Pipeline for harmonizing SNP data

Author: Marilijn van Rumpt - marilijn@live.nl (2024)

Project description:

To be able to merge genotype files from different platforms, the files must be in a uniform format. This genotype processing pipeline (command line tool) creates files which are in the same format, ready to be merged.

This command line utility can be used for the following platforms:

- Embark
- Neogen 170K array (Illumina CanineHD BeadChip)
- Neogen 220K array (Illumina CanineHD BeadChip)
- Lupa 170K array
- MyDogDNA (mdd)
- Wisdom
- WGS canfam 3 (vcf file)
- WGS canfam 4 (vcf file)
- Affymetrix

Command line utility convert.sh

Input files should be in same folder as convert.sh script. In this folder should also be the convert_files folder.

Usage: bash convert.sh [-i|e|a|o|f|n|w|v|t|l|p|h] Command line tool to convert input files to a uniform format

- Syntax options:
  - -i <filename.bim> Specify full name of .bim file
  - -e <filename.bed> Specify full name of .bed file
  - -a <filename.fam> Specify full name of .fam file
  - -o <prefix_filename> Specify prefix for output files. Obligatory
  - -f <prefix_filename> Specify prefix for .bed + .fam +. bim file
  - -n <filename> Specify full name of final report file of Neogen220K or 170K
  - -w <filename> Specify full name of Wisdom file (xlsx)
  - -v <filename> Specify full name of VCF.gz canfam 3 or 4 file (WGS)
  - -t To indicate a .tbi file of the VCF file is already present, and skip the step of indexing the raw vcf
  - -l To indicate a file with filtered locations from raw vcf file is present, and skip step of filtering locations
  - -p <platform> Specify platform, options: embark, neogen170, neogen220, lupa170, mdd, wisdom, vcf3, vcf4. Obligatory
  - -h Print the help overview
- Examples:
  - bash convert.sh -f inputfile -p embark -o newfilename
  - bash convert.sh -a inputfile.fam -i inputfile.bim -e inputfile.bed -p mdd -o newfilename
  - bash convert.sh -n inputfile -p neogen220 -o newfilename
  - bash convert.sh -v inputfile.vcf.gz -t -p vcf3 -o newfilename
  - bash convert.sh -v inputfile_filtered_locations.vcf -l -p vcf3 -o newfilename
- Dependencies needed:
  - python3, with packages pandas and openpyxl (only needed for converting wisdom files)
  - perl
  - plink 1.9

Syntax options per platform:

- -p embark:
  - For specifying input files, use either -i,-e-,a together, or only -f.
  - Options -n, -w, -v, -t, -l cannot be used
  - For platform use -p embark
- -p neogen220:
  - For specifying input files, use -n
  - Options -i,-e-,a,-f, -w, -v, -t, -l cannot be used
  - For platform use -p neogen220
- -p neogen170:
  - For specifying input files, use -n
  - Options -i,-e-,a,-f, -w, -v, -t, -l cannot be used
  - For platform use -p neogen170
- -p wisdom:
  - For specifying input files, use -w
  - Options -i,-e-,a,-f, -n, -v, -t, -l cannot be used
  - For platform use -p wisdom
- -p mdd:
  - For specifying input files, use either -i,-e-,a together, or only -f.
  - Options -n, -w, -v, -t, -l cannot be used
  - For platform use -p mdd
- -p lupa170:
  - For specifying input files, use either -i,-e-,a together, or only -f.
  - Options -n, -w, -v, -t, -l cannot be used
  - For platform use -p lupa170
- -p vcf3:
  - For specifying input files, use -v
  - Use -t to indicate .tbi (indexed vcf) file is already present. Use in combination with -v.
  - Use -l to indicate .vcf file with filtered locations from the raw vcf file is already present. Specify file with -v.
  - Options -i,-e-,a,-f, -n, -w cannot be used
  - For platform use -p vcf3
- -p vcf4:"
  - For specifying input files, use -v
  - Use -t to indicate .tbi (indexed vcf) file is already present. Use in combination with -v.
  - Use -l to indicate .vcf file with filtered locations from the raw vcf file is already present. Specify file with -v.
  - Options -i,-e-,a,-f, -n, -w cannot be used
  - For platform use -p vcf4
- -p affymetrix
  - For specifying input files, use either -i,-e-,a together, or only -f.
  - Options -n, -w, -v, -t, -l cannot be used
  - For platform use -p affymetrix

Operating system: Linux

This command line utility operates in Linux and is tested in Ubuntu, so it is advised to use Ubuntu. If you have a Windows system, a Windows Subsystem for Linux (WSL) should be used. Installation information can be found on the ubuntu website.

