Documentation for Genotype Format Converter: fcGENE

Nab Raj Roshyara
email: roshyara@yahoo.com
Universitaet Leipzig
Leipzig Research Center for Civilization Diseases(LIFE)
Institute for Medical Informatics, Statistics and Epidemiology (IMISE)
Group: Statistical Genetics and Systems Biology
Group Leader: Prof. Markus Scholz

Version: 1.0.7

June 25, 2014

Contents

Chapter 1

Introduction

fcGENE is a free, open-source **f**ormat **c**onverting tool for **GEN**otyp**E** SNP data. Our newly developed format converting tool fcGENE offers user-friendly and efficient tool for fast generation of genotype data required by many GWA analysis tools. Current Version 1.0.7 of **fcGENE** can read gzipped files. The ending of filenames of gzipped data should be ".gz". This program can be used to simplify the process of imputation-based Genome wide association (GWA) studies. More precisely, its main purpose is twofold. First, it performs SNP-wise and individual-wise quality control of genotype SNP data and generates the files required for different imputation tools. fcGENE calculates different summary statistics in one step and gives a one-command line solution for different steps of quality control issues. Second, it converts the imputation results back to different GWA analysis tools. While converting the imputation results, it can filter SNPs based on pre-determined cut-off level of imputation quality measures. **fcGENE** can read and convert sets of genotype SNP data having formats of software listed below.

- PLINK[1], a tool for analyzing genotype/phenotype data,
- SNPTEST[4], used for analyzing single SNP association in GWA studies,
- HAPLOVIEW[8]
- EIGENSOFT[9],
- r-formatted genotype data (i.e. genotypes are given as count of allele),
- vcf-format,
- GenABEL
- and the genotype imputation tools like
  - MaCH[2],
  - minimac[2],
  - IMPUTE[3],
  - BEAGLE[6], and
  - BIMBAM[7].

PLINK is the most popular free open-source program which can be used as GWA analysis toolset[1], and is designed to perform a range of basic, large-scale analyses in a computationally efficient manner. Genotype imputation has also been recognized as an important statistical and technical step for GWA studies. MACH, IMPUTE, BEAGLE and BIMBAM are the frequently used software for imputation purpose. Similarly, SNPTEST takes into account of genotype imputation uncertainty when performing a test for association between genotypes and phenotypes. That is why this program seems to be useful for the analysis of SNP association in GWA studies.

The newly developed **fcGENE** software is designed especially for:

- the data management like:
  - merging two or more than two sets of genotype data at a time,
  - snp-wise and/or individual-wise splitting of a genotype data,
  - and exclusion SNPs and individuals
- performing snp-wise and individual-wise quality control of genotype SNP data,
- converting the formats of genotyped SNP data from one kind of previously mentioned GWA analysis tools to other,
- generating templates of commands which are necessary for the selected tool including the imputation tools
- converting the imputation-reference panel into plink-format and into the formats of other imputation tools,
- transforming back the imputed genotype data (files generated by imputation tool) into plink- and other necessary formats,
- changing input data of any imputation tool into the formats of any other imputation software,

Files converted by **fcGENE** can be directly used for subsequent association analysis. Therefore it simplifies significantly the process of imputation-based GWA studies. Sequential order of using different software for a GWA study may be considered as shown in Figure 1.1.
Figure 1.1 is a graphical representation of how fcGENE can be implemented in GWA studies together with the other GWA tools. In the figure, arrows towards the box in which fcGENE is written imply that the data files of the programs, from where the arrows start, can be loaded by fcGENE. Similarly arrows going out form fcGENE-box, are showing the programs whose files can be generated. Moreover, fcGENE can also perform a two way transformation among different imputation tools as shown in Figure 1.2.
Figure 1.2 is also a graphical representation showing the formats of different imputation tools that can be read by **fcGENE**, and the formats into which fcGENE converts afterwards.

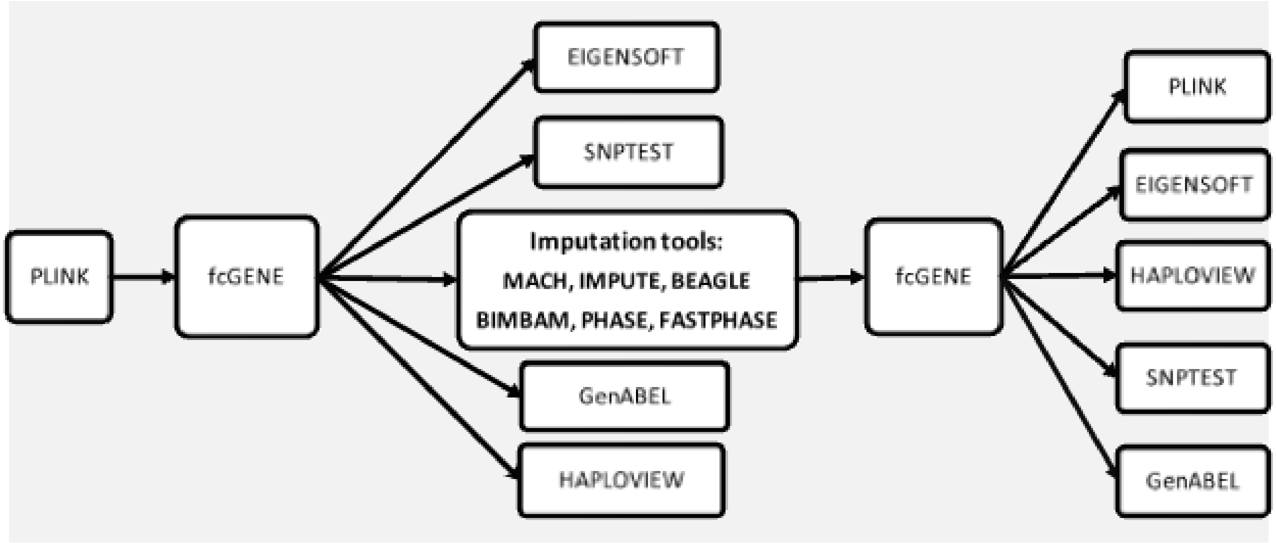

Figure 1.1: Flowchart of possible conversion steps for genotype data and the use of fcGENE during this process.

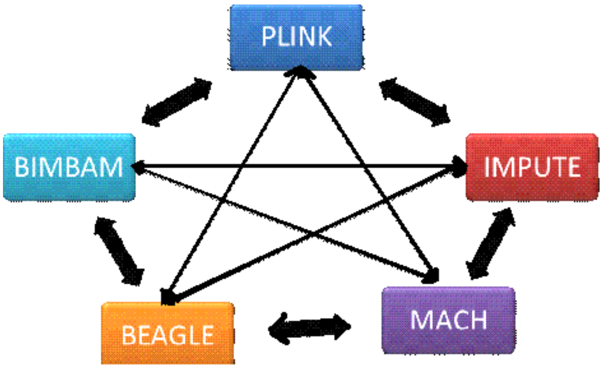

Figure 1.2: A graphical representation of genotype data that fcGENE can convert.

1.1 Basic properties of fcgene-commands

- Commands of **fcGENE** are inspired by plink commands. For example just like as in PLINK, plink-ped and map files can be read in fcGENE with command options “- -ped" and “- -map". If the ped and map file has same starting name, then we can also use the command “- -file” (e.g. - -file filename).

- fcGENE can read gzipped files. The gzipped files should have names ending with “.gz”.

- To convert any inputs into plink format, command option “- -oformat plink" is used. If we want to create binary plink-formatted data then we can use “- -oformat plink-bed”. Moreover fcGENE supports multiple format conversion at a time. We can read genotype data given into many other formats and convert them into any other formats mentioned previously.For example plink-formatted ped- and map-files can be converted by using “- -oformat tool_name” into the formats of

- plink-dosage files,
- plink-raw files,
- HAPLOVIEW,
- EIGENSOFT,
- GenABEL,
- SNPTEST,
- r-formatted data(count of reference allele)
- and different imputation tools like MaCH, IMPUTE, BEAGLE, BIMBAM.

- Command option “--oformat tool_name”, “tool_name” represents different tool specific words as described in the following table 1.1. In this table, the first column describes how files (e.g. plink ped and map files) can be read in *fcGENE*. The second column describes that genotype data read in *fcGENE* can be converted into any of the different possible formats by specifying the format name after “ --oformat”. Here, the command option:
“./*fcgene* `--ped example.ped --map example.map` ”
or “./*fcgene* `--file example` ”

Commands to read files i.e. to make input in fcGENE	Commands to generate files	
./fcgene --ped example.ped\ --example.map or ./fcgene --file example	Command option to generate files	Name of program(format) who files are to be generated
	--oformat plink	PLINK
	--oformat plink-bed	PLINK-binary
	--oformat plink-dosage	PLINK-dosage
	--oformat plink-recodeAD (--recodeAD type)	PLINK-raw
	--oformat plink-recodeA (--recodeA type)	PLINK-raw
	--oformat mach	MaCH
	--oformat minimac	minimac
	--oformat impute	IMPUTE
	--oformat snptest	SNPTEST
	--oformat haploview	HAPLOVIEW
	--oformat eigensoft	EIGENSOFT
	--oformat beagle	BEAGLE
	--oformat bimbam	BIMBAM
	--oformat r (0,1,2 coding)	r-format
	--oformat r-dose	dose files
	--oformat genable	GenABEL
	--oformat phase	PHASE/fastPHASE
	--oformat vcf	VCF

Table 1.1: Table showing the commands to generate files with fcGENE.

can be combined with any row of second column of Table 1.1. The third column of the table explains the names of the programs whose files can be generated with *fcGENE*. Moreover, the second column can be combined with any kind of commands used to upload data in *fcGENE*.

- The sequential order of command options is unimportant.
- fcGENE can perform multiple independent commands at a time. Each command except the first, starts with the command option "- -new-start" and ends with "- -new-end". The command outside of "- -new-start" and "- -new-end" is considered as the first command. While merging two separate data having same or different format can be combined with to one by using "- -merge" option inside the "- -new-start" and "- -new-end" option. In other words,if "- -merge" is given in between "- -new-start" and with "- -new-end", then this particular data is combined with the first data. An example of the command is given as

```
./fcgene --file example --oformat mach --out mach/example\
--new-start --dosage example2.dose --fam example2.fam\
--map example2.map --oformat impute --out impute/example\
--new-end
```

Table 1.2 shows all type of input files which fcGENE can read and, also the output files which can be generated with fcGENE after format conversion. One can make any combination of possible input file-formats and output file formats. Some more commands are given in in Table 1.3. These commands must be used in combination of any input files as explained in **Table** 1.2. For more details of these commands, I recommend to read the software specific chapter given in this document.

program	input options	input files	output option	output files
PLINK	--ped and --map --covar	*.ped,*.map file covariate file	--oformat plink	*.ped & *.map
PLINK binary	--bed,--bim and --fam	*.bed,*.fam, *.bim file	--oformat plink-bed	*.bed,*.bim and*.fam file
PLINK dosage	--dosage,--fam --map	*.dosage, *.fam and *.map file	--oformat plink-dosage	*.dosage,*.fam and *.map file
PLINK-raw (recodeAD)	--recodeAD --snpinfo,--map	*.raw, *.map and *.snpinfo file	--oformat plink-recodeAD	*.raw, and *.map file
PLINK-raw (recodeA)	--recodeA --snpinfo --map	*.raw, *.map and *.snpinfo file	--oformat plink-recodeA	*.raw, and *.map file
MACH inputs	--ped and --map	*.ped and *.dat files	--oformat mach	*.ped and *.dat files
MACH references	--mach-hap --mach-snp	mach_ref.hap mach_ref.snp		
MACH outputs	--mach-geno and --mach-info --mach-mlgeno and --mach-mlinfo --mach-mlprob and --mach-mlinfo	*.geno and *.info files *.mlgeno and *.mlinfo files *.mlprob and *.mlinfo files	- - -	- - -
minimac inputs	--ped and --snps	*.ped and *.snps	--oformat minimac	*ped and *.snps files
minimac outputs	--minimac-prob and --minimac-info	*.prob and *.info	-	-
IMPUTE	--gens	*.gens file	-- oformat impute	*.gens file *.strand.txt file
IMPUTE references	--impute-hap --impute-legend	impute_ref.hap impute_ref.legend	-	-
SNPTEST	--gens and --sample	*.gen and *.sample file	--oformat snptest	*.gen and *.sample file
BEAGLE inputs	--bgl	*.bgl file	--oformat beagle	*.bgl file
BEAGLE references	--bgl	beagle_ref.bgl		
BEAGLE outputs	--bgl --bgl-gprobs	*.bgl.phased file *.bgl.grpobs file		
BIMBAM input	--geno --pos	*.geno.txt file *.pos.txt	--oformat bimbam	*.geno.txt file *.pheno.txt file *.pos.txt file
output best-guess	--wbg --pos	best.guess.genotype snpinfo.txt		
output gprobs	--wgd --pos	genotype.distribution snpinfo.txt		

Table 1.2: Table showing the list command options and files which fcGENE can upload and write as output.

Functions	**Commands**
[Summary statistics	**Commands**
To calculate p-values of Hardy Weinberg equilibrium test	- -hardy
To calculate snp-wise call rate	- -crate
To calculate individual-wise call rate	- -crate
To calculate Minor allele frequencies	- -freq
To filter SNPs with quality	- -filter-snp maf=0.1,hwe=0.2,crate=0.98
To filter Individuals with call rate	- -filter-indiv crate=0.98
Format file generation	**Commands**
To generate files of HAPLOVIEW	- -oformat haploview
To generate files of EIGENSOFT	- -oformat eigensoft
To generate standard 012-formatted genotype files	- -oformat r
Standard text file with expected doses of minor allele	- -oformat r-dose
To generate VCF-format files	- -oformat vcf
To generate fast-phase files	- -oformat fastphase
To generate GenABEL files	- -oformat genabel
Data Management	**Commands**
To generate a list of SNPs	- -write-snplist
To extract SNP data only with particular SNPs	- -extract
To generate a list of individuals	- -write-pedlist
To write SNP information	- -write-snpinfo
To write pedigree information	- -write-pedinfo
To update pedigree information	- -pedinfo
To update snp information	- -snpinfo
To exclude SNPS	- -exclude snplist.txt
To remove individuals	- -remove individual.txt
To extract or split individuals	- -isplitt n1-n2,n3-n4
To extract or split SNPs	- -ssplitt n1-n2,n3-n4
To merge family id and individual ids for generating new individual ids	- -iid fid,iid,sep=_ New iid = “fid_iid”
To merge fid, patid and matid to iid with separator ’->’	- -iid fid,patiid,matid,sep=-> New iid = “fid->patid->matid->iid”
To force fcGENE for assigning all individuals to a particular phenotype	- -force pheno=aff or - -force pheno=unaff
To force fcGENE for assigning all individuals to a particular sex	- -force sex=1 (for male) or - -force sex=2 (for female) and - -force sex=0 (for undefined)

Table 1.3: Description of optional commands that can be combined with the commands given in Table 1.2.

Chapter 2

Instruction to download,compile and start program: fcGENE

One can either use the pre-compiled version of fcGENE or can use the source codes for self compilation.

1. Go to the homepage of *fcGENE* documentation `http://sourceforge.net/projects/fcgene/`.

2. Download the latest version of fcGENE and its pdf documentation file.

3. Decompress the source code using unzip command.

   - Use of pre-compiled version of the program:

     (a) **To use in Windows system:**

     $*->$ Open command promt (cmd.exe).

     $*->$ Go the directory of fcgene and copy “fcgene.exe” to your working directory

     $*->$ Start using the program fcGENE :(type fcgene.exe and <ENTER>)

     (b) **To use in Linux system:**

     $*->$ Open Linux console.

     $*->$ Use the following command to download:
     wget http://sourceforge.net/projects/fcgene/files/fcgene-1.0.7.tar.gz/download

     $*->$ To unpack use command : “tar -zxvf fcgene-1.0.7.tar.gz”.

     $*->$ Go the directory of fcgene-1.0.7 with command “cd fcgene-1.0.7”.

     $*->$ If you want to use static version of the program, type command :
     “mv fcgene_static fcgene”,

     $*->$ For the dynamic version type command: “mv fcgene_dynamic fcgene”.

   - Self-compilation of the program:

     (a) **For Linux users:**

     $*->$ Open Linux console.

     $*->$ Use the following command to download:
     wget http://sourceforge.net/projects/fcgene/files/fcgene-1.0.7.tar.gz/download

     $*->$ To unpack use command : “tar -zxvf fcgene-1.0.7.tar.gz”.

     $*->$ Go the directory of fcgene-1.0.7 with command “cd fcgene-1.0.7”.

- ∗− > Execute the following commands:
    1. ./configure (type " ./configure" and press <enter>).
    2. make (type " make" and press <enter>).
    3. sudo make install (If you want to install the program in your system)
    4. fcgene
    5. make clean (To delete *.o files and *gch files from folder src)
- ∗− > If you don't want to install, then you can copy "fcgene" from src folder to your working directory using "cp src/fcgene working_directoy/.".
- ∗− > Start using fcGENE commands as mentioned in fcGENE-documentation file.
- ∗− > If you get a problem(something like error: cannot find install-sh or install.sh in) when using command "./configure", then execute the command: "automake –add-missing –copy". while using this command, you may get error but don't worry. The necessary script "install-sh" script is already created. Now type again "./configure", "make", "sudo make install" and finally "fcgene" . (If you don't have automake program, this can be installed with the command "apt-get install automake".)

(b) **For Windows users:**

- ∗− > Download and install MINGW program from http://www.mingw.org/, if this is not installed on your computer.
- ∗− > Open mSys console installed together with MINGW program
- ∗− > execute the command:
  " ./configure" (type " ./configure" and press <enter>).
- ∗− > execute the command: "make" (type " make" and press <enter>).
- ∗− > Type 'make clean' to delete *.o and *.gch types of files.
- ∗− > Start using fcGENE by typing fcGENE-commands either in MSYS console or in windows command promt (cmd.exe).

(c) **If you have a compilation error something like:**
/usr/bin/ld: cannot find -lm
collect2: ld returned 1 exit status
make: *** [fcGENE] Error 1,
then it may be that you do not posses math dynamic library and/or glibc-dynamic library which is used for linking c ro c++ files statically. The solution is to compile dynamically.

(d) **For static compilation:** Instead of using just "make" in Linux command line, type:
make AC_CFLAGS1= "-static"
in the Linux-command line and press <ENTER>.

