HiggsGene Solutions

FImpute

User's Guide

Version 3.0

Mehdi Sargolzaei & Flavio Schenkel
HiggsGene Solutions Inc. & University of Guelph
Dec 2018

Disclaimer

The FImpute software is distributed "AS IS". The authors and their organizations will not be liable for any general, special, incidental or consequential damages arising from using FImpute. By the use of this software the user agrees to bear all risks resulting from using the software.

Citing FImpute:

Sargolzaei, M., J. P. Chesnais and F. S. Schenkel. 2014. A new approach for efficient genotype imputation using information from relatives. BMC Genomics, 15:478 (DOI: 10.1186/1471-2164-15-478).

Contact:

Mehdi Sargolzaei @ hgs.msargolzaei@gmail.com

IMPORTANT: If you have a problem with a specific imputation run, please include "report.txt" and control files with your message.

Link for academic version.

What is new in version 3.0:

- Increased ability to handle bigger reference size resulting in higher accuracy with big data
- Faster imputation with lower memory requirement through implementation of thread parallel processing; parallel processing can be optimized across and within chromosomes.
- Improved imputation accuracy for situations where pedigree error rate is high.
- For ease of use, chip with the same number of SNP (or very close number of SNPs) can be merged automatically

- Imputed genotypes for a user defined list of individuals or panel can be outputted instead of outputting genotypes for all individuals
- Minimum number of informative SNP for parentage test can be specified by the user
- Faster parentage verification and discovery
- Fast parentage verification and discovery can be carried out on a user defined list of target individuals
- keep_og command prevents change in original genotypes during imputation process
- Larger number of panels (up to 100) can be handled
- Reference chip can now be specified by the user; This feature makes two-step imputation easier and allows for flexible genotype input
- Allele frequency correlations between chips are now reported before and after imputation
- Distribution of the difference in allele frequencies between reference and target groups is outputted. This will help to identify SNPs with low accuracy in pure breed imputation.
- Reporting individuals with low genotyping call rate
- Reporting proportion of SNPs imputed with random sampling based on allele frequency for each individual
- Random filling based on allele frequency can be turned off (i.e. more accurate imputation but with some missing genotypes)
- Two haplotypes per individual can be outputted
- Imputed genotype, map and pedigree files can be outputted in a format required by snp1101 software
- When a SNP is excluded, the exclusion reason is now reported
- Memory issue has been fixed
- Fix in parentage module
- Better error/warning reporting
- Support for ZW sex-determination (i.e. for poultry)
- Gender identifier in pedigree file can now include ‘U’ for individuals with not known gender for imputation of autosomes.
- And more …

Overview

FImpute (ef-impute) was mainly developed for large scale genotype imputation in livestock where hundreds of thousands of individuals are genotypes with different panels. FImpute uses an overlapping sliding window approach to efficiently exploit relationships or haplotype similarities between target and reference individuals. The process starts with long windows to capture haplotype similarity between close relatives. After each chromosome sweep, the window size is shrunk by a constant factor allowing for shorter haplotype similarity (arising from more distant relatives) to be taken into account. Because closer relatives usually share longer haplotypes while more distant relatives share shorter haplotypes the algorithm simply assumes that all individuals are related to each other at different degrees. Note that if pedigree information is provided FImpute makes use of this information for more accurate imputation. Pedigree information becomes more important as the low density panel becomes sparser. High input genotype quality is the key for accurate imputation, therefore FImpute is not recommended for sequence imputation with low coverage. The current version of FImpute can handle SNP markers (i.e. bi-allelic) only.

Input control file

The program requires a control file, in which various parameters for imputation should be specified. The input parameter file must be in ASCII format. C++ like comments can be used to add descriptive comments anywhere in the parameter file. All commands end with a semicolon.

title

Description: Set an arbitrary title.
Usage: title = "string";
string indicates an arbitrary title.
Type: Optional
Default: None

genotype_file

Description: Input genotype file.

Usage: genotype_file = "filename" option;

filename	is input genotype file name.	
option	/phased	Indicates that input genotypes are already phased.

