PBAP Version 1 User Manual

Alejandro Q. Nato, Jr.[1], Nicola H. Chapman[1], Harkirat K. Sohi[1], Hiep D. Nguyen[1], Zoran Brkanac[2], and Ellen M. Wijsman[1,3,4,*]

[1]Division of Medical Genetics, Department of Medicine, University of Washington, Seattle, WA 98195, USA
[2]Department of Psychiatry and Behavioral Sciences, University of Washington, Seattle, WA 98195, USA
[3]Department of Biostatistics, University of Washington, Seattle, WA 98195, USA
[4]Department of Genome Sciences, University of Washington, Seattle, WA 98195, USA

Last Modified: 8/7/2015

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Pedigree-based analysis pipeline (PBAP) suite of programs is a unified comprehensive system geared towards SNPs and sequence data. It is composed of Perl scripts (developed under the Linux environment) that:

1) implement basic quality control (QC) checks on genetic data
2) carefully select markers for linkage analysis
3) set up files for MORGAN (Thompson, 2011)
4) access software (*i.e.*, gl_auto) to sample inheritance vectors (IVs), and
5) format files for family-based analyses.

PBAP is not intended to replace existing software that perform file manipulations or QC checks, *e.g.*, Mega2 (Mukhopadhyay *et al.*, 2005), PLINK (Purcell *et al.*, 2007). In fact, we developed this pipeline for datasets that should have already undergone some level of QC.

1.1. Citation

Nato AQ, Chapman NH, Sohi HK, Nguyen HD, Brkanac Z, and Wijsman EM. (2015) PBAP: A pipeline for file processing and quality control of pedigree data with dense genetic markers. *Bioinformatics*. First published online July 30, 2015. doi:10.1093/bioinformatics/btv444.

1.2. Software URL

http://faculty.washington.edu/wijsman/software.shtml

1.3. Definition of terms and acronyms

Complex pedigrees
Pedigrees with cross-generational marriages, inbreeding, or marriage loops

Exclusion marker
A marker that the user wants to be absent in the marker subpanel.

Founder genome labels or founder gene labels (FGLs)
Unique identifiers assigned to each of the two haploid genomes of each founder with the assumption that the founders of a pedigree are unrelated

ID translation file
A file containing original family IDs and individual IDs as well as new family IDs and individual IDs

Identity by descent (IBD)
A matching segment of DNA shared by two or more individuals that was inherited from a common ancestor without recombination.

Inclusion marker
A marker that the user wants to be present in the first marker subpanel. There are two types of inclusion markers in PBAP: core and auxiliary (2.4.2.2).

Inheritance vectors (IVs)
Labels that specify the flow of founder alleles in a pedigree. They represent the flow of chromosomes through pedigrees.

Marker subpanel
A subset of markers selected from an original marker panel which contains many markers.

Markov Chain
A random process whose future only depends on the current state of such process and not the past, which means that it is memoryless.

Markov chain Monte Carlo (MCMC) methods
Algorithms used to sample from a probability distribution based on constructing a Markov chain with an equilibrium distribution that is based on a distribution of interest.

Meiosis indicators (MIs)
Labels used to track the descent of genes through the pedigree at the marker and trait loci

Minor allele frequency (MAF)
Frequency of the least common allele in a given population

Monte Carlo methods
Computational algorithms or sets of instructions that randomly sample from a specific process.

Normal file format
A format in which each row of the genotype file represents one individual.

Outdir
A shorter term used by PBAP for output directory

Short tandem repeats (STRs) or microsatellites
Short sequences of DNA (usually two to thirteen base pairs), that are repeated numerous times in a head-tail manner.

Single nucleotide polymorphisms (SNPs)
DNA sequence variations that occur commonly within a population in which a single nucleotide on a chromosome differs between members of a particular species

Transposed file format
A format in which each row of the genotype file represents one marker

1.4. Schematic diagram of PBAP

A schematic diagram of the pipeline is shown in Figure 1.

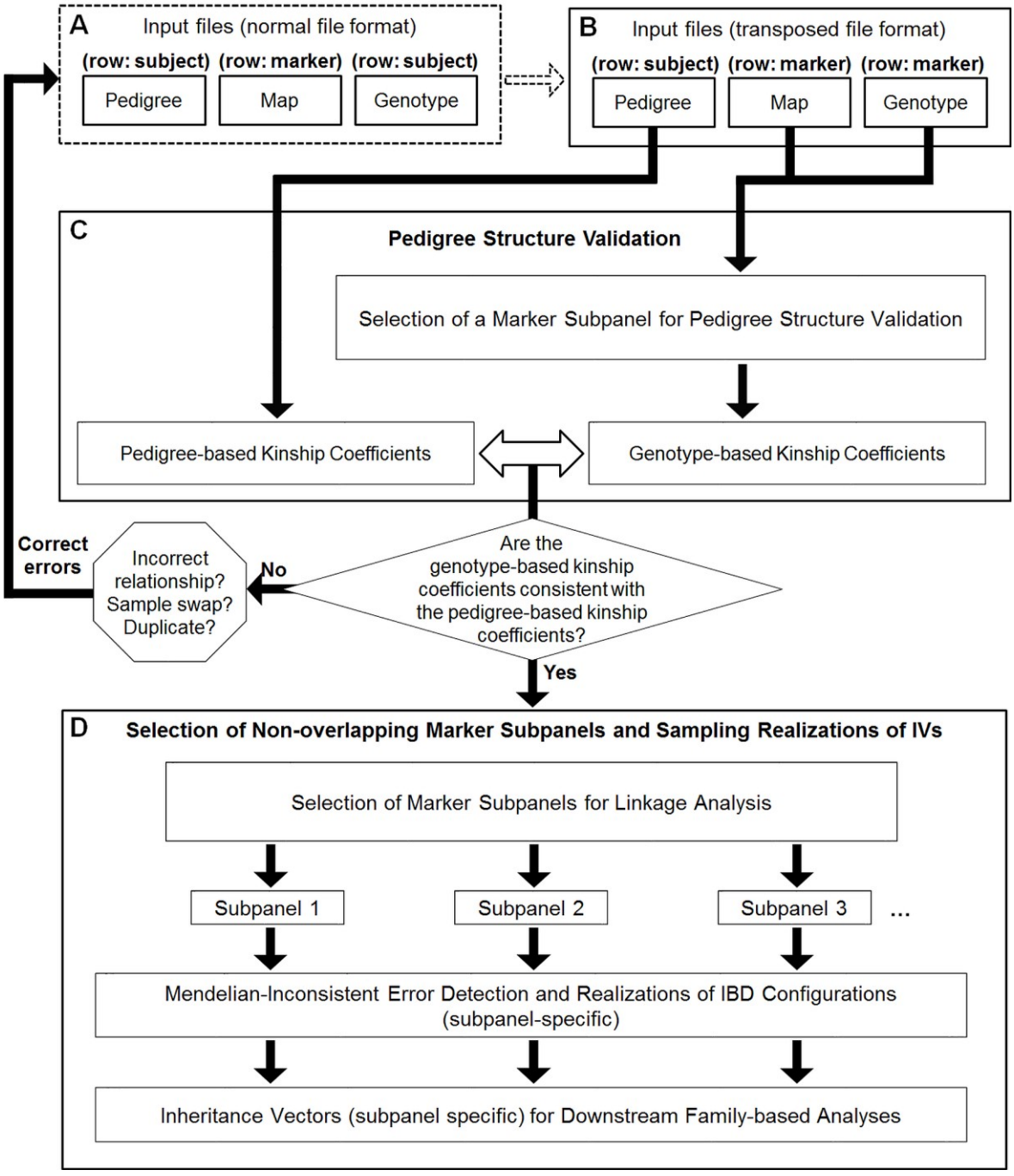

Figure 1. Schematic diagram of PBAP. (A-B) Input files that are in the normal file format are first converted to the transposed file format. (C) Relationship or pedigree errors are identified. (D) One or more non-overlapping marker subpanels suitable for linkage analysis may be selected from the dense panels. The IVs for each marker subpanel are sampled by gl_auto of the MORGAN (Thompson, 2011) package. The applications used in each of these steps are described in 2.4.

1.5. What's Next?

We are currently working on applications that will access GIGI (Cheung *et al.*, 2013) and gl_lods of the MORGAN package (Koepke and Thompson, 2013; Thompson, 2011).

We will also add new command-line options for run_gl_auto.pl (2.4.5) to allow more flexibility. The output directory of run_gl_auto.pl is currently the same as the output directory of setup_gl_auto.pl (2.4.4). In one of the upcoming updates, we will allow users to specify a different output directory at the run_gl_auto.pl step. Aside from this option, there are others that we will add to allow the user to change the parameter file of gl_auto (chr*.glauto.par) at the run_gl_auto.pl step.

A quick internal check that determines the dimensions of both input and main output files will also be included in future versions of our scripts. This check is currently present in Application 1 (2.4.1).

2. RUNNING PBAP

2.1. Installing PBAP

Download PBAP from the link below into the directory of your choice (*e.g.*, /home/username) and navigate to that directory.

https://faculty.washington.edu/wijsman/progdists/pbap/pbap_v1.00.tar.gz

Unzip the files by typing

tar –zxvf pbap_v1.00.tar.gz

Navigate to the PBAP directory. You should see eight Perl scripts, a text file, and four folders.

2.2. Files in the PBAP Suite of Programs

PBAP is composed of the following Perl scripts:

1) transpose_fileset.pl
2) marker_subpanels.pl
3) generation_numbers.pl
4) kped.pl
5) kstat.pl
6) relationship_check.pl
7) setup_gl_auto.pl
8) run_gl_auto.pl

Each of these scripts has a screen menu that may be viewed by executing the script without any arguments in the command line. For example, if you want to view the screen menu of transpose_fileset.pl, navigate to your PBAP directory and simply type

./transpose_fileset.pl

or you may also prefer to include the complete path, *e.g.*,

/home/username/pbap_v1.00/transpose_fileset.pl

Screen menus of these Perl scripts are in the text file "README_screen_menus." **We recommend that you print this text file in landscape format before proceeding since it will be a useful guide for each of the PBAP scripts.**

In addition, there are two folders, namely: essentials and tools. The essentials folder contains kstat (Choi *et al.*, 2009) and kinship.pl, which are described in 2.4.3.2.3, and five files (curve_fit.*) described in 2.4.3.3. The tools folder will be used for ancillary Perl scripts – it currently has one script, exclude_indels_dups.pl, which we used in preparing the reference genotype files described in 2.3.2.2. The par_templates folder contains template parameter files. We are also in the process

of creating sample files that you may use to familiarize yourself with PBAP and these will be placed in another folder, *i.e.*, example folder.

For the reference data files needed for PBAP, see 2.3.2.

2.3. Data Input Files

2.3.1. Project Data Input Files

You may use either normal or transposed file format as input for this pipeline. Differences between these two formats are discussed below. The input files may be either space- or tab-delimited. To avoid duplicate individual IDs in your input files, concatenate the family IDs with the individual IDs by a dot, hyphen, or underscore (*i.e.*, family ID_individual ID, family ID_father ID, and family ID_mother ID except for founders whose father and mother IDs are both zeroes). In all the examples in this manual, we will use an underscore to concatenate the family IDs with the individual IDs. For each of the scripts of this pipeline, LOG files are created to accompany the output files. Always read these LOG files whenever you examine your results. For each of these files, the required columns and their corresponding order are specified within a parenthesis.

For the **normal file format** (row-individual format), input files are as follows:

1) pedigree file (family ID, individual ID, father ID, mother ID, sex)
2) genotype (family ID_individual ID, genotypes) – one file per chromosome
3) map (chromosome, marker, genetic location, physical position) – one file per chromosome
4) phenotype (family ID_individual ID, phenotype(s), (covariate(s))) – optional

We will refer to files 1, 2, 3, and 4 above as PED, GENO, MAP, and PHENO files, respectively.

For the **transposed file format** (row-marker format), you should have the following input files:

1) pedigree (family ID, individual ID, father ID, mother ID, sex)
2) genotype (chromosome, marker, genotypes) – one file per chromosome
3) map (chromosome, marker, genetic location, physical position) – one file per chromosome
4) IDs of genotyped individuals (family ID_individual ID) – order is important
5) phenotype (family ID_individual ID, phenotype(s), (covariate(s))) – optional

Suggested extensions of filenames for the transposed file format are *.tpedo, *.tgen, *.tmap, *.tind, and *.tphen, for 1, 2, 3, 4, and 5 above, respectively. We will refer to files 1, 2, 3, 4, and 5 above as TPEDO, TGEN, TMAP, TIND, and TPHEN files, respectively.

The main differences between these two file formats above are the following:

1) For the normal file format, each row of the genotype file represents one individual while the columns represent the genotype data for each of the markers. In contrast, for the PBAP

transposed file format, each row of the genotype file represents one marker while the columns represent the genotype data for each of the individuals in the dataset.

2) There is an extra file in the PBAP transposed file format which contains the IDs of individuals with genotype data. The order of the individual IDs in this file should match the order of individuals in the genotype file.

Individual IDs in the pedigree file come in different forms and may have one or more delimiters. For simplicity, we recommend use of only one delimiter, which separates the family ID from the individual ID. Furthermore, MORGAN implements a maximum of 15 characters for the ID of an individual. PBAP, therefore, employs a maximum of 15 characters to be compatible with the MORGAN format. This maximum includes the delimiter so the total number of characters for Family ID_Individual ID should be ≤14. Note that if you intend to use makeped, it implements a maximum of 11 characters. To facilitate handling files with these restrictions, PBAP allows use of an ID translation file as additional input (see 2.4.1.1).