Operating system and used program versions

- convert.sh
  - This command line utility operates in Linux and is tested in Ubuntu, so it is advised to use Ubuntu. If you have a Windows system, a Windows Subsystem for Linux (WSL) should be used. Installation information can be found on the ubuntu website.
- convert_GB.sh
  - This utility operates in git bash.
- convert_W.sh
  - This utility operates in the Windows Powershell

- Used versions to test the tools:
    - Ubuntu 22.04.2 LTS
    - Git bash 2.40.1.windows.1
    - Windows Powershell PSVersion 5.1.22621.2506
    - Perl 5, version 34, subversion 0 (v5.34.0)
    - Python 3.10.12
        - pandas 2.0.3
        - openpyxl 3.1.2
    - Plink 1.9

SNP information

- How alleles are called can differ. They can be coded in Forward/dbsnp, TOP, A/B and 1/2. To merge files, all files must be in the same allele coding.
- The dog has 38 autosomal chromosomes. The additional SNPs are placed on a 'chromosome' as followed: On 39 are the non-pseudo-autosomal X chromosome SNPs, on 40 are the Y-chromosome SNPs, on 41 are the pseudo-autosomal X-chromosome SNPs and on 42 are the mitochondrial SNPs.

File descriptions (common files folder):

- SNPsToExcludeMerge.list
    - This file contains SNPs that have to be excluded when merging files from different platforms
    - SNPs on this list have locations that do not match between files (for example, SNP123 has different location in Embark, compared to Illumina)
    - These SNPs are added to each platforms ExcludedSNPs file, on which are also SNPs that have to be removed for this platform specifically.
- SNP_Table_Big.txt
    - snp table used in perl script to change allele calling. All SNPs are in ATCG format, also the indel SNPs
- SNP_Table_Big_Forward.bim
    - SNPs from the SNP_Table_Big.txt in their forward calling, is used for checking correct allele calls in WGS files

Plink settings

- --chr-set 38 is used in command (not --dog)
    - By doing this, the chromosome coding will remain the same (all in numbers from 1 to 42).
    - Dog has 38 autosomes
- --allow-no-sex
    - prevents errors because sex is missing in .fam or .ped file when merging files

Overview differences between arrays

PLATFORM	EMBARK	NEOGEN 170K	NEOGEN 220K	LUPA 170K	LUPA 50K	MYDOGDNA	WISDOM	AFFYMETRIX
SIZE ARRAY	229988 229991 232268	173662	220853	174376 174419 174450 174810	49658	15654	15788	913936
ALLELE CALLING	TOP	TOP/forward/AB	TOP/forward/AB	Forward	Forward	1/2	0-1-2 (forward)	Forward
RS NUMBERS	Yes	Yes	Yes	No	No	Yes	Yes	No
CANFAM	3.1	2	3.1	2	2	3.1	3.1	3.1
NON-AUTOSOMAL CHROMOSOMES	39-40-41-42	X-Y	X-Y-MT	39-40	39	None	X-Y	39,42
EXCLUDED SNPS	7-8	181	558	303-319	1533	708	215-222	702437
OTHER	900 - 922 indel SNPs 213 different SNP IDs 1534 SNPs with _ilmndup1 36 SNPs with wrong alleles 5 Duplicate SNPs 3 snps with canfam 2 location	1 snps that calls wrong allele	887 indel SNPs 12 different SNP IDs 2 duplicate snps 1 SNP calls wrong allele	53 SNPs have no CF3 location 146 SNPs not in TOP		702 unknown SNPs	13 snps with wrong alleles in translation table	No Y SNPs 211499 SNPs remain

For each platform, the actions performed by the command line utility are explained below.