Type: Mandatory

Input Format: ID, chip number, genotype calls
First line is header line.
Chip number starts from 1 and should be the order of chip in SNP info file.
There is no space between genotypes and genotypes should be coded as: 0 and 2 for homozygotes, 1 for heterozygote and 5 for missing genotypes.
The number of genotypes for each individual must be exactly the same as the number of SNP on the chip for which the individual was genotyped with.
Genotype calls:
0: A1A1
1: A1A2 or A2A1
2: A2A2
5: missing
Maximum ID length is 30 characters.

Note: Multiple genotype files can be read in as:
genotype_file = "filename1" "filename2" ... ;

snp_info_file

Description: This file contains SNP map information.

Usage: snp_info_file = "filename" option;

filename	is input SNP map file name.	
option	/chrx = v	specifies chromosome X. Note that v should not contains pseudo-autosomal regions of X.

Type: Mandatory

Input Format: SNP ID, chromosome number, base pair position, order of SNP for each chip
First line is header line.
Maximum SNP ID length is 50 characters.
Maximum number of chips is 100.

Note: Positions of SNP on each chromosome should be defined as accurate as possible since FImpute uses base pair position to model recombination. 1,000,000 base pairs is considered as 1 cM.

ped_file

Description: Pedigree file.

Usage: ped_file = "filename" option;

filename is input pedigree file name.

option	/remove_parent = v	Remove parents for individuals genotyped with certain chip. v is chip number(s).
option	/dob	This option is used when the fifth column in pedigree is date of birth in YYYYMMDD format.

Type: optional

Input Format: ID, sire ID, dam ID, sex
First line is header line.
IDs can be alphanumeric and do not need to be sorted.
sex should be coded as ‘M’ and ‘F’
Maximum ID length is 30 characters.

Note: Multiple pedigree files can be read in as:
ped_file = "filename1" "filename2" ... ;
Pedigree files with overlap will be combined to create one pedigree with unique IDs.
If pedigree file is not defined family imputation is automatically turned off.
If sex chromosome is to be analyzed, pedigree file should always be defined. In this case, if pedigree is not known set parents to missing but provide correct sex.

hap_lib_file

Description: Haplotype library file.

Usage: hap_lib_file = "filename" option;

filename is input haplotype library file name.

option	/diplotype	Compressed format. Two haplotypes are combined in one line.
	/mr = value	Missing rate threshold. Haplotypes with large missing rate will be discarded. Default is 0.2.

Type: Optional

Input Format: First line should contain SNP IDs
Haplotypes start from the second line. There should be no space between haplotype codes.
Haplotype codes:

1: A1
2: A2
5: missing

When "diplotype" option is specified the codes are:
0: A1A1
1: treated as missing
2: A2A2
3: A1A2
4: A2A1
5: missing
6: A1? (second haplotype is missing)
7: A2? (second haplotype is missing)
8: ?A1 (first haplotype is missing)
9: ?A2 (first haplotype is missing)

Note: Multiple haplotype library files can be inputed as:
Hap_lib_file = "filename1" "filename2" ... ;

output_folder

Description: Output folder.
Usage: output_folder = "foldername";
foldername is output folder name.
Type: Mandatory

output_group

Description: Output group of individuals.
Usage: output_group = v;
v is chip number(s).
Type: optional
Note: With this command, genotypes for specified individuals are outputted, which results in smaller imputed genotype file.

save_chr

Description: Output imputed genotyped for each chromosome in a separate file.
Usage: save_chr;
Type: optional

write_ref_target

Description: Output list of individuals in reference and target groups.
Usage: write_ref_target;
Type: optional

merge_chip

Description: Merge different chips into one chip.

Usage: merge_chip option;

option	/min_overlap = v	v is minimum required overlap for two chips to be merged.

Type: optional

Note: This command is very useful when two genotyping chips are very similar and highly overlapped. They are merged into one chip, which makes the imputation process faster.

ref_chip

Description: Specify reference chip for imputation (i.e. high density chip).

Usage: ref_chip=v;

v	is chip number to be used as reference chip.

Type: optional

add_ungen

Description: Add ungenotyped individuals in imputation process and try to impute genotypes for these individuals.