Note: If you intend to run linkage analysis on the sampled inheritance vectors (IVs) using gl_lods (Koepke and Thompson, 2013; Thompson, 2011), and would like PBAP to prepare the necessary MORGAN-format files, you must include an input phenotype file (2.4.1) at the very beginning of the pipeline.

2.3.2. Reference Data Input Files

There are two main types of reference data input files used by PBAP: map files and main population genotype files. We have constructed such files from publicly available data but you may construct and use any reference files of your choice that contain the necessary information. These reference files should be prepared prior to using PBAP.

2.3.2.1. Reference Map Files

We combined the Rutgers smoothed framework map and the Rutgers map of all dbSNP Build 134 variants (Matise *et al.*, 2007; Nato *et al.*, in prep) that we downloaded from http://compgen.rutgers.edu/download_maps.shtml. On the merged map files, we converted Kosambi distances between markers to Haldane distances, thus providing Haldane genetic locations (cM). The latest version of the Rutgers Maps now contain both Kosambi and Haldane genetic locations.

2.3.2.2. Reference Main Population Genotype Files

We downloaded the Phase I integrated release version 3 (November 2010 data freeze) of the 1000 Genomes Project (1000G) data (Altshuler *et al.*, 2010) from http://archive.sph.umich.edu/csg/abecasis/MACH/download/1000G.2012-03-14.html, which have already been separated into four main populations (African (AFR), Admixed American (AMR), East Asian (ASN), and European EUR)). If the link above has changed, you may try: ftp://share.sph.umich.edu/1000genomes/fullProject/2012.03.14/ and download the files whose filenames contain one of the three-letter code for the main populations. We used a custom Perl script (exclude_indels_dups.pl), which is included in the PBAP release, to (1) pre-process the 1000G genotype files (*i.e.*, to exclude indels and duplicate entries) by using vcftools (http://vcftools.sourceforge.net/) to remove indels and convert the VCF files into PLINK

transposed file format (PLINK *.tped and*.tfam), (2) perform simple subsequent check, and (3) save the output files into the corresponding main population folder. You may prefer at the command line by typing:

vcftools --vcf chr#.*.population.vcf --remove-indels --plink-tped --out population_chr#

where population would be the three-letter code for one of the four main populations (i.e., AFR, AMR, ASN, or EUR), # would be the chromosome number, and * represents the long part of the filename.

These reference files (*.tped and *.fam) will be needed for applications 2 and 4 of the pipeline. Folders (named using their three-letter codes) for each of these populations should be created under the main 1000G folder. After running exclude_indels_dups.pl, copy the output files (minimally just the *.tped and *.tfam files) to their respective main population folders.

In all examples presented in this manual, we used the 1000G genotype data for the European population. In selecting marker subpanels (2.4.2), PBAP **requires** pedigree files, these pre-processed reference genotype files, and uses PLINK (Purcell *et al.*, 2007) to calculate main population allele frequencies and to generate LD estimates.

2.4. PBAP Applications

PBAP comprises five main applications, namely:

1) transposition of normal file format
2) selection of marker subpanels
3) genotype-based kinship estimation
4) preparation of files for MORGAN
5) execution of gl_auto to sample inheritance vectors (IVs)

If your input files are in the transpose file format (see 2.3.1), skip application 1 but make sure that you follow the required file format in the subsequent applications of PBAP.

2.4.1. Application 1: Transposition of Normal File Format

This application uses the script transpose_fileset.pl, which transposes files that are in the normal file format (see 2.3.1) and generates files that will be used by subsequent scripts of the pipeline. Run transpose_fileset.pl by typing

./transpose_fileset.pl <chromosome> <parameter file*> <family ID>**

*use absolute path
**optional: specify the family ID if you want to generate transposed file format files for one specific family

The parameter file for transpose_fileset.pl should contain the entries shown in Table 1. **For all of the parameter files, blank lines or lines that start with a pound (#) sign will be ignored. To see the complete details regarding each line in this parameter file as well as in the parameter files of the succeeding scripts of PBAP, see "README_screen_menus."**

Table 1. Parameter file for transpose_fileset.pl

Line #	Description
1	Directory containing transpose_fileset.pl[a]
2	Directory for output files[a]
3	Additional codes for missing data[b]
4	Input pedigree filename[a,c]
5	Specify whether input pedigree file has a header or not: header=[T|F]
6	Input phenotype filename[a,c,d]
7	Specify whether input phenotype file has a header or not: header=[T|F]
8	Columns in phenotype file containing FamilyID and Individual ID[e]
9	Contiguous columns in phenotype file for phenotypes/covariates of interest
10	Column in phenotype file to use for normal fileset ped file and PLINK-format tfam file
11	Description of phenotype: pheno=[description of phenotype]
12	Phenotype conversion file [a,c,d] (for MORGAN-format pedigree file) – described further below
13	Directory containing map files[f]
14	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where map files are located[g,h]
15	[(prefix=) (suffix=)] of chromosome number in map filenames[g]
16	Specify whether input map files have headers or not: header=[T|F]
17	Columns in map files for chromosome, marker, genetic location, and physical position
18	Directory containing genotype files[f]
19	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where genotype files are located[g,h]
20	[(prefix=)(suffix=)] of chromosome number in genotype filenames[g]
21	Specify whether input genotype files have headers or not: header=[T|F]
22	Columns in genotype files containing family ID and individual ID[e]
23	Column containing the allele of the first marker[i]
24	Delete normal file format (ped/map) after running the script: [Y|N]
25	Delete PLINK-format transposed file format (tfam/tped) after running the script: [Y|N]
26	Marker type: [SNP|STR]
27	Option: [include|exclude] markers without genetic locations
28	Option: [include|exclude] markers without physical positions
29	Priority: option for [genetic|physical] in line 27 or 28 overrides the other
30	Family ID translation file[a,j]
31	Renumber individual IDs within each family[k]: [Y|N]
32	Family ID - Individual ID translation file[a,l]

[a]Use absolute path
[b]Current PBAP codes for missing data: -0-, MISSING, MIS, miss, NA, -9, -1, 0, and – so if you don't have additional codes, put 'no_extra_code'
[c]Use one file that includes all families or for the entire dataset
[d]Put 'none' if you don't have this input file
[e]If column 1 or 2 contains Family ID_Individual ID, put '1' or '2', respectively. If columns 1 and 2 contain family ID and individual ID, respectively, put '1 2'
[f]If there is a subdirectory for each chromosome, do not include it here
[g]Indicate the portion of the filename that precedes (prefix) and succeeds (suffix) the chromosome number (*e.g.*, if chromosome subdirectory is chr22, then the chromosome number is '22' so put 'prefix=chr suffix=none'; if filename is chr22-panelA-map.txt, put 'prefix=chr suffix=-panelA-map.txt' for this line).
[h]For subdirectories in lines 14 and 19, if files in lines 15 and 20 are directly under the directories in lines 13 and 18, respectively, put 'no dir'
[i]If you are using only one *.ped file for the pedigree, phenotype, and genotype input files, where genotype data starts at column 7, put '7'. If your genotype file is already in the PBAP format, *i.e.*, columns are Family ID_Individual ID and Genotypes, put '2'.
[j]Format (space-delimited): Current Family ID, New Family ID. If you don't have a Family ID translation file, put 'none'
[k]If you specified a family ID translation file (line 30), put 'Y' to renumber individual IDs per family. Otherwise, put 'N'
[l]Format (space-delimited): Current family ID, Current individual ID, New Family ID, New Individual ID. If you don't have Family ID-Individual ID translation file, put 'none'

2.4.1.1. Input Files

Lines 4-23, 30, and 32 of the parameter file (and in 2.3.1) focus on the input files, namely:

1) pedigree file (lines 4-5)
2) map file (lines 13-17)
3) genotype file (lines 18-23)
4) phenotype file (lines 6-11) - optional
5) phenotype conversion file (line 12) – optional
6) family ID translation file (line 30) - optional
7) family ID - individual ID translation file (line 32) - optional

For the pedigree file and the optional files above, simply specify the absolute path and indicate whether the file has a header or not. For the map and genotype files, PBAP requires a breakdown of the location and filenames to allow use of only one parameter file for the entire genome. **This is very useful when you want to run all 22 chromosomes at the same time in a cluster without the need to create 22 separate parameter files.**

Although we specified formats for the project data input files in 2.3.1, PBAP actually accepts any of the formats for the input pedigree, genotype, and phenotype files listed below. In fact you may use the same PLINK-format *.ped file per chromosome as the input pedigree, genotype, and phenotype files. Specify the appropriate columns accordingly (lines 8, 9, and 23). As stated earlier, all input files should be space- or tab-delimited.

Input pedigree file formats

1) Family ID, Family ID_Individual ID, Family ID_Father ID, Family ID_Mother ID, Sex
2) Family ID, Individual ID, Father ID, Mother ID, Sex
3) Family ID_Individual ID, Family ID_Father ID, Family ID_Mother ID, Sex
4) Family ID, Individual ID, Father ID, Mother ID, Sex, Affectation Status (or Phenotype), Genotypes (use this format only when all of the individuals in your pedigree are in this file)

Input genotype file formats

1) Family ID_Individual ID, Genotypes
2) Family ID, Individual ID, Father ID, Mother ID, Sex, Affectation Status (or Phenotype), Genotypes

Input phenotype file formats

1) Family ID_Individual ID, Phenotypes
2) Family ID, Individual ID, Father ID, Mother ID, Sex, Affectation Status (or Phenotype), Genotypes (using this format will result in only one phenotype in your output file)

Input map file format

For the input map file, you may use any format as long as the file has the following information:

Chromosome, Marker, Genetic Location, Physical Position

Marker genetic locations should be in Haldane cM. Once you have identified the columns in the map file containing the information needed, specify the column numbers (line 17).

Phenotype conversion file

If you intend to run linkage analysis using gl_lods of the MORGAN package (Koepke and Thompson, 2013; Thompson, 2011), we reiterate that you must include a phenotype file. We recommend that you include all columns of phenotypes or covariates in your input phenotype file that you may use in downstream analyses (line 9) since transpose_fileset.pl creates a MORGAN-

format pedigree file that includes all the phenotypes that you specify. For binary phenotypes, MORGAN uses three values for affectation status, namely: 0 (missing), 1 (unaffected), and 2 (affected). If any of your binary phenotypes/covariates are not in this format, you must provide a phenotype conversion file (line 12), which has the format (space-delimited):

> Column Number in Phenotype File, Type [binary|continuous|integer|string], Affectation Status, Value(s)

You must not use a header for this file. Each phenotype in phenotype file should have one to three rows in the phenotype conversion file. For a binary phenotype, the phenotype conversion file should have three rows specifying the three value(s) that correspond to Affectation Status as 0, 1, and 2. If a phenotype is a continuous variable, an integer, or a string and you want to keep it as is, use one row to specify value(s) only for Affectation Status 0 (missing). However, if you want to convert it into a binary variable, use three rows to specify value(s) for Affectation Status 0, 1, and 2. For Value(s), you can specify a single value, a set of values (comma-delimited, *e.g.*, '2,5,7,8'), or a range of values (dash-delimited, *e.g.*, '2-10').

Consider a phenotype file for a family composed of 10 individuals with five phenotypes where the first column contains the family ID_individual ID, *e.g.*,

```
A_001 8.3675 1 60 0 AA
A_002 4.1045 0 75 D 0
A_003 7.5843 0 57 Y BB
A_004 1.3295 0 77 G BB
…
```

For this phenotype file, the phenotype variables are continuous (real), binary, integer, one-character string, and multi-character string, respectively. The phenotype conversion file would be similar to the one below.

```
2 continuous 0 NA,0
3 binary 0 NA,0
3 binary 1 1
3 binary 2 2
4 integer 0 NA,0
4 integer 1 50-65
4 integer 2 66-80
5 string 0 NA,0
5 string 1 A-M
5 string 2 N-Z
6 string 0 NA,0
6 string 1 AA,BB
6 string 2 CC,DD
```

There is only one row for the continuous variable and it instructs PBAP to convert 0 or NA to Affectation Status 0. For the binary variable, which is already in MORGAN-compatible format, the same value is placed for both the Affectation Status 1 and 2. For the integer variable, integers from 50-65 and 66-80 will be recoded as Affectation Status 1 and 2, respectively. For the one-character string variable, letters from A-M and N-Z will be recoded as Affectation Status 1 and 2, respectively. For the multi-character string variable, phenotypes AA or BB will be recoded as Affectation Status 1 while phenotypes CC or DD will be recoded as Affectation Status 2. PBAP converts these phenotypes and rearranges the columns (*i.e.*, set of integers first followed by set of

real numbers) to create a MORGAN-format pedigree file that contains your phenotypes of interest (see below). At this stage, the components of the pedigree, discussed in 2.4.4 and 2.4.5, have not been dealt with. For individual A_005 who has no phenotype data, all phenotypes are coded as missing (0).

```
input pedigree size 10
input pedigree record names 3 integers 5 reals 1
input pedigree record father mother
******
A_001 A_002 A_003 1 1 1 0 1 8.3675
A_002 A_010 A_011 1 0 2 1 0 4.1045
A_003 0 0 2 0 1 2 1 7.5843
A_004 A_002 A_003 1 0 2 1 1 1.3295
A_005 A_002 A_003 2 0 0 0 0 0
…
```

ID translation files

If the family IDs in your files are too long or complex, use of a family ID translation file (line 30) is recommended, which has the format (space-delimited):

Current Family ID, New Family ID

If you want to renumber the individual IDs per family from 1 up to the number of individuals in that family (n), put 'Y' in line 31. Otherwise, put 'N'.