Embark

Summary raw data:

- SNP array of 229988, 229991 or 232268 SNPs
- Type of allele calling: TOP
- Contains data of X (chr 39), X-pseudo autosomal (chr 41), Y (chr 40) and depending on array mitochondrial (chr 42)
- 213 SNPs in EMBARK have a different SNPid in other arrays
- Embark has 36 SNPs that call a different allele compared to alleles in other arrays
- Not all SNPs are written completely in uppercase
- Some SNPs have _rs numbers in SNP id
- CanFam 3.1

Specific array characteristics

- 229988 array
    - Has 5 duplicate SNPs, under 2 different SNP ids, but the same location
        - Of 3 of these SNPs, one of the duplicate pair calls wrong allele
    - Has 3 snps with location still in canfam 2
    - Has a number of SNPs with a wrong location, a 2 bp deviation form the correct location
    - 1534 SNPs have _ilmndup1 in SNP id
    - 900 indel SNPs
- 229991 array
    - Has 5 duplicate SNPs, under 2 different SNP ids, but the same location
        - Of 3 of these SNPs, one of the duplicate pair calls wrong allele
    - Has 3 snps with location still in canfam 2
    - 900 indel SNPs
    - 1534 SNPs have _ilmndup1 in SNP id

- 232268 array
    - Has no duplicate SNPs
    - Has no _ilmndup1 in SNP ids
    - Has no mitochondrial SNPs
    - 916 indel SNPs
    - Has 280 SNPs with a wrong location, a 2 bp deviation form the correct location or still in old canfam 2
        - 276 of these locations are corrected in the script
        - Of the remaining 4 locations, no correct location is available (will be removed)

Embark files

- EmbarkSNPIdConversion
    - File with the SNP id's that have multiple names between arrays, contains the embark ID and the ID used in other platforms
    - This file contains the SNPs for which the names are changed.
- EmbarkSNPIdCorrespondingOtherArrayId
    - Full list of SNPs that have multiple names, but some of these are excluded and are hence not in the EmbarkSNPIdConversion file
- EmbarkDuplicatesOrWrongAllele
    - File with SNP id's of duplicate snps and if one of the pair calls the wrong alleles
- EmbarkCorrectSNPPositions.map
    - File with SNPs of which correct locations and chromosomes are known.
        - locations that did not match with other arrays were checked in ensembl or by doing liftover if canfam 2 or 4 locations were available
- EmbarkCorrectAlleles.bim
    - File with SNPs that call wrong alleles in raw data files, with their correct alleles
- EmbarkIndelSNPs.txt
    - file with SNPs that are coding for indels
- EmbarkIlmndupList.txt
    - file with SNPs with _ilmndup1 in SNP id

Steps performed by the command line utility for getting the right format for Embark:

1. EMBARKconvertBIM.py script
    - Creates a new file with a list of SNP id's to be excluded using --exclude in plink
        - SNPs on chromosome 0 or bp position 0 (= no location available) or with a wrong location that could not be corrected
        - SNPs that are in the list 'SNPsToExcludeMerge'
        - SNPs that are duplicate
    - Creates a new BIM file with these changes:
        - Removes _ilmndup and _rsnumber from SNP id
        - Makes SNP id's uppercase
        - adds _INDEL to SNP id (for insertion/deletion SNPs)
        - for SNPs without or wrong location, location and chromosome is updated using EmbarkCorrectSNPPositions.map
        - changes wrong alleles to correct alleles using EmbarkCorrectAlleles.bim
        - Change SNP id if SNP also has different ID in other arrays (some SNPs in embark are the same as for example in Neogen and have the same location, but have a different SNP id)
2. The generated ExcludedSNPs.list is used in plink --exclude to remove SNPs
    - --chr-set 38 is used to make sure the chromosome coding remains the same
    - plink --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --make-bed --exclude file_exclude --chr-set 38 --out new_file
3. The log of the python and plink scripts is put in a new log file.
4. The temporary files are removed

Neogen 170K

Summary raw data:

