EZImputer

User Guide and Cookbook for 1000 genome imputation.

Version 1.0

Hugues Sicotte, Naresh Prodduturi 2013, last Updates 6/12/2014

EZImputer User Guide

Table of Contents

EZImputer User Guide

The EZImputer workflow is a modular set of scripts for genome-wide imputation. It automates many of the steps regularly needed for an imputation project. We have extensively tested the following workflow and even created work-around to compensate for bugs in the various external tools we use. We provide a cookbook with fully worked out examples (full scripts are provided in the downloadable package). This workflow is compatible with linux operating system.

The modules in this workflow should be run in the following sequence. Steps with a * are optional, based on the specific project.

1) Get_impute_reference.pl : To download impute2 1000 genome reference files.
2) Upgrade_inputmarkers_to_build37_by_DBSNP.pl*: To convert input data which is on a previous build (e.g. build 36) to the current build (e.g. build 37) coordinates using DBSNP files.
3) Filter_inputdata_by_impute_ref.pl*: To filter the input data markers based on the different 1000 genome populations minor allele frequency ranges.
4) Check_plink_data_ezimputer.pl: To prepare plink datasets for the QC and imputation scripts i.e. changing the names of the duplicate samples and duplicate markers.
5) QC_fwd_structure.pl*: To perform initial sample & marker level QC, perform plink sex check, perform forward strand mapping, and run the STRUCTURE program to get the predicted ethnicity (compared to Hapmap samples) for the samples.
6) Create_fwdstrd_ind_4ambiInputMarkers.pl*: To generate the indicator file for ambiguous markers which is used by the impute workflow to deal with ambiguous markers in the two step process.
7) Filter_1000genome_reference_by_maf.pl*: To filter and create a new impute2 imputation reference after filtering out some SNPs based thresholds for the minor allele frequency in one or more reference populations. For best quality, do not filter out SNPs except, perhaps, for monomorphic. This filtering is mainly for performace optimization. For details see paper.
8) Phase_Impute_by_parallel_proc.pl : To perform phasing and imputation.
9) Wrapper.pl: This script is a wrapper to run all the above modules serially.
10) VCF_to_plink_transpose_TPED_TFAM.pl: This script is to convert vcf files to transpose plink file set TFAM/TPED.

EZImputer User Guide

Prerequisites:

This program is only available in unix, linux, Mac OS X .

Installation

Downloading

Download the EZImputer from google code https://code.google.com/p/ezimputer/ (Using SVN)

Decide on an installation directory for ezimputer (you will need about 80MB for the Easy imputer tools and external tools).

export EZIMPUTER=”full_path_to_installation_directory”

cd ${EZIMPUTER}

# Untar the code(the name of the tar file may change based on the version you have)

tar –zxvf ezimputer.tar.gz

# You can then either install the required external tools yourself (see below) or run the following two commands.

EXTERNAL tools dependencies

A number of external tools need to be installed prior to running the workflow.

plink,impute2,shapeit,check_strands,gprobs_metric,structure

To let the workflow know where each tool is installed, you need to specify its location in a tool info (text file with any name) configuration file

We provide a binary whenever the license terms allow, but you may want to download the latest versions and copy them in the bin directory. The following script automates the installation and creates the config file.

./install_tools.sh ${PATH TO INSTALL DIRECTORY} {TOOL INFO FILE}

Appendix I: Downloading External Tools (TOOLINFO FILE)

Get_impute_reference.pl

This script downloads the impute2 reference files, untars them, and generates the reference file as a subdirectory in the path specified. The name of the downloaded dataset is to be used as the parameter for -IMPUTEREF_VERSION for the various modules. If you want to create or use different reference convert your files to impute2 reference format accordingly. You can visit this website for further information https://mathgen.stats.ox.ac.uk/impute/impute_v2.html#reference

USAGE : perl $EZIMPUTER/Get_impute_reference.pl -OUT_REF_DIR <TARGET IMPUTE REFERENCE DIRECTORY> -DOWNLOAD_LINK <DOWNLOAD LINK>

-OUT_REF_DIR -> Path to the impute reference directory (Output directory will be created if not existing)

-DOWNLOAD_LINK-> Download link for the impute [see below or visit impute website to get the download link]

Links for Impute2 1000 genome Reference data

http://mathgen.stats.ox.ac.uk/impute/data_download_1000G_phase1_integrated.html

http://mathgen.stats.ox.ac.uk/impute/ALL_1000G_phase1integrated_v3_impute.tgz

SAMPLE EXECUTION

```
#Example command (all on one line)
perl $EZIMPUTER/Get_impute_reference.pl
 -OUT_REF_DIR  /home/EzImputer_Sample_project/impute_ref
-DOWNLOAD_LINK
http://mathgen.stats.ox.ac.uk/impute/ALL_1000G_phase1integrated_v3_impute.tgz


#Will download the reference and create one file per chromosome.
#/home/EzImputer_Sample_project/impute_ref/ALL_1000G_phase1integrated_v3_chr*_impute.h
ap.gz
#The script will echo both complete path to reference and reference keyword

IMPUTE REF DIRECTORY : /home/EzImputer_Sample_project/impute_ref
IMPUTE REF KEYWORD : ALL_1000G_phase1integrated_v3

#In this case, the "VERSION" parameter for many scripts will be ALL_1000G_phase1integrated
```

INPUT FILES

The input data for the scripts should be in plink transposed fileset format (The following description of this format is taken from the plink website: http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml)

A transposed fileset contains two text files:
one (TPED) containing SNP and genotype information where one row is a SNP;
one (TFAM) containing individual and family information, where one row is an individual.

The first 4 columns of a TPED file are the same as a standard 4-column MAP file. All genotypes are listed for all individuals for each particular SNP on each line. The TFAM file is just the first six columns of a standard plink PED file.

Example

```
<------------- trans.tped ------------->      <- trans.tfam ->
1 snp1 0 5000650 A A A C C C A C C C C C      1  1  0  0  1  1
1 snp2 0 5000830 G T G T G G T T G T T T      2  1  0  0  1  1
                                              3  1  0  0  1  1
                                              4  1  0  0  1  2
                                              5  1  0  0  1  2
                                              6  1  0  0  1  2
```

This kind of format can be convenient to work with when there are very many more SNPs than individuals (i.e. WGAS data).
To read a transposed fileset (mydata.tped and mydata.tfam), using plink

```
plink --tfile mydata

http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml#tr
```

PLINK COMMANDS

(i) Convert BED(Binary plink files) to TPED (Transpose plink files)

```
plink –bfile <INPUT BINARