivID	Y	identifies individuals in the pedigree and groups files
seqID	Y	matches *IndivID* to the identifier in the genotype file
FatherID	N	Recorded *IndivID* of father
MotherID	N	Recorded *IndivID of mother*
FatherGroup	N	Group label for group of potential fathers for the given IndivID

MotherGroup	N	Group label for group of potential mothers for the given IndivID

Father and mother group labels should be distinct. If required, they are entered for the progeny. The information linking these labels to the set of possible parents is placed in the groups file.

Groups file

If parent matching is required, then a groups file describing the group labels in the pedigree file is required. This is a comma separated file (csv). The columns (both required) of this file are specified below. The names must be exactly as specified. Additional columns may be present in the file.

Variable name	Description
IndivID	identifier for potential parent, matching *IndivID* in the pedigree file
ParGroup	Group label for the group that *IndivID* belongs to

There should be one row for each group a potential parent belongs to.

Optional packages

The software has been designed to run without the need for any R packages to be installed, but can use such packages if available. Sometimes there will be messages relating to these packages, but these messages can be ignored. A list of optional packages and their use follows.

Package name	Usage
Rcpp	Various functions have C++ versions for improved efficiency, but require this package
RcppArmadillo	This package is required for some of the C++ functions to be used
data.table	Reading tagdigger files, writing VCF files
plotly	Interactive graphics output from calcG
heatmaply	Interactive heatmap from calcG
parallelDist	Parallelized calculation of distance for the heatmap in calcG
MethComp	Bland-Altman plots in GCompare

Example

This folder contains an example run (possibly using an earlier version). Files in directory :
GBSRun.R HapMap.hmc.txt.gz Ped-GBS.csv Ped-Groups.csv

GBSRun.R

```
genofile <- "HapMap.hmc.txt.gz"

source("<source directory>/GBS-Chip-Gmatrix.R")
Gfull <- calcG()
GHWdgm.05  <-  calcG(which(HWdis  >  -0.05),"HWdgm.05",  npc=4)     #
recalculate using Hardy-Weinberg disequilibrium cut-off at -0.05

pedfile <- "Ped-GBS.csv"
groupsfile <- "Ped-Groups.csv"

rel.thresh <- 0.2
GCheck <- "GHWdgm.05$G5"
```

```
source("<source directory>/GBSPedAssign.R")
```

<source directory> should be replaced with the location of the relevant .R files before running.
linux command:
R CMD BATCH --no-save GBSRun.R &

Files in directory after running code:

```
AlleleFreq.png            GcompareHWdgm.05.png     MotherVerify.png
Best2FatherMatches.png    Gcompare.png             PC1v2G5HWdgm.05.png
Best2MotherMatches.png    Gdiagdepth.png           PCG5HWdgm.05.pdf
BestFatherMatchesE.png    G-diag.png               Ped-GBS.csv
BestFatherMatches.png     GHWdgm.05diagdepth.png   Ped-Groups.csv
BestMotherMatchesE.png    GHWdgm.05-diag.png       PedVerify.csv
BestMotherMatches.png     GroupsParentCounts.csv   SampDepthCR.png
CallRate.png              HapMap.hmc.txt.gz        SampDepthHist.png
Co-call-HWdgm.05.png      Heatmap-G5HWdgm.05.png   SampDepth.png
Co-call-.png              Heatmap-G5.png           SampDepth-scored.png
ExpMM-Father.png          HighRelatedness.csv      SampleStats.csv
ExpMM-Mother.png          HWdisMAFsig.png          seqID.csv
FatherMatches.csv         LRT-hist.png             SNPCallRate.png
FatherVerify.png          LRT-QQ.png               SNPDepthHist.png
finplot.png               MAFHWdgm.05.png          SNPDepth.png
GBSRun.R                  MAF.png
GBSRun.Rout               MotherMatches.csv
```

A workshop using this example was given at the 2015 MapNet meeting. Instructions and course notes are available.

ParExample

The folder gives an example of the code for a parentage analysis, based on the example given in Dodds *et al*. (2019). Example code is given in GBSParDeer.R. The example code assumes all necessary files are in the working directory. The example data (allele counts, pedigree file, groups file) can be obtained from https://gsajournals.figshare.com/s/7ca45accf6ae82047c86. An annotated description of the command s in GBSParDeer.R is in GBSParentage-Annotated.pdf.

Acknowledgement

This project was supported by the Ministry of Business, Innovation and Employment via its funding of the “Genomics for Production & Security in a Biological Economy” programme (Contract ID C10X1306).

Rudi Brauning, Rachael Ashby, Timothy Bilton, Alice Chappell and David Winter have all contributed to the code development.

References

Altman, D G and Bland, J M (1983) Measurement in medicine: the analysis of method comparison studies. *The Statistician* **32**, 307-337.

Bilton, T P, Chappell, A J, Clarke, S M, Brauning, R, Dodds, K G, McEwan, J C and Rowe, S J (2019) Using genotyping-by-sequencing to predict gender in animals. *Animal Genetics* **50**, 307-310. doi:10.1111/age.12782

Carstensen, B, Gurrin, L, Ekstrom, C and Figurski, M (2015). MethComp: Functions for Analysis of Agreement in Method Comparison Studies. R package version 1.22.2. http://CRAN.R-project.org/package=MethComp

Clark, L V and Sacks, E J (2016) TagDigger: user-friendly extraction of read counts from GBS and RAD-seq data. *Source Code for Biology and Medicine* **11**, 1-6. doi:10.1186/s13029-016-0057-7

Dodds, K G, McEwan, J C, Brauning, R, Anderson, R A, Van Stijn, T C, Kristjánsson, T and Clarke, S M (2015) Construction of relatedness matrices using genotyping-by-sequencing data. *BMC Genomics* **16**, 1047.

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. *bioRxiv* 582585. doi:10.1101/582585

Li, H (2011) A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* **27**, 2987-2993.

Lu, F, Lipka, A E, Glaubitz, J, Elshire, R, Cherney, J H, Casler, M D, Buckler, E S and Costich, D E (2013) Switchgrass Genomic Diversity, Ploidy, and Evolution: Novel Insights from a Network-Based SNP Discovery Protocol. *PLoS Genetics* **9**, e1003215.

Korneliussen, T S, Albrechtsen, A and Nielsen, R (2014) ANGSD: Analysis of Next Generation Sequencing Data. *BMC Bioinformatics* **15**, 356. doi:10.1186/s12859-014-0356-4