- SNP array of 173661 SNPs
- Type of allele calling: TOP, forward and A/B
- No duplicate SNPs
- Contains data of X and X-pseudo-autosomal, Y, no mitochondrial
    - SNPs on X (39) chromosome are not yet divided over 'chromosome' 39 and 41
- Not all SNPs are written completely in uppercase
- Some SNPs have _rsnumber in ID name
- Canfam 2
- Has 1 SNP that calls a wrong allele

Neogen 170K files:

- Neogen170KsnpsPresentInCF3.map
    - map file with snps in canfam 3.1 that could be liftover. The snps that failed are not in this file.
- Neogen170KsnpsNotPresentInCF3.map
    - map file with snps that could not be liftover
- Neogen170KCorrectAlleles.bim
    - file with correct alleles for snps that call wrong alleles

Steps performed by the command line utility for getting the right format for neogen 170 K:

1. NEOGEN170Kconvert.py script
    - Creates a PED file with these changes:
        - For missing alleles, - is changed to 0
        - Corrects wrong alleles
    - Creates a MAP file with these changes:
        - Removes _rsnumber from SNP id
        - Makes SNP id's uppercase
        - Changes chromosome X to 39 or 41 if pseudo-autosomal (if location < 6640000 bp)
        - Changes chromosome Y to 40
        - Updates location and chromosome to canfam 3.1
    - Creates a new file with a list of SNP id's to be excluded using --exclude in plink
        - SNPs without location
        - SNPs that could not be liftover to canfam 3.1
        - SNPs on the SNPsToExcludeMerge.list
2. The generated ExcludedSNPs list is used in plink --exclude to remove SNPs
    - --chr-set 38 is used to make sure the chromosome coding remains the same
    - plink --map inputfile.map --ped inputfile.ped --make-bed --exclude file_exclude --chr-set 38 --out new_file
3. The log of the python and plink scripts is put in a new log file.
4. The temporary files are removed

Neogen 220K

Summary raw data:

- SNP array of 220853 SNPs
- Type of allele calling: TOP, forward and A/B
- Has 2 duplicate SNPs, under 2 different SNP ids, but the same location
- 12 SNPs in neogen have a different SNPid in other arrays
- Contains data of X and X-pseudo-autosomal, Y, and mitochondrial
    - SNPs on X (39) chromosome are not yet divided over 'chromosome' 39 and 41
    - Mitochondrial SNPs are on chromosome 'MT'
- Not all SNPs are written completely in uppercase
- Some SNPs have _rsnumber in ID name
- 887 SNPs code for indels

- 1 SNP calls wrong allele
- CanFam 3.1

Neogen 220K files:

- Neogen220KDuplicates.txt
  - File with duplicate snps
- Neogen220KSNPsMissingLocation
  - file with locations of snps derived from other arrays, to use as location for snps in neogen without location.
- Neogen220KSNPsIdConversion
  - File with the SNP id's that have multiple names between arrays, contains the neogen ID and the ID used in other platforms
  - This file contains the SNPs for which the names are changed.
- Neogen220KSNPIdCorrespondingOtherArrayId.txt
  - Full list of SNPs that have multiple names, but some of these are excluded and are hence not in the Neogen220KSNPsIdConversion file
- Neogen220K_Indels.txt
  - Indel snps present in Neogen 220K
- Neogen220KCorrectAlleles.bim
  - file with correct alleles for snps that call wrong alleles

Steps performed by the command line utility for getting the right format for Neogen 220K:

1. NEOGEN220Kconvert.py
   - Creates a PED file with these changes:
     - For missing alleles, - is changed to 0
     - For indel snps, changes the I/D calls to I=A, D=G
     - uprdates wrong alleles
   - Creates a MAP file with these changes:
     - Removes _rsnumber from SNP id
     - adds _INDEL to SNP id (for insertion/deletion SNPs)
     - Makes SNP id's uppercase
     - Changes chromosome X to 39 or 41 if pseudo-autosomal (if location < 6640000 bp)
     - Changes chromosome Y to 40
     - Changes chromosome MT to 42
     - for SNPs without location, get location on chromosome from SNP id or from Neogen220KSNPsMissingLocation file
   - Creates a new file with a list of SNP id's to be excluded using --exclude in plink
     - SNPs on chromosome 0 or bp position 0 (= no location available)
     - SNPs that are in the list 'SNPsToExcludeMerge'
2. The generated ExcludedSNPs list is used in plink --exclude to remove SNPs
   - --chr-set 38 is used to make sure the chromosome coding remains the same
   - plink --map inputfile.map --ped inputfile.ped --make-bed --exclude file_exclude --chr-set 38 --out new_file
3. The log of the python and plink scripts is put in a new log file.
4. The temporary files are removed