Another option is to use a family ID – individual ID translation file (line 32) instead of a family ID translation file. Use this to specify the new family and individual IDs in your dataset. This file has the format (space-delimited):

Current Family ID, Current Individual ID, New Family ID, New Individual ID

2.4.1.2. Options and Specifications

Specify the family ID in the command line (2.4.1) if you want to generate transposed files for only one family.

Lines 24-29 and 31 of the parameter file focus on the different options for output files that transpose_fileset.pl would create depending on what the user needs.

For lines 24 and 25, we recommend that you keep the normal file format (ped/map) and PLINK-format transposed files (tfam/tped) files by putting 'Y' unless you're absolutely sure that you won't use them in any of your analyses.

Line 26 allows the user to indicate the marker type and it will be included in the LOG file.

Lines 27 and 28 allow the user to control whether or not a marker will be included or not. If you want to exclude all markers that have no genetic location in your map file, put 'exclude' in line 27. If you want to include all markers that have no physical position in your map file, put 'exclude' in line 28. Line 29 allows you to prioritize the option you chose for either line 27 or 28. If you want to prioritize a marker with genetic location (*i.e.*, include marker if genetic location is present

regardless of whether or not physical position is present), put 'genetic'. If you want to prioritize a marker with physical position (*i.e.*, include marker if physical position is present regardless of whether or not genetic location is present), put 'physical'. Tables 2 and 3 indicate what PBAP does based on the options chosen for lines 27-29. User-specified parameters are shown in italics.

Table 2. Prioritize markers based on presence of genetic location (line 29 is 'genetic')

			Genetic Location		
			absent (-)		present (+)
		line 27 / **line 28**	***Include***	***Exclude***	
Physical Position	absent (-)	***Include***	include	exclude	include
		Exclude	include	exclude	include
	present (+)		include	exclude	include

Table 3. Prioritize markers based on presence of physical position (line 29 is 'physical')

			Genetic Location		
			absent (-)		present (+)
		line 27 / **line 28**	***Include***	***Exclude***	
Physical Position	absent (-)	***Include***	include	include	include
		Exclude	exclude	exclude	exclude
	present (+)		include	include	include

2.4.1.3. Output Files

All output folders and files we be saved under the output directory (outdir) that you specified. **We will use the term "outdir" for all output directories in this manual. For all PBAP applications, the outdir will be created if it does not exist.** If you specified a family ID (*e.g.*, famA) in the command line (2.4.1), all output files will be saved under /outdir/pedfamA. Otherwise, they will be saved under /outdir/allpeds. Under this directory, folders for the chromosome that you ran will be created. For example, if you ran transpose_fileset.pl for chromosome 22 and you specified fam22, output files will be under /outdir/pedfamA/chr22.

Depending on the options that you made, you should have four up to ten space-delimited output files, a LOG file (*e.g.*, transpose_fileset_chr22.log), and possibly an aux folder. All of these files except for the LOG and MORGAN-format pedigree files have no headers. Individual IDs, father IDs, and mother IDs in these output files now have the family ID concatenated with them, *i.e.*, Family ID_Individual ID, Family ID_Father ID, and Family ID_Mother ID, respectively, except for the parents of founders who are coded as zeroes. The main output files (also see 2.3.1) are the only main files generated if you opted to delete some of the output files (lines 24 and 25) and did not include an input phenotype file. If you included a phenotype file and a phenotype conversion file, the phenotype-related output files will be listed with the main output files in the LOG file. If you ran transpose_fileset.pl on chromosome 22, your output files and how we will refer to them throughout this manual are shown in Table 4 below. In addition to these output files, auxiliary files are generated depending on your dataset. In the case of famA, your auxiliary files will be under /outdir/pedfamA/chr22/aux.

Table 4. Output files of transpose_fileset.pl

File Group	Filename (Designation)	Format or Description
PBAP-format (main)	chr22.tpedo (TPEDO)	Family ID, Individual ID, Father ID, Mother ID, Sex
	chr22.tmap (TMAP)	Chromosome, Marker, Genetic Location, Physical Position
	chr22.tgen (TGEN)	Chromosome, Marker, Genotypes
	chr22.tind (TIND)	List of genotyped individuals[a]
Normal file format	chr22.ped (PED)	Family ID, Individual ID, Father ID, Mother ID, Sex, Phenotype, Genotypes
	chr22.map (MAP)	Chromosome, Marker, Genetic Location, Physical Position
PLINK-format (transposed)	chr22.tfam (TFAM)	Family ID, Individual ID, Father ID, Mother ID, Sex, Phenotype
	chr22.tped (TPED)	Chromosome, Marker, Genetic Location, Physical Position, Genotypes
Phenotype-related	chr22.tphen (TPHEN)	Individual ID, Phenotype(s)
	chr22.mped (MPED)	Individual ID, Father ID, Mother ID, Sex, Phenotype(s)
Auxiliary files	chr22.trans (TRANS)	Old Family ID, Old Individual ID, New Family ID, New Individual ID[b]
	chr22.nogen (NOGEN)	List of individuals without genotype data
	chr22.gnip (GNIP)	List of individuals with genotype data who are not in the input pedigree file
	chr22.pnip (PNIP)	List of individuals with phenotype data who are not in the input pedigree file

[a]The order of the individuals in this file corresponds to the order of individuals in the TGEN file
[b]Family ID - Individual ID translation file

Lastly, the LOG file contains the following information:

1) parameters that you specified in the command line and in the parameter file
2) location (absolute path) and format of output files
3) date and time that you started running transpose_fileset.pl
4) total real, system, and user times

As indicated earlier, PBAP scripts usually run one chromosome at a time. Although some of the files will be identical across chromosomes, it would be best to have copies under each chromosome folder since these would be used separately in succeeding steps. The filenames include the chromosome number so that, in the event that you want to combine all of these files in one folder, you won't need to rename each of these files.

2.4.2. Application 2: Selection of Marker Subpanels

This application uses the script marker_subpanels.pl, which carefully selects non-overlapping subsets of markers from a dense marker panel. Marker_subpanels.pl uses the following criteria/parameters:

1) LD (r^2)
2) Minor allele frequency (MAF)
3) Monomorphic markers
4) Direction of marker processing
5) Starting marker
6) Minimum intermarker distance (MID)
7) Main population (*i.e.*, AFR, AMR, ASN, or EUR)
8) Source of MAF information
9) Types of markers with genotype data
10) Number of marker subpanels
11) STRs as core inclusion markers
12) Gap filling

Run marker_subpanels.pl by typing

./marker_subpanels.pl <chromosome> <parameter file*>

*use absolute path

The parameter file for marker_subpanels.pl should contain the entries shown in Table 5.

Table 5. Parameter file for marker_subpanels.pl

Line #	Description
1	Chromosome(s) that have genotype data
2	Directory containing marker_subpanels.pl[a]
3	Directory for output files[a,b]
4	Inclusion filename for SNPs[a,c,d,e]
5	Inclusion filename for STRs and minimum distance (cM) from SNPs[a,c,d,f]
6	Exclusion filename[a,c,d,g]
7	Prepare files for pedigree structure validation (kstat.pl): [Y|N]
8	Minimum intermarker distance (cM)
9	Marker completion (%) threshold
10	Minor allele frequency (MAF) minimum and maximum cut-offs[h]
11	Maximum LD threshold[i]
12	Exclude monomorphic markers: [Y|N][j]
13	Number of marker subpanels[k]
14	Sequence of methods for each subpanel specified in line 13[l]
15	Path of PLINK that will be used (*e.g.*, /usr/bin/plink or /home/username/bin/plink)
16	Directory containing external reference files (*e.g.*, 1000G files)[a,m]
17	Population: [AFR|AMR|ASN|EUR]
18	Directory containing map files[n]
19	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where map files are located[o,p]
20	[(prefix=) (suffix=)] of chromosome number in map filenames[o]
21	Specify whether input map files have headers or not: header=[T|F]
22	Directory containing pedigree files[n]
23	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where pedigree files are located[o,p]
24	[(prefix=) (suffix=)] of chromosome number in pedigree filenames[o]
25	Specify whether input pedigree file have headers or not: header=[T|F]
26	Directory containing genotype files[n]
27	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where genotype files are located[o,p]
28	[(prefix=) (suffix=)] of chromosome number in genotype filenames[o]
29	Specify whether input genotype files have headers or not: header=[T|F]
30	Option: [1|2] for input file(s) containing IDs of genotyped individuals[q]
31	Input file(s) with IDs of genotyped individuals[r]
32	[(prefix=) (suffix)] of chromosome number in names of subdirectories where files with IDs of genotyped individuals are located[o,p]
33	[(prefix=) (suffix)] of chromosome number in files with IDs of genotyped individuals[o]
34	Specify whether input files with IDs of genotyped individuals have headers or not: header=[T|F]

[a]Use absolute path
[b]Under this directory, subdirectories for marker subpanel number, main population, and chromosome will be created
[c]Use one file that applies to the entire dataset. Remove any header(s).
[d]Put 'none' if you don't have this input file
[e]Format (space-delimited): Chromosome, Marker, [core|aux]
[f]Format (space-delimited): For line 5, Inclusion Filename, Minimum Distance. For inclusion file, Chromosome, Marker, Genetic Location, Physical Position
[g]Format (space-delimited): Chromosome, Marker
[h]Format (space-delimited): Minimum, Maximum; Recommended values: for pedigree structure validation ("0.3 0.5"), for linkage analysis ("0.2 0.5")
[i]Recommended values: for pedigree structure validation ("0.25"), for linkage analysis ("0.04" or "0.01")
[j]By default, markers that are monomorphic will be excluded. If these markers actually have very low MAFs and only happen to be monomorphic in this dataset due to small sample size and you want to keep them, include them in one the inclusion files (lines 4-5)
[k]Recommended values: for pedigree structure validation ("1"), for linkage analysis ("3" or more)
[l]Format (space-delimited): Subpanel Number, Direction, Marker Number (Main Panel), Marker Number (Pre-Subpanel), *e.g.*, "1 fwd 1 1" means "for subpanel 1, use forward direction, start at marker 1 on main panel, and start at marker 1 on pre-subpanel." Separate sequence of methods between panels by a ' | '.
[m]Make sure that you have placed the subfolders for the AFR, AMR, ASN, and EUR populations of your reference files under this directory prior to running marker_supbanels.pl
[n]If there is a subdirectory for each chromosome, do not include it here
[o]Indicate the portion of the filename that precedes (prefix) and succeeds (suffix) the chromosome number (*e.g.*, if chromosome subdirectory is chr22, then the chromosome number is '22' so put 'prefix=chr suffix=none'; if filename is chr22.tmap, put 'prefix=chr suffix=.tmap' for this line).
[p]For subdirectories in lines 14 and 19, if files in lines 15 and 20 are directly under the directories in lines 13 and 18, respectively, put 'no dir'
[q]Option 1: Use output files of transpose_fileset.pl. Option 2: Use a file containing the list of genotyped individuals (supplied by user). If you chose option 2, line 31 should contain the absolute path of input file and leave lines 32-34 blank.
[r]If option 1 was chosen in line 30, indicate directory containing files with IDs of genotyped individuals[n]. If option 2 was chosen in line 30, indicate input file with list of IDs of genotyped individuals[a].

2.4.2.1. Input Files

Reference main population genotype files (2.3.2.2), *e.g.*, 1000G files, should have been prepared or downloaded before running marker_subpanels.pl. The subfolders for each of the different main populations should be in the same external reference directory (line 16).

Lines 4-6 and 18-34 of the parameter file focus on the input files, namely:

1) inclusion file for SNPs (line 4)
2) inclusion file for STRs (line 5)
3) exclusion file (line 6)
4) map files (lines 18-21)
5) pedigree files (lines 22-25)
6) genotype files (lines 26-29)
7) list of genotyped individuals (lines 30-34)

For the inclusion and exclusion files, simply specify the absolute path. For all other files, PBAP requires a breakdown of the location and filenames to allow use of only one parameter file for the entire genome. The inclusion file for SNPs has the format (space-delimited, see explanation for 'core' and 'aux' at 2.4.2.2 below):

Chromosome, Marker, [core|aux]

Short tandem repeats (STRs), which were previously available at NCBI UniSTS with other sequence-tagged sites (STSs), are now available at NCBI Probe (http://www.ncbi.nlm.nih.gov/probe) and may be searched by using the search term 'unists[properties]'. Flat files may also be obtained from the NCBI FTP site in the UniSTS repository (ftp://ftp.ncbi.nih.gov/pub/ProbeDB/legacy_unists). The inclusion file for STRs has the format:

Chromosome, Marker, Genetic Location, Physical Position

2.4.2.2. Options and Specifications

Lines 4-5, 7-14, 17, and 30 of the parameter file focus on the different options and specifications for the user. For lines that require values, do not put the units (*e.g.*, cM, %, etc).

For line 4, indicate the absolute path of the inclusion file for SNPs and for each line in the inclusion file, you have to indicate whether a marker should always be included in the first marker subpanel (*i.e.*, core inclusion marker so put 'core') or a marker will be given priority but should pass thresholds for minimum MAF (in the reference dataset), minimum marker completion, and should not be monomorphic in the dataset (*i.e.*, auxiliary inclusion marker so put 'aux'). As long as these inclusion markers exist in the external reference sources (*e.g.*, 1000G) or markers with different names (*i.e.*, different rsIDs) but with the same physical positions from the same build exist, marker_subpanels.pl will be able to effectively include these markers in the first marker subpanel.