Usage: add_ungen option;

option	/min_fsize = c	Add ungenotyped individuals with minimum family size of c. Default is 4.
	/output_min_fsize = d	Save imputed genotypes for ungenotyped individuals with minimum family size of d. Default is 4.
	/output_min_call_rate = e	Save imputed genotypes for ungenotyped individuals with minimum call rate e. Default is 0.9.

Type: Optional

Note: Adding ungenotyped individuals improves the overall imputation accuracy but imputation might not be highly successful for ungenotyped individuals with small family size.

crt4w

Description: Specify call rate threshold for QC check. Program gives warning for individuals with call rate less than this threshold but these individuals are not removed.

Usage: crt4w=v;

v	is call rate threshold.

Type: optional

Default: 0.85

parentage_test

Description: Check for parentage errors.

Usage: parentage_test option;

option	/target = v	Specify target group for parentage test. v can be the chip number(s) or can be file name containing list of ID for target individuals.
option	/chip = v	Chip to be used for parentage test. v can be the chip number or can be file name pointing to pre-defined SNP list.
	/min_isnp = n	Minimum number of informative SNP required for parentage test can be specified by this command (default is 100).
	/find_match_cnflt	Find match for individuals having conflict with their parent
	/find_match_mp	Find match for individuals with missing parent (might be time consuming)
	/find_match_ugp	Find match for individuals with ungenotyped parent (might be time consuming)
	/find_identical	Find individual pairs with identical genotypes
	/ert_mm = v1	Error rate threshold to find progeny-parent mismatches (default is 0.01).
	/ert_m = v2	Error rate threshold to find progeny-parent matches (default is 0.005).
	/ert_i = v3	Error rate threshold to find individuals with identical genotypes (default is 0.001).
	/ert_s = v4	Error rate threshold to find sex conflict for males only (default is 0.05).
	/remove_conflict	When a progeny-parent conflict is detected, set the conflicting parents to missing.
	/pseudo_ped_off	When pedigree information is

not available or pedigree is not complete the program as default creates a pseudo pedigree, which is only used in population imputation part. This command skips search for pseudo pedigree.

/off Skip parentage test

Type: Optional
Default: Parentage test is on and /remove_conflict is off

keep_og

Description: Original genotypes (i.e. input genotypes) is kept intact in output imputed file
Usage: keep_og;
Type: Optional
Note: When there is Mendelian inconsistency between progeny and parents in most cases genotypes of progeny and parent are set to missing and re-imputed. This process alter some original genotypes especially if "/remove_conflict" is not specified in parentage_test. "keep_og" prevents change in original genotypes. The percentage change in original genotypes for each individual is reported in "org_vs_imp.txt" file.

exclude_snp

Description: Exclude user defined SNP
Usage: exclude_snp = "filername";
filename is the file name that contains SNP list to be excluded (no header line).
Type: Optional

exclude_chr

Description: Exclude SNP that are located on specified chromosomes.
Usage: exclude_chr = c1 c2 c3 …;
c1 c2 c3 … are chromosome numbers.
Type: Optional

exclude_chip

Description: Exclude the specified chip(s)
Usage: exclude_chip = c1 c2 c3 …;
c1 c2 c3 … are chip numbers.
Type: Optional

njob

Description: Number of jobs to run in parallel.
Usage: njob = n;
Type: Optional
Default: 1
Note: The number of job cannot be larger than the number of chromosomes. For servers with more CPU cores than the number of chromosomes you may use “nthread” to make better use of cores for faster imputation. The optimum number of jobs in many scenarios was close to 5 with “nthread” set to the number of available cores. Please note that fewer number of jobs results in lower memory requirement.

nthread

Description: Number of threads to run in parallel.
Usage: nthread = n;
Type: Optional
Default: Determined automatically.
Note: This command helps with big data to reduce both computing time and memory requirement.

chmod

Description: Set desired permission on output folder and files.
Usage: chmod = value;
value is a 3 digit number similar to that of Unix's chmod.
Type: Optional
Default 700 for folder and 600 for files.
Note: Always set read and write permissions for the owner. Because the output files are not executable the execute permission is not allowed. If execute permission is specified the program automatically ignore it. However, the execute permission is always set for the output folder.