Lupa 170K

Summary raw data:

- SNP array of 174376 - 174810 SNPs
- Type of allele calling: Forward
- No duplicate SNPs
- Contains data of X and X-pseudo-autosomal SNPs, no Y or mitochondrial
  - SNPs on X (39) chromosome are not yet divided over 'chromosome' 39 and 41
- SNPs are all in uppercase
- No _rsnumber in SNPid
- CanFam 2

Lupa 170K files:

- Lupa174KSNPsNotInTop.txt
  - File with SNPs that could not be converted from forward to TOP calling
- Lupa174KSNPsPresentInCF3.map
  - File with SNPs that have a location in CanFam 3.1
- Lupa174SNPsNotPresentInCF3.map.txt
  - File with SNPs that have no location in CanFam 3.1, only have location in CanFam 2
- convert_bim_allele.pl
  - perl script to change allele calling
- SNP_Table_Big.txt
  - needed for the perl script, contains information about the forward or top calling of alleles

Steps performed by the command line utility for getting the right format for Lupa 170K:

1. LUPA174Kconvert.py
   - Creates a BIM file with these changes:
     - Changes chromosome X to 39 or 41 if pseudo-autosomal (if location < 6640000 bp)
     - Updates location and chromosome to canfam 3.1
   - Creates a new file with a list of SNP id's to be excluded using --exclude in plink
     - SNPs that could not be liftover to canfam 3.1
     - SNPs that can not be converted to TOP calling
     - SNPs on the SNPsToExcludeMerge.list
2. The generated ExcludedSNPs list is used in plink --exclude to remove SNPs
   - --chr-set 38 is used, to make sure the chromosome coding remains the same
   - plink --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --make-bed --exclude file_exclude --chr-set 38 --out new_file
3. Perl script convert_bim_allele.pl to convert forward allele calling to TOP calling
   - uses SNP_Table_Big.txt
   - convert_bim_allele.pl --intype dbsnp --outtype top --outfile new_bim_file input_bim_file SNP_Table_Big.txt
4. The log of the python, plink and perl scripts is put in a new log file.
5. The temporary files are removed

MyDogDNA (mdd)

Summary raw data:

- SNP array of 15654 SNPs
- Type of allele calling: illumina 1/2
- No duplicate SNPs
- Does not contain X or Y data
- Has 702 unknown SNPs (wisdom gives the SNP id 'unknown', because they are secret SNPs, MDD does not give information about their location)
- Not all SNPs are written completely in uppercase
- Some SNPs have _rsnumber in ID name
- CanFam 3.1

Files:

- convert_bim_allele.pl
  - perl script to change allele calling
- SNP_Table_Big.txt
  - needed for the perl script

Steps performed by the command line utility for getting the right format for MyDogDNA:

1. MDDconvert.py script
   - Creates a new BIM file with these changes:
     - Removes _rsnumber in SNP id
     - Makes SNP id's uppercase

    - Creates a new FAM file with these changes:
        - if family id is 0, change this value to individual id
    - Creates a new list file with SNPs to exclude and use in --exclude plink:
        - Unknown SNPs
        - SNPs that are in the list 'SNPsToExcludeMerge'
2. The generated ExcludedSNPs list is used in plink --exclude to remove SNPs
    - --chr-set 38 is used to make sure the chromosome coding remains the same
    - plink --bim inputfile.bim --fam inputfile.fam --bed inputfile.bed --make-bed --exclude file_exclude --chr-set 38 --out new_file
3. Perl script convert_bim_allele.pl to convert ilumina 1 2 allele calling to TOP calling
    - uses SNP_Table_Big.txt
    - convert_bim_allele.pl --intype ilmn12 --outtype top --outfile new_bim_file input_bim_file SNP_Table_Big.txt
4. The log of the python, plink and perl scripts is put in a new log file.
5. The temporary files are removed