For line 5, you have to indicate the inclusion file for STRs and the minimum distance (cM) of these STRs from the SNPs. Since you have given the genetic locations of these STRs within the inclusion

file, PBAP forces these STRs into the pre-subpanel and adds SNPs that pass the various parameters and are beyond the minimum distance that you have specified. STRs in this file are automatically core inclusion markers.

For line 7, put 'Y' if you want to generate one marker subpanel for pedigree structure validation. Marker_subpanels.pl should be executed for all chromosomes before performing pedigree structure validation. To generate marker subpanel(s) for linkage analysis, put 'N'.

For line 8, specify a minimum intermarker distance (MID) in cM. To avoid or to minimize MCMC-mixing issues (Sieh *et al.*, 2005; Thompson and Heath, 1999; Wilcox *et al.*, 2005) when sampling IVs using gl_auto (2.4.5), we recommend an MID of 0.5 cM.

Line 9 allows the user to specify a minimum value for marker completion (%) of the dataset. We recommend values of at least 80%. If you want to be really stringent, put about 95%.

For lines 10 and 11, the recommended values are specified in Table 5 footnotes h and i. In line 10, we recommend use of variants that are more common (MAF≥0.3), for pedigree structure validation. For linkage analysis, a MAF≥0.2 may be used. Note that this is for the MAF of the marker in the external reference files (*i.e.*, 1000G). Line 11 allows the user to specify a maximum LD (r^2) between markers spanning ~1cM in the marker subpanel. For linkage analysis, markers should be in linkage equilibrium so an r^2 of 0.01 or 0.04 is recommended.

With regard to monomorphic markers in the dataset, specify whether you want to exclude ('Y') or include ('N') in line 12.

For line 13, specify the number of marker subpanels that you want to generate. For pedigree structure validation, you would need only one subpanel each for the 22 chromosomes. For linkage analysis, you can specify one or more non-overlapping subpanels. Use the first subpanel for usual linkage analysis then you can come back later to use the other two for regions of interest or for regions which potentially have genotyping errors.

Line 14 allows the user to introduce more variables that will add to the variability of the non-overlapping subpanels generated by specifying: (1) direction of marker processing, (2) starting marker in the main dense panel, and (3) starting marker in the pre-subpanel. Format is shown below (separate sequence of methods between panels by a ' | '):

Subpanel Number, Direction, Marker Number (Main Panel), Marker Number (Pre-Subpanel)

For example, "1 fwd 1 1 | 2 fwd 5 1" means "for subpanel 1, use forward direction, start at marker 1 on main panel, and start at marker 1 on pre-subpanel and for subpanel 2, use forward direction, start at marker 5 on main panel and start at marker 1 on pre-subpanel" (Table 5, footnote l).

Specify the main population in line 17. The pre-processed 1000G files (2.3.2.2) are divided into four main populations, namely: (1) African (AFR), Admixed American (AMR), East Asian (ASN), and European (EUR).

For line 30, put '1' if you want to use the output files of transpose_fileset.pl (2.4.1) or put '2' if you want to specify a file which contains the list of individuals with genotype data.

2.4.2.3. Output Files

All output folders and files we be saved under the outdir that you specified. Under this directory, folders for the marker subpanel number, main population, and chromosome that you ran will be created. For example, if the marker subpanel number, main population, and chromosome are 1, EUR, and 22, respectively, the output files will be under /outdir/panel1/EUR/chr22, the LOG (*.log), pedigree (*.tfam), and genotype (*.tped) files will be under /outdir, while the auxiliary/intermediate files will be under /outdir/panel1/EUR/aux/chr22 and /outdir/panel1/EUR/aux2/chr22 (Table 6). Various PLINK-format TFAM and TPED files are created in the process of selecting markers.

You should have two space-delimited output files, a LOG file, and auxiliary folder(s) (aux and aux2). The auxiliary folder aux2 is only created when you are selecting marker subpanels for linkage analysis. All of these files except for the LOG, MATCH (*.match), DESC (*.desc), and PLINK-format FREQ (*.frq) files discussed below have no headers.

The main output file for this example (*i.e.*, chr22.pmap (PMAP)) is a marker subpanel map file that contains information for each marker. Columns 1-4 can easily be parsed to create a regular map file of the marker subpanel. Columns 5-6 indicate the marker type (SNP or STR) and marker completion, respectively. Columns 7-9 and 10-12 each indicate the MAF, expected heterozygosity (H_E), and observed heterozygosity (H_O) for 1000G and your dataset, respectively. Columns 13-14 indicate reference alleles 1 and 2 (1000G) while columns 15-16 indicate dataset alleles 1 and 2. Columns 17-18 indicate the intermarker distance (ID) in cM and Mbp, respectively.

The marker subpanel description file (*e.g.*, chr22.desc (DESC)) contains a summary of the different parameters in the SPMAP file, namely: Intermarker distance, Mbp/cM, marker completion, MAF, H_E, and H_O.

All the other output and auxiliary files are described in Table 6.

Table 6. Output files of marker_subpanels.pl

Directory	Filename (Designation)	Format (Description)
/outdir	marker_subpanels_EUR_chr22.log (LOG)	(Contains information for all marker subpanels generated, parameters specified, input files, output directory, markers excluded, reasons for marker exclusion, date and time that you started running marker_supbanels.pl, and total real, system, and user times)
	chr22.tfam (TFAM)	Family ID, Individual ID, Father ID, Mother ID, Sex, Phenotype (PLINK-format pedigree file of dataset)
	chr22.tped (TPED)	Chromosome, Marker, Genetic Location, Physical Position, Genotypes (PLINK-format SNP-genotype file of dataset)
/outdir/panel1/EUR/chr22	chr22.pmap (PMAP)	Chromosome, Marker, Genetic Location, Physical Position, Marker Type, Marker Completion, MAF_{1000G}, $H_{E,1000G}$, $H_{O,1000G}$, $MAF_{dataset}$, $H_{E,dataset}$, $H_{O,dataset}$, $A1_{1000G}$, $A2_{1000G}$, $A1_{dataset}$, $A2_{dataset}$, Intermarker Distance (cM), Intermarker Distance (Mbp) (Marker subpanel map file)
	chr22.desc (DESC)	Parameter, Number of Markers, Mean, Variance, Standard Deviation, Minimum, Lower Quartile, Median, Upper Quartile, Maximum (Marker subpanel map description file)
	chr22.tgen (TGEN)[a]	Chromosome, Marker, Genotypes
	chr22.tind (TIND)[a]	List of genotyped individuals
/outdir/panel1/EUR/aux/chr22	chr22.tfam (TFAM)	Format same as that of TFAM file above (PLINK-format pedigree file (1000G) in which the Father ID, Mother ID, Sex, and Phenotype are all missing (0).)

	chr22.tped (TPED)	Format same as that of TPED file above (PLINK-format SNP-genotype file (1000G) based on the marker names or physical positions in the dataset genotype file under /outdir. Genetic location of the markers in this file is missing (0). This is the dense marker panel with genotypes from 1000G.)
	chr22_KGpos.match (MATCH)	Dataset rsID, Physical Position, 1000G rsID (List of markers in the dataset that matched that of 1000G based on physical position)
	chr22_den.frq (FREQ)	Chromosome, SNP, Allele1, Allele2, MAF, NCHROBS (PLINK-format allele frequency file for the dense marker panel)
	chr22_den.log (LOG)	(PLINK log file for obtaining allele frequencies)
	chr22_den.pmap (PMAP)	Format same as that of PMAP file above (Dense marker panel map file)
	chr22_den.desc (DESC)	Format same as that of DESC file above (Dense marker panel description file)
	chr22_ext.tmap (TMAP)	Chromosome, Marker, Genetic Location, Physical Position, Intermarker Distance (Based on 1/5 of the user-defined intermarker distance, this is a map file of the markers that will be extracted from chr22.tped (1000G TPED file above; marker names still match that of the original map file)
	chr22_ext.in (IN)	(List of markers that will be extracted from chr22.tped (1000G TPED file above; markers matched based on physical position now use the marker names in the 1000G TPED file)
	chr22_red.tped (TPED) chr22_red.tfam (TFAM)	Format same as that of TPED and TFAM files above (PLINK-format SNP-genotype file of markers extracted using chr22_ext.in from 1000G TPED file. We call this the reduced set of markers. Some of the markers in chr22_ext.in may not be in the 1000G TPED file.)
	chr22_ext1.tmap (TMAP)	Chromosome, Marker, Genetic Location, Physical Position, Reason for Exclusion (Map file of markers that were excluded due to marker completion, MAF, for being monomorphic in the dataset, or for being in the list of exclusion markers)
	chr22_10_4_04.prune.in (IN) chr22_10_4_04.prune.out (OUT) chr22_10_4_04.log (LOG)	(PLINK-format prune.in, prune.out, and LOG files created by PLINK after LD-based SNP pruning. The three numbers after "chr22_" correspond to window size (number of markers), increment (number of marker), and maximum LD threshold (r^2), respectively. The prune.in file contains the list of markers that will be extracted from chr22_red.tped file.)
	chr22_pru.tped (TPED) chr22_pru.tfam (TFAM)	Format same as that of TPED and TFAM files above (PLINK-format SNP-genotype file of markers extracted using prune.in file above from chr22_red.tped file. We call this the pruned set of markers)
	chr22_pru_f.frq (FREQ)	Format same as that of FREQ file above (PLINK-format allele frequency file for the pruned set of markers)
	chr22_pru_f.log (LOG)	(PLINK log file for obtaining allele frequencies)
	chr22_pru.tmap (TMAP) chr22_prurev.tmap (TMAP)	Chromosome, Marker, Genetic Location, Physical Position, Marker Type (From the pruned set of markers, inclusion markers are included only in panel 1. The *pru.tmap and *prurev.tmap files are map files where the genetic locations (cM) are in increasing and decreasing order, respectively. From these map files, the pre-subpanel map file is generated based on in the user-specified minimum intermarker distance (MID).)
	chr22_pre.pmap (PMAP)[b]	Format same as that of PMAP file above (Pre-subpanel map file)
	chr22_pre.desc (DESC)[b]	Format same as that of DESC file above (Pre-subpanel description file)
	chr22_pre.gaps (GAPS)[b]	Gap Start (cM), Gap End (cM), Gap Length (cM) (Gaps in the pre-subpanel map file are listed in this file.)
	chr22_pre.gaps (NGAPF)[b]	Gap Start (cM), Gap End (cM), Gap Length (cM), MID, Number of Gap Fillers, Range (Information in this file will be used to fill in gaps in the pre-subpanel and create the marker subpanel. Range (cM) indicates the size of the region where markers will be searched for potential gapfillers.)
/outdir/panel1/EUR/aux2/chr22	chr22_gap_*_*_*.tped (TPED) chr22_gap.tfam (TFAM) chr22_gap_*_*_*.log (LOG) chr22_gap_*_*_*.ld (LD)	Format same as that of TPED, TFAM and LOG files above. The format of the PLINK LD file is: CHR A, BP A, SNP A, CHR B, BP B, SNP B, r^2

chr22_gap.nosex	(The three asterisks represent the gap start genetic location start, range minimum, and range maximum, where region specified by the range is where SNPs will be chosen to fill the gap. The chr22_gap.nosex contains the list of individuals from 1000G files, all of which have no sex specified)

[a]Created only when selecting marker subpanels for pedigree structure validation
[b]Created only when selecting marker subpanels for linkage analysis
MAF: minor allele frequency
H_E: expected heterozygosity
H_O: observed heterozygosity
A1: Allele 1
A2: Allele 2
1000G: 1000 Genomes Project data (Altshuler *et al.*, 2010)
NCHROBS: number of observations
MID: minimum intermarker distance
CHR: chromosome
BP: base pair

After you have finished running marker_subpanels.pl, examine the following items in your marker subpanel PMAP file (*e.g.*, chr22.pmap in Table 6):

1) Columns 6-7 (Marker completion and MAF (1000G)): All markers should have marker completion and MAF greater than or equal to the marker completion and minimum MAF that you specified in your parameter file (lines 9-10).
2) Columns 7-12 (MAF, H_E, and H_O in 1000G and dataset, respectively): All values should be ≥0 if you excluded monomorphic markers (line 12). MAF (dataset) can be less than the minimum MAF that you specified (line 10) since PBAP only filters based on MAF (1000G).
3) Columns 13-16 (alleles 1 and 2 in 1000G and dataset, respectively): The alleles in your dataset should match that of 1000G (*e.g.*, A G A G, C T C T, etc). If you observe that a lot of the alleles in your dataset do not match that of 1000G, revisit your input files since they may not be in the forward genomic orientation. If that is the case, recode your genotypes and go back to the beginning of the pipeline (2.4.1).
4) Columns 17-18 (ID in cM and Mbp, respectively): The only row that should have zeroes is the first row. If there are values ≤0 for these columns in the other markers, it is possible that the physical position of those markers in the dbSNP build that you are using is different from that of 1000G and this resulted to a change in the order of markers on your map. Include these markers in your exclusion file (line 6) and rerun marker_subpanels.pl.