- ∗− > Command "make AC_CFLAGS1='-static' "
  will compile the program statically.

- The alternative way to use fcGENE in windows or Linux system is to use fcGENE in "R-commands". For this put fcgene (for windows fcgene.exe) file in the working directory. Then you can execute fcGENE by using R-command option: "system".
  An example of how to use fcGENE under R, is given below.

```
system("./fcgene –– bfile example ––oformat r-dose")
```

4. **How to start the program:**

- If you have installed fcGENE in your system,then use “fcgene”to start the program (without ./ i.e. type “fcgene” and <ENTER>).
- If the program is not installed and you have copied the program in your current working directory, then use “./fcgene”

Remark 2.1 *In this manual, I have always used “./fcgene” while explaining the different commands of fcGENE. If you have installed the program in your system with “sudo make install” then use “fcgene” only. “./” means the executable program “fcgene” is contained in the current working directory.If don’t have executable "fcgene" in your current directory and you use ./fcgene, it won’t work. At such a case just use "fcgene ...".*

Chapter 3

Data Management

3.1 Running fcGENE from command line

- *fcGENE* supports plink users to convert the format of genotype SNP data into any of the formats used by different imputation tools. Commands used to run fcGENE are inspired by PLINK[1]. Thus just like as the PLINK commands, each command line of fcGENE contains commands separated by two dashes (− −). For example, one can use

  ./fcgene - -help

  to display the information on how commands are used in fcGENE.

- PED and MAP files are two basic files used by software PLINK. To read these two files using fcGENE, any one of the following two commands can be applied.

  1. If the ped- and map- files are saved as “example.ped” and “example.map, then these two files can be read by fcGENE using command option:

     ```
     ./fcgene --map example.map  --ped example.ped or ./fcgene -file example
     ```

  2. If these two files are saved for example in a directory named “plink" then use the following command is used to read the plink-formatted files.

     ```
     ./fcgene --map plink/example.map  --ped plink/example.ped
     ```

     Remark 3.1 *Note that different command options used in fcGENE have no sequential effect. That means previous command can also be applied as*

     ```
     ./fcgene --ped example.ped  --map example.map
     ```

- In order to convert sets of genotype SNP data from one kind of format into other format, option “- -oformat" is used. The full form of “- -oformat" is *out format*.

- Some options have only a single phrase (for example - -help) but some options are double phrased. Double phrased options must contain another phrase after it. For example:

  - To write the output files with name “plink_out", use “- -out plink_out”.
  - To change plink formatted files into mach formatted files use “- -oformat mach”

- If the command is too long to write in one line, it can be divided into two lines by using backslash(\)as mentioned in the following example. This example is used to convert plink-dosage file into impute formatted files by filtering SNPs with snp-wise quality and individual-wise quality control.

```
./fcgene --dosage plink/example.dosage  --fam plink/example.fam\
--filter-snp crate=0.9,hwe=1e-6,maf=0.1\
--filter-indiv crate=0.9\
--oformat impute\
-- out impute/plink_impute
```

- fcGENE accpets multiple task at a time. When a fcGENE's command contains multiple format conversing tasks then each new tasks, except the first, is separated by command identifiers "--new-start" and "--new-end". The following command reads two separate types of plink-formatted files and convert the first into mach and second into impute format.

```
./fcgene --ped example.ped --map example.map \
--oformat mach --out mach/example\
--new-start\
--dosage example.dose --fam example.fam\
--map example.map --oformat impute --out impute/example\
--new-end
```

3.2 Execution of multiple tasks at a time

fcGENE can execute multiple independent tasks at a time. That means fcGENE can consider two or more than two tasks as one command. In such a case, each new tasks, except the first, starts with identifier "--new-start" and, ends with "--new-end". This type of commands can be used for example to merge two or more than two sets of genotype data. If option "--merge" is also given within "--new-start" and "--new-end" then the genotype data mentioned within these two identifiers, is merged with the genotype data given as first task (i.e. command given outside of "--new-start" and "--new-end"). The following command reads three genotype data; merge first the first two of them before the merged data is converted into plink dosage format. Call rate, HWE and MAFs are calculated for the third data first and then it is converted into beagle format.

```
./fcgene --ped example.ped --map example.map\
--new-start\
--ped mach/example.ped --dat example/example.dat\
--snpinfo mach/example_snpinfo --filter-snp hwe=1e-2\
--merge\
--new-end\
--new-start \
--gens impute/example.gens --pedinfo impute/example_pedinfo.txt \
--hardy --crate --freq --oformat beagle \
--out beagle/impute_beagle\
--new-end \
--out plink/example_dosage --oformat plink-dosage
```

3.3 Merging and splitting of genotype data

- **To merge two genotype data given in same or different formats:**
  We mention each the two sets of genotype data in between command options "`--new-start`" and "`--new-end`" and use the command option "$--merge''$. While merging two sets of genotype data, all SNPs and individuals contained in first set of genotype data, are accepted as basics. Only those SNPS and individuals which are not in first, are merged into the first. Moreover, those SNPs and/or individuals which are not in the first data also not in the second data are considered as missing so that the newly merged data has a form of a matrix. An example of merging two data can be found in previous example.

- **Excluding SNPS form the genotype data:**
  The following command first exclude SNPs mentioned in the file snplist.txt and then performs other commands such as converting format into BIBAM or calculating HWE and MAF.

```
./fcgene --ped example.ped --map example.map\
--exclude snplist.txt --hardy --maf --oformat bimbam\
--out bimbam/plink_bimbam
```

  Note that *"snplist.txt"* file used for extracting and removing SNPs from a given dataset, contains a list of SNP names one in one line as given below.

```
snp15
snp100
snp103
snp1023
```

- **Excluding Samples from genotype data:**
  Using command option "- -remove", we can remove individuals from further analysis. The following command first individuals mentioned in the file indivlist.txt and then performs other commands such as converting format into IMPUTE format or calculating call rates.

```
./fcgene --ped example.ped --map example.map\
--remove indivlist.txt --crate --oformat impute\
--out impute/plink_impute
```

  Similarly file *"indivlist.txt"* used for extracting and removing individuals contains two columns. First column contains family id and individual ids are given in the second column. An example of *"indivlist.txt"*file is given below.

```
Ind_03
Ind_13
Ind_149
Ind_198
```

- **SNP-wise and/or individual-wise Splitting of genotype data: fcGENE** can split a genotype data into its different subsets. The splitting can be snp-wise or individual-wise or both. Snp-wise splitting can be done in two ways.
  - first: using index of SNPs given in the genotype data.This can be done with "--ssplit" command (e.g. "--ssplit 1-10" means the first 10 SNPs).
  - second: using the basepair position. This can be done with "--bpsplit".(e.g. "--bpsplit 1-20000,15800-203000").

Similarly individual-wise splitting command is given with option "- -isplit". (e.g. "- -isplit 1-20,5-25" This command tells that split the original genotype data into two subsets first with first 20 individuals and second with 5-25 individuals). Snp-wise and individual-wise commands can be given at a time to extract subsets with given range of SNPs and individuals. A full command of splitting may look like as the following .
"./fcgene - -gens example.gens - -ssplit 1-500, 2000-300 - -isplit 1-300,200-800 - - oformat plink" This command splits the data in the given range and converted each of the subsets of genotype data into plink format.

3.4 Strand alignment

- Correct strand of SNPs is extremely important in GWA analysis and imputation of genotyped data. Strand orientation can be '+' or '-'. If imputation with given reference panel is planned then the strand information of SNPs in both genotyped SNP data and the reference panel must be matched with each other. Generally HapMap reference panels are given according as '+' strand but sometimes it may differ. To change the strand alignment of any SNP with PLINK, it just need a list of SNPs whose strand should be flipped. Flipping strand means changing the alleles:

  ▶ A -> T

  ▶ C ->G

  ▶ G -> C

  ▶ T -> A

  If SNPs in cases and controls are genotyped in different platform, genotyped data may contain a number of SNPs for which the allele coding differs between cases and controls. At such a situation the rare SNPs may show a very strong association with disease. If two Individuals at a SNP are genotyped in different strand, the difference can be recognized except SNPs which have C and G or A and T as first and second allele. However, SNPs having C and G, or A and T allele may look similar even if they are genotyped in different strand. To detect the opposite strand alignment among individuals at any particular SNP, the following plink command can be used. This command uses differential patterns of LD in cases versus controls:

  ▶ `./plink -- file mydata --flip-scan --out mydata`

  This command produces an output file named *t'mydata.flipscan'* . For more information, we recommend plink website and plink documentation.
  `http://pngu.mgh.harvard.edu/ purcell/plink/dataman.shtml#flipscan`

Strand alignment between genotype data and imputation reference panel

To impute genotype data with a given reference panel, both the reference panel and genotype data must be coded with respect to the same strand. If the strand orientation differs between them, we can flip the strand in one of these two sets by using previously mentioned plink command. We can also detect the strand mismatch of C/G and A/T SNPs Between SNP data and reference panel, by first using fcGENE and then PLINK. The exact steps are written below.

1. Merge the two genotype data and reference panel and convert the new merged data into plink-format. While merging them, all individual of reference panel should be assigned to a dummy phenotype. This dummy phenotype must be the same (either case or control) for all individuals. With command option "- -force", one can force fcGENE to assign all individuals either to cases or controls. Similarly assign all individuals of study data to a phenotype opposite to the reference panels.
2. Check strand the mismatches by using previously mentioned plink's "- -flip-scan" or "- -flip-scan-verbose" command.

For more information on format conversion of reference panel into plink format, I recommend to visit Chapter 5.

3.5 Updating pedigree and SNP information

When reading genotypes given in any formats mentioned earlier, one can also update the pedigree information and SNP information using commands: *- -pedinfo* and *- -snpinfo*. For example:

```
▶ ./fcgene --map plink/example.map --ped plink/example.ped\
  --pedinfo plink/pedinfo.txt
```

```
▶ ./fcgene --map plink/example.map --ped plink/example.ped --snpinfo plink/snpinfo.txt
```

```
▶ ./fcgene --map plink/example.map --ped plink/example.ped\
  --snpinfo plink/snpinfo.txt --pedinfo plink/pedinfo.txt
```

Remark 3.2 • *The first line of pedinfo.txt file must contain column specifiers. The order of specifiers may be different but the specifiers must be the words given in follwoing line.*

```
famid  indid  matid  patid  sex  phenotype
```

- *The sequential order of columns in "pedinfo.txt" is not important. Also it is not necessary to have all of the columns, but the words in the first line must contain some or all of the previously mentioned specifiers. Moreover the specifiers must represent the corresponding columns. Similarly the notation of each of pedigree information should be given in plink format. For example sex information should be coded as (1=male; 2=female; other=unknown), diseases status as (-9= missing, 0=missing, 1= unaffected , 2= affected).*
- *While updating SNP information, "snpinfo.txt" file should contain column specifiers in the first row. The column specifier should be given as*

rsID (or rsid) snpID (or snpid) position (or bp) cm_pos a0 (or allele1) a1 (or allele2)

However the sequential position of the columns and existence of all of the columns is not important. An example of snpinfo.txt file is illustrated below.

snpid	*rsid*	*position*	*a0*	*a1*
snp1	*rs4819391*	*14550436*	*A*	*G*
snp2	*rs11089128*	*14560203*	*C*	*T*
snp3	*rs11912265*	*14715506*	*A*	*C*

Command option	New IDs
- -iid famid,iid,sep=_	famid_iid
- -iid famid,iid,patid,matid,sep=_	famid_iid_patid_matid
- -iid famid,iid,patid,matid,sep=-	famid-iid-patid-matid
- -iid famid,iid,patid,matid	famidiidpatidmatid

Table 3.1: Table showing command options to crate new individual ids.

- *If `snpinfo` file contains columns having allele information like explained above, then the alleles are also updated. That means, the order of genotype probabilities (if given) will also be changed according as the allele order given in `snpinfo` file. For example if your original ped file contains the three probabilities* $(p(AA), p(AB), p(BB))$ *of a genotype as* 0.3, 0.5, 0.2 *with allele order as* C, T *and if `snpinfo` file contains the allele order as* T, C *then the probabilities of the genotype will be considered as* 0.2, 0.5, 0.3 *for further analysis.*

3.6 Creation of new individual ids for family data

Individuals in plink or mach (merlin) formatted family genotype data may be observed as unique only if we consider the corresponding family ids , individual ids, paternal and /or maternal ids collectively. This type of uniqueness may be lost if we observe individual or family ids only. However some imputation software like SHAPEIT or Eigensoft, which observe only individual ids, may not work properly unless we create new unique id for each individual. In this case, we may have to combine family-ids, and /or individual-ids, and/or paternal ids and/or maternal ids. We can specify a rule, with which fcGENE can create unique individual ids while converting genotype data into the required format. Command option " - -iid " specifies this type of rule. For example if we want to create new individual ids by combining family ids and individual ids with a character " _ " as separator, then we can mention this rule in fcGENE as

"- - iid fid,iid,sep=_",

which creates new individual ids in the form: "familyid_individualid". Any other character like "-", ".", "->" or even a string or nothing can be used as separator between family ids and individual ids and/or others. One can also combine patids and matids together with famids and individual ids by using the command option for example

" -iid fid,iid,patid,matid,sep=->"

Some examples of new individual ids are given in Table 3.6.

3.7 Extracting pedigree and SNP Information

In order to extract pedigree and SNP information from a genotype data given in any kind of format, we can use the following commands.

- To extract a list of individuals, use “--write-pedlist" as follows:
  ```
  ./fcgene --map plink/example.map --ped plink/example.ped \
  --write-pedlist --out plink/outName
  ```
- To extract a list of snps/rsids, from for example mach formatted files, use “--write-snplist" command as follows:
  ```
  ./fcgene --dat mach/example.dat --ped mach/example.ped\
  --write-snplist --out mach/outName
  ```
- To extract whole pedigree information :
  ```
  ./fcgene --map plink/example.map --ped plink/example.ped\
  --write-pedinfo --out plink/outName
  ```
- To extract whole snp information information :
  ```
  ./fcgene --map plink/example.map --ped plink/example.ped\
  --write-snpinfo --out plink/outName
  ```
- One can also get both SNP and pedigree lists at one time:
  ```
  ./fcgene --map plink/example.map --ped plink/example.ped\
  --write-pedlist\
  --write-snplist\
  --out plink/outName
  ```

  Remark 3.3 *Command option option “--out " is optional. It is used just to specify the name of output file. Above commands will save `outName_snplist.txt` and `outName_pedlist.txt` respectively.*

- To extract whole SNP information from IMPUTE formatted file:
  ```
  ./fcgene --gens impute/example.gens\
  --write-snpinfo \
  --out impute/outName
  ```
- One can also get both SNP and pedigree information at one time:
  ```
  ./fcgene --map plink/example.map --ped plink/example.ped\
  --write-pedinfo\
  --write-snpinfo\
  --out plink/outName
  ```

Remark 3.4 *Above commands will save `outName_snpinfo.txt` and `outName_pedinfo.txt` respectively.*

3.8 Quality control: snp-wise and sample-wise

fcGENE can calculate different quality measures. SNP-wise quality measures are the p-value of Hardy Weinberg equilibrium (HWE), call rate and minor allele frequency. Similarly call rates for every samples can also be calculated. Calculation of p-values is according as described and implemented by Wigginton et al [10]. This is the same test implemented in PLINK and Haploview and is considered as more accurate for rare genotypes.

fcGENE also accepts options that allow us to filter SNPs on the basis of snp-wise and individual-wise quality measures. The following commands are used to calculate quality measures and to filter SNPs before the transformation of data into the preferred format.

- To calculate allele frequencies of SNPs of impute-formated data, use “--freq" as follows:
```
./fcgene --gens impute/example.gens\
--freq --out impute/outName
```
- To calculate call rate of plink-formated data :
```
./fcgene --map plink/example.map --ped plink/example.ped\
--crate --out plink/outName
```
- For the p-values of Hardy Weinberg equilibrium:
```
./fcgene --map plink/example.map --ped plink/example.ped\
--hardy --out plink/outName
```
- One can calculate all these quality measures simultaneously.
```
./fcgene --map plink/example.map --ped plink/example.ped\
--crate --hardy --freq\
--out example/outName
```
- To filter plink-formatted SNPs according as snp-wise call rate and convert them for example into BEAGLE format :
```
./fcgene--map plink/example.map --ped plink/example.ped\
--filter-snp crate=0.9\
--oformat beagle\
--out beagle/outName
```
- To filter minimac-formatted SNPs according as snp-wise callrate, allele frequency and p-values of HWE:
```
./fcgene--snps minimac/example.snps --ped minimac/minimac.ped\
--filter-snp hwe=1e-2,crate=0.9,maf=0.1 \
--oformat beagle --out beagle/outName
```
- To filter impute-formatted SNPs according as individual-wise call rate and snp-wise call rate, allele frequency and p-values of HWE:
```
./fcgene--gens imptue/example.gens\
--filter-snp hwe=1e-2,crate=0.9,maf=0.1\
--filter-indiv crate=0.9\
--oformat beagle\
--out beagle/outName
```

Remark 3.5 *Some command option are necessary to upload files and to transform the data. However other commands such as commands used for updating pedigree and snp-information, calculating quality measures etc are optional. They can be used only when you are interested in that specific task.*

3.9 FST calculation

In order to calculate “FST” just use an additional command “--fst” with the commands to upload the data in fcGENE.

Chapter 4

Quality control and format conversion of plink-formatted data

PLINK can read different kinds of genotype data. Plink-related file formats that can be read and generated by fcGENE are

1. pedigree file format (*.ped and‘*.map files)
2. binary file format (*.bim, *.fam and *.bed files)
3. dosage file format (*.dosage and *.fam) files
4. raw file-format: This format has two forms.
    - plink-rawAD file format (*.raw file): This type of file format can be generated by - -recodeAD option in PLINK.
    - plink-rawA file format (*.raw file): This type of file format can be generated by - -recodeA option in PLINK.

For more details about plink-formatted files, I recommend to visit PLINK’s official website. Commands used to read plink-formatted files are given in Table 4. Moreover previously mentioned file-formats can be read by fcGENE using commands as follows.

