Software for GBS-based relationship calculations

Author: Ken Dodds
Address: Invermay Agricultural Centre, Puddle Alley, Private Bag 50034, Mosgiel 9053, New Zealand
Email: ken.dodds at agresearch.co.nz
Date: 11 March 2017

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). Unless specified, relatedness estimates in this documentation refer to those using the 'G5' method of that paper.

The code is still undergoing development.

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.

Calling program (GBSRun.R)

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" 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.
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.
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. Default is 9 (all available output)
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.
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. This has a default value of 0.4.
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 estimated inbreeding to exclude a parent-pair match with the inbreeding check. This has a default value of 0.2.
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.
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.

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 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.

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
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

Function to read TagDigger format files (*readTD*)

This function is for reading TagDigger files. It is used by the main program, but can be used to read additional files (e.g. to compare results in two different files). The variables *nsnps, seqID, nind,* and *alleles* are defined. The ability to generate the other variables required for further analysis is planned. See the section on the TagDigger format for more information.

Arguments:

genofilefn — the name of the file to read. Defaults to *genofile*.

Value: NULL

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: *depth2Kbb* (depthvals, alph=Inf)
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 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)
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

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). Only values of mergeID with nresults >1 are included.
Details: The function displays pairs of results where the estimated relatedness is ≤ *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 waws returned by the function
SelfRel<sfx>.png	a plot of *meanrel* against *selfrel*. The line of identity is shown in red.

Function for merging results for the same individual (*mergeSamples*)

A function, *mergeSamples*, for merging samples from the same individual.
Usage *mergeSamples* (mergeIDs)
Arguments:

mergeIDs	a vector of identifiers, ordered as in *Guse*, such that samples that have the same identifier are to be merged

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
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*) and that the diagnostics produced when sourcing GBS-Chip-Gmatrix.R are not re-done by this function.

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)
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 *indsubset*) should be greater than 0 and less than 1. This is for the full set of snps (it is subsetted using *indsubset*).
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).

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.

Details: The function also produces a set of output files, as follows.

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 and their G5 relatedness, where this 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.

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>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.

PC1v2G5<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.

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

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 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)

Arguments:

Guse	the G matrix of relationships to output, should be a square matrix
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
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 n the output, corresponding to the order in *Guse*, the default is to use values of *seqID* as the identifiers

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 *seqID* 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 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.

Pedigree program (GBSPedAssign.R)

This program uses a relatedness matrix to verify given pedigrees and/or to find the best matching parents from groups of potential parents. 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.

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* 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. 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. 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.thresh.*
E	Excess mismatch rate for best matching parent exceeds *emm.thresh.*
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.thresh.*
M	Mother assigned, father not assigned.
F	Father assigned, mother not assigned.
E	Excess mismatch rate for best matching parent-pair exceeds *emm.thresh2.*
A	An alternate parent-pair appears acceptable. This pair has excess mismatch rate that is lower than that for the best parent-pair by more than *emmdiff.thresh2*. The alternate 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.
I	The best parent-pair relatedness exceeds the offspring inbreeding by at least *inb.thresh.*
Y	Best matching parent passes all assignment criteria

Where 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
FatherMatch	TRUE if *FatherRel* > *rel.thresh*
MotherRel	Relatedness estimate between individual and it's specified mother
MotherMatch	TRUE if *MotherRel* > *rel.thresh*

FatherVerify.png is a scatterplot matrix showing *FatherRel* (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
FatherMatch2nd	IndivID of the father from the *FatherGroup* having the 2nd highest estimated relatedness to the individual
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.thresh* 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. 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) 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*.

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 serve as the SNP identifier), respectively. As with the “uneak” format, this is a tab- separated flat text file with a header row.

A python helper script vcf2ra_ro_ao.py is available to convert .vcf files (containing either AD – allelic depth – or AO and RO fields) to the ‘Tassel’ format.

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 preceeding 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.

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.

Example

The example given is based on an earlier version of the code. 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.

References

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.

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.

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.