2.4.3. Application 3: Pedigree Structure Validation

This application is composed of three steps: (1) pedigree-based kinship calculation, (2) genotype-based kinship estimation, and (3) relationship or sample error detection. The following scripts are used for this application and are further discussed below:

1) generation_numbers.pl
2) kped.pl
3) kstat.pl
4) relationship_check.pl

2.4.3.1. Pedigree-based Kinship Calculation

This step uses two scripts, namely: generation_numbers.pl and kped.pl. The script generation_numbers.pl determines the generation numbers for individuals in a pedigree while kped.pl subsequently determines pairwise kinship coefficients based on the pedigree structure. First, run generation_numbers.pl by typing

./generation_numbers.pl <pedigree file*> <[T|F]> <output directory*>**

*use absolute path
**specify whether input pedigree file has a header or not: header=[T|F]

2.4.3.1.1. Input File for generation_numbers.pl

This script needs one input file: a pedigree file. If you transposed your dataset by using transpose_fileset.pl, use its output pedigree file (*.tpedo) as your input pedigree file. Otherwise, use a pedigree file which has the format:

Family ID, Individual ID, Father ID, Mother ID, Sex

2.4.3.1.2. Output Files of generation_numbers.pl

The output and LOG files will be saved under the outdir that you specified and are shown in Table 7. The main output file (gen_num.out) is a pedigree file with generation numbers and has the format:

Family ID, Individual ID, Father ID, Mother ID, Sex, Generation Number

The second output file (pedigree.info) contains information for each of the pedigrees in the dataset and has the format:

Family ID, Number of Individuals, Number of Males, Number of Females, Number of Founders, Number of Non-founders, Number of Singletons, Maximum Number of Generations

This file is useful in letting you know if you have singletons in certain pedigrees so you can fix them first and start at the beginning of the pipeline (2.4.1). In cases where there are complex pedigrees whose generation numbers cannot be determined, a pedigree file containing these pedigrees will be in a third output file (gen_num.pnd).

The LOG file (gen_num.log) contains the following information:

1) parameters that you specified in the command line
2) location (absolute path) of input and output files
3) dataset summary
   a. total number of pedigrees
   b. total number of individuals
   c. total number of males
   d. total number of females
   e. total number of founders
   f. total number of non-founders
   g. total number of singletons
   h. highest generation number
   i. number of pedigrees skipped by kped.pl (PBAP does not handle super complex relationships)
4) individuals with more than one partner

5) date and time that you started running generation_numbers.pl
6) total real, system, and user times

Table 7. Output files of generation_numbers.pl

Filename (Designation)	Format/Description
gen_num.out (OUT)	Family ID, Individual ID, Father ID, Mother ID, Sex, Generation Number
pedigree.info (INFO)	Family ID, Number of Individuals, Number of Males, Number of Females, Number of Founders, Number of Non-founders, Number of Singletons, Maximum Number of Generations
gen_num.pnd (PND)*	Format same as that of a TPEDO file
gen_num.log (LOG)	Contains information about input files, output directory, dataset summary, individuals with more than one partner, date time that you started running generation_numbers.pl, total real, system, and user times

*created only when there are complex pedigrees whose generation numbers cannot be determined

After running generation_numbers.pl, use the second script, kped.pl to calculate the pedigree-based kinship coefficients. If there was no pedigree file without generation numbers (gen_num.pnd) created when you ran generation_numbers.pl, run kped.pl by typing

./kped.pl <pedigree file with generation numbers*> <[T|F]> <output directory*>**

*use absolute path
**specify whether input pedigree file has a header or not: header=[T|F]

If there was gen_num.pnd created when you ran generation_numbers.pl, run kped.pl by typing

./kped.pl <pedigree file with generation numbers*> <[T|F]> <pedigree file without generation numbers*> <[T|F]**> <output directory*>**

*use absolute path
**specify whether input pedigree file has a header or not: header=[T|F]

2.4.3.1.3. Input File(s) for kped.pl

Use the main output file of generation_numbers.pl (gen_num.out) as the input file for kped.pl. If you prefer to generate your own pedigree file with generation numbers, use an input file which has the format:

Family ID, Individual ID, Father ID, Mother ID, Sex, Generation Number

as specified in Table 7. A second optional input file, which is a pedigree file without pedigree numbers for complex pedigrees (gen_num.pnd) may also be needed.

2.4.3.1.4. Output Files of kped.pl

The output and LOG files will be saved under the outdir that you specified and are shown in Table 8. The main output file (kped.out) has the format:

Individual 1, Individual 2, Kinship Coefficient, Relationship

Codes/symbols used for the different relationships are as follows:

1) PO: parent-offspring
2) FS: first cousins

3) HS: half-siblings
4) AV: avuncular
5) GG: grandparent-grandchild
6) DFC: double first cousins
7) FC: first cousins
8) GAV: grand avuncular
9) GGG: great-grandparent-great-grandchild
10) FC1R: first cousin once removed
11) DR: distant relatives
12) CX: complex relationship
13) U: unrelated
14) ND: not determined

A relationship that is not as close as FC1R is no longer determined and just classified as distant relatives (DR). If a relationship is too complicated to determine (*e.g.*, cross generational marriages are present), kped.pl classifies this as a complex relationship (CX). For families whose generation numbers were not determined (*i.e.*, skipped by generation_numbers.pl), all relationship pairs except PO and U are not determined (ND).

An error file (kped.err) is created if there are errors in the input pedigree file, *e.g.*, a specific pedigree has more than one component since certain individuals are not correctly connected. If this file is created, fix your pedigree file and start at the beginning of the pipeline (2.4.1).

The LOG file (kped.log) contains the following information:

1) parameters that you specified in the command line
2) location (absolute path) of input and output files
3) specifies whether all components or specific pairs are checked for each of the pedigrees
4) date and time that you started running kped.pl
   total real, system, and user times

Table 8. Output files of kped.pl

Filename (Designation)	Format/Description
kped.out (OUT)	Individual 1, Individual 2, Kinship Coefficient, Relationship
kped.err (ERR)*	Specified pedigrees with more than one component
kped.log (LOG)	Contains information about input files, output directory, component checked, date time that you started running kped.pl, total real, system, and user times

*created only when there are errors in the pedigree file

2.4.3.2. Genotype-based Kinship Estimation

This step uses the script kstat.pl, which recodes genotype files and executes a likelihood-based estimator (*i.e.*, kstat) for computing genotype-based kinship coefficients ($\hat{\varphi}$) and identity by descent probabilities of sharing one allele ($\hat{k}_1$) (Choi *et al.*, 2009). Run kstat.pl by typing

./kstat.pl <parameter file*>

*use absolute path

The parameter file for kstat.pl should contain the entries shown in Table 9. If you have selected marker subpanels for pedigree structure validation (2.4.2, Table 5 line 7), use two of the four output files of marker_subpanels.pl, *i.e.*, TGEN and TIND, in each of the chromosomes as your input files this step.

Table 9. Parameter file for kstat.pl

Line #	Description
1	Chromosome(s) that have genotype data[a]
2	Directory for output files[b]
3	Additional codes for missing data[c]
4	Directory containing both kstat and kinship.pl (*e.g.*, /home/username/pbap_v1.00/essentials)[d] and the operating system bit count[e]
5	Directory containing pedigree files[f]
6	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where pedigree files are located[g,h]
7	[(prefix=) (suffix=)] of chromosome number in pedigree filenames[g]
8	Specify whether input pedigree file have headers or not: header=[T|F]
9	Directory containing genotype files[f]
10	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where genotype files are located[g,h]
11	[(prefix=) (suffix=)] of chromosome number in genotype filenames[g]
12	Specify whether input genotype files have headers or not: header=[T|F]
13	Option: [1|2] for input file(s) containing IDs of genotyped individuals[i]
14	Input file(s) with IDs of genotyped individuals[j]
15	[(prefix=) (suffix)] of chromosome number in names of subdirectories where files with IDs of genotyped individuals are located[g,h]
16	[(prefix=) (suffix)] of chromosome number in files with IDs of genotyped individuals[g]
17	Specify whether input files with IDs of genotyped individuals have headers or not: header=[T|F]

[a]It would be best to use genotype data of selected markers from all 22 chromosomes. For example, if there is genotype data for chromosomes 1-22, put '1-22' or if you only have genotype data only for chromosome 3, put '3'

[b]Use absolute path

[c]Current PBAP codes for missing data: -0-, MISSING, MIS, miss, NA, -9, -1, 0, and – so if you don't have additional codes, put 'no_extra_code'

[d]kstat is a C program written by Yoonha Choi (Choi *et al.*, 2009). The Perl script kinship.pl was also written by Yoonha Choi to process the output file of kstat

[e]Put '32' for 32-bit and '64' for 64-bit since we included both kstat32 and kstat64 in the essentials folder. For example, if the scripts are under /home/username/pbap_v1.00/essentials and you are using a 64-bit computer, put '/home/username/pbap_v1.00/essentials 64'

[f]If there is a subdirectory for each chromosome, do not include it here

[g]Indicate the portion of the filename that precedes (prefix) and succeeds (suffix) the chromosome number (*e.g.*, if chromosome subdirectory is chr22, then the chromosome number is '22' so put 'prefix=chr suffix=none'; if filename is chr22.tgen, put 'prefix=chr suffix=.tgen' for this line).

[h]For subdirectories in lines 6, 10, and 15, if files at lines 7 11, and 16, are directly under the directories in lines 5, 9, and 14, respectively, put 'no dir'

[i]Option 1: Use output files of transpose_fileset.pl or marker_subpanels.pl. Option 2: Use a file containing the list of genotyped individuals (supplied by user). If you chose option 2, line 14 should contain the absolute path of input file and leave lines 15-16 blank.

[j]If option 1 was chosen in line 13, indicate directory containing files with IDs of genotyped individuals[f]. If option 2 was chosen in line 13, indicate input file with list of IDs of genotyped individuals[b].

2.4.3.2.1. Input Files

Lines 5-17 of the parameter file focus on the input files, namely:

1) pedigree files (lines 5-8)
2) genotype files (lines 9-12)
3) list of genotyped individuals (lines 13-17)

PBAP requires a breakdown of the location and filenames to allow use of only one parameter file for the entire genome. Genotypes from several chromosomes up to all the autosomes are needed to have a better estimate of the genotype-based kinship coefficients. More genotypes mean more information that result in better approximation.

We recommend a minimum of ~20 families **and** ~100 founders in your dataset and select markers using marker_subpanels.pl (opt for creating subpanel for pedigree structure validation in line 7 of the marker subpanel parameter file (Table 5)). Using a dataset with less than ~20 families and ~100 founders may still allow you to detect sample swaps during the relationship checking step (especially when it involves a parent-offspring pair) but the estimated genotype-based kinship coefficients may not be that close to the expected kinship coefficients. The markers should span

the 22 chromosomes (or at least 3 chromosomes) to achieve a better estimate of the kinship coefficients.

Note: Before running kstat, make sure that the files containing the list of genotyped individuals (TIND) are exactly the same across all chromosomes that you will include in estimating the genotype-based kinship coefficients.

2.4.3.2.2. Specifications

Line 4 allows you to specify the location of kstat and kinship.pl, which are currently released under the essentials folder, as well as the operating system bit count of your computer (format: space-delimited). We included both kstat32 and kstat64 in the essentials folder so put '32' for 32-bit and '64' for 64-bit. For example, if the scripts are under /home/username/pbap_v1.00/essentials and you are using a 64-bit computer, put '/home/username/pbap_v1.00/essentials 64'.

2.4.3.2.3. Output Files

All output folders and files will be saved under the outdir that you specified and are shown in Table 10. The main output file (kstat.out) has the format:

Individual 1, Individual 2, $\hat{k}_0$, $\hat{k}_1$, $\hat{k}_2$, Kinship Coefficient ($\hat{\varphi}$)

The LOG file (kstat.log) contains the following information:

1) parameters that you specified in the parameter file
2) location (absolute path) of input and output files
3) number of individuals and markers genotyped for each pedigree
4) total number of individuals and total number of markers
5) markers that were excluded for having more than two unique alleles
6) date and time that you started running kstat.pl
7) total real, system, and user times

The auxiliary/intermediate files will be under /outdir/geno and outdir/kstat. For each pedigree in the dataset, PBAP recodes the genotypes into kstat-compatible format in which genotypes 11, 12, and 22 are coded as 0, 1 and 2, respectively. Missing genotypes are recoded as -1. These are saved in /outdir/ped*_kstat.geno. Genotypes from all pedigrees are combined and saved in one file (/outdir/allped_kstat.geno), and kstat (Choi *et al.*, 2009) is executed, which produces an output file (/outdir/kstat.tmp). kstat is a C program originally written by Yoonha Choi (Choi *et al.*, 2009).

The header of kstat.tmp is replaced (/outdir/kstat/kstat.txt) and used as input file for kinship.pl (written by Yoonha Choi), which produces /outdir/kstat/kc.txt. Finally, kstat.pl combines information from kstat.txt and kc.txt to create the main output file (kstat.out).

Table 10. Output files of kstat.pl

Directory	Filename (Designation)	Format (Description)
/outdir	kstat.out (OUT) kstat.log (LOG)	Individual 1, Individual 2, $\hat{k}_0$, $\hat{k}_1$, $\hat{k}_2$, Kinship Coefficient ($\hat{\varphi}$)

		(Contains information about numbers of individuals and markers in the dataset, parameters specified, input files, output directory, markers excluded for having more than two unique alleles, date and time that you started running kstat.pl, and total real, system, and user times)
/outdir/geno	ped*_kstat.geno (GENO) allped_kstat.geno (GENO)	Individual ID, Genotypes Individual ID, Genotypes
/outdir/kstat	kstat.tmp (TMP) kstat.txt (TXT) kc.txt (TXT)	ID1, ID2, $\hat{k}_0$, $\hat{k}_1$, $\hat{k}_2$ (Output of kstat (Choi *et al.*, 2009)) Individual 1, Individual 2, $\hat{k}_0$, $\hat{k}_1$, $\hat{k}_2$ (Changed header before using kinship.pl) Subject 1 (sub_1), Subject 2 (sub_2), Kinship Coefficient ($\hat{\varphi}$) (kin) (Output of kinship.pl)

2.4.3.3. Relationship or Sample Error Detection

This step uses the script relationship_check.pl, which compares pedigree-based kinship coefficients with the genotype-based kinship coefficients by using the output files of kped.pl (2.4.3.1) and kstat.pl (2.4.3.2). Run relationship_check.pl by typing

./relationship_check.pl <kped.pl output file*> <kstat.pl output file*> <number of markers> <[80|90|95|99|99.5]> <essentials directory*> <output directory*>**

*use absolute path
**percent confidence interval (%CI)

For the other arguments in the command line, here are some tips or comments:

1) Number of markers

   Obtain or verify the total number of markers above from kstat.log (2.4.3.2.3).