- To read ped and map files:
  ```
  ./fcgene -- ped plink/example.ped --map plink/example.map
  ```
  or
  ```
  ./fcgene -- file example
  ```
- To read binary files:
  ```
  ./fcgene -- bim plink/example.bim --map plink/example.fam --map plink/example.bed
  ```
  or
  ```
  ./fcgene -- bfile example
  ```
- To read dosage and fam files:
  ```
  ./fcgene -- dosage plink/example.dosage --fam plink/example.fam
  ```
- To read raw (- -recodeAD type) file:
  ```
  ./fcgene -- recodeAD plink/example.raw --snpinfo plink/example_snpinfo.txt
  ```
- To read raw (- -recodeA type) file:
  ```
  ./fcgene -- recodeA plink/example.raw --snpinfo plink/example_snpinfo.txt
  ```

Format name	Command option	File name
plink-pedigree	–file	example
plink- pedigree	--ped	example.ped
	--map	example.map
plink-binary	–bfile	example
plink-binary	--bim --fam --bed	example.bim example.fam example.bed
plink- dosage	--dosage	example.dosage
	--fam	example.fam
plink- raw(recodeAD-type)	--recodeAD	example.raw
	--snpinfo	example_snpinfo.txt
plink- raw(recodeA-type)	--recodeA	example.raw
	--snpinfo	example_snpinfo.txt

Table 4.1: Table showing necessary files and command options to read plink-formatted data.

A detailed information on "snpinfo" file can be found in Section 3.5. Snpinfo file containing minor and major allele is necessary to read raw files. Otherwise 0 1 and 2 coding can not be recognized.

4.1 Quality control

Quality control(QC) is an important step for GWA analysis. Most of the imputation software-provider suggest to filter SNPs and individuals on the basis of their quality measures. Quality of SNPs is typically based on the QC variables e.g. p-value of Hardy-Weinberg Equilibrium (HWE), missing percent (or call rate) and minor allele frequency (MAF). fcGENE can calculate these quality measures and also filter SNPs and individuals according as pre-defined cutoff. Commands used to calculate HWE,MAF and callrate of different plink-formatted data are given below.

- To calculate call rate, p-values of HWE and MAF with pedigree data
  ```
  ./fcgene -- ped plink/example.ped --map plink/example.map\
  --freq --hardy --crate\
  --out plink/example
  ```
- To calculate call rate, p-values of HWE and MAF with dosage file
  ```
  ./fcgene -- dosage plink/example.dosage --fam plink/example.fam\
  --freq --hardy --crate\
  --out plink/example
  ```
- To calculate call rate, p-values of HWE and MAF with plink-rawAD file
  ```
  ./fcgene --recodeAD plink/example.raw --snpinfo plink/example_snpinfo.txt\
  --freq --hardy --crate\
  --out plink/example
  ```
- To calculate call rate, p-values of HWE and MAF with plink-rawA file
  ```
  ./fcgene --recodeA plink/example.raw --snpinfo plink/example_snpinfo.txt\
  ```

```
--freq --hardy --crate\
--out plink/example
```

Similarly some examples to filter SNPs and individuals with pre-defined cutoff are given below.

-
  ```
  ./fcgene -- ped plink/example.ped --map plink/example.map\
  --filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  --filter-indiv crate=0.95\
  --oformat plink-dosage --out plink/example
  ```

-
  ```
  ./fcgene -- dosage plink/example.dosage --fam plink/example.fam\
  --filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  --filter-indiv crate=0.95\
  --write-snplist --out plink/example
  ```

-
  ```
  ./fcgene --recodeAD plink/example.raw --snpinfo plink/example_snpinfo.txt\
  --filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  --filter-indiv crate=0.95\
  --oformat beagle --out plink/example
  ```

-
  ```
  ./fcgene --recodeA plink/example.raw --snpinfo plink/example_snpinfo.txt\
  --filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  --filter-indiv crate=0.95\
  --oformat haploview --out plink/example
  ```

-
  ```
  ./fcgene --bfile plink/example --filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  --filter-indiv crate=0.95\
  --oformat vcf --out plink/example
  ```

-
  ```
  ./fcgene --file plink/example --filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  --oformat r --out plink/example
  ```
  After filtration process, the filtered data will be converted into r-formatted data (i.e. gentoype allele counts. 0, 1,2).

-
  ```
  ./fcgene --file plink/example --ssplit 1-500,30-1000 --isplit 1-100,100-200
  --filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  --filter-indiv crate=0.95\
  --oformat r-dose --out plink/example
  ```
  This commands first split the data into different SNPs and individuals and then filters each of the split data according as the given filtering commands.

4.2 Format Conversion

fcGENE can read files having PLINK's pedigree,binary, dosage and raw-format, and then convert the uploaded genotype data into the formats of of MaCH, IMPUTE, BEAGLE, BIMBAM, SNPTEST, HAPLOVIEW, EIGENSOFT, GenABEL, VCF etc. Table 4.2 shows the possible commands that can be applied to generate files with fcGENE. One can convert plink-binary, plink-dosage and plink-raw format files into any other formats using the same procedure as explained in Table 4.2. While converting plink-formatted dosage files, it is important to match family id and sample id both in fam and dosage file. If family id is not given, fcGENE creates a family id like "famid+index". This may sometimes differ from the real family id. So The best solution is always to update pedigree information with "--pedinfo" command.

Commands to read files i.e. to make input in fcGENE	Command option to generate files	Assignment of output file name
./fcgene - - ped plink/example.ped\\ - -map plink/example.map\\ - -filter-snp maf=0.1,crate=0.95,hwe=1e-2\\ - -filter-indiv crate=0.95 Command options related to Quality control are optional.	- -oformat plink	- -out outFileName
	- -oformat plink-bed	
	- -oformat plink-dosage	
	- -oformat plink-recodeAD (- -recodeAD type)	
	- -oformat plink-recodeA (- -recodeA type)	
	- -oformat recodeA-dose (- -recodeA type, but with doses)	
	- -oformat mach	
	- -oformat minimac	
	- -oformat impute	
	- -oformat snptest	
	- -oformat haploview	
	- -oformat eigensoft	
	- -oformat beagle	
	- -oformat bimbam	
	- -oformat r (0,1,2 coding)	
	- -oformat r –transpose	
	- -oformat phase	
	- -oformat vcf	
	- -oformat genable	

Table 4.2: Table showing commands to generate files from plink-ped and map file.

4.3 Coding genotypes as count of a given reference allele:

We realized that coding genotypes as the counts of minor alleles may be very ideal when we apply different statistical models like regression and ANOVAs. PLINK supports to produce this type of format by generating two forms of raw-formatted files by using command options "- -recodeA" and "- -recodeAD", but these two options support only the hard calls of genotype data: meaning use of only the numbers 0, 1 and 2, to represent genotypes of the form homozygote major, heterozygote and homozygote minor respectively. Alongside the support to PLINK-users by converting different types of imputation results into previously mentioned two forms of raw-formatted files.

fcGENE provides the facility also to transform the data with genotype probability distribution into the form of PLINK's *recodeA-formatted* files but filled with *expected minor-allele-doses*. This means this type of raw files transformed by fcGENE can contain not only 0,1 and 2 as minor allele counts but also the expected allele dose of minor allele, which can be any fractional number between 0 and 2. If "A" and "B" are two alleles of a SNP with "B" as minor allele, then the expected minor allele dose can be calculated as

$$0 * p(AA) + 1 * p(AB) + 2 * p(BB) = p(AB) + 2p(BB),$$

where $p(AA)$, $p(AB)$ and $p(BB)$ are the probabilities having genotypes A , AB and BB at an individual respectively. GWA analysis with this type of coding may be very useful especially when the accuracy of imputation results is low because it can account for the uncertainty in the imputed genotypes.

Using command options “- -oformat plink-recodeA” and “- -oformat plink-recodeAD”, we can obtain PLINK’s raw files. In order to produce raw file filled with “minor-allele-expected dose”, fcGENE requires the command option “- -oformat recodeA-dose”. This type of file format has the same form as previously mentioned PLINK raw files but provides expected allele doses of reference allele instead of genotypes resulting in numbers between 0 and 2. By default, minor-allele is taken as reference allele.

One can force fcGENE to change the reference allele with command option “- -force ref-allele=”. Possible options for forcing reference allele are:

- “- -force ref-allele=minor-allele” (this is default),
- “- -force ref-allele=major-allele”,
- “- -force ref-allele=allele1” and
- “- -force ref-allele=allele2”.

Some examples of Coding genotype doses

▶
```
./fcgene -- file example \
--oformat --recodeA-dose --out plink/example
```

▶
```
./fcgene --bed example.bed --bim example.bim --fam example.fam \
--oformat --recodeA-dose \
--force ref-allele=major-allele, --out plink/example
```

Chapter 5

Strand allignment and format conversion of imputation references

5.1 Strand alignment

Before we start the imputation process in any imputation tool, we may need to check the similarities (for example the strand alignment) between genotype SNP data and the reference panel. Strand alignment between genotype dataset and reference data set is crucial for GWA analysis and imputation. Generally, reference panels such as HapMap are given as '+' strand but they might be genotyped with respect to negative strand. fcGENE supports strand flipping by the following approach:

1. Use fcGENE first to merge the genotype SNP data and the given reference panel

2. Use fcGENE to convert the merged data into plink format.
   While merging these two data, individuals of genotype SNP data and reference panel should be assigned with dummy cases and control status respectively. This can be done by applying command option "–force" in fcGENE (see below examples and section).

3. Finally, strand mismatches can be detected by applying "–flip-scan" or "– flip-scan-verbose" commands of PLINK **??**.

In order to perform above steps for example in mach-formatted references and plink-binary formatted genotype data, we can use the following command

```
./fcgene --ped example.ped --map example.map
--new-start\
--mach-hap mach/mach_ref.hap \
--mach-snp mach/mach_ref.snps \
--snpinfo mach/example_snpinfo \
--force pheno=unaff,sex=m --merge \
--new-end \
--force pheno=aff,sex=m --out plink/plink_mach_ref --oformat plink-bed
```

Above command will combine mach-formatted reference file and plink-formatted example file and will convert it into plink binary file and will save converted files in the folder named "plink".

Similarly, we may also need to check the differences in allele frequencies. To perform these analysis, the first step would be to convert the given reference panel into plink format or in the format of any other imputation tool. The following sections explain the commands to convert the HapMap formatted reference; mach formatted references and impute formatted references.

5.2 HapMap-formatted references

This option has not been included yet.

5.3 Mach-formatted references

Mach-formatted reference data consist of two files

1. *.hap file
2. *.snp file

These two files can be downloaded from mach homepage:
`http://www.sph.umich.edu/csg/abecasis/MACH/download/`. Table 5.1 shows the necessary files and commands to upload mach-formatted references in fcGENE.

file name	command option
mach_ref.hap	--mach-hap
mach_ref.snp	--mach-snp

Table 5.1: Table showing necessary files and command options to read mach-formatted references.

Examples of format conversion of MaCH-references

- To read mach formatted hap and snp file with fcGENE, the following command can be used.
```
./fcgene --mach-hap mach/mach_ref.hap\
--mach-snp mach/mach_ref.snp
```
- For calculating quality measures like MAF and HWE:
```
./fcgene --mach-hap mach/mach_ref.hap\
--mach-snp mach/mach_ref.snp\
--hardy --freq --out mach/mach_ref
```
- To convert mach formatted hap and snp file into plink-formatted genotype data:
```
./fcgene --mach-hap mach/mach_ref.hap\
--mach-snp mach/mach_ref.snp \
--oformat plink \
--force pheno=unaff,sex=m \
--out plink/mach_ref
```

or

```
./fcgene --mach-hap mach/mach_ref.hap\
--mach-snp mach/mach_ref.snp \
--oformat plink-bed \
--force pheno=unaff,sex=m \
--filter-snp hwe=1e-6,maf=0.05\
--out plink/mach_ref
```

These commands help us to check the strand alignment between genotype SNP data and HapMap reference.

- Mach formatted references do not contain the phenotype and sex information of its individuals. However, using “--force” command, we can force fcGENE to assign all individuals to a particular phenotype and sex. This assignment is necessary if we want to make strand alignment with PLINK between references (e.g. assigned as controls) and SNP data (e.g. assigned as cases). PLINK categories individuals as unspecified if sex information is not given.
- The command option after “--force” should not contain a space character and if you give more option to force then they should be separated by a comma (,).
- To change phenotype type information, one can use either “pheno=1” or “pheno=unaff" and “pheno=2”or “pheno=aff”. Default would be missing but if you want express it specifically, then you can use “pheno=-9”.
- To change sex information, either “sex=1” and “sex=2” or “sex=m” and “sex=f” or “sex=M” and “sex=F” can be used. For missing, use “sex=0”.
- We can also convert mach references into the format of for example SNPTEST and any other GWA analysis tool. For these type of conversion we need to specify “--oformat ” option accordingly. Some examples of such conversion are given below.
  - ▶ Conversion into impute-format:
    ```
    ./fcgene --mach-hap mach/mach_ref.hap\
    --mach-snp mach/mach_ref.snp \
    --oformat impute \
    --out impute/mach_ref
    ```
  - ▶ Conversion into beagle-format:
    ```
    ./fcgene  --mach-hap mach/mach_ref.hap\
    --mach-snp mach/mach_ref.snp \
    --oformat beagle \
    --force pheno=unaff,sex=m\
    --out beagle/mach_ref
    ```
  - ▶ Conversion into snptest-format:
    ```
    ./fcgene  --mach-hap mach/mach_ref.hap\
    --mach-snp mach/mach_ref.snp \
    --oformat snptest \
    --force pheno=unaff,sex=m \
    --out snptest/mach_ref
    ```
  - ▶ Conversion into bimbam-format:
    ```
    ./fcgene  --mach-hap mach/mach_ref.hap\
    --mach-snp mach/mach_ref.snp \
    ```

```
--oformat bimbam \
--force pheno=unaff,sex=m \
--out bimbam/mach_ref
```

- Conversion into eigensoft-format:
  ```
  ./fcgene  --mach-hap mach/mach_ref.hap\
  --mach-snp mach/mach_ref.snp \
  --oformat eigensoft \
  --force pheno=unaff,sex=m\
  --out eigensoft/mach_ref
  ```
- Conversion into haploview-format:
  ```
  ./fcgene  --mach-hap mach/mach_ref.hap\
  --mach-snp mach/mach_ref.snp \
  --oformat haploview \
  --force pheno=unaff,sex=m \
  --out haploview/mach_ref
  ```
- Conversion into plink-raw-format:
  ```
  ./fcgene  --mach-hap mach/mach_ref.hap\
  --mach-snp mach/mach_ref.snp \
  --oformat plink-recodeA \
  --force pheno=unaff,sex=m\
  --out plink/mach_ref_rawA
  ```
- Conversion into plink-raw-format:
  ```
  ./fcgene  --mach-hap mach/mach_ref.hap\
  --mach-snp mach/mach_ref.snp \
  --oformat plink-recodeAD \
  --force pheno=unaff,sex=m\
  --out plink/mach_ref_rawAD
  ```
- Conversion into plink-dosage-format:
  ```
  ./fcgene  --mach-hap mach/mach_ref.hap\
  --mach-snp mach/mach_ref.snp \
  --oformat plink-dosage \
  --force pheno=unaff,sex=m \
  --out plink/mach_ref_dosage
  ```
- Conversion into binary plink data:
  ```
  ./fcgene  --mach-hap mach/mach_ref.hap\
  --mach-snp mach/mach_ref.snp \
  --oformat plink-bed \
  --force pheno=unaff,sex=m \
  --out plink/mach_ref_binary
  ```
- Conversion into r formatted data:
  ```
  ./fcgene  --mach-hap mach/mach_ref.hap\
  --mach-snp mach/mach_ref.snp \
  --oformat r \
  --force pheno=unaff,sex=m \
  --out plink/mach_ref_binary
  ```

Remark 5.1 *A list of other possible command options that can be used in combination with "- -oformat" are given in Table 4.2 of Section 4.2.*

5.4 Impute-formated references

Impute formatted references are given with two files namely

1. *.hap file
2. *.legend file

Table 5.2 describes the files and command option necessary to read impute-formatted data. Other commands are optional and are used to perform the specific tasks. Impute formatted reference files can be read with fcGENE as follows

file name	command option
impute_ref.hap	--impute-hap
impute_ref.legend	--impute-legend

Table 5.2: Table showing necessary files and command options to read impute-formatted references.

- To convert impute-references into plink-format:
```
./fcgene  --impute-hap impute/impute_ref.hap\
--impute-legend impute/impute_ref.legend
```
- To calculate quality measures:
```
./fcgene  --impute-hap impute/impute_ref.hap\
--impute-legend impute/impute_ref.legend\
--freq --hardy --out impute/impute_ref
```
- To filter SNPs according as MAF, HWE and call rate
```
./fcgene  --impute-hap impute/impute_ref.hap\
--impute-legend impute/impute_ref.legend\
--filter-snp maf=0.1,hwe=1e-6,crate=0.99 --out impute/impute_ref
```

5.4.1 Strand alignment with IMPUTE references

For this purpose,we can use the following command

```
./fcgene --gens example.gens
--new-start\
--impute-hap impute/impute_ref.hap\
--impute-legend impute/impute_ref.legend\
--force pheno=unaff,sex=m --merge \
--new-end \
--force pheno=aff,sex=m --out impute/impute_data_ref --oformat plink-bed
```

To convert impute-references into different formats of GWA analysis tools,we can use the following commands. Note that the command options outside of “--new-start” and “--new-end” are assumed as the first command.

Other useful commands

- To convert impute-references into plink-format:
  ```
  ./fcgene  --impute-hap impute/impute_ref.hap\
  --impute-legend impute/impute_ref.legend\
  --oformat plink\
  --force pheno=unaff,sex=m\
  --out plink/impute_ref
  ```
- To convert impute-references into mach-format:
  ```
  ./fcgene --impute-hap impute/impute_ref.hap\
  --impute-legend impute/impute_ref.legend\
  --oformat mach\
  --force pheno=unaff,sex=m\
  --out mach/impute_ref
  ```
- To convert impute-references into bealge-format:
  ```
  ./fcgene  --impute-hap impute/impute_ref.hap\
  --impute-legend impute/impute_ref.legend\
  --oformat beagle --out beagle/impute_ref
  ```
- To convert haploview-references into bealge-format:
  ```
  ./fcgene  --impute-hap impute/impute_ref.hap\
  --impute-legend impute/impute_ref.legend\
  --oformat haploview --out haploview/impute_ref
  ```
- To convert impute-references into eigensoft-format:
  ```
  ./fcgene  --impute-hap impute/impute_ref.hap\
  --impute-legend impute/impute_ref.legend\
  --oformat eigensoft --out eigensoft/impute_ref
  ```
- To convert impute-references into plink-dosage-format:
  ```
  ./fcgene  --impute-hap impute/impute_ref.hap\
  --impute-legend impute/impute_ref.legend\
  --oformat plink-dosage  --out plink/impute_ref_dosage
  ```

For more information on "--force" command option, I recommend to read section 5.3.