ped_depth

Description: Set maximum generations to be traced for family imputation.
Usage: ped_depth = value;
value is the number of generations.
Type: Optional
Default 10
Note If set to zero only parents are used. In this case the accuracy is higher but the missing rate is also higher.

min_nprg_imp

Description: Set minimum number of progeny required for imputation from progeny
Usage: min_nprg_imp = value;
value is the number of progeny.
Type: Optional
Default 4

min_nsib_imp

Description: Set minimum number of sibs required for sib imputation
Usage: min_nsib_imp = value;
value is the number of sibs with minimum value of 1.
Type: Optional
Default 1
Note Set “value” to -1 to skip sib imputation. Only the 30 most informative sibs are used.

min_segm_len_fam

Description: Set minimum segment length for family imputation
Usage: min_segm_len_fam = L1 L2 L3 ...;
L1, L2 and L3 are segment lengths (in the same order of the chips).
Type: Optional

trim_segm_fam

Description: Trim head and tail of segment in family imputation
Usage: trim_segm_fam = v;
v is the portion of segment to be trimmed.
Type: Optional
Default 0.05

ref

Description: Set parameters for population imputation
Usage1: ref = n options;
n is the number of reference individuals.
option /parent Consider only individuals with progeny
option /male Consider only male individuals
option /female Consider only female individuals
Usage2: ref = "filename";
filename contains user defined list of reference individuals (multiple files can be selected; Files should be separated by space).
Usage3: ref = ped_pick; With this command, first parents of target individuals

are selected and then individuals with progeny are selected.

Type: Optional
Default ref= 50000;
Note One great feature of FImpute version 3 is the ability to efficiently handle large reference size (hundreds of thousands) for accurate imputation especially for rare variants.

target

Description: Specify list of individuals to be imputed using population information.
Usage1: target = "filename";
filename is user defined list of target individuals (multiple files can be selected; Files should be separated by space).
Usage2: target = c1 c2 c3 ...;
c1 c2 c3 ... are chip numbers.
Note This command is ignored for family imputation (i.e. all individuals are considered for family imputation).

sw_shrink_factor

Description: Shrink factor (0.02 - 0.5) for sliding windows.
Usage: sw_shrink_factor = v1 v2 v3 ...;
v1, v2 and v3 are shrink factors (in the same order of the chips).
Type: Optional
Default 0.08

sw_overlap

Description: Set amount of overlap (0.01 – 0.95) for sliding windows.
Usage: sw_overlap = v1 v2 v3 ...;
v1, v2 and v3 are overlap values (in the same order of the chips).
Type: Optional
Default 0.75

sw_min_size

Description: Set minimum sliding window size.
Usage: sw_min_size = v1 v2 v3 ...;
v1, v2 and v3 are the numbers of overlap SNP (in the same order of the chips).
Type: Optional
Default 4

sw_max_size

Description: Set maximum sliding window size.
Usage: sw_max_size = v1 v2 v3 ...;
v1, v2 and v3 are the maximum numbers of SNP (in the same order of

the chips).

Type: Optional
Default Automate
Note If set to zero for a specified chip, the program uses default value.

trim_segm_pop

Description: Trim head and tail of segment in population imputation
Usage: trim_segm_pop = v;
v is the portion of segment to be trimmed.
Type: Optional

turnoff_fam

Description: This command turns off family imputation
Usage: turnoff_fam;
Type: Optional

turnoff_pop

Description: This command turns off population imputation
Usage: turnoff_pop;
Type: Optional

turnoff_random_fill

Description: This command turns off random filling (imputation) based on allele frequency.
Usage: turnoff_random_fill;
Type: Optional

reset_hap_before_pop

Description: Specify list of individuals to be unphased after family imputation and before population imputation. This is not recommended for low density panels.
Usage1: target = "filename";
filename is user defined list of individuals (multiple files can be selected; Files should be separated by space).
Usage2: target = c1 c2 c3 ...;
c1 c2 c3 ... are chip numbers.

save_partial

Description: Save partial calls (6, 7, 8 and 9; See **hap_lib_file** command for partial codes).
Usage: save_partial;
Type: Optional
Note: In output statistics, partial calls are treated as missing.

save_genotype

Description: Saves genotypes instead of haplotypes (heterozygous loci are saved as code 1)

Usage: save_genotype;

File format: ID, chip, genotype codes
Genotype codes:
0: A1A1
1: A1A2 or A2A1
2: A2A2
5: missing

Type: Optional

save_2hap

Description: Saves two haplotypes for each individual

Usage: save_2hap;

File format: ID, haplotype codes
Allele codes:
1: A1
2: A2
5: missing

Type: Optional

save_hap_lib

Description: Save haplotype library built from reference individuals.