2) Percent confidence interval (% CI)

   Lower and higher values of % CI are for stringent and relatively loose checks, respectively. Since we intend to use relationship_check.pl for detecting possible sample swaps and duplicates within close relatives, choosing either 99% CI or 99.5% CI would be appropriate.

3) Essentials directory

   The essentials directory contains files (curve_fit.*) that are used by relationship_check.pl to determine the boundaries of the confidence intervals.

2.4.3.3.1. Input File for relationship_check.pl

This script uses the output files of kped.pl (2.4.3.1) and kstat.pl (2.4.3.2) as input files. The output of kped.pl has the format:

Individual 1, Individual 2, Kinship Coefficient, Relationship

While the output of kstat.pl has the format:

Individual 1, Individual 2, $\hat{k}_0$, $\hat{k}_1$, $\hat{k}_2$, Kinship Coefficient ($\hat{\varphi}$)

2.4.3.3.2. Output Files of relationship_check.pl

The output and LOG files will be saved under the outdir that you specified and are shown in Table 11. The main output file (relcheck_*.out) is a file containing pairs of individuals flagged for potential relationship errors and has the format:

Individual 1, Individual 2, Known Relationship, ExpKC, ExpMinKC, ExpMaxKC, EstKC, Expk1, ExpMink1, ExpMaxk1, Estk1, Reason(s) for Flagging

where

* = user-specified % CI
ExpKC = expected φ (based on pedigree structure)
ExpMinKC = minimum value of φ at user-specified % CI
ExpMaxKC = maximum value of φ at user-specified % CI
EstKC = estimated φ based on genotype data
Expk1 = expected k_1 (based on pedigree structure)
ExpMink1 = minimum value of k_1 at user-specified % CI
ExpMaxk1 = maximum value of k_1 at user-specified % CI
Estk1 = estimated k_1 based on genotype data

Only the relationships with $\varphi \geq 0.03$ are reported in the output file. For $\varphi < 0.03$, kstat (Choi *et al.*, 2009) and other estimators are not very reliable in estimating the kinship coefficients from the genotype data (not because of the limitations of these software but because of a limitation of the approach itself) so we discourage use of differences in kinship coefficients to detect Mendelian inconsistencies between more distant relatives.

The second and third output files (relcheck_*.pss and relcheck_*.puip) have the format:

Individual 1, Individual 2, Known Relationship, ExpKC, EstKC, Expk1, Estk1, Estk0, Estk2

The extension *.pss (PSS) stands for "**p**otential **s**ample **s**waps." If the PSS file was generated, you will see pairs of individuals between two pedigrees that are supposed to be unrelated, but have EstKC >= 0.06 and are therefore potentially related. The extension *.puip (PUIP) stands for "**p**otentially **u**nrelated **i**ndividuals in the **p**edigree." If the PUIP file was generated, you will see pairs of individuals within the same pedigree that are supposed to be related, but have EstKC <= 0.015625 and are therefore potentially unrelated.

The fourth output file (allpairwise_*.txt), which is an intermediate file that contains all expected and estimated pairwise φ and k_1, has the format:

Individual 1, Individual 2, Known Relationship, ExpKC, EstKC, Expk1, Estk1

The LOG file (relcheck_*.log) contains the following information:

1) location (absolute path) of input and output files
2) list of pairs of individuals whose relationships were not checked due to complexity
3) date and time that you started running relationship_check.pl

4) total real, system, and user times

Table 11. Output files of relationship_check.pl

Filename (Designation)	Format/Description
relcheck_*.out (OUT)	Individual 1, Individual 2, Known Relationship, ExpKC, ExpMinKC, ExpMaxKC, EstKC, Expk1, ExpMink1, ExpMaxk1, Estk1, Reason(s) for Flagging
relcheck_*.pss (PSS)	Individual 1, Individual 2, Known Relationship, ExpKC, EstKC, Expk1, Estk1, Estk0, Estk2
relcheck_*.puip (PUIP)	Individual 1, Individual 2, Known Relationship, ExpKC, EstKC, Expk1, Estk1, Estk0, Estk2
allpairwise_*.txt (TXT)	Individual 1, Individual 2, Known Relationship, ExpKC, EstKC, Expk1, Estk1
relcheck_*.log (LOG)	Contains information about input files, output directory, list of pairs of individuals whose relationships were not checked due to complexity, date time that you started running relationship_check.pl, total real, system, and user times

*percent confidence interval (% CI)

2.4.4. Application 4: Preparation of Files for MORGAN

This application uses the setup_gl_auto.pl, which prepares file for gl_auto of the MORGAN package (Thompson, 2011). Setup_gl_auto.pl uses the following parameters:

1) Maximum number of meioses for exact computation
2) Total number of IBD graphs per component for exact computation
3) Total number of sequential imputation realizations for setup
4) Total number of Monte Carlo (MC) iterations
5) Percentage of MC iterations for burn-in
6) L-sampler probability
7) Output score every nth scored MC iteration
8) Main population (*i.e.*, AFR, AMR, ASN, or EUR)
9) Source of MAF information
10) Marker type for first (and possibly a second) set of genotype data

Run setup_gl_auto.pl by typing

./setup_gl_auto.pl <chromosome> <marker subpanel number> <parameter file*> <family ID>**

*use absolute path
**optional: specify the family ID if you want to execute setup_gl_auto.pl for one specific family

The parameter file for setup_gl_auto.pl should contain the entries shown in Table 12.

Table 12. Parameter file for setup_gl_auto.pl

Line #	Description
1	Chromosome(s) that have genotype data[a]
2	Directory containing setup_gl_auto.pl[b]
3	Directory for output files[b,c]
4	Split setup_gl_auto output files by pedigree: [Y|N][d]
5	Family ID exclusion file[a,e,f]
6	Genotype exclusion file[a,e,f,g]
7	Maximum number of meioses for exact computation[h]
8	Total number of IBD graphs per component for exact computation[i]
9	Total number of sequential imputation realizations for setup[j]
10	Total number of Monte Carlo (MC) iterations[k]
11	Percentage (%) of MC iterations for burn-in[l]
12	L-sampler probability[m]
13	Output score every nth scored MC iterations[n]

14	Population: [AFR|AMR|ASN|EUR]
15	Source of minor allele frequency (MAF) information: [dataset|1KG|MAF][o]
16	Directory containing MAF files[p]
17	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where MAF files are located[q,r]
18	[(prefix=) (suffix=)] of chromosome number in MAF filenames[q]
19	Specify whether input MAF files have headers or not: header=[T|F]
20	Directory containing map files[p]
21	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where map files are located[q,r]
22	[(prefix=) (suffix=)] of chromosome number in map filenames[q]
23	Specify whether input map files have headers or not: header=[T|F]
24	Directory containing pedigree files[p]
25	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where pedigree files are located[q,r]
26	[(prefix=) (suffix=)] of chromosome number in pedigree filenames[q]
27	Specify whether input pedigree file have headers or not: header=[T|F]
28	Number of types of markers with genotype data: [1|2][s]
29	Marker type for first set of genotype data: [SNP|STR]
30	Directory containing genotype files[p]
31	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where genotype files are located[q,r]
32	[(prefix=) (suffix=)] of chromosome number in genotype filenames[q]
33	Specify whether input genotype files have headers or not: header=[T|F]
34	Option: [1|2] for input file(s) containing IDs of genotyped individuals[t]
35	Input file(s) with IDs of genotyped individuals[u]
36	[(prefix=) (suffix)] of chromosome number in names of subdirectories where files with IDs of genotyped individuals are located[q,r]
37	[(prefix=) (suffix)] of chromosome number in files with IDs of genotyped individuals[q]
38	Specify whether input files with IDs of genotyped individuals have headers or not: header=[T|F]
39	Marker type for second set of genotype data: [SNP|STR]
40	Directory containing genotype files[p]
41	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where genotype files are located[q,r]
42	[(prefix=) (suffix=)] of chromosome number in genotype filenames[q]
43	Specify whether input genotype files have headers or not: header=[T|F]
44	Option: [1|2] for input file(s) containing IDs of genotyped individuals[t]
45	Input file(s) with IDs of genotyped individuals[u]
46	[(prefix=) (suffix)] of chromosome number in names of subdirectories where files with IDs of genotyped individuals are located[q,r]
47	[(prefix=) (suffix)] of chromosome number in files with IDs of genotyped individuals[q]
48	Specify whether input files with IDs of genotyped individuals have headers or not: header=[T|F]

[a]Allows PBAP to check whether list of IDs of genotyped individuals are the same among these chromosomes. If there is genotype data for chromosomes 1-22, put '1-22', if you only have genotype data only for chromosome 3, put '3', or if you only have genotype data only for chromosomes 3 and 7, put '3 7'

[b]Use absolute path

[c]Under this directory, subdirectories for marker subpanel number, family, and chromosome will be created

[d]Most of the downstream analyses perform well when ran per pedigree so it would be practical to split your data by pedigree (put 'Y'). However, if you want to have one huge file (which would not be parallelizable and hence, would require more time in downstream applications), put 'N'

[e]Use one file that applies to the entire dataset. Remove any header(s).

[f]Put 'none' if you don't have this input file

[g]Format (PBAP tpedo file, space-delimited): Family ID, Individual ID, Father ID, Mother ID, Sex

[h]Recommended values: 10 through 14, which can be increased up to 20 (depending on the processor, memory, and system bit)

[i]This will be maximum number of IBD graphs that will be saved for each component. Usual value: 1000. Depending on pedigree size and structure, you may need to increase this value

[j]Recommended values: for relatively complete data ('20'), for sparse data, *i.e.*, a lot of missing data ('50' or more)

[k]Recommended values: for test runs ('30000' or more), for regular runs ('100000' or more)

[l]Recommended values: for relatively complete data ('10'), for sparse data, *i.e.*, a lot of missing data ('11' or more)

[m]Recommended values: for test runs ('0.5'), for regular runs ('0.2')

[n]Quotient of line 10/line 13 should be an integer since this would be the number of sampled IVs. If you placed '50000' in line 10, and you want 1,000 sampled IVs, put '50' in this line (*i.e.*, line 10/line 13 = 50,000 total MC iterations/50 = 1,000 sampled IVs)

[o]If you want to use genotypes of the dataset (minimum of ~20 families AND ~100 founders) to calculate the allele frequencies, put 'dataset' and put 'none' in lines 16-19. If you want to use MAFs of unrelateds from 1000G obtained from running marker subpanels.pl (2.4.2), put '1KG'. Check your subpanels to make sure that allele 1 (minor allele) of your dataset matches allele 1 of 1000G. If you want to use your own MAF file, put 'MAF' and use the format (space-delimited): SNP, Allele 1, Allele 2, Allele Frequencies of Alleles 1 to n

[p]If there is a subdirectory for each chromosome, do not include it here

[q]Indicate the portion of the filename that precedes (prefix) and succeeds (suffix) the chromosome number (*e.g.*, if chromosome subdirectory is chr22, then the chromosome number is '22' so put 'prefix=chr suffix=none'; if filename is chr22.tmap, put 'prefix=chr suffix=.tmap' for this line).

[r]For subdirectories in lines 17, 21, 25, 31, 36, 41, and 47, if files in lines 18, 22, 26, 32, 37, 42, and 48 are directly under the directories in lines 16, 20, 24, 31, 35, 40, and 46, respectively, put 'no dir'

[s]If you have SNPs or STRs but not both, put '1' and leave lines 39-48 blank. If you have both SNPs and STRs, put '2'.

[t]Option 1: Use output files of transpose_fileset.pl. Option 2: Use a file containing the list of genotyped individuals (supplied by user). If you chose option 2, lines 34 (and 45, if applicable) should contain the absolute path of input file and leave lines 36-38 (and lines 46-48, if applicable) blank, respectively.

[u]If option 1 was chosen in line 34 (and 45, if applicable), indicate directory containing files with IDs of genotyped individuals[p]. If option 2 was chosen in line 34 (and 45, if applicable), indicate input file with list of IDs of genotyped individuals[b].

2.4.4.1. Input Files

Lines 5-6, 16-27, 30-33, 35-38, 40-43, and 40-48 of the parameter file focus on the input files, namely:

1) exclusion files (lines 5-6)
2) MAF files (lines 16-19)
3) map files (lines 20-23)
4) pedigree files (lines 24-27)
5) genotype files (lines 30-33 and lines 40-43)
6) list of genotyped individuals (lines 35-38 and lines 45-48)

For the exclusion files (and if preferred for the list of genotyped individuals), simply specify the absolute path. For all other files, PBAP requires a breakdown of the location and filenames to allow use of only one parameter file for the entire genome. The family ID exclusion file (line 5) is useful in excluding families with only 3 genotyped individuals and/or when they are trios while the genotype exclusion file (line 6) is useful in excluding genotypes of certain individuals based on the results from pedigree structure validation (2.4.3) specifically after relationship or sample error detection (2.4.3.3) was performed. For the map files, specify the location of the main PMAP output file of marker_subpanels.pl. For the genotype files and list of genotyped individuals, you will notice that there are two sets to allow merging two different datasets (or two types of markers (SNPs or STRs). **If you only have one set of genotype data, leave lines 39-48 blank.**

2.4.4.2. Options and Specifications

Lines 4, 7-15, 28-30, 34, 39-40, and 44 of the parameter file focus on the different options and specifications for the user. For lines that require values, do not put the units (*e.g.*, meioses, iterations, etc).