Remark 5.2 *A list of other possible command options that can be used in combination with "--oformat" are given in Table 4.2 of Section 4.2.*

A list of

5.5 Beagle-formatted references

Beagle reference has the same format as Beagle input data excluding the phenotype information row. Necessary files to upload beagle-formatted reference in fcGENE is described in Table 5.3. Other Commands are function-specific and are optional.
fcGENE reads Beagle reference file, which also ended with "*.bgl", by using the following commands.

command option	file name
- -bgl	beagle_ref.bgl

Table 5.3: Table showing necessary files and command options to read beagle-formatted references.

- To read beagle-formatted references:
  ./fcgene - -bgl beagle/beagle_ref.bgl - -force pheno=unaff,sex=m
- To compute quality measurs like MAF,HWE and callrate:
  ./fcgene - -bgl beagle/beagle_ref.bgl
  - -freq - -hardy - -crate - -out beagle/beagle_ref
- For snp-wise and individual-wise filtering:
  ./fcgene - -bgl beagle/beagle_ref.bgl
  - -filter-snp hwe=0.002,maf=0.1,crate=0.95
  - -filter-indiv crate=0.99 - -out beagle/beagle_ref

5.5.1 Strand alignment with Beagle references

For this purpose,we can use the following command.

```
./fcgene --bgl beagle/example.bgl\
--new-start\
--bgl beagle/beagle_ref.bgl\
--force pheno=unaff,sex=m --merge \
--new-end \
--force pheno=aff,sex=m\
--out beagle/beagle_data_ref \
--oformat plink-bed
```

Other Examples

Similarly format conversion of beagle-formatted references can be performed as followings.

- Conversion into plink-format:
  ./fcgene - -bgl beagle/beagle_ref.bgl
  *- -oformat plink - -force pheno=unaff,sex=m *
  - -out plink/beagle_ref
- Conversion into mach-format:
  ```
  ./fcgene  --bgl beagle/beagle_ref.bgl\
  --oformat mach\
  --force pheno=unaff,sex=m --out mach/beagle_ref
  ```
- Conversion into impute-format:
  ```
  ./fcgene  --bgl beagle/beagle_ref.bgl\
  --oformat impute\
  --force pheno=unaff,sex=m  --out impute/beagle_ref
  ```

- Conversion into snptest-format:
  ```
  ./fcgene  --bgl beagle/beagle_ref.bgl\
  --oformat snptest\
  --force pheno=unaff,sex=m  --out snptest/beagle_ref
  ```
- Conversion into bimbam-format:
  ```
  ./fcgene  --bgl beagle/beagle_ref.bgl --oformat bimbam\
  --force pheno=unaff,sex=m --out bimbam/beagle_ref
  ```
- Conversion into haploview-format:
  ```
  ./fcgene  --bgl beagle/beagle_ref.bgl --oformat bimbam\
  --force pheno=unaff,sex=m --out haploview/beagle_ref
  ```
- Conversion into eigensoft-format:
  ```
  ./fcgene  --bgl beagle/beagle_ref.bgl\
  --oformat eigensoft\
  --force pheno=unaff,sex=m --out eigensoft/beagle_ref
  ```
- Conversion into plink-raw-format:
  ```
  ./fcgene  --bgl beagle/beagle_ref.bgl\
  --oformat plink-recodeA \
  --force pheno=unaff,sex=m  --out plink/beagle_ref_rawA
  ```
- Conversion into plink-dosage-format:
  ```
  ./fcgene  --bgl beagle/beagle_ref.bgl\
  --oformat plink-dosage \
  --force pheno=unaff,sex=m  --out plink/beagle_ref_dosage
  ```
- Conversion into plink-raw-format:
  ```
  ./fcgene  --bgl beagle/beagle_ref.bgl\
  --oformat plink-recodeAD\
  --force pheno=unaff,sex=m  --out plink/beagle_ref_rawAD
  ```

Remark 5.3 *A list of other possible command options that can be used in combination with "- -oformat" are given in Table 4.2 of Section 4.2.*

Chapter 6

Quality control, format conversion and imputation with MaCH and minimac

6.1 Data transformation to mach format

If you decide to use imputation software “MaCH " (and “mimimac") to impute your genotype data set and the data are given in plink format, then the imputation process can be proceeded by using software as follows.

PLINK ⇒ fcGENE ⇒ MaCH (and minimac) ⇒ fcGENE ⇒ PLINK

Similar process can be used if the data are given in other format.

Examples to convert into mach format

Following examples explains how to convert genotype data into mach format.

▶
```
./fcgene --map plink/example.map --ped plink/example.ped\
--oformat mach\
--out mach/example
```

▶
```
./fcgene --bfile plink/example --oformat mach\
--out mach/example
```

▶
```
./fcgene --rgeno example_genotype_data.txt \
--snpinfo alleleinfo.txt \
--oformat mach\
--out mach/example
```

Remark 6.1 *fcGENE can create mach-formatted data from any type of other formats which fcGENE can read. Note that “--rgeno” commands is used to read r-formatted genotype data whose genotypes are counts of reference allele. For more information, on this type of format, I recommend to read Section 13.*

- Above commands write two mach formatted files namely “*example/mach.ped*" and “*mach/example.dat*" . These are the two files needed for the imputation process with MaCH.

- At the same time, it also saves two extra files “*example_pedinfo.txt*" and “*example_snpinfo.txt*" in the directory “mach".
- “*mach/example_pedinfo.txt*" contains pedigree information (individual id, sex information, diseases status etc) of all individuals contained the genotype file.
- Similarly “*mach/example_snpinfo.txt*" contains SNP information (rsid, snpid, base pair position, distance in centimorgan first and second allele).
- These two “*pedinfo.txt*" and “*snpinfo.txt*" files can be used as extra information when converting imputed genotypes back into the plink format.
- As explained previously, “*--out*" option is optional. If this option does not exist, out files will be saved with the name “*fcgene_out*".

In order to upload mach-formatted ped- and dat-files into fcGENE, commands given in the Table 6.1.

Format name	Command option	File name
merlin- pedigree or mach-pedigree	--ped	example.ped
	--dat	example.dat
mach-pedigree ped and dat file together	--mfile	example

Table 6.1: Table showing necessary files and command options to upload mach-formatted data.

Other commands used in mach-pedigree format are optional and they can be used only when you are interested to perform specific function.

Remark 6.2 *If you are interested in performing the snp-wise and/or individual-wise quality control process, then the following commands should be used.*

```
▶ ./fcgene --map plink/example.map --ped plink/example.ped\
  --filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  --filter-indiv crate=0.95\
  --oformat mach\
  --out mach/example
```

```
▶ ./fcgene --bfile example \
  --filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  --filter-indiv crate=0.95\
  --oformat mach\
  --out mach/example
```

Converting data from other than plink-pedigree format is also possible. In such a case, I recommend to read Chapter 1 to know how files given in other formats can be uploaded in fcGENE. Any type of uploaded files can be converted into mach-format with or without quality control process.

6.2 Imputation with MaCH and minimac

1. Convert plink formated data into mach format as explained previously.

2. After converting genotype data into mach format, The imputation process with software “MaCH" can be performed as follows.

   - Make sure that you have downloaded software“MaCH" in your working directory and saved with name “mach1".

   - A simple way to impute the missing genotypes without using any kind of references, is to use the following commands.

     ```
     ▶ ./mach1 -p mach/example.ped -s mach/example.dat \
     - -rounds 100\
     - -states 200 - -phase - -geno \
     - -prefix mach/example_output
     ```

   - To carry out the maximum likelihood genotype imputation without using *HapMap* references, you can use - -*mle* and `--mldetails` option as stated below.

     ```
     ▶ ./mach1 -p mach/example.ped -d mach/example.dat \
     - -rounds 100\
     - -states 200 - -phase - -mle - -mldetails \
     - -prefix mach/example_output
     ```

   - To impute your genotype data using available references (e.g. HapMap or 1000 Genomes), the imputation can be performed with the following two steps method.
     step1:

     ```
     ▶ ./mach1 -p mach/mach.ped -s mach/mach.dat\
     -h hapmap.phased -s hapmap.legend - -hapmapFormat - -greedy \
     - -rounds 100\
     - -prefix mach/step1
     ```

     step2:

     ```
     ▶ ./mach1 -p mach/example.ped -s mach/example.dat\
     -h hapmap.phased -s hapmap.legend - -hapmapFormat - -greedy\
     - -crossover mach/step1.rec - -errormap mach/step1.erate\
     - -mle - -mldetails - -phase\
     - -prefix mach/step2
     ```

   - More information on imputation using software “MaCH", can be obtained from the website `http://www.sph.umich.edu/csg/abecasis/MaCH/tour/`

   If you prefer to use “minimac" in which a low memory and computationally efficient MaCH algorithm is implemented, the imputation process should be done as follows.

   - **step1:** Determine the haplotypes of your sample using “MaCH" as follows:

     ```
     ▶ ./mach1 -p mach/example.ped -d mach/example.dat \
     - -rounds 50 - -states 200\
     - -sample 5 - -phase - -compact\
     - -prefix mach/step1
     ```

Command written in step1 will save the haplotypes of your genotype data as *"mach/step1.gz"*.

- To use minimac, you also need snplist existed in the previously haplotypes. The snp list can be obtained by:

∗− > either using any one of the following command *fcGENE*.

```
./fcgene --dat mach/example.dat --ped mach/example.ped \
--write-snplist \
--out mach/example
```

This command will save the list of SNPs with name "example.snplist" in the directory "mach".

```
./fcgene --map plink/example.map --ped plink/example.ped \
--oformat minimac \
--out mach/example
```

∗− > Or you can use the following Linux commands.

```
cut -f 2 -d " " mach/example.dat > mach/example.snplist
```

- **step2:** Once you have obtained two files namely file containing haplotypes(step1.gz) and file containing SNP list(example.snps), minimac can be applied to impute the genotype data as follows.

▶
```
./minimac --refHaps ref.hap.gz\
--refSnps ref.snps.gz --haps mach/step1.gz\
--snps mach/example.snplist\
--rounds 100 --states 200\
--phased --probs\
--prefix mach/step2
```

The command options *--refHaps ref.hap.gz* and *--refSnps ref.snps.gz* are used to specify reference haplotypes and References snp list respectively. These data can be downloaded from MaCH download page.

3. After the completion of imputation process, **fcGENE** can be used to convert the imputed genotypes back to plink format. This can be done as explained in next subsection.

6.3 Format conversion of mach-imputed data

Important output files which we obtain from imputation processes performed with MaCH and minimac are listed in Table 6.3.

- To change the format of the genotype data imputed with option *--geno*, we can use the following command.

▶
```
./fcgene --mach-geno mach/example_output.geno\
--mach-info mach/example_output.info\
--oformat plink \
--out plink/geno_plink
```

If you prefer to update the original “pedigree information " and “SNP information " contained in original genotype data, then following command should be applied.

Output file Names	**Description**
example_output.geno example_output.info	Imputation is performed with MaCH using - -*geno* option.
example_output.mlgeno example_output.mlinfo example_output.mldose example_output.mlprob example_output.mlqc	Imputation is performed with MaCH using - -*mle* - -*mldetails* option.
minimac_output.info minimac_output.prob	Imputation is performed with minimac using - -*probs* - -*phased* option.

Table 6.2: Table showing the outputs from imputation with MaCH.

```
► ./fcgene - -mach-geno mach/example_output.geno - -mach-info mach/example_output.info\
  - -pedinfo mach/example_pedinfo.txt\
  - -snpinfo mach/example_snpinfo.txt\
  - -oformat plink\
  - -out plink/geno_plink
```

- *example_output.mlgeno* file contains the best-guess (i.e., most likely) genotype for each individual at each SNP and is generated after completion of imputation process with MaCH. In order to change the format of MaCH mlgeno back into the plink format, the following command can be applied.

```
► ./fcgene - -mach-mlgeno mach/example_output.mlgeno\
  - -mach-mlinfo mach/example_output.mlinfo \
  - -pedinfo mach/example_pedinfo.txt\
  - -snpinfo mach/example_snpinfo.txt\
  - -oformat plink \
  - -filter-snp hwe=1e-2 - -rsq 0.3 - -maf-thresh 0.1\
  - -out plink/mlgeno_plink
```

Remark: Options: - -*rsq 0.3* and - -*maf 0.1* are optional. You can also use other values for these options. Option - -*rsq 0.3* will filter those SNPs which have *Rsq* value smaller or equal to 0.3 and change the format of only those SNPs which have $Rsq \geq 0.3$. *Rsq* is a measure which estimates the squared correlation between imputed and true genotypes[2]. Similarly, - -*maf 0.1* will filter those SNPs which have minor allele frequency smaller or equal to 0.1 and Those SNPs which have minor allele frequency greater than 0.1 will be written in the new genotype file.

- *example_output.mlprob* file contains the posterior probabilities for the “AA" and “AB" genotypes at each marker for each individual. If you prefer to use this file for changing the genotype format back into the plink format then, you can use the following command.

```
► ./fcgene - -mach-mlprob mach/example_output.mlprob \
  - -mach-mlinfo mach/example_output.mlinfo \
  - -pedinfo mach/example_pedinfo.txt \
  - -snpinfo mach/example_snpinfo.txt\
    - -oformat plink - -filter-snp hwe=1e-2\
    - -rsq 0.3 - -maf-thresh 0.1 - -out plink/mlprob_plink
```

This command considers the genotype having the maximum probability value among the three possible genotypes (“AA",“AB", and “BB") as the true genotype and will convert it into plink format. Table 6.3 shows the necessary commands to upload mach-imputed files in fcGENE. After uploading the imputed data, we can use further command options for the function we are interested in.

Format name	Command option	File name
mach-imputed:	- -mach-geno	example.geno
geno and info files:	- -mach-info	example.info
mach-imputed:	- -mach-mlinfo	example.mlinfo
mlgeno and mlinfo files:	- -mach-mlgeno	example.mlgeno
mach-imputed:	- -mach-mlprob	example.mlprob
mlprob and mlinfo files:	- -mach-mlinfo	example.mlinfo

Table 6.3: Table showing necessary files and command options to upload mach-imputed data.

Remark 6.3 *Mach-imputed data can be converted not only into plink formatted data but also all other possible formats. A list of command options that can be used to get output from fcGENE are given for example in Table 6.5.*

6.4 Format conversion of minimac-imputed data

minimac_output.prob file obtained from the imputation process with *minimac*, is similar to *example_output.mlprob* obtained from the imputation with MaCH. Only *minimac_output.info* has more information than in *mach_output.mlinfo*. Using the following command, one can convert the genotype format of minimac imputed data into plink format.

```
► ./fcgene - -minimac-prob mach/minimac_output.prob \
  - -minimac-info mach/minimac_output.info \
  - -pedinfo mach/example_pedinfo.txt \
  - -filter-snp hwe=1e-2\
  - -snpinfo mach/example_snpinfo.txt\
  - -oformat plink - -rsq 0.3 - -maf-thresh 0.1 - -filter-snp hwe=1e-2\
  - -out plink/minimac_plink
```

This command considers the genotype having the maximum probability value among the three possible genotypes (“AA",“AB", and “BB") as the true genotype and will convert it into plink format. Similarly the necessary commands required to upload minimac-imputed data into fcGENE is given in Table 6.4.

Remark 6.4 *Detailed information on the command options used here is given in section 6.3.*

To convert minimac results into the formats other thank plink’s ped and map file one can use different options after “- -oformat”. Different outputs that can be obtained with option “- -oformat” are listed also in Table 6.5.

Format name	Command option	File name
minimac-imputed: prob and info files:	- -minimac-prob	example.prob
	- -minimac-info	example.info

Table 6.4: Table showing necessary files and command options to upload minimac-imputed data.

Commands to read files i.e. to make input in fcGENE	Command option to generate files	Assignment of output file name
./fcgene - -minimac-prob minimac_output.prob\ - -minimac-info minimac_output.info\ - -pedinfo mach/example_pedinfo.txt \ - -snpinfo mach/example_snpinfo.txt\ - -filter-snp hwe=1e-2\ - -rsq 0.3 - -maf-thresh 0.1\ - -filter-snp hwe=1e-2	- -oformat haploview	- -out outFileName
	- -oformat plink-dosage	
	- -oformat plink-recodeAD (- -recodeAD type)	
	- -oformat plink-recodeA (- -recodeA type)	
	- -oformat mach	
	- -oformat minimac	
	- -oformat impute	
	- -oformat snptest	
	- -oformat eigensoft	
	- -oformat beagle	
	- -oformat bimbam	
	- -oformat r (0,1,2 coding)	
	- -oformat r –transpose (0,1,2 coding)	
	- -oformat r-dose –transpose (0,1,2 coding)	
	- -oformat recodeA-dose plink-type recodeA	
	- -oformat vcf vcf format	
	- -oformat genable GenABEL format	
	- -oformat phase fastPhase format	
	- -oformat plink-bed plink binary format	
	- -oformat plink plink format	

Table 6.5: Table showing commands to generate files from minimac-imputed data.

Remark 6.5 *Genotype transformations explained in Table 6.5 can also be performed for all kinds of mach-imputed results.*

Chapter 7

Quality control, format conversion and imputation with IMPUTE

7.1 Data transformation

Some examples of data transformation into IMPUTE-format are given below.

►
```
./fcgene --ped plink/example.ped --map plink/example.map\
--snpinfo impute_ref.legend \
--oformat impute\
--out impute/plink_impute
```

►
```
./fcgene --ped mach/example.ped --dat mach/example.dat\
--oformat impute\
--snpinfo impute_ref.legend \
--out impute/mach_impute
```

►
```
./fcgene --bfile plink/example\
--oformat impute\
--out impute/mach_impute\
--snpinfo impute_ref.legend \
```

►
```
./fcgene --rgeno example_genotype.txt\
--snpinfo alleleinfo.txt \ --oformat impute\
--out impute/mach_impute
```

- Above commands write the following impute-formatted files.