Wisdom

Summary raw data:

- SNP array of 15788 SNPs
- Type of allele calling: 0-1-2 (after conversion it is in forward)
- No duplicate SNPs
- Contains data of X and X-pseudo-autosomal and Y snps, no mitochondrial snps
- Not all SNPs are written completely in uppercase
- Some SNPs have _rsnumber in ID name
- CanFam 3.1

Wisdom files:

- Raw input xlsx file
- WisdomTranslationTable.txt
    - snps and the alleles that correspond to 0 - 1 - 2
    - the wrong alleles are corrected
- WisdomTranslationTableUnchanged.txt
    - snps and the alleles that correspond to 0 - 1 - 2
    - the wrong alleles are not corrected
- convert_bim_allele.pl
    - perl script to change allele calling
- SNP_Table_Big.txt
    - needed for the perl script
- WisdomWrongAlleleInTranslationTable.txt
    - file with SNPs that have a different allele in the translation table, compared to other platform arrays (maybe these snps have wrong alleles in translation table)

Steps performed by the command line utility for getting the right format for Wisdom:

1. WisdomConvert.py script
    - Creates a new MAP file with these changes:
        - SNP names to uppercase
        - RS number removed from SNPname
        - X chromosome divided over chromosome 41 (pseudo-autosomal) and 39
    - Creates a new PED file with these changes:
        - 0-1-2 allele coding of wisdom is converted to ACTG forward
        - missing alleles are changed to 0
    - Creates a new list file with SNPs to exclude and use in --exclude plink:
        - SNPs without location
        - SNPs not in translation table
        - SNPs with wrong allele in translation table
        - SNP AMELOGENIN_C_SEX
2. The generated ExcludedSNPs list is used in plink --exclude to remove SNPs

    - --chr-set 38 is used to make sure the chromosome coding remains the same
    - plink --map inputfile.map --ped inputfile.ped --make-bed --exclude file_exclude --chr-set 38 --out new_file
3. Perl script convert_bim_allele.pl to convert forward allele calling to TOP calling
    - uses SNP_Table_Big.txt
    - convert_bim_allele.pl --intype dbsnp --outtype top --outfile new_bim_file input_bim_file SNP_Table_Big.txt
4. The log of the python, plink and perl scripts is put in a new log file.
5. The temporary files are removed

VCF canfam 3

Summary raw data:

- Type of allele calling: forward

Files:

- Raw VCF file, with .vcf.gz extension or a VCF file with filtered locations from the raw VCF file as input
- Optional a .tbi file (indexed file of the raw VCF file, made by using tabix)
- Filter file
    - To filter the snps with known SNP ids out of the VCF file
    - Has chromosome's coded as for example 1 and chr1, so both formats can be filtered
- Bim file with SNP alleles in forward to check if SNPs need te be flipped or are wrong
- SNP table for perl script

Steps performed by the command line utility for getting the right format for VCF in canfam 3: First 2 steps are skipped if option -l was used and input file is a vcf file with already filtered locations (which is normally done at step 2 with tabix).

1. Index raw VCF file using tabix (only if .tbi file does not exist yet)
    - tabix -p vcf filename.vcf.gz
2. Filter vcf file for known SNPs using tabix
    - tabix -h -R VCFFilterFile.txt filename.vcf.gz > newfilename.vcf
3. Make BED BIM FAM format from VCF in plink
    - plink --vcf filename.vcf --chr-set 38 --make-bed --out newfilename
4. VCF3convert.py script
    - Creates a new bim file with these changes:
        - adds SNP id for known snps, based on base pair position
        - SNPs on chromosome 39 are divided over 39 and 41 (pseudo-autosomal)
        - flippes strands when needed
        - changes alleles of indel IDs to fictional alleles A (insertion) and G (deletion)
    - Creates a file with SNPs to extract
        - SNPs with a SNP id
        - bi-allelic SNPs
        - SNPs that are SNPs and not indels, except for the known indels
5. The generated VCF3_extract list is used in plink --extract to extract SNPs
6. Perl script convert_bim_allele.pl to convert forward allele calling to TOP calling
    - uses SNP_Table_Big.txt
    - convert_bim_allele.pl --intype dbsnp --outtype top --outfile new_bim_file input_bim_file SNP_Table_Big.txt
7. The log of the python, plink and perl scripts is put in a new log file.
8. The temporary files are removed