Line 4 allows the user to split the input files by pedigree. Since most of the downstream analyses perform well when ran per pedigree, it would be practical to split your data by pedigree, so put 'Y'. However, if you want to have one huge file (which would not be parallelizable and hence, would require more time in downstream applications), put 'N'.

Lines 7-8 allow the user to specify parameters for exact computation and the recommended values are specified in Table 12 footnotes h and i. For line 7, specify the maximum number of meioses within which gl_auto will use exact computation instead of MCMC. You may specify any value from 10 through 14 although this may be increased up to 20 depending on the processor, memory, and system bit of your computer. For line 8, specify the maximum number of IBD graphs that will be saved for each pedigree component when exact computation is used. MORGAN uses the term "component" for each connected pedigree (*i.e.*, distinct family) in a pedigree file (http://faculty.washington.edu/eathomp/Anonftp/PANGAEA/MORGAN/morgan3-tut/morgan-tut_V33.html) (Thompson, 2011). Although the usual value used is '1000', the maximum number of IBD graphs do not have to be the same as the total number of sampled IVs when MCMC is used (which involves parameters specified in lines 10 and 13 below). It will entirely depend on the user to select the number of IBD graphs to be saved based on the pedigree size and structure. For more complex pedigrees, use a higher value.

By default, PBAP includes the phrase "use sequential imputation for setup" in the parameter file for gl_auto. Sequential imputation (with the trait treated as unlinked) is the default method used by gl_auto to find a starting configuration for the meiosis indicators prior to MCMC. Line 9 allows the user to specify the number of sequential imputation realizations that will be used for setup. For a relatively complete data, put '20' which is the default value used by gl_auto. However, if you have a lot of missing data (*i.e.*, you have sparse data), use a higher value, *e.g.*, '25' up to around '50', where 50 is considerably a huge number for this parameter (Table 12 footnote j). Using a much higher value (*e.g.*, '100' or '200') would not make much difference.

Lines 10-13 allow the user to specify parameters for the MC iterations and the recommended values are specified in Table 12 footnotes k through n. In line 10, we recommend a total of 30,000 MC iterations for test runs, although sometimes we use 50,000 MC iterations. In these cases, put '30000' or '50000'. For regular runs, use ≥100,000 MC iterations, so put '100000' or more. In line 11, if you have relatively complete data, we recommend 10% of MC iterations for burn-in, so put '10'. If you have sparse data, use ≥11%, so put '11' or more. In line 12, the probability (between 0.0 and 1.0) of using the locus sampler (L-sampler) instead of the meiosis sampler (M-sampler) in each MCMC iteration is specified by the user. By default, gl_auto uses 0.0 (*i.e.*, use M-sampler only). For test runs, we recommend '0.5' while for regular runs, you may put a lower value, *e.g.*, '0.2'. In line 13, specify the nth scored MC iteration that will be saved by gl_auto. The quotient of line 10/line 13 should be an integer since this would be the number of sampled IVs in your output file. For example, if you placed '100000' in line 10, and you want 1,000 sampled IVs, put '100' in this line (*i.e.*, line 10/line 13 = 100,000 total MC iterations/100 = 1,000 sampled IVs).

Specify the main population in line 14. The pre-processed 1000G files (2.3.2.2) are divided into four main populations, namely: (1) African (AFR), Admixed American (AMR), East Asian (ASN), and European (EUR).

Line 15 allows the user to specify the source of the MAF information: dataset, 1KG or MAF. Put 'dataset' in line 15 and 'none' in lines 16 through 19 if you want PBAP to use the genotypes of the dataset to calculate the allele frequencies. For this option, we recommend a minimum of ~20 families and ~100 founders in your dataset. If you want to use MAFs of unrelateds from 1000G obtained from running marker subpanels.pl (2.4.2), put '1KG'. Check your subpanels to make sure that allele 1 (minor allele) of your dataset matches allele 1 of 1000G. If you want to use your own MAF file, put 'MAF' and use the format (space-delimited):

SNP, Allele 1, Allele 2, Allele Frequencies of Alleles 1 to n

For line 28, specify the number of types of markers with genotype data. If you have SNPs or STRs but not both, put '1' and leave lines 39-48 blank. If you have both SNPs and STRs or if you have two datasets, put '2' and make sure that your map files (lines 20-23) is the union of markers from both datasets.

For lines 29 and 39, specify the marker type: SNP or STR.

For lines 34 and 44, put '1' if you want to use the output files of transpose_fileset.pl (2.4.1) or put '2' if you want to specify a file which contains the list of individuals with genotype data.

2.4.4.3. Output Files

All output folders and files we be saved under the outdir that you specified. Under this directory, folders for the marker subpanel number, main population, pedigree, and chromosome that you ran will be created. For example, if the marker subpanel number, main population, family, and chromosome are 1, EUR, AA, and 22, respectively, the output files will be under /outdir/panel1/EUR/pedAA/chr22. If you didn't split by pedigree, the pedigree subdirectory won't be created and the chromosome directory will be directly under the main population subdirectory, *i.e.*, /outdir/panel1/EUR/chr22. If your input pedigree file has several families, and you opted to split by pedigree (line 4 of parameter file), all the ped* subdirectories will be under /outdir/panel1/EUR with the info_tind subdirectory, which contains the chr*.tind file used. The main LOG file (*e.g.*, setup_gl_auto_panel1_EUR_chr22.log) will be under /outdir/panel1/log.

The main output files are the input files for gl_auto (filenames given are for the example above), namely: (1) chr22.geno (GENO), chr22.glauto.par (PAR), and pedAA.txt (TXT). All output files, except for the LOG files, have no headers and are described in Table 13.

Table 13. Output files of setup_gl_auto.pl

Directory	Filename (Designation)	Description and/or Format
/outdir/panel1/log	setup_gl_auto_panel1_EUR_chr22.log (LOG)	Contains the parameters specified, input files, output directory, individuals whose genotypes were excluded (based on the genotype exclusion file), families excluded (based on family ID exclusion file or if there are <3 genotyped individuals in the family), date and time that you started running setup_gl_auto.pl, and family-specific or total real, system, and user times
/outdir/panel1/EUR/pedAA/chr22	chr22.geno (GENO)	MORGAN-format genotype file which has three parts (space-delimited): Part 1: Marker positions 'map marker positions' followed by the genetic locations of the markers in the marker subpanel Part 2: Allele frequencies having 1 row per marker in the same order as that of the marker subpanel used and marker names are replaced with marker numbers from 1 to n 'set markers', Marker Number, 'allele freq', AF1, AF2 Part 3: Genotype data 'set markers', Last Marker Number, 'data' (first row) Individual ID, Genotypes (succeeding rows)
	pedAA.txt (TXT)	MORGAN-format pedigree file which contains three header lines (space-delimited) as follows: 'input pedigree size', Pedigree Size 'input pedigree record names 3 integer', Number of Additional Columns (usually 2) 'input pedigree record father mother' The header lines are succeeded by the main content of the pedigree file (trio information) and has the format (space-delimited): Individual ID, Father ID, Mother ID, Sex, Phenotype
	chr22.glauto.par (PAR)	MORGAN-format parameter file for gl_auto which contains the absolute paths of the input and output files as well as the different parameters for gl_auto
	pedAA.log (LOG)	Pedigree-specific LOG file which contains the information generated when sex of individuals in the pedigree are being checked to make sure that the format (Father ID is followed by the Mother ID) is being implemented correctly.
	chr22.mnum	Marker number file which has the format: Chromosome, rsID, Genetic Location, Physical Position, MAF_{1000G}, $MAF_{dataset}$, $A1_{1000G}$, $A2_{1000G}$, $A1_{dataset}$, $A2_{dataset}$, Marker Number in Subpanel, Marker Number in MORGAN-format Genotype file, Remarks

	chr22.arec	Chromosome, rsID, Genetic Location, Physical Position, $A1_{1000G}$, $A2_{1000G}$, First Allele Seen, Recode Value 1, Second Allele Seen, Recode Value 2
	chr22.si	Chromosome, rsID, Genetic Location, Physical Position, $A1_{1000G}$, $A2_{1000G}$, $A1_{dataset}$, $A2_{dataset}$

AF1: allele frequency 1
AF2: allele frequency 2
A1: Allele 1
A2: Allele 2
1000G: 1000 Genomes Project data (Altshuler *et al.*, 2010)

After you have finished running setup_gl_auto.pl, examine (spot check) the following items in your main output files before going to the next application (2.4.5):

1) Genotype file (*e.g.*, chr22.geno (GENO)) should contain the three parts described in Table 13, *i.e.*, marker positions, allele frequencies, and genotype data. For the allele frequencies, if you used the option '1KG' in line 15 of the parameter file, the frequency of allele 1 (AF1) of a certain marker should match MAF of that marker in the marker subpanel PMAP file.
2) Pedigree file (*e.g.*, pedAA.txt (TXT)) should have at least 5 columns of a regular pedigree file (without the Family ID) and has 3 header lines.
3) Parameter file (*e.g.*, chr22.glauto.par (PAR)) should indicate the correct absolute paths for the input and output files and also show the important parameters for gl_auto (*e.g.*, 'use multiple meiosis sampler', maximum number of meioses for exact computation, number of sequential imputation realizations for setup, etc).
4) Marker number file (chr22.mnum (MNUM)) should indicate the marker numbers in both the marker subpanel and in the MORGAN-format genotype file (*e.g.*, chr22.geno). This file is very useful when there are markers excluded while setting up files for gl_auto due to strand inconsistency or missing MAF.
5) Allele recoding file (chr22.arec (AREC)) should indicate the recoded alleles of markers present in the MORGAN-format genotype file (*e.g.*, chr22.geno). Check whether the allele 1 of the markers in the MORGAN-format genotype file corresponds to that of 1000G (Altshuler *et al.*, 2010). For a specific marker, there would be two unique alleles for an entire dataset that has several families. In most cases, there is at least one unique allele present for that specific marker in a particular family, so missing data (coded as "0") is not recoded (*i.e.*, remains as "0" in the MORGAN-format genotype file) and is not specified in the AREC file. In cases where there is no data for that specific marker in a particular family, setup_gl_auto.pl puts "0" under First Allele Seen and "0" under Recoded Value 1 in the AREC file. This makes it easier for the user to see which markers in the subpanel have two alleles in the entire dataset but are missing for a specific family within that dataset.
6) Strand inconsistency file (chr22.si (SI)) should indicate markers that have strand inconsistency which are also present in the MNUM file. This file will only be created if there are markers that have been excluded due to strand inconsistency.

2.4.5. Application 5: Execution of gl_auto to Sample IVs

This application uses the run_gl_auto.pl, which executes gl_auto of the MORGAN package (Thompson, 2011). Run_gl_auto.pl currently only uses one parameter, *i.e.*, main population (AFR, AMR, ASN, or EUR).

Run run_gl_auto.pl by typing

./run_gl_auto.pl <chromosome> <marker subpanel number> <parameter file*> <family ID>**

*use absolute path
**optional: specify the family ID if you want to execute run_gl_auto.pl for one specific family

The parameter file for run_gl_auto.pl should contain the entries shown in Table 14.

Table 14. Parameter file for run_gl_auto.pl

Line #	Description
1	Directory containing run_gl_auto.pl[a]
2	Output files of setup_gl_auto.pl were split by pedigree: [Y|N][b]
3	Path of gl_auto that will be used (*e.g.*, /usr/bin/gl_auto or /home/username/bin/gl_auto)
4	Population: [AFR|AMR|ASN|EUR]
5	Directory containing marker_subpanel.pl PMAP files[a]
6	Output directory specified in the parameter file for setup_gl_auto.pl[a,c]
7	Directory containing pedigree files[d]
8	[(prefix=) (suffix=)] of chromosome number in names of subdirectories where pedigree files are located[e,f]
9	[(prefix=) (suffix=)] of chromosome number in pedigree filenames[e]
10	Specify whether input pedigree file have headers or not: header=[T|F]

[a]Use absolute path
[b]If you split your data by pedigree in setup_gl_auto.pl, put 'Y'. To run gl_auto one family at a time (pedigree-specific) for all families, indicate the location of pedigree files used for setup_gl_auto.pl in lines 7-10. To run specific families, list the Family IDs (space-delimited) after 'Y', *e.g.*, 'Y Family1 Family2 Family3 ... FamilyN'. If you didn't split by pedigree, put 'N' and leave lines 7-10 blank.
[c]This should be the directory above the panel number, main population, family ID, and chromosome number and should be the same as the directory specified in Table 12 line3.
[d]If there is a subdirectory for each chromosome, do not include it here
[e]Indicate the portion of the filename that precedes (prefix) and succeeds (suffix) the chromosome number (*e.g.*, if chromosome subdirectory is chr22, then the chromosome number is '22' so put 'prefix=chr suffix=none'; if filename is chr22.tmap, put 'prefix=chr suffix=.tmap' for this line).
[f]For subdirectory in line 8, if files in line 9 are directly under the directory in line 7, put 'no dir'

2.4.5.1. Input Files

Lines 5-10 of the parameter file focus on the input files, namely:

1) marker subpanel (PMAP) files (line 5)
2) output files of setup_gl_auto.pl (line 6)
3) pedigree files (lines 7-10)

For the directories containing the marker subpanel PMAP files (2.4.2.3) and the output files of setup_gl_auto.pl (2.4.4.3), simply specify the absolute path. For the pedigree files, PBAP requires a breakdown of the location and filenames to allow use of only one parameter file for the entire genome.