  *− > “*plink_impute.gens*"

  *− > “*plink_impute.strand.txt*"

  *− > “*plink_impute.command.txt*"

  *− > “*plink_impute_pedinfo.txt*"

  *− > “*plink_impute_snpinfo.txt*"

- "*plink_impute.gens*" file contains SNP genotypes in the form of genotype probabilities. This is the main file used in IMPUTE and has a one-line-per-SNP format and generally it possesses a suffix $*.gens$. The first 5 entries of each line contain SNP ID, RS ID of the SNP, base-pair position of the SNP, the allele coded A and the allele coded B[3] respectively. Genotypes of impute formatted data are described as the probabilities of three genotypes AA, AB and BB at each SNP. For the first individual in the cohort, the first SNP contains these three probabilities on the sixth, seventh and eighth column of the file. Similarly the next three numbers contained in ninth, tenth and eleventh columns are the genotype probabilities for the second individual in the cohort. The whole genotype file contains a matrix with size no of SNPS $\times$ $(5+3\times$ no of individuals$)$.
- "*plink_impute.strand.txt*" file contains strand Information. Since fcGENE updates the first and second allele, strand information will be always positive("+").
- "*plink_impute.command.txt*" contains impute command that can be used for imputation. Note that command written in the file is just an example telling how you can impute your genotype data with IMPUTE. You can modify the command at necessary places so that the command can be executed with IMPUTE properly. If your genotype data has a region of larger than 7Mb, IMPUTE software providers recommend to split the genotyping region into small chunks for imputation purpose. These chunks can be imputed in parallel on multiple computer processors. fcGENE writes a table in "*plink_impute.command.txt*" file which describes the lower are upper boundaries (in base pair position) of the region in which imputation should be carried out. Moreover, you will find all the commands for each of the splitted region.
- "*plink_impute_pedinfo.txt*" contains pedigree information (individual id, sex information, diseases status etc) of all individuals contained the genotype file.
- Similarly "*plink_impute_snpinfo.txt*" contains SNP information (rsid, snpid, base pair position, distance in centimorgan first and second allele).
- These two "*pedinfo.txt*" and "*snpinfo.txt*" files can be used as extra information when converting imputed genotypes back into the plink format. For more information on "*pedinfo.txt*" and "*snpinfo.txt*" files, please go to sub section 3.

Impute-formatted chiamo-formatted gens file can be converted into different formats by using command option "- -gens". Moreover the necessary files and command option to upload impute-formatted data is given in Table 7.1.

Format name	Command option	File name
impute-formatted or chiamo-formatted	- -gens	example.gens

Table 7.1: Table showing necessary files and command options to upload impute-formatted data.

7.2 Quality control and format conversion

To perform quality control of the data first and then to generate impute-formatted data, one should use "- -filter-snp" and "- -filter-indiv" option to give the quality threshold as follows.

```
► ./fcgene - -ped plink/example.ped - -map plink/example.map\
  - -filter-snp maf=0.1,hwe=1e-2\
  - -oformat impute - -out impute/plink_impute
```

```
► ./fcgene - -ped plink/example.ped - -map plink/example.map\
  - -filter-indiv crate=0.95\
  - -oformat impute\
  - -out impute/plink_impute
```

```
► ./fcgene - -ped plink/example.ped - -map plink/example.map\
  - -filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  - -filter-indiv crate=0.95\
  - -oformat impute\
  - -out impute/plink_impute
```

If the original genotype data is given in other format thank PLINK, then impute-formatted files can be generated by uploading the files with corresponding command options.

7.3 Data transformation with HapMap-reference panel

If imputation is planned with IMPUTE using HapMap References, then it is most important to update first the allele information according to HapMap legend file. This file can be downloaded from IMPUTE homepage[3]. Note that the legend file should contain at least two columns namely the columns allele information. Moreover, just like as given in the HapMap legend file(given IMPUTE homepage[3]), the first line of “hapmap.legend" must be a header specifier as below .

rsID (or rsid)   snpID (or snpid)   position (or bp)   cm_pos   a0 (or allele1)   a1 (or allele2)

However the sequential order of the columns can be different and it is also not necessary to exist all of the columns. An example of HapMap legend file is given below.

snpid	rsid	position	a0	a1
snp1	rs4819391	14550436	A	G
snp2	rs11089128	14560203	C	T
snp3	rs11912265	14715506	A	C

Using the HapMap legend file as reference, one can update first the SNP information and then convert the genotype format into the format used by IMPUTE software as below.

```
► ./fcgene - -ped plink/example.ped - -map plink/example.map\
  - -oformat impute\
  - -snpinfo hapmap.legend - -out impute/plink_impute
```

Remark 7.1 *It is not necessary that all the SNPs containing in legend file are also contained in the used genotype data. Also the sequential order of SNPs, in plink formatted data and legend file must not be the same.*

7.4 Imputation with IMPUTE

Imputing genotype data with IMPUTE is a bit more complicated than MaCH. This is because IMPUTE needs a couple of files and many parameters that also need to be determined first. However, to facilitate the imputation process, the software provider has prepared a couple of necessary files required by IMPUTE. The necessary files are listed below.

- A genetic map file containing position, combined genetic recombination rate and genetic map in centimorgen. This file is based on hapmap based or 1000 Genome base SNPs and may certainly contain more SNPs than required. A typical map file is given a form illustrated below.

position	COMBINED_rate.cM.Mb.	Genetic_Map.cM.
20299958	0.0056794611	15.0106101777
20301041	0.0056794611	15.0106163286
20303319	0.0056794611	15.0106292664

- A SNP legend file which contains rsid, SNP position and first and second allele information. A Typical legend file is given as

rsID	position	a0	a1
rs4819391	14550436	A	G
rs11089128	14560203	A	G
rs11912265	14715506	A	C
rs4321465	14836970	C	T

  It is important to make sure that your genotype data is compatible with rsid, base-pair positions and allele information given in the legend and genetic map file. That is why this file is also necessary to use with - -*snpinfo* option while converting from other genotype format to IMPUTE format.

- a file containing haplotype as the references.
  Remark: This haplotype file including genetic map and legend file can be downloaded from IMPUTE website
  `https://mathgen.stats.ox.ac.uk/impute/impute_v2.html#download_reference_data`

- a strand file containing SNP position in first column and strand in second column as shown in below.

15638313	+
15643854	+
15644565	+
15270159	+
15272858	+

- A gens file containing genotype probability for each SNPs at each individual: The gens file containing three SNPs and three individuals is illustrated below.

rs5994011	rs5994011	15638313	A	T	1	0	0	0.5	0.3	0.2	0	1	0
rs5748621	rs5748621	15643854	C	G	0	1	0	0.8	0.1	0.1	1	0	0
rs5748622	rs5748622	15644565	G	T	0.3	0.1	0.6	0	0	1	0	0	1

Remark 7.2 *These two files(gens and strand) can be obtained form using Fcgene software using command given at the start of the previous sub section.*

7.4.1 Commands for imputation

Before starting the imputation process, make sure that current version IMPUTE2 is saved in your current directory as impute2 and you have downloaded HapMap or 1000 Genome references. Then one can use the following command.

```
► ./impute2 -m hapmap3_r2_b36/genetic_map_chr10_combined_b36.txt\
  -h hapmap3_r2_b36/hapmap3.r2.b36.all.chr22.haps\
  -l hapmap3_r2_b36/hapmap3.r2.b36.all.chr22.legend\
  -g example/example.chr22.gens\
  -strand_g example/example.chr22.strand\
  -int 20.4e6 20.5e6 -Ne 20000\
  -pgs\
  -o example/example.chr22.impute2
```

Above example command illustrates typical applications of IMPUTE2.

- Three files namely *genetic_map_chr10_combined_b36.txt*, *hapmap3.r2.b36.all.chr22.haps* and *hapmap3.r2.b36.all.chr3.legend* are HapMap reference files and contained in the directory *hapmap3_r2_b36*.

- Similarly *example.chr22.gens* and *example.chr22.strand* can be obtained by converting the format of genotype data with FCgene.

- Two numbers after *-int* commands are lower are upper boundaries (in base pair position) of the region in which imputation should be carried out.

- *-Ne* option is an internal parameter of IMPUTE and it sets effective population size that scales the fine-scale recombination map for the given population.

- The option *-pgs* (meaning "Predict Genotyped SNPs") tells the program to impute all the missing and non missing genotypes and replace the input genotypes of gens file with imputed genotypes and write them in the -o file. Instead of using the option *-pgs*, one can use *-pgs_miss* which will predict only the missing genotypes of gens file.

- Option given with *-o* tells impute where to save the impute output files.

Similarly If you prefer to impute your genotype data without using any kind of reference panel, this is also possible using option *-phase* in IMPUTE. However, you still needs the genetic map file to impute genotype data with IMPUTE. A typical example for imputing genotypes without using references is given below.

```
► ./impute2 -phase \
  -m hapmap3_r2_b36/genetic_map_chr10_combined_b36.txt\
  -g example/example.chr22.gens\
  -strand_g example/example.chr22.strand\
  -int 20.4e6 20.5e6 \
  -Ne 20000 -pgs -call_thresh 0.8\
  -o example/example.chr22.impute2
```

For more information on imputation process through IMPUTE, we recommend to visit the website `https://mathgen.stats.ox.ac.uk/impute/impute_v2.html`.

7.4.2 Imputing large genomic region

As explained previously, if the genomic region is larger than 7Mb, it is recommended to split first the region int small chunks and then to impute each chunk by mentioning the lower and upper boundaries of the region with the parameter *-Ne*. *fcGENE* provides you the upper and lower boundaries of each of the chunks and the commands necessary to use for each chunk. Each chunk will have the size in nearly 5 Mb. We recommend you

- to look on the command file that you obtain from *fcGENE* while converting your genotype data into impute format.
- Make necessary modification on the commands (if necessary) of each chunk to make sure that the command suits with your data.
- Impute each chunk of the chromosome separately or parallel on different computer.
- At the end of the imputation process, the impute outputs of each chunk imputed separately should be combined into a single whole region of chromosome. On Linux-based systems, you can simply type a command like this:
  cat impute2_chunk1 impute2_chunk2 impute2_chunk3 >impute2_all

7.5 Format conversion of impute-imputed data

Imputation through IMPUTE provides us with the following main important files.

- main output: *example.chr22.impute2*
- SNP QC info: *example.chr22.impuete_info*
- Sample QC info: *example.chr22.impuete_info_by_sample*

Main output file contains the imputed genotype probabilities of each SNP at each individual.SNP QC info file contains an information metric in the fifth column. This metric is similar to the r-squared metrics reported by other programs like MaCH and Beagle[3] and can be used for the SNP filtering after imputation. Generally the cutoffs used for post-imputation SNP filtering lies in between 0.3 and 0.5, but the impute software providers warn that the right threshold for your analysis may differ. Our recommendation is to use 0.3. Using the two files namely main output and SNP QC info, we can obtain best guess genotypes for plink format as follows.

- Decide which cutoffs you want to use for your post-imputation SNP filtering. *fcGENE* takes this cutoffs with the option *- -info-thresh* .
- Use the following *fcGENE* command to convert impute-imputed genotypes into plink formatted best-guess genotypes.

  ```
  ► ./fcgene --gens example/example.chr22.impute2 --thresh 0.9 \
    --info example/example.chr22.impute2_info \
    --info-thresh info_thresh_value \
    --maf-thresh maf_thresh_value\
    --filter-snp hwe=1e-2\
  ```

```
--pedinfo example/impute.pedinfo\
--oformat plink  --out example/impute_plink
```

- In above Command Command, option *--thresh* is optional, if this option is not used then *Fcgene* uses 0.0 for *maxProb* value and does not set to any genotypes to missing. For more information see remarks below.
- Option *--info* and *--info-thresh* are also optional. If you do not use these option, *fcGENE* takes 0.0 as default value and does not make any kind of post filtering of SNPs. If you prefer post-imputation filtering of SNPs, then make sure that you have used this option with your preferred cutoffs value. Moreover, Options *--info* and *--info-thresh* can be uses only when both options are given.
- Filtering of SNPs with *--maf-thresh* option is also not a compulsory option. If you use this option then SNPs having *maf* lower than the given value will be filtered out while doing further analysis.

Remark 7.3

- *Impute formatted genotype data is fully dependent of first and second allele used when creating the data. If the first and second alleles should be exchanged, it is important first to update the allele information before converting the impute formatted data into any other formats.*
- *While changing the format of impute formatted genotype data, one can either choose the best guess genotype (genotype that has the maximum probability) at each SNP and at each individual level or one can use a threshold option* --thresh maxProb *to make sure that the corresponding genotype has a genotyping probability greater than* maxProb. *If the maximum probability of predicted genotypes is less than* maxProb*, then the particular genotype at the SNP and individual will be set to missing. This kind of option will enrich the confidence of choose genotypes at the SNP.*

To convert imputed-imputed data into other formats, one can use different options for “--oformat ” as described in Table 7.2.

Commands to read files i.e. to make input in fcGENE	Command option to generate files	Assignment of output file name
./fcgene - -gens example/example.chr22.impute2\ - -thresh 0.9 \ - -info example/example.chr22.impute2_info \ - -info-thresh info_thresh_value \ - -filter-snp hwe=1e-2 - -maf-thresh 0.1 \ - -pedinfo example/impute.pedinfo	- -oformat haploview	- -out outFileName
	- -oformat plink	
	- -oformat plink-dosage	
	- -oformat plink-recodeAD (- -recodeAD type)	
	- -oformat plink-recodeA (- -recodeA type)	
	- -oformat mach	
	- -oformat minimac	
	- -oformat impute	
	- -oformat snptest	
	- -oformat eigensoft	
	- -oformat beagle	
	- -oformat bimbam	
	- -oformat r (0,1,2 coding)	
	- -oformat r –transpose (0,1,2 coding)	
	- -oformat r-dose (0,1,2 coding)	
	- -oformat recodeA-dose plink-type recodeA	
	- -oformat vcf vcf format	
	- -oformat genable GenABEL format	
	- -oformat phase fastPhase format	
	- -oformat plink-bed plink binary format	

Table 7.2: Table showing commands to generate files from impute-imputed data.

Chapter 8

Quality control, format conversion and imputation with BEAGLE

BEAGLE is written in Java and can be run on any computer(windows, Unix, Linux, Mac) with a Java version 1.6 interpreter. Since BEAGLE is also designed to analyze large-scale data sets with hundreds of thousands of markers genotyped on thousands of samples[6], it can phase genotype data for unrelated individuals as well as related individuals. With BEAGLE, one can also estimates identity-by-descent (IBD) probabilities and homozygosity-by-descent (HBD) probabilities from called genotypes. If you prefer BEAGLE for your imputation process, then you can use the softwares in a sequential way as mentioned below.

PLINK ⇒ fcGENE ⇒ BEAGLE ⇒ fcGENE ⇒ PLINK

If the original data are given in a format other than plink pedigree, the same procedure can be used for the analysis. The process for changing different data into beagle format and the information about the necessary files is given in following subsection.

8.1 Data transformation to beagle-format

-
  ```
  ./fcgene --ped plink/example.ped --map plink/example.map\
  --oformat beagle --out beagle/plink_beagle
  ```

-
  ```
  ./fcgene --ped mach/example.ped --dat mach/example.map\
  --oformat beagle --out beagle/plink_beagle
  ```

-
  ```
  ./fcgene --mfile mach/example \
  --oformat beagle --out beagle/plink_beagle
  ```

-
  ```
  ./fcgene --rgeno example_genotypes.txt \
  --snpinfo alleleinfo.txt \
  --oformat beagle --out beagle/r_beagle
  ```

- Above commands write BEAGLE format file namely "*plink_impute.bgl*". This file is needed to impute the missing genotype data and to infer the haplotypes of unphased data.
- At the same time, two extra files are saved: "*plink_beagle_pedinfo.txt*" and "*plink_beagle_snpinfo.txt*".
- These two "*pedinfo.txt*" and "*snpinfo.txt*" files can be used as extra information when converting imputed genotypes back into the plink format. For more information on "*pedinfo.txt*" and "*snpinfo.txt*", we recommend you to visit subsection 3.

Remark 8.1 *Beagle-formatted file can also be uploaded in fcGENE to convert it into many other formats of GWA analysis tools. Table 8.1 shows the necessary files and command to upload the data. One can use many other optional command-options as well.*

Format name	Command option	File name
beagle- bgl	- -bgl	example.bgl

Table 8.1: Table showing necessary files and command options to read beagle-formatted data.

8.2 Quality control and format conversion

In order to exclude those SNPs which pass the quality criteria, we do not need to calculate snp-wise quality measures like HWE,MAF and call rate separately. We can use "- -filter-snp" and/or "- -filter-indiv" commands for by giving the cutoff threshold for each measure as mentioned in the following examples.

► *./fcgene - -ped plink/example.ped - -map plink/example.map*
- -oformat beagle - -out beagle/plink_beagle
- -filter-snp maf=0.1,crate=0.95,hwe=1e-2

► *./fcgene - -ped plink/example.ped - -map plink/example.map*
- -oformat beagle - -out beagle/plink_beagle
- -filter-indiv crate=0.95

► *./fcgene - -ped plink/example.ped - -map plink/example.map*
- -oformat beagle - -out beagle/plink_beagle
- -filter-snp maf=0.1,crate=0.95,hwe=1e-2
- -filter-indiv crate=0.95

Remark 8.2 *We can also convert other type of file formats into beagle-format. For this purpose we first need to upload data by using the command option explained in Table 1.2 and the corresponding chapter of the format in which genotype data are given.*

8.3 Imputation with BEAGLE

Before starting imputation with beagle, make sure that you have downloaded BEAGLE program and the downloaded program is executable. Downloaded files or programs do not allow them to execute automatically. If you are not permitted to execute them, then you can change permission by using *chmod* command in Linux/Unix based computer. For example, if the program is saved as "beagle.bgl", then you can use the command "*chmod 755 beagle.bgl* " which allows you to execute the program.

8.3.1 Imputation without using references

To perform imputation with Beagle without using any kind of references, the necessary arguments for the beagle command are:

- use of argument for specifying input files,
- use of the output prefix, and
- use of missing allele code,

A typical beagle command used for imputation process can be written as

```
► java -jar beagle.jar unphased=beagle/example.bgl\
  missing="?" niterations=20 gprobs=true out=beagle/example_output
```

- The genotype file *example/beagle.bgl* is given with *unphased*= option,
- Missing genotypes are denoted by "?" and this is explained in the command line with option *missing*=. If you have denoted the missings with other symbols like "NA" or "0", then you can use *missing=NA* or *missing =0.*
- *niterations* is an optional argument which gives the number of iterations of the phasing algorithm. If an odd integer is specified, the next even integer is used. The default value of *niterations* is 10.
- The argument *grobs=true* is also optional. If you use this option, BEAGLE will produce genotype probabilities for each genotype and it also calculates the imputation accuracy for each SNPs given in a separate file.
- Finally option *out*= determines the output file name. Above mentioned command is applied when you have only one genotype data to impute and you are are interested in imputing it without using any kind of references.