VCF canfam 4

Summary raw data:

- Type of allele calling: forward

Files:

- Raw VCF file, with .vcf.gz extension or a VCF file with filtered locations from the raw VCF file as input
- Optional a .tbi file (indexed file of the raw VCF file, made by using tabix)
- Filter file
    - To filter the snps with known SNP ids out of the VCF file
    - Has chromosome's coded as for example 1 and chr1, so both formats can be filtered
- Bim file with SNP alleles in forward to check if SNPs need te be flipped or are wrong
    - this file is based on forward alleles in build canfam 3. The forward alleles in build canfam 4 are sometimes different, hence the number of flips becomes higher.
- SNPs_CF3_CF4.txt with SNPs and their locations in canfam 3 and 4
- SNP table for perl script

Steps performed by the command line utility for getting the right format for VCF in canfam 4: First 2 steps are skipped if option -l was used and input file is a vcf file with already filtered locations (which is normally done at step 2 with tabix).

1. Index raw VCF file using tabix (only if .tbi file does not exist yet)
    - tabix -p vcf filename.vcf.gz
2. Filter vcf file for known SNPs using tabix
    - tabix -h -R VCFFilterFile.txt filename.vcf.gz > newfilename.vcf
3. Make BED BIM FAM format from VCF in plink
    - plink --vcf filename.vcf --chr-set 38 --make-bed --out newfilename
4. VCF4convert.py script
    - Creates a new bim file with these changes:
        - adds SNP id for known snps, based on base pair position
        - SNPs on chromosome 39 are divided over 39 and 41 (pseudo-autosomal)
        - flippes strands when needed
        - changes locations from canfam 4 to 3
        - changes alleles of indel IDs to fictional alleles A (insertion) and G (deletion)
    - Creates a file with SNPs to extract
        - SNPs with a SNP id
        - bi-allelic SNPs
        - SNPs that are SNPs and not indels, except for the known indels
5. The generated VCF4_extract list is used in plink --extract to extract SNPs
6. Perl script convert_bim_allele.pl to convert forward allele calling to TOP calling
    - uses SNP_Table_Big.txt
    - convert_bim_allele.pl --intype dbsnp --outtype top --outfile new_bim_file input_bim_file SNP_Table_Big.txt
7. The log of the python, plink and perl scripts is put in a new log file.
8. The temporary files are removed

Affymetrix

Summary raw data:

- SNP array of 913936 SNPs
- Type of allele calling: forward
- Contains data of X (chr 39), X-pseudo autosomal (chr 41, in raw array still 39) and mitochondrial (chr 42)
- Genome build: canfam 3

Files:

- Raw BED, BIM, FAM files
- Bim file with SNP alleles in forward to check if SNPs need te be flipped or are wrong
    - this file is based on forward alleles in build canfam 3.
- SNP table for perl script

Steps performed by the command line utility for getting the right format for affymetrix:

1. AffymetrixConvert.py script

    - Creates a new bim file with these changes:
        - adds SNP id for known snps, based on base pair position
        - SNPs on chromosome 39 are divided over 39 and 41 (pseudo-autosomal)
        - flippes strands when needed
        - changes alleles of indel IDs to fictional alleles A (insertion) and G (deletion)
    - Creates a file with SNPs to extract
        - SNPs with a SNP id
        - bi-allelic SNPs
        - SNPs that are SNPs and not indels, except for the known indels
2. The generated Affymetrix_extract list is used in plink --extract to extract SNPs
3. Perl script convert_bim_allele.pl to convert forward allele calling to TOP calling
    - uses SNP_Table_Big.txt
    - convert_bim_allele.pl --intype dbsnp --outtype top --outfile new_bim_file input_bim_file SNP_Table_Big.txt
4. The log of the python, plink and perl scripts is put in a new log file.
5. The temporary files are removed

Credits

This project is part of the Expertise Centre Genetics of Companion Animals (Faculty veterinary medicine, Utrecht University).

License