2.4.5.2. Options and Specifications

Lines 2 and 4 of the parameter file focus on the two types of options for the user.

For line 2, if you split your data by pedigree in setup_gl_auto.pl, put 'Y'. To run gl_auto one family at a time (pedigree-specific) for all families, indicate the location of pedigree files used for setup_gl_auto.pl in lines 7-10. To run specific families, list the Family IDs (space-delimited) after 'Y', *e.g.*,

Y, Family 1, Family 2, Family 3, ..., Family N

If you didn't split by pedigree, put 'N' and leave lines 7-10 blank.

Specify the main population in line 4. The pre-processed 1000G files (2.3.2.2) are divided into four main populations, namely: (1) African (AFR), Admixed American (AMR), East Asian (ASN), and European (EUR).

2.4.5.3. Output Files

All output files will be saved under the outdir of setup_gl_auto.pl (line 6 of parameter file). Under this directory, folders for the marker subpanel number, main population, pedigree, and chromosome that you ran already exist. For example, if the marker subpanel number, main population, family, and chromosome are 1, EUR, AA, and 22, respectively, the output files will be under /outdir/panel1/EUR/pedAA/chr22 or /outdir/panel1/EUR/chr22 (the same as the examples in 2.4.4.3) depending on whether you split by pedigree or not. The LOG file (*e.g.*, run_gl_auto_panel1_EUR_pedAA_chr22.log) will be under /outdir/panel1/log. All output files, except for the LOG files, have no headers and are described in Table 15.

The main output files are the output files of gl_auto, namely: (1) founder genome label file, *e.g.*, chr22.fgl (FGL), (2) meiosis indicator file, *e.g.*, chr22.mi (MI), and sampler seed file, *e.g.*, chr22.sampler.seed (SEED). The vector of a meiosis indicator at a particular locus over all the meioses of a pedigree is known as the inheritance vector (IV) at that specific locus. PBAP's run_gl_auto.pl also saves the screen output of gl_auto, *e.g.*, chr22.glauto.out (OUT). In the same directory, you will also see the output files of setup_gl_auto.pl that were used as input files of gl_auto, namely: (1) chr22.geno (GENO), chr22.glauto.par (PAR), and pedAA.txt (TXT).

If there are Mendelian inconsistencies detected by gl_auto, run_gl_auto.pl zeroes out the genotypes for all individuals in the pedigree for that particular marker in a stepwise manner. First, the OUT and GENO files are backed up and saved as *.out1 (*e.g.*, chr22.glauto.out1) and *.geno.bak1 (*e.g.*, chr22.geno.bak1), respectively, and the information regarding inconsistent family data are saved in an error (ERR) file, *e.g.*, chr22.err1 which has the format (space-delimited):

> Error Number, Marker Number, rsID, Genetic Location (cM), Physical Position (bp), Father ID:[Genotype], Mother ID:[Genotype], Individual ID:[Genotype]

Second, run_gl_auto.pl uses this information to zero out the genotypes for all individuals in the pedigree for this particular marker and saves it as the new GENO file then executes gl_auto again. If there are Mendelian inconsistencies that are still present, OUT and GENO files are backed up again and saved as *.out2 (*e.g.*, chr22.glauto.out2) and *.geno.bak2 (*e.g.*, chr22.geno.bak2), respectively, and the same steps are performed and repeated up to maximum of 10 iterations until the FGL and MI files are finally generated. If gl_auto fails to generate the FGL and MI files, there is a high possibility that there is something wrong with your data or your files. Based on experience, it would be best to check your main input files and the output files of transpose_fileset.pl (2.4.1.3). If gl_auto has generated FGL and MI files then you now have your sampled IVs that may be used for downstream analyses.

Table 15. Output files of run_gl_auto.pl

Directory	Filename (Designation)	Format (Description)
/outdir/panel1/log	run_gl_auto_panel1_EUR_chr22.log (LOG)	(Contains the parameters specified (path of gl_auto, PMAP file, and output directory), error notes (if applicable), date and time that you started running run_gl_auto.pl, and family-specific or total real, system, and user times)
/outdir/panel1/EUR/pedAA/chr22	chr22.fgl (FGL)	Individual ID, FGL at beginning of chromosome, Marker Number at First Recombination, FGL (from this marker up to downstream markers), Marker Number at Second Recombination, FGL (from this marker up to downstream markers), and so on until the end of the chromosome (MORGAN-format founder genome label file)
	chr22.mi (MI)	Individual ID, Individual Index[a], Maternal (0) or Paternal (1) Meiosis, First Unlinked Marker[b], Meiosis Indicator at the First Marker, Number of Switches[c], Recombination Positions[d] (MORGAN-format meiosis indicator file)
	chr22.sampler.seed (SEED)	Seed generated by gl_auto for the sampler (hexadecimal)
	chr22.glauto.out (OUT)	Screen output of gl_auto
	chr22.glauto.out# (OUT)	Back-up of latest screen output of gl_auto
	chr22.geno.bak# (BAK)	Back-up of MORGAN-format genotype file
	chr22.geno.err# (ERR)	Error Number, Marker Number, rsID, Genetic Location (cM), Physical Position (bp), Father ID:[Genotype], Mother ID:[Genotype], Individual ID:[Genotype] (Error file generated when Mendelian inconsistencies are detected)

[a]Internal MORGAN 0-origin index of the individual
[b]Unlinked null locus
[c]Number of recombinations across the chromosome
[d]Marker numbers where recombinations occur. The number of integers correspond to the number of recombinations.
#: iteration number for each cycle performed by run_gl_auto.pl until FGL and MI files are generated by gl_auto. These files are only created when Mendelian inconsistencies are detected by gl_auto.

3. EXAMPLES

We are still in the process of creating example files for PBAP. They will be discussed in this chapter.

Table 16. Scripts Included in PBAP Release 1.0.

Directory	Script	Function / Use
pbap_v1.00/	transpose_fileset.pl marker_subpanels.pl generation_numbers.pl kped.pl kstat.pl relationship_check.pl setup_gl_auto.pl run_gl_auto.pl	
pbap_v1.00/essentials	kstat (Choi *et al.*, 2009) kinship.pl curve_fit.*	
pbap_v1.00/tools	exclude_indels_dups.pl	

4. FREQUENTLY ASKED QUESTIONS (FAQ)

4.1. Application 1: Transposition of Normal File Format

1) When would a phenotype file be necessary?

If you intend to delete the PLINK-format transposed files (tfam/tped) (Table 1, line 25) such that the phenotype column for the tped file is not important, you may choose any valid column number for line 10 in Table 1 from the phenotype file since the output tped file will be deleted anyway.

If you choose not to delete the PLINK-format transpose files (tfam/tped) and phenotype data is available, a phenotype column will be included in these tped/tfam files so you should select the appropriate column for your phenotype of interest. If phenotype data is missing, it will be replaced by "-9" in the PLINK-format transposed files. In PLINK, the missing phenotype value for quantitative traits is, by default, -9 (which can also be a number used for disease traits, just like 0). To avoid confusion, it can be recoded within PLINK by including the --missing-phenotype option: '--missing-phenotype NA'.

If you intend to run linkage analysis on the sampled IVs using gl_lods, you would need a phenotype file. Application 1 (transpose_fileset.pl) prepares a file compatible with gl_lods.

4.2. Application 2: Selection of Marker Subpanels

1) If I want to obtain a marker subpanel that I will use for linkage analysis, what is the minimum intermarker distance (MID) that you would recommend? Why?

To avoid or to minimize MCMC-mixing issues (Sieh *et al.*, 2005; Thompson and Heath, 1999; Wilcox *et al.*, 2005) when sampling IVs using gl_auto (2.4.5), which would be used for linkage analysis, we recommend an MID of 0.5 cM.

4.3. Application 3: Pedigree Structure Validation

1) Why do I need genotype data from several chromosomes when I run kstat.pl?

Genotypes from several chromosomes up to all the autosomes are needed to have a better estimate of the genotype-based kinship coefficients. More genotypes across several chromosomes mean more information result in better approximation. There may be more IBD sharing in one particular chromosome than the other chromosomes so focusing on one chromosome only may not result in good estimates of relatedness.

2) Is it really necessary for me to have exactly the same TIND files across chromosomes when I run kstat.pl, *i.e.*, the list of genotyped individuals for a particular dataset should be the same for all chromosomes?

Yes, this is to make sure that there would be complete genotype data for all markers in the marker subpanel used for kstat.pl and obtain a better estimate of the genotype-based kinship coefficients. If you prepared your own files instead of using PBAP's transpose_fileset.pl (2.4.1), make sure that you have placed your files in consistently-named folders across chromosomes to allow use of the same subdirectory name in the parameter file.

4.4. Application 4: Preparation of Files for MORGAN

1) Question here

Answer here

4.5. Application 5: Execution of gl_auto to Sample IVs

1) Question here

Answer here

4.6. General Questions

1) In your parameter files, why does PBAP require several items (directory, prefix, suffix, etc) for a particular input file instead of simply asking for the absolute path of the file?

PBAP requires a breakdown of the location and filenames to allow use of only one parameter file for the entire genome. This is very useful when you want to run all 22 chromosomes at the same time in a cluster without the need to create 22 separate parameter files.

2) In the parameter file, why do I have to indicate whether an input file has header or not?

Aside from other details, we included this in the parameter file to keep everything modular. This means that you can use the scripts separately as long as you have the correct set and format of input files needed and it won't really matter whether your file has a header or not since you can indicate it in the parameter file. However, in most of the latter parts of the pipeline, the output files produced by PBAP have no header to facilitate concatenation of files from different chromosomes. This may change in some of the output files where it would be better to have headers.

3) If I discover a bug, what should I do?

Inform us (wijsman at uw dot edu or aqnato at uw dot edu) immediately so that we can fix it as soon as we can.

5. LICENSE

PBAP Suite of Programs is free. You may modify it or redistribute it.

6. ACKNOWLEDGEMENTS

We acknowledge inclusion of two external programs in PBAP, namely: kstat and kinship.pl (both written by Yoonha Choi). We also gratefully thank discussions with Mohamad Saad and Sulgi Kim.

7. FUNDING

This work was supported by the National Institutes of Health (R01 MH092367, R01 MH094293, R01 AG039700, R37 GM046255, P50 AG005136, and U01 AG016976).

8. REFERENCES

Altshuler, D., Durbin, R.M., Abecasis, G.R., Bentley, D.R., Chakravarti, A., Clark, A.G. *et al.* (2010) A map of human genome variation from population-scale sequencing, *Nature*, **467**, 1061-1073.

Cheung, C.Y.K., Thompson, E.A. and Wijsman, E.M. (2013) GIGI: An Approach to Effective Imputation of Dense Genotypes on Large Pedigrees, *Am. J. Hum. Genet.*, **92**, 504-516.

Choi, Y., Wijsman, E.M. and Weir, B.S. (2009) Case-control Association Testing in the Presence of Unknown Relationships, *Genet. Epidemiol.*, **35**, 668-678.

Koepke, H. and Thompson, E. (2013) Efficient identification of equivalences in dynamic graphs and pedigree structures, *J. Comput. Biol.*, **20**, 551-570.

Matise, T.C., Chen, F., Chen, W.W., De la Vega, F.M., Hansen, M., He, C.S. *et al.* (2007) A second-generation combined linkage-physical map of the human genome, *Genome Research*, **17**, 1783-1786.

Mukhopadhyay, N., Almasy, L., Schroeder, M., Mulvihill, W.P. and Weeks, D.E. (2005) Mega2: data-handling for facilitating genetic linkage and association analyses, *Bioinformatics.*, **21**, 2556-2557.

Nato, A.Q., Buyske, S. and Matise, T.C. (in prep) The Rutgers Map: A third generation combined linkage-physical map of the human genome. pp. Manuscript in preparation.

Purcell, S., Neale, B., Todd-Brown, K., Thomas, L., Ferreira, M.A.R., Bender, D. *et al.* (2007) PLINK: A tool set for whole-genome association and population-based linkage analyses, *Am. J. Hum. Genet.*, **81**, 559-575.

Sieh, W., Basu, S., Fu, A.Q., Rothstein, J.H., Scheet, P.A., Steward, W.C.L. *et al.* (2005) Comparison of marker types and map assumptions using Markov chain Monte Carlo-based analysis of COGA data, *BMC Genet*, **6 (Suppl 1)**, S11.

Thompson, E.A. (2011) The structure of genetic linkage data: from LIPED to 1M SNPs, *Hum. Hered.*, **71**, 86-96.

Thompson, E.A. and Heath, S.C. (1999) Estimation of conditional multilocus gene identity among relatives. In Seillier-Moseiwitch, F., Donnelly, P. and Waterman, M. (eds), *Statistics in Molecular Biology and Genetics: Selected Proceedings of the 1997 Joint AMS-IMS-SIAM Summer Conference on Statistics in Molecular Biology*. Institute of Mathematical Statistics, Hayward, CA, pp. 93-113.

Wilcox, M.A., Pugh, E.W., Zhang, H., Zhong, X., Levinson, D.F., Kennedy, G.C. *et al.* (2005) Comparison of single-nucleotide polymorphisms and microsatellite markers for linkage analysis in the COGA and simulated data sets for Genetic Analysis Workshop 14: Presentation groups 1, 2, and 3, *Genet. Epidemiol.*, **29 (Suppl 1)**, S7-S28.