8.3.2 Imputation using references

A typical command of BEAGLE for imputing missing and ungenotyped markers can be described as

```
► java -jar beagle.jar unphased=example/fileA.bgl \
  phased= example/fileB.bgl markers=markers.txt \
  missing="?" niterations=20 gprobs=true \
  out=example/beagle_output
```

Above mentioned file “*fileA.bgl*" is supposed to be imputed using a reference panel given in the file “*fileB.bgl*". The markers file named as “*marker.txt*" lists all the markers contained in file *fileB.bgl* in chromosomal order. As mentioned previously missing are denoted by “?". If you are using unphased data as your reference panel, then use the option “*unphased = filleB.bgl* instead of using *unphased=fileB.bgl.*

For more information on imputation process with BEAGLE, we recommend you to visit the official beagle-website:
`http://faculty.washington.edu/browning/beagle/beagle.html`.

8.4 Format conversion of beagle-imputed data

After the completion of imputation process with BEAGLE, it creates a log file (.log) that summarizes the analysis. Other files created by using the imputation command mentioned above are

- a phased file (.phased) (example.bgl.phased.gz).
- a file containing genotype probabilities (example.bgl.gprobs.gz)
- Genotype dosage file (example.bgl.dose.gz)
- the allelic r^2 file (example.bgl.r2)

The phased file gives the most likely pair of phased haplotypes for each sample. Beagle-grpobs contains three possible probabilities for each genotype if input Beagle file contains only diallelic markers. Genotype dose file contains genotype dose for each genotype. For example, if the genotype probabilities of a sample are given as $p(AA), p(AB)$ and $p(BB)$, then the allele dose is defined as $0p(AA) + 1p(AB) + 2p(BB)$. The file containing estimated squared correlation R^2 is given for each marker. This correlation value can be used to detect the imputation quality. In order to convert the genotype format of beagle output into PLINK format, *fcGENE* considers the phased file or file containing genotype probabilities as main input file and the allelic R^2 file (.r2) as an optional input. If allelic R^2 file (beagle.bgl.r2) file is also given as input in *fcGENE*, then it is necessary to use the command option - -*rsq-thresh* as well. Typical commands to change the beagle-outputs (if phased data is used) into PLINK format are given below.

- If you prefer just to convert the format of beagle phased data, the following command can be used in *fcGENE.*

  ► *./fcgene - -bgl beagle/example.bgl.phased*\
  - -oformat plink - -out plink/beagle_plink

- To convert the beagle-imputed file containing genotype probabilities (example.bgl.gprobs):

  ► *./fcgene - -bgl-gprobs beagle/example.bgl.gprobs*\
  - -oformat plink - -out plink/beagle_plink

- Generally we recommend you to update pedigree information and SNP information before converting the beagle-imputation output into PLINK format. To change the format of beagle-imputed data together with updating the pedigree and SNP information, the following commands can be used.

```
► ./fcgene - -bgl beagle/example.bgl.phased\
- -oformat plink\
- -out plink/beagle_plink\
- -pedinfo beagle/plink_beagle_pedinfo.txt\
- -snpinfo beagle/plink_beagle_snpinfo.txt
```

```
► ./fcgene - -bgl-gprobs beagle/example.bgl.gprobs\
- -oformat plink\
- -out plink/beagle_plink\
- -pedinfo beagle/plink_beagle_pedinfo.txt\
- -snpinfo beagle/plink_beagle_snpinfo.txt
```

- If you prefer also to use beagle allelic r^2 file to check the quality of imputed SNPs, then the following commands can be used.

```
► ./fcgene - -bgl beagle/example.bgl.phased\
- -bgl-rsq beagle/example.bgl.r2\
- -rsq-thresh 0.3 \
- -filter-snp hwe=1e-2,maf=0.1\
- -oformat plink\
- -out plink/beagle_plink\
- -pedinfo beagle/plink_beagle_pedinfo.txt - -snpinfo beagle/plink_beagle_snpinfo.txt
```

```
► ./fcgene - -bgl-gprobs beagle/example.bgl.gprobs\
- -bgl-rsq beagle/example.bgl.r2\
- -rsq-thresh 0.3  - -oformat plink \
- -filter-snp hwe=1e-2,maf=.1\
- -out plink/beagle_plink\
- -pedinfo beagle/plink_beagle_pedinfo.txt\
- -snpinfo beagle/plink_beagle_snpinfo.txt
```

To remove the poorly imputed SNPs, it is recommended to use *- -rsq-thresh 0.3*. That means only those SNPs which has satisfy the condition $R^2 > 0.3$ will be considered as well imputed SNPs.

Remark 8.3 *If you convert the "beagle.gprobs" data into plink format,* fcGENE *will compare the probabilities of three possible genotypes (AA, AB and BB) and take the genotype which as the maximum probability among them.*

Aside converting into plink-formatted pedigree files, we can convert the beagle-imputed data into the formats of many other GWA analysis tools. This can be done by changing command option after "- -oformat". Different combination from Table 8.3 can be used in fcGENE to convert beagle-imputed data.

Remarks 8.4

- *To convert "beagle.gprobs" data into other formats, the same procedure as explained in Table 8.3 can be applied.*
- *Table 8.4 shows the necessary files and command options required to upload beagle-imputed data into fcGENE. However one can also use other command options like quality measure threshold and update of snp- and individual information etc.*

Format name	Command option	File name
beagle- phased data	- -bgl	example.bgl.phased
genotype probability distribution data	- -bgl-gprobs	example.bgl.gprobs

Table 8.2: Table showing necessary files and command options to read beagle-imputed data.

Commands to read files i.e. to make input in fcGENE	Command option to generate files	Assignment of output file name
./fcgene - -bgl beagle/example.bgl.phased\\ - -bgl-rsq beagle/example.bgl.r2\\ - -rsq-thresh 0.3 \\ - -pedinfo beagle/plink_beagle_pedinfo.txt\\ - -snpinfo beagle/plink_beagle_snpinfo.txt\\ - -filter-snp hwe=1e-2,maf=0.1	- -oformat haploview	- -out outFileName
	- -oformat plink	
	- -oformat plink-dosage	
	- -oformat plink-recodeAD (- -recodeAD type)	
	- -oformat plink-recodeA (- -recodeA type)	
	- -oformat mach	
	- -oformat minimac	
	- -oformat impute	
	- -oformat snptest	
	- -oformat eigensoft	
	- -oformat beagle	
	- -oformat bimbam	
	- -oformat r (0,1,2 coding)	
	- -oformat r –transpose	
	- -oformat r-dose	
	- -oformat recodeA-dose plink-type recodeA	
	- -oformat vcf	
	- -oformat genable GenABEL format	
	- -oformat phase fastPhase format	
	- -oformat plink-bed plink binary format	

Table 8.3: Table showing commands to generate files from beagle-imputed data.

Chapter 9

Quality control, format conversion and imputation with BIMBAM

BIMBAM implements Bayesian method to impute to fill in missing genotypes or untyped genotypes. BIMBAM tries first to capture the patterns of Linkage disequilibrium information from the genotyped SNPs and reference panel (if given) and then performs the imputation by providing

1. genotype distribution (with -*wgd* option)
2. mean genotype (with -*wmg* option)
3. best guess genotypes (with -*wbg*) option.

A possible order of using GWA analysis software while using BIMBAM as imputation tool is given below.

PLINK ⇒ fcGENE ⇒ BIMBAM ⇒ fcGENE ⇒ PLINK

9.1 Data transformation to bimbam-format

BIMBAM software generally needs three basic files as input.

- *example.geno.txt*: *geno.txt* file contains the genotype information of a cohort. An example of this basic file is given below.

  ```
  5
  3
  IND,  ind1, ind2, ind3, ind4, ind5
  snp1, ??,   CT,   CC,   ??,   ??
  snp2, GT,   TG,   TT,   TG,   ??
  snp3, GG,   AA,   AG,   GG,   AA
  ```

- *example.pheno.txt pheno.txt* contains just one column indicating the phenotype value for each individual. The individual's phenotype value should have the same order as in the Genotype file. Binary phenotypes like in case control study contain 0 or 1 for each individual. If the phenotype is missing then , the individual's phenotype should be denoted by "NA". An example of *pheno txt*

file can be given as
0
1
1
0
NA

- *example.pos.txt* This file should contain two, or three, columns, with the first column being the SNP name, and the second column being its physical location. Third column can contain the chromosome number. Order of SNPs rows is not necessary but all the SNPs contained in *geno.txt* must contain also in *post.txt*. An example of *pos.txt* is described below.

*snp*1,	14560203	22
*snp*2,	14715506	22
*snp*3,	14836970	22

For more information on the format of BIMBAM input files, we recommend you to visit, the official website of BIMBAM

`http://www.bcm.edu/cnrc/mcmcmc/bimbam`

To generate these files needed by BIMBAM from the plink-formatted data, the following command can be used.

```
► ./fcgene - -ped plink/example.ped\
  - -map plink/example.map\
  - -oformat bimbam\
  - -out bimbam/plink_bimbam
```

Above commands saves three basic files required for the imputation with BIMBAM (*“geno.txt"*,*“pheno.txt"* and *“pos.txt"*) and two extra files namely *pedinfo.txt*and *snpinfo.txt*. Previously mentioned two extra files: *pedinfo.txt*and *snpinfo.txt* can be used while converting the format of bimbam-imputed data back to PLINK format.

Remarks 9.1

- *Bimbam input files can also be generated from data given in other formats e.g. MaCH,impute, beagle etc.*

- *Bimbam-formatted input data can be converted into other different formats.Table 9.1 shows the necessary files and command options required to upload bimbam-formatted data into fcGENE. However one can also use other optional command options like quality control and/or update of snp- and individual information etc.*

9.2 Quality control and format conversion

In order to generate bimbam-formatted data with quality control process, we can apply following commands to fcGENE.

Format name	Command option	File name
bimbam-formatted data	- -geno - -pos	example.geno.txt example.pos.txt

Table 9.1: Table showing necessary files and command options to upload bimbam-formatted data.

```
► ./fcgene - -ped plink/example.ped\
  - -map plink/example.map\
  - -filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  - -oformat bimbam\
  - -out bimbam/plink_bimbam
```

```
► ./fcgene - -ped plink/example.ped \
  - -map plink/example.map\
  - -filter-indiv crate=0.95\
  - -oformat bimbam\
  - -out bimbam/plink_bimbam
```

```
► ./fcgene - -ped plink/example.ped\
  - -map plink/example.map\
  - -filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  - -filter-indiv crate=0.95\
  - -oformat bimbam\
  - -out bimbam/plink_bimbam
```

9.3 Imputation with BIMBAM

A simple type of imputation with BIMBAM can be performed by using the following command.

```
► ./bimbam -g bimbam/plink_bimbam.geno.txt\
  -p bimbam/plink_bimbam.pheno.txt\
  -pos bimbam/plink_bimbam.pos.txt\
  -e 10 -s 20 -c 15 -wbg -wmg -wgd\
  -o bimbam_imputed
```

Option *-g bimbam/plink_bimbam.geno.txt* is used to provide genotype file as input. With option *-p* and *-pos*, we can use phenotype information file *pheno.txt* and SNP location file *pos.txt*. With command option *-e 10*, we can ask BIMBAM to run EM algorithms 10 times. Option *-s 20* means that each EM should run 20 steps. To give the number of clusters made while performing the algorithm, we can use for example the option *-c 15*. Command option *- -nobf* is saying not to calculate Bayes factors at the end of the EM runs, and *-wmg* and *-wgd* are used to ask to write both genotype distribution and mean genotype files. Finally option *-o* is saying that each output file should have name starting from *“bimbam_imputed"*.

Remark 9.2 *Since BIMBAM creates automatically an directory named “output", current version of BIMBAM does not allow to use directory as output name with option* -o.

9.3.1 Imputation with BIMBAM using reference panel

BIMBAM software provider recommends to impute genotype data by using reference panel even though BIMBAM has the facility to impute genotype data without using reference panel. The command used for imputation with reference panel is basically the same as the command used for imputation without references. The only extra option used here is to give the reference panel as input. A typical BIMBAM command for imputation with reference panel has the form:

```
./bimbam -g reference/reference_panel.geno.txt -p 0\
-g bimbam/plink_bimbam.geno.txt\
-p bimbam/plink_bimbam.pheno.txt -pos bimbam/plink_bimbam.pos.txt\
-e 10 -s 20 -c 15 -nobf -wmg -wbg -wgd\
-o bimbam_imputed
```

Above command involves with the two genotype input files:

1. reference panel. The reference panel contains densely genotyped individuals, e.g. HapMap, 1000 Genomes. To mention that the first genotype file is not a cohort but a reference panel command option *-p 0* (Numeric 0 after -p) is used. To download bimbam-formatted HapMap or 1000 Genome reference panel, we recommend you to download from BIMBAM official website or you can write an email to `yongtaog@bcm.edu` requesting for these data.
2. cohort data. This the genotyped data which you want to impute.

If you have cohort input file not as the genotypes but as the genotype probability distribution (see its format in BIMBAM website), then the following command would be used for imputation purpose.

```
./bimbam -g reference/reference_panel.geno.txt -p 0\
-gmode 2 -g bimbam/plink_bimbam.genotype.distribution.txt\
-p bimbam/plink_bimbam.pheno.txt\
-pos bimbam/plink_bimbam.pos.txt\
-e 10 -s 20 -c 15 -nobf\
-wmg -wbg -wgd\
-o bimbam_imputed
```

Output files produced by BIMBAM imputation are listed below.

- SNP info file: *bimbam_imputed.snpdata.txt*. This file contains six columns namely the columns containing SNP ID, minor allele, major allele, minor allele frequency, chromosome, and position.
- if option *-wbg* is used, then a file *bimbam_imputed.best.guess.genotype.txt*, containing best guess genotypes will be saved.
- if option *-wmg* is used, then a file *bimbam_imputed.mean.genotype.txt*, containing best guess genotypes will be saved. The first column of this file contains SNP ID, the second and third columns contain allele types with minor allele first. The rest columns are the mean genotypes of different individuals - numbers between 0 and 2 that represents the (posterior) mean genotype, or dosage of the minor allele. An example of mean genotypes file of two SNPs and three individuals follows.[7]

- Using option *-wgd*, one can obtain the genotype probability distribution for each SNP at each individual namely *bimbam_imputed.genotype.distribution.txt*. The first three columns of this type of file are identical to those of the mean genotype file. The fourth and fifth columns denote the posterior probabilities ($p(AA)$ and $p(AB)$) of the SNP at first individual. Similarly six and seventh columns denote genotype probabilities ($p(AA)$ and $p(AB)$) of second individual and so on.

9.4 Format conversion of bimbam-imputed data

fcGENE can read *best.guess.genotype* file and *genotype.distribution* file using respectively the same options *–wbg* and *–wgd* used by BIMBAM for the imputation process. *snpinfo.txt* file obtained as output of BIMBAM can also be fed to *fcGENE* by using option *--snp-info* to update the allele SNP information. The possible commands of *fcGENE* to read bimbam-output files are given below.

- Reading *best.guess.genotype.txt* and Converting its format:

  ►
  ```
  ./fcgene --wbg output/bimbam_imputed.best.guess.genotype.txt\
  --pos output/bimbam_imputed.snpdata.txt\
  --maf-thresh 0.1 --filter-snp hwe=1e-6
  ```

  ►
  ```
  ./fcgene --wbg output/bimbam_imputed.best.guess.genotype.txt\
  --pos output/bimbam_imputed.snpdata.txtt\
  --maf-thresh 0.1 --oformat plink --out plink/bimbam_plink\
  --filter-snp hwe=1e-6
  ```

- To read *genotype.probability.distribution.txt* file and to convert its format:

  ►
  ```
  ./fcgene --wgd output/bimbam_imputed.genotype.probability.distribution.txt\
  --pos output/bimbam_imputed.snpdata.txtt\
  --maf-thresh 0.1
  ```

  ►
  ```
  ./fcgene --wgd output/bimbam_imputed.genotype.probability.distribution.txt\
  --pos output/bimbam_imputed.snpinfo.txt\
  --maf-thresh 0.1 --pedinfo bimbam/plink_bimbam_pedinfo.txt
  ```

  ►
  ```
  ./fcgene --wgd output/bimbam_imputed.genotype.probability.distribution.txt\
  --pos output/bimbam_imputed.snpinfo.txt\
  --maf-thresh 0.1 --pedinfo bimbam/plink_bimbam_pedinfo.txt\
  --oformat plink --out plink/bimbam_imputed --filter-snp hwe=1e-6
  ```

Remark 9.3 *Command option* --maf-thresh 0.1 *is given so that SNPs having minor allele frequency less than or equal to 0.1 are marked as poorly genotyped or low quality SNPs. The alternative way is to use "--filter-snp".*

To convert bimbam-imputed data (for example genotype probability distribution)into other GWA analysis tools, we can apply different combination of commands given in the columns of Table 9.2.

Remarks 9.4

- *Other type of outputs of bimbam-imputed data can be converted into different formats as explained in Table 9.2*

Commands to read files i.e. to make input in fcGENE	Command option to generate files	Assignment of output file name
./fcgene\ -wgd example.genotype.probability.distribution.txt\ --pos example.snpinfo.txt --maf-thresh 0.1\ --pedinfo bimbam/plink_bimbam_pedinfo.txt\ --oformat plink --out plink/bimbam_imputed\ --filter-snp hwe=1e-6	--oformat haploview	--out outFileName
	--oformat plink	
	--oformat plink-dosage	
	--oformat plink-recodeAD (--recodeAD type)	
	--oformat plink-recodeA (--recodeA type)	
	--oformat mach	
	--oformat minimac	
	--oformat impute	
	--oformat snptest	
	--oformat eigensoft	
	--oformat beagle	
	--oformat bimbam	
	--oformat r (0,1,2 coding)	
	--oformat r –transpose	
	--oformat r-dose	
	--oformat recodeA-dose plink-type recodeA	
	--oformat vcf vcf format	
	--oformat genable	
	--oformat phase fastPhase format	
	--oformat plink-bed plink binary format	

Table 9.2: Table showing commands to generate files from bimbam-imputed data.

- *Table 9.3 shows the necessary files and command options required to upload bimbam-imputed data into fcGENE. However one can also use other optional command options like quality control and/or update of snp- and individual information etc.*

Format name	Command option	File name
Best-guess genotype	--wbg --pos	example.best.guess.genotype.txt example.snpinfo.txt
genotype probability distribution	--wgd --pos	example.genotype.probability.distribution.txt example.snpinfo.txt

Table 9.3: Table showing necessary files and command options to upload bimbam-imputed data.

9.5 Format conversion of bimbam-input data

If the original files are given in bimbam-format and you are interested in processing your analysis for example with PLINK, you can use the following command to convert the bimbam-formatted data into plink-format.

```
► ./fcgene --geno example/bimbam_geno.txt --pos example/bimbam_pos.txt\
  --filter-snp hwe=1e-2,maf=0.1,crate=0.95\
  --oformat plink --out bimbam_plink
```

Chapter 10

Quality control and format conversion to SNPTEST

SNPTEST analyses the association of a SNP in GWA studies and this program is very popular because it was one of the programs used in the analysis of the 7 genome-wide association studies carried out by the Wellcome Trust Case-Control Consortium**??**. A pre-compiled version of the program can be downloaded from its homepage **??**.

10.1 SNPTEST's input files

SNPTEST basically needs two files namely:

1. A genotype file (ended with "*.gen*"). This file is similar to the ".gens" file used in imputation software "IMPUTE" and has only different ending name ".gen".
2. A file containing covariates and phenotype information. The name of this file should end with ".sample". The sample file must contain at least three columns namely family id, individual id and call rate for each individual. Moreover it consist of three parts
   (a) a header line,
   (b) a line containing the types of covariates and
   (c) lines containing covariate information.

   Each line should have same number of columns. The first three entries of header line must be "ID_1", "ID_2" and "missing" and the rest of the entries of this line can have any names representing the names of covariates and phenotypes given for each individual. The second line of sample file should determine the types of covariates and types of phenotypes. Covariate can be either of type "D", meaning discrete or of type "C", meaning continuous. Similarly Phenotypes can be either of type "P", meaning continuous or "B", meaning binary (for example 0=controls and 1=cases, sex: 0=female, 1=male etc) Note that the phenotype columns must appear after the covariate columns. That means C and D should appear before P and B.

A detail of the format of these files can be found in
`http://www.stats.ox.ac.uk/ marchini/software/gwas/file_format.html`

10.2 Converting files into snptest format

FCgene can convert the plink formatted files as well as the output of any imputation tools into snptest format.Therefore, one can test an association with SNPTEST directly after imputing the genotyped data with the preferred imputation tool. FCgene also gives you an option whether to filter first the SNPS which have low imputation quality. Moreover, the standard plink formatted covariate file can be feed to FCgene together with the output of an imputation tool so that FCgene can create snptest's ".gen" and ".sample" files.

10.2.1 Converting plink formatted data

To change the plink formatted "ped" and "map" file into snptest format, the following command is necessary.

```
./fcgene - -ped example/plink.ped - -map example/plink.map\
- -oformat snptest - -out example/plink_snptest
```

If you are interested in adding covariates from plink formatted covariate file (e.g. named "plink_cov.txt") to the snptest ".sample" file, then the following command should be used.

```
./fcgene - -ped example/plink.ped - -map example/plink.map\
- -covar example/plink_cov.txt\
- -covar-name pheno1,pheno2,covar_A,covar_B\
- -covar-type P,B,D,C\
- -oformat snptest - -out example/plink_snptest
```

Remarks 10.1

- *The family id and individual id in covariate file "plink_cov.txt" and must match with the individual ids and family ids given in ".ped" file.*
- *The covariate file "plink_cov.txt" can have more than necessary phenotypes and covariates. If you do not mention any variables with option "–covar-name" and "–covar-tpye", then FCgene just leaves the columns of these variables.*
- *As mentioned previously, snptest ".sample" file should contain the columns of covariates before the phenotype variables,however FCgene takes care of this fact alone and put the columns of phenotype variables at end. So you do not need to worry about the sequential order of variables.*

If some of covariates and phenotypes which are given in adjacent columns of plink formatted covariate file, have the same type then the command for converting into snptest format can also be written as

```
./fcgene - -ped example/plink.ped - -map example/plink.map\
- -covar example/plink_cov.txt\
- -covar-name pheno1-pheno2,covar_A-covar_C\
- -covar-type B,D\
- -oformat snptest - -out example/plink_snptest
```

In this command FCgene takes all the variables given in columns starting from pheno1 till the column of pheno2 and includes them in snptest ".sample" file as "B" type variables. Similarly, the variables given in the columns between covar_A and covar_C are set to the type "D" while writing them in the snptest ".sample" file.

10.2.2 Converting MACH formatted data

Conversion of MACH input files

To convert mach input files into snptest format,the following command can be used.

```
./fcgene - -ped example/mach.ped - -dat example/mach.map\
- -oformat snptest - -out example/mach_snptest
```

If you are interested in adding covariates from plink formatted covariate file (e.g. named “plink_cov.txt") to the snptest “.sample" file, then the command can be written as

```
./fcgene - -ped example/mach.ped - -dat example/mach.dat\
- -covar example/plink_cov.txt\
- -covar-name pheno1,pheno2,covar_A,covar_B\
- -covar-type P,B,D,C\
- -oformat snptest - -out example/mach_snptest
```

Remarks 10.2

- *The family id and individual id in covariate file “plink_cov.txt" and must match with the individual ids and family ids given in “.ped" file.*
- *The covariate file “plink_cov.txt" can have more than necessary phenotypes and covariates. If you do not mention any variables with option “–covar-name" and “–covar-tpye", then FCgene just leaves the columns of these variables.*
- *As mentioned previously, snptest “.sample" file should contain the columns of covariates before the phenotype variables,however FCgene takes care of this fact alone and put the columns of phenotype variables at end. So you do not need to worry about the sequential order of variables.*

If some of covariates and phenotypes which are given in adjacent columns of plink formatted covariate file, have the same type then the command for converting into snptest format can also be written as

```
./fcgene - -ped example/mach.ped - -dat example/mach.dat\
- -covar example/plink_cov.txt\
- -covar-name pheno1-pheno2,covar_A-covar_C\
- -covar-type B,D\
- -oformat snptest - -out example/mach_snptest
```

In this command FCgene takes all the variables given in columns starting from pheno1 till the column of pheno2 and includes them in snptest “.sample" file as “B" type variables. Similarly, the variables given in the columns between covar_A and covar_C are set to the type “D" while writing them in the snptest “.sample" file.

Converting mach output: geno and info files

-
  ```
  ./fcgene - -mach-geno example/mach_output.geno\
  - -mach-info example/mach_output.info\
  - -rsq 0.3 - -maf-thresh 0.1\
  - -oformat snptest - -out example/geno_snptest
  ```

-
  ```
  ./fcgene - -mach-geno example/mach_output.geno\
  - -mach-info example/mach_output.info\
  - -rsq 0.3 - -maf-thresh 0.1\
  - -pedinfo example/mach_pedinfo.txt\
  - -snpinfo example/mach_snpinfo.txt\
  - -covar example/plink_cov.txt\
  - -covar-name pheno1,pheno2,covar_A,covar_B\
  - -covar-type P,B,D,C\
  - -oformat snptest - -out example/geno_snptest
  ```

-
  ```
  ./fcgene - -mach-geno example/mach_output.geno\
  - -mach-info example/mach_output.info\
  - -rsq 0.3 - -maf-thresh 0.1\
  - -pedinfo example/mach_pedinfo.txt\
  - -snpinfo example/mach_snpinfo.txt\
  - -covar example/plink_cov.txt\
  - -covar-name pheno1-pheno2,covar_A-covar_C\
  - -covar-type B,D\
  - -oformat snptest - -out example/geno_snptest
  ```

Remarks 10.3

- *Information on the options* “- -covar", “- -covar-name", “- -covar-type" *can be found at the start of this section.*

- *Information on the options* “- -snpinfo", “- -pedinfo" *is given on chapter* **??**.

- *A description on the options* “- -rsq", “- -maf-thresh" *can be found in Section 6.3.*

Converting mach output: mlgeno and mlinfo files

-
  ```
  ./fcgene - -mach-mlgeno example/mach_output.mlgeno\
  - -mach-mlinfo example/mach_output.mlinfo\
  - -rsq 0.3 - -maf-thresh 0.1\
  - -oformat snptest - -out example/mlgeno_snptest
  ```

-
  ```
  ./fcgene - -mach-mlgeno example/mach_output.mlgeno\
  - -mach-mlinfo example/mach_output.mlinfo\
  - -rsq 0.3 - -maf-thresh 0.1\
  - -pedinfo example/mach_pedinfo.txt\
  - -snpinfo example/mach_snpinfo.txt\
  - -covar example/plink_cov.txt\
  - -covar-name pheno1,pheno2,covar_A,covar_B\
  - -covar-type P,B,D,C\
  - -oformat snptest - -out example/mlgeno_snptest
  ```

-
  ```
  ./fcgene - -mach-mlgeno example/mach_output.mlgeno\
  - -mach-mlinfo example/mach_output.mlinfo\
  - -rsq 0.3 - -maf-thresh 0.1\
  - -pedinfo example/mach_pedinfo.txt\
  - -snpinfo example/mach_snpinfo.txt\
  ```

```
--covar example/plink_cov.txt\
--covar-name pheno1-pheno2,covar_A-covar_C\
--covar-type B,D\
--oformat snptest --out example/mlgeno_snptest
```

Remarks 10.4

- *Information on the options* “--covar", “--covar-name", “--covar-type" *can be found at the start of this section.*
- *Information on the options* “--snpinfo", “--pedinfo" *is given on chapter* **??**.
- *A description on the options* “--rsq", “--maf-thresh" *can be found in Section 6.3.*

Converting mach output: mlprob and mlinfo files

-
  ```
  ./fcgene --mach-mlprob example/mach_output.mlprob\
  --mach-info example/mach_output.mlinfo\
  --rsq 0.3 --maf-thresh 0.1\
  --oformat snptest --out example/mlprob_snptest
  ```

-
  ```
  ./fcgene --mach-mlprob example/mach_output.mlprob\
  --mach-mlinfo example/mach_output.mlinfo\
  --rsq 0.3 --maf-thresh 0.1\
  --pedinfo example/mach_pedinfo.txt\
  --snpinfo example/mach_snpinfo.txt\
  --covar example/plink_cov.txt\
  --covar-name pheno1,pheno2,covar_A,covar_B\
  --covar-type P,B,D,C\
  --oformat snptest --out example/mlprob_snptest
  ```

-
  ```
  ./fcgene --mach-mlprob example/mach_output.mlprob\
  --mach-mlinfo example/mach_output.mlinfo\
  --rsq 0.3 --maf-thresh 0.1\
  --pedinfo example/mach_pedinfo.txt\
  --snpinfo example/mach_snpinfo.txt\
  --covar example/plink_cov.txt\
  --covar-name pheno1-pheno2,covar_A-covar_C\
  --covar-type B,D\
  --oformat snptest --out example/mlprob_snptest
  ```

Remarks 10.5

- *Information on the options* “--covar", “--covar-name", “--covar-type" *can be found at the start of this section.*
- *Information on the options* “--snpinfo", “--pedinfo" *is given on chapter* **??**.
- *A description on the options* “--rsq", “--maf-thresh" *can be found in Section 6.3.*

Converting minimach output: prob and info files

-
  ```
  ./fcgene --minimach-prob example/minimach_output.prob\
  --minimach-info example/minimach_output.mlinfo\
  --rsq 0.3 --maf-thresh 0.1\
  --oformat snptest --out example/minimach_snptest
  ```

-
  ```
  ./fcgene --minimach-prob example/minimach_output.prob\
  --minimach-info example/minimach_output.mlinfo\
  --rsq 0.3 --maf-thresh 0.1\
  --pedinfo example/mach_pedinfo.txt\
  --snpinfo example/mach_snpinfo.txt\
  --covar example/plink_cov.txt\
  --covar-name pheno1,pheno2,covar_A,covar_B\
  --covar-type P,B,D,C\
  --oformat snptest --out example/minimach_snptest
  ```

-
  ```
  ./fcgene --minimach-prob example/minimach_output.prob\
  --minimach-info example/minimach_output.mlinfo\
  --rsq 0.3 --maf-thresh 0.1\
  --pedinfo example/mach_pedinfo.txt\
  --snpinfo example/mach_snpinfo.txt\
  --covar example/plink_cov.txt\
  --covar-name pheno1-pheno2,covar_A-covar_C\
  --covar-type B,D\
  --oformat snptest --out example/minimach_snptest
  ```

Remarks 10.6

- *Information on the options* “--covar", “--covar-name", “--covar-type" *can be found at the start of this section.*
- *Information on the options* “--snpinfo", “--pedinfo" *is given on chapter* **??**.
- *A description on the options* “--rsq", “--maf-thresh" *can be found in Section 6.3.*
- *One should be very cautious in using* “--snpinfo" *option if any kind of references are used for further analysis. The allele order in “snpinfo" file and references must match with other.*

10.2.3 Converting IMPUTE formatted data

Output of impute can be directly used in SNPTEST unless if low imputation quality SNPs are to be filtered. If you prefer to filter first the low quality SNPs then to run SNPTEST, then the following command can be used.

-
  ```
  ./fcgene --gens example/example.impute2\
  --thresh maxProb\
  --info example/example.impute2_info\
  --info-thresh info_thresh_value\
  --maf-thresh maf_thresh_value\
  --pedinfo example/example.pedinfo\
  --oformat snptest --out example/impute_snptest
  ```

-
  ```
  ./fcgene - -gens example/example.impute2\
  - -thresh maxProb \
  - -info example/example.impute2_info\
  - -info-thresh info_thresh_value\
  - -maf-thresh maf_thresh_value\
  - -pedinfo example/example.pedinfo\
  - -covar example/plink_cov.txt\
  - -covar-name pheno1-pheno2,covar_A-covar_C\
  - -covar-type B,D\
  - -oformat snptest - -out example/impute_snptest
  ```

Remarks 10.7

- *Information on the options* “- -covar", “- -covar-name", “- -covar-type" *can be found at the start of this section.*
- *Information on the options* “- -snpinfo", “- -pedinfo" *is given on chapter* **??**.
- *A description on the options* “- -thresh", “- -info-thresh", “- -maf-thresh" *can be found in Section 7.5.*
- *One can also use “–snpinfo option" in above examples. However, one should be very cautious if any kind of references are used for further analysis. The allele order in “snpinfo" file and references must match with other. If there is a need to update SNP information, the best way is to use legend file of impute reference. The legend file of impute formatted references can be directly fed to FCgene with “–snpinfo " option.*

10.2.4 Converting BEAGLE formatted data

Input as well as the output of the imputation tool Beagle can be converted to snptest format. Beagle has two types of outputs: first : “.bgl.phased" file and the second “.bgl.grpobs" file. The following commands are used for converting the format of beagle output files.

-
  ```
  ./fcgene  - -bgl example/beagle.bgl.phased\
  - -pedinfo example/plink_beagle_pedinfo.txt\
  - -snpinfo example/plink_beagle_snpinfo.txt\
  - -oformat snptest - -out example/beagle_snptest
  ```

-
  ```
  ./fcgene  - -bgl-gprobs example/beagle.bgl.gprobs\
  - -oformat snptest  - -out example/beagle_snptest\
  - -pedinfo example/plink_beagle_pedinfo.txt\
  - -snpinfo example/plink_beagle_snpinfo.txt
  ```

-
  ```
  ./fcgene - -bgl example/beagle.bgl.phased\
  - -bgl-rsq example/beagle.bgl.r2\
  - -rsq-thresh 0.3 - -oformat snptest - -out example/beagle_snptest\
  - -pedinfo example/plink_beagle_pedinfo.txt\
  - -snpinfo example/plink_beagle_snpinfo.txt
  ```

-
  ```
  ./fcgene - -bgl-gprobs example/beagle.bgl.gprobs\
  - -bgl-rsq example/beagle.bgl.r2\
  ```

```
--rsq-thresh 0.3  --oformat snptest --out example/beagle_snptest\
--pedinfo example/plink_beagle_pedinfo.txt\
--snpinfo example/plink_beagle_snpinfo.txt
```

-
  ```
  ./fcgene --bgl example/beagle.bgl.phased\
  --bgl-rsq example/beagle.bgl.r2\
  --covar example/plink_cov.txt\
  --covar-name pheno1-pheno2,covar_A-covar_C\
  --covar-type B,D\
  --pedinfo example/plink_beagle_pedinfo.txt\
  --snpinfo example/plink_beagle_snpinfo.txt \
  --rsq-thresh 0.3 --oformat snptest --out example/beagle_snptest
  ```

-
  ```
  ./fcgene --bgl-gprobs example/beagle.bgl.gprobs\
  --bgl-rsq example/beagle.bgl.r2\
  --rsq-thresh 0.3  --oformat snptest --out example/beagle_snptest\
  --covar example/plink_cov.txt\
  --covar-name pheno1-pheno2,covar_A-covar_C\
  --covar-type B,D\
  --pedinfo example/plink_beagle_pedinfo.txt\
  --snpinfo example/plink_beagle_snpinfo.txt
  ```

Remarks 10.8

- *Information on the options* “--covar", “--covar-name", “--covar-type" *can be found at the start of this section.*

- *Information on the options* “--snpinfo", “--pedinfo" *is given on chapter* **??**.

- *A description on the options* “- rsq-thresh" *can be found in Section 8.4.*

- *One should be very cautious in using* “--snpinfo" *option if any kind of references are used for further analysis. The allele order in “snpinfo" file and references must match with other.*

10.2.5 Converting BIMBAM formatted data

-
  ```
  ./fcgene  --wbg output/bimbam_imputed.best.guess.genotype.txt\
  --pos output/bimbam_imputed.snpdata.txt --maf-thresh 0.1 --covar example/plink_cov.txt\
  --covar-name pheno1-pheno2,covar_A-covar_C\
  --covar-type B,D\
  --pedinfo example/plink_beagle_pedinfo.txt\
  --snpinfo example/plink_beagle_snpinfo.txt\
  --oformat snptest  --out example/bimbam_snptest
  ```

-
  ```
  ./fcgene  --wbg output/bimbam_imputed.best.guess.genotype.txt\
  --pos output/bimbam_imputed.snpdata.txt  --maf-thresh 0.1\
  --covar example/plink_cov.txt\
  --covar-name pheno1-pheno2,covar_A-covar_C\
  --covar-type B,D\
  --pedinfo example/plink_beagle_pedinfo.txt\
  --snpinfo example/plink_beagle_snpinfo.txt\
  --oformat snptest  --out example/bimbam_snptest
  ```

Chapter 11

Format conversion to EIGENSOFT-format

11.1 Introduction to EIGENSOFT

EIGENSOFT represents software SMARTEIGENSTRAT and SMARTPCA **??**, which are popular for:

- correcting for population stratification in association studies,
- and detecting and analyzing population structure.

Principal component analysis(PCA) is used for these purpose. The authors of EIGENSOFT have provided both source code and compiled version of the program and also the necessary Perl scripts for creating data in necessary formats. SMARTPCA accepts file format similar to PLINK but before the start of the program, some changes in the plink-formatted files may be necessary. For example SMARTPCA uses "-99“ to code missing phenotype information, which in PLINK generally is expressed as "-9“. Similarly if we analyze population structure not in between cases and controls but among individuals from different population/ethnic group, then it is better to replace the phenotype information by group/population-label for each individuals. We can first use SMARTPCA to determine the principal components and then upload SMARTPCA's outputs into SMARTEIGENSTRAT to correct stratification. The outputs of SMARTPCA can't be uploaded in SMARTEIGENSTRAT directly. It also needs a parameter file. Another programs like "convertf“ and "evec2pca.perl“ can create files for SMARTEIGENSTRAT from SMARTPCA's output. However applying these different programs may be confusing for a straight forward analysis of the data.

fcGENE can creates necessary files for EIGENSOFT not only from plink formatted data but also from the input and outputs of previously mentioned imputation software. fcGENE converts any type of uploaded genotype data into the formats necessary for EIGENSOFT. Aside from this, necessary files that fcGENE writes are:

- parameters file for SMARTPCA
- parameters file for SMARTEIGENSTRAT
- R-script for PCA-Plot and modifying outputs of SMARTPCA into the compatitable form of SMARTEIGENSTRAT
- Linux-script(alternative way of using R) to run SMARTPCA, SMARTEIGENSTRAT, eigenplot,twstats

11.2 Format conversion

To convert plink-formatted data into EIGENSOFT format, we can use the following command.

```
./fcgene --map plink/example.map --ped plink/example.ped \
--oformat eigensoft --out eigensoft/plink_eigensoft
```

Marking group/population label to individuals

If we make PCA not in between cases and controls but among the individuals from different population/ethnic group, then each individual can be assigned to a certain group-label by using command option “--group-label". To mark group-label for each study-individual,we can prepare a File containing individual ids in the first column and group-label ids in the second column. fcGENE can read such a group-labeling file with command option “--group-label". fcGENE can write a list of individuals in a specified file when we use the command option “--write-pedlist". To prepare a individual list file for example from mach-formatted data, we can use the following command.

```
./fcgene --dat mach/examle.dat --ped mach/example.ped \
--write-pedlist --out mach/example_pedlist
```

This command will create a “mach_pedlist.txt" file. One can add individual’s group/population label in this file as second column and update this with fcGENE as follows

```
./fcgene --dat mach/example.dat --ped mach/example.ped \
--oformat eigensoft\
--group-label mach/mach_pedlist.txt\
--out eigensoft/mach_eigensoft
```

The process for converting data from other formats is similar to these commands. Regardless of the type of data uploaded to fcGENE, it can convert the uploaded data into the format specified by command option “--oformat". Some more examples of format conversion into eigensoft-formatted data are given below.

11.3 Commands for differently formatted genotype data

11.3.1 Command to convert plink-formatted data

- General command:
  ```
  ./fcgene --map plink/example.map --ped plink/example.ped \
  --oformat eigensoft --out eigensoft/plink_eigensoft
  ```
- With updates of group/population-label:
  ```
  ./fcgene --map pink/example.map --ped plink/example.ped \
  --oformat eigensoft\
  --group-label plink/plink_pedlist.txt \
  --out eigensoft/plink_eigensoft
  ```
  Note that command option “group-label" is optional used to mark individuals to a certain population/ethnic group.

- Format conversion with quality control:
  ```
  ./fcgene - -map pink/example.map - -ped plink/example.ped \
  - -filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  - -filter-indiv crate=0.95\
  - -oformat eigensoft\
  - -group-label plink/plink_pedlist.txt \
  - -out eigensoft/plink_eigensoft
  ```

11.3.2 Command to convert mach-formatted data

- General command:
  ```
  ./fcgene - -dat mach/mach.dat - -ped mach/mach.ped \
  - -filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  - -filter-indiv crate=0.95\
  - -pedinfo mach_pedinfo.txt - - snpinfo mach_snpinfo.txt\
  - -oformat eigensoft - -out mach/mach_eigensoft\
  ```

Remark 11.1 *Note that command option "- -pedinfo" and "- -snpinfo" are optional but we recommend here to update pedigree information and information related to SNPs because mach-formatted genotype data does not contain all information like phenotype information, base pair position, genetic distance in centimorgan and allele infos, required by EIGENSOFT.*

- With updates of group/population-label:
  ```
  ./fcgene - -dat mach/example.dat - -ped mach/example.ped \
  - -pedinfo mach/mach_pedinfo.txt - -snpinfo mach/mach_snpinfo.txt\
  - -filter-snp maf=0.1,crate=0.95,hwe=1e-2\
  - -filter-indiv crate=0.95\
  - -oformat eigensoft\
  - -out eigensoft/mach_eigensoft\
  - -group-label mach/mach_pedlist.txt
  ```

- Conversion of MaCH's imputation output: geno:
  ```
  ./fcgene - -mach-geno mach/mach_output.geno - -mach-info\
  mach/mach_output.info - -pedinfo mach/mach_pedinfo.txt\
  - -snpinfo mach/mach_snpinfo.txt - -oformat eigensoft\
  - -filter-snp maf=0.1,hwe=1e-2\
  - -group-label mach/mach_pedlist.txt\
  - -out mach/geno_eigensoft
  ```

- Conversion of MaCH's imputation output: mlgeno:
  ```
  ./fcgene - -mach-mlgeno mach/mach_output.mlgeno\
  - -mach-mlinfo mach/mach_output.mlinfo - -pedinfo\
  mach/mach_pedinfo.txt - -snpinfo mach/mach_snpinfo.txt\
  - -group-label mach/mach_pedlist.txt - -oformat eigensoft\
  - -filter-snp hwe=1e-2\
  - -filter-snp hwe=1e-2\
  - -rsq 0.3 - -maf-thresh 0.1 - -out mach/mlgeno_eigensoft
  ```

- Conversion of MACH's imputation output: prob /mlprob:
```
./fcgene - -mach-mlprob mach/mach_output.mlprob \
- -mach-mlinfo mach/mach_output.mlinfo - -pedinfo\
mach/mach_pedinfo.txt - -snpinfo mach/mach_snpinfo.txt\
- -group-label mach/mach_pedlist.txt - -oformat eigensoft\
- -filter-snp hwe=1e-2\
- -rsq 0.3 - -maf-thresh 0.1 - -out mach/mlprob_eigensoft
```
- Conversion of minimac's imputation output: prob /info:
```
./fcgene - -minimac-prob mach/minimac_output.prob \
- -minimac-info mach/minimac_output.info - -pedinfo\
mach/mach_pedinfo.txt - -snpinfo mach/mach_snpinfo.txt\
- -group-label mach/mach_pedlist.txt - -oformat eigensoft\
- -filter-snp hwe=1e-2\
- -rsq 0.3 - -maf-thresh 0.1 - -out mach/minimac_eigensoft
```

Remark 11.2 *Detailed information on the command options used here is given in section 6.3.*

11.3.3 Commands to convert minimac-formatted data

- Conversion of minimac's imputation output: prob /info:
```
./fcgene - -minimac-prob minimac/minimac_output.prob \
- -minimac-info minimac/minimac_output.info - -pedinfo\
minimac/mach_pedinfo.txt - -snpinfo minimac/mach_snpinfo.txt\
- -group-label minimac/mach_pedlist.txt - -oformat eigensoft\
- -filter-snp hwe=1e-2\
- -rsq 0.3 - -maf-thresh 0.1 - -out minimac/minimac_eigensoft
```

Remark 11.3 *Detailed information on the command options used here is given in section 6.3.*

11.3.4 Commands to convert impute-formatted data

- General format conversion:
```
./fcgene - -gens impute/example.gens - -thresh maxProb \
- -pedinfo impute/impute.pedinfo\
- -filter-snp maf=0.1,crate=0.95,hwe=1e-2\
- -filter-indiv crate=0.95\
- -oformat eigensoft - -out eigensoft/impute_eigensoft
```
- Format conversion imputed data with group/population-label:
```
./fcgene - -gens impute/example.chr22.impute2 - -thresh maxProb \
- -info impute/example.chr22.impute2_info \
- -filter-snp hwe=1e-2\
- -info-thresh info_thresh_value \
- -maf-thresh maf_thresh_value\
- -group-label impute/impute_pedlist.txt - -oformat eigensoft\
- -pedinfo impute/impute.pedinfo\
- -oformat eigensoft - -out eigensoft/impute_eigensoft
```

Remark 11.4 *For detailed informaiton the command option used here, please visit section see 7.5.*

11.3.5 Commands to convert beagle-formatted data

- Command convert beagle phased data:
  ```
  ./fcgene --bgl beagle/beagle.bgl.phased --oformat eigensoft --out beagle/beagle_eigensoft
  ```
- To convert beagle.output file containing genotype probabilities (beagle.bgl.gprobs):
  ```
  ./fcgene --bgl-gprobs example/example.bgl.gprobs --oformat eigensoft --out eigensoft/beagle_eigensoft
  ```
- Format conversion with updates of pedigree and SNP information and group-label:

  -
    ```
    ./fcgene --bgl beagle/example.bgl.phased --out eigensoft/beagle_eigensoft\
    --pedinfo beagle/example_pedinfo.txt --snpinfo beagle/example_snpinfo.txt\
    --filter-snp maf=0.1,crate=0.95,hwe=1e-2\
    --filter-indiv crate=0.95\
    --group-label beagle/example_pedlist.txt --oformat eigensoft
    ```

  -
    ```
    ./fcgene --bgl-gprobs beagle/example.bgl.gprobs\
    --out eigensoft/beagle_eigensoft\
    --pedinfo beagle/example_pedinfo.txt --snpinfo beagle/example_snpinfo.txt\
    --filter-snp maf=0.1,crate=0.95,hwe=1e-2\
    --filter-indiv crate=0.95\
    --group-label beagle/example_pedlist.txt --oformat eigensoft
    ```

  -
    ```
    ./fcgene --bgl beagle/beagle.bgl.phased --bgl-rsq beagle/beagle.bgl.r2\
    --rsq-thresh 0.3 --out eigensoft/beagle_eigensoft\
    --filter-snp hwe=1e-2\
    --pedinfo beagle/beagle_pedinfo.txt --snpinfo beagle/beagle_snpinfo.txt\
    --group-label beagle/beagle_pedlist.txt --oformat eigensoft
    ```

  -
    ```
    ./fcgene --bgl-gprobs beagle/example.bgl.gprobs --bgl-rsq beagle/example.bgl.r2\
    --rsq-thresh 0.3 --out eigensoft/beagle_eigensoft\
    --filter-snp maf=0.1,hwe=1e-2\
    --pedinfo beagle/example_pedinfo.txt --snpinfo beagle/example_snpinfo.txt\
    --group-label beagle/example_pedlist.txt --oformat eigensoft
    ```

Remark 11.5 *For detailed informaiton the command option used here, please visit chapter see 8.*

11.3.6 Command to convert bimbam-formatted data

Chapter 12

Data transformation to HAPLOVIEW-format

To generate genotype data into the format of Haploview, one can first upload the data in fcGENE and then use "- -oformat haploview". Haploview performs linkage disequilibrium (LD) analysis and estimates haplotype population frequency estimation [8]. Haploview basically needs two files:

1. a pedigree genotype data, and
2. a SNP information file with columns of marker names and base pair position.

Plink-formatted pedigree data (PED) is accepted by Haploview. However it also requires an extra file with marker names and position information. A format conversion process is necessary also for the genotype data given in formats like imputation outputs.

Remark 12.1 *For more details of the commands to upload data, please see previous chapters.*

Chapter 13

Genotype data as count of reference alleles:

13.1 Matrix with cout of reference allele

Coding genotypes as the counts of minor alleles may be useful if we use statistical models like regression and ANOVAs. PLINK has support to produce this type of files by using command options "- -recodeA" and "- -recodeAD", but these two options support only the hard calls of genotype data meaning use of only the numbers 0, 1 and 2 to represent genotypes of the form homozygote major, heterozygote and homozygote minor respectively. fcGENE supports to transform the data with genotype probability distribution into the form of PLINK's recodeA-formatted files but filled with expected minor- allele-doses. This means the raw files transformed by fcGENE can contain not only 0,1 and 2 as minor allele counts but also the expected allele dose of minor allele, which can be any fractional number between 0 and 2. If "A" and "B" are two alleles of A SNP with "B" as minor allele, then the expected minor allele dose can be calculated as

$$0.p(AA)+1.p(AB)+2.p(BB) = p(AB)+2p(BB),$$

where $p(AA), p(AB)$ and $p(BB)$ are the probabilities having genotypes AA ,AB and BB at an individual respectively. Using command options "- -oformat plink-recodeA" and “–oformat plink-recodeAD” , we can obtain PLINK's raw files while to produce raw file filled with minor-allele-expected dose, fcGENE requires the command option "- -oformat recodeA-dose". Similarly researchers who are interested to start an analysis in statistical package R, can use fcGENE to convert genotype data from different formats into standard text files with genotyped codes either as the counts of minor allele as the expected dose of minor allele-counts of a SNP at each individuals. By standard text files, we mean rows for observations (individuals) and columns for variables (SNPs), with simple headers for gene names (rsids) and first column to identify subjects (i.e. pedigree ids). An sized genotype data matrix with dosage values for genotype (number of minor alleles) would look like: Similarly *fcGENE* writes an affection status file having 0 for cases and 1 for control, which would look like: Upon loading a dataset in fcGENE,

SMAPLE_ID	rsid_1	rsid_2	⋯	rsid_M
ID_1	0	2	⋯	1
ID_2	0	1	⋯	0
…	…	…	…	…
ID_3	NA	1	⋯	0

```
SMAPLE_ID  AFFSTAT
ID_1 1
ID_2 0
...        ...
ID_N 1
```

previously mentioned two text files can be generated as output by using the command option "- -oformat R" or "- -oformat r". Similarly to obtain a matrix where each row is given as a SNP, not as an individual mentioned previously,we can use an additional command option “- -transpose”.

13.1.1 Uploading genotype data with allele count

In order to upload genotype data with counts of reference allele, we can use option “- -rgeno” and “- -snpinfo”. An example is given below

- *./fcgene - -rgeno example/example_genotype.txt - -snpinfo example/example_alleleinfo.txt\ - -oformat plink-bed*

Summary of Commands

Function	Command option	File name
To upload r-formatted or standard data with counts of genotypes	- -rgeno	example_genotype.txt
	- -snpinfo	alleleinfo.txt
To write r-formatted or standard data with counts of genotypes	- - oformat r - - oformat R	
To write genotype data with SNPs in rows of genotypes	- - oformat r - -transpose	
To write data with allele dose	- - r-dose - - R-dose	

Table 13.1: Table showing necessary files and command options to upload r-formatted data.

13.2 Matrix with allele dose

To obtain previously mentioned type of standard text files filled with expected dose of minor allele for every SNP at each individuals , fcGENE requires command option:

- "- -oformat r-dose" or
- "- -oformat R-dose".

These types of files may be useful when we analyze the data in R. There are R packages for the analysis of genetic data however it is often useful to get data into previously mentioned format when using self-planned analysis in R.

13.3 Change of reference allele

fcGENE takes minor allele as default reference allele and writes the count of reference allele for each genotype. If a SNP at any individual is coded as homozygote of minor allele, then its allele count is 2. Similarly 1 and 0 is used to code heterozygote and homozygote of alternative allele. To change the reference allele, one can force fcGENE with an additional command option "- -force ref-allele=". Possible command options for forcing reference allele are:

- "- -force ref-allele=minor-allele" (this is default),
- "- -force ref-allele=major-allele",
- "- -force ref-allele=allele1" and
- "- -force ref-allele=allele2".

Chapter 14

Data transformation to PHASE/fastPHASE-format

To generate genotype data into the format of PHASE/fastPHASE, we can apply “- -oformat phase” command.

Bibliography

[1] Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, de Bakker PIW, Daly MJ, Sham PC, 2007. PLINK: a toolset for whole-genome association and population-based linkage analysis. American Journal of Human Genetics, 81. PMCID: PMC1950838,
URL: `http://pngu.mgh.harvard.edu/purcell/plink/`

[2] Abecasis G, Homepage of Imputation software MACH1.0
URL: `http://www.sph.umich.edu/csg/abecasis/MACH/tour/imputation.html`

[3] J. Marchini, B. Howie Homepage of software: IMPUTE
URL:`http://mathgen.stats.ox.ac.uk/impute/impute.html`

[4] J. Marchini, B. Howie Homepage of software: SNPTEST
URL:`https://mathgen.stats.ox.ac.uk/genetics_software/snptest/snptest.html`

[5] The Wellcome Trust Case Control Consortium (2007) Genomewide association study of 14,000 cases of seven common diseases and 3,000 shared controls. Nature 447;661-78. PMID: 17554300 DOI: 10.1038/nature05911

[6] B L Browning and S R Browning Homepage of software: BEAGLE,
URL: `http://faculty.washington.edu/browning/beagle/beagle.html`

[7] Yongtao Guan , Matthew Stephens, Homepage of software BIMBAM,
URL: `http://www.bcm.edu/cnrc/mcmcmc/bimbam`

[8] Barrett JC, Fry B, Maller J, Daly MJ. Haploview: analysis and visualization of LD and haplotype maps. Bioinformatics. 2005 Jan 15 [PubMed ID: 15297300]

[9] Patterson N, Price AL, Reich D (2006) Population Structure and Eigenanalysis. PLoS Genet 2(12): e190. doi:10.1371/journal.pgen.0020190

[10] Wigginton, JE, Cutler, DJ, and Abecasis, GR (2005) A Note on Exact Tests of Hardy-Weinberg Equilibrium. American Journal of Human Genetics. 76:000 - 000