Usage: save_hap_lib option;

Option		
	/diplotype	This options force the program to combine two haplotypes together to save memory.
	/id	Corresponding individual ID is outputted.

File format: SNP IDs are listed in the first line.
Haplotypes start from the second line with no space between haplotype codes.
Haplotype codes:
1: A1
2: A2
5: missing

When "diplotype" option is specified the codes are:
0: A1A1
2: A2A2
3: A1A2

4: A2A1
5: missing
6: A1– (second haplotype is missing)
7: A2– (second haplotype is missing)
8: –A1 (first haplotype is missing)
9: –A2 (first haplotype is missing)

Type: Optional

random_fill

Description: Random filling (imputation) based on allele frequency. This command is useful to access minimum accuracy by random sampling of alleles based on their frequency.

Usage: random_fill;

Type: Optional

system

Description: Run a system command after FImpute finishes all processes.

Usage: system = "command";
Command is a system command.

Type: Optional

Output files:

genotypes_imp.txt

Contains ID, chip number, haplotypes.
Haplotype codes:
0: A1A1
1: Unphased heterozygous
2: A2A2
3: A1A2
4:A2A1
5: missing
6:A1–
7:A2–
8: –A1
9: –A2
First allele is paternal and the second is maternal.
If "save_genotype" is specified in control file, program outputs only genotype codes (i.e., 3 and 4 are converted to 1 and 6, 7, 8 and 9 are set to 5).

genotypes_imp_chip0.txt

Contains ID, chip number(0), imputed genotypes for ungenotyped individuals.
This file is created if command “add_ungen” with option “save_sep” is specified.

haplotypes_imp.txt

Contains ID and haplotypes. The two haplotypes for each individual are saved in two rows. The haplotype codes are 1, 2 and 5 for allele 1, allele2 and missing, respectively.
This file is created with command “save_2hap”.

snp_info.txt

Contains SNP ID, chromosome number, position and SNP order for each chip .

excluded_snp_list.txt

Contains list of excluded SNPs.

stat_snp.txt

Reports statistics on SNPs: SNP ID, chromosome number, positions, call frequencies, missing rate and minor allele frequency. Missing calls are ignored for statistics on MAF and calls 0, 1 and 2.

stat_snp_imp.txt

Reports statistics on SNPs after imputation.

stat_anim.txt

Reports statistics on individuals' genotypes: ID, chip number, call frequencies, homozygosity and missing rate. Missing calls are ignored for statistics on homozygosity and calls 0, 1 and 2.

stat_anim_imp.txt
Reports statistics on individuals' genotypes after imputation.

org_vs_imp.txt
Reports the difference between original genotypes and imputed genotypes. Large changes in the original genotypes may indicate progeny-parent conflict (use of /remove_conflict is recommended when there are parentage conflicts to increase imputation accuracy). Individuals are sorted by change% and individuals with 0% are not reported.

ref_pop.txt
Contains list of reference individuals used for population phasing and imputation.

report.txt
Detailed report on the steps carried out by the software.

af_fill_rate.txt
This file contains proportion of SNPs that are imputed randomly by sampling allele frequency of reference population for each individual. Individuals are sorted from largest to smallest proportion and individuals with 0% are not reported.

afreq_diff_dist.txt & afreq_diff_dist_imp.txt
This file reports distribution of differences in allele frequency among chips (before and after imputation). This is quality check when combining different chips from difference sources/labs.

reference_pop.txt & target_pop.txt
This files list only ID of reference and target populations.

Running the application

FImpute [control filename] -o

If *control file name* is not specified, program will prompt the user to enter it. Option *-o* forces the program to overwrite output folder if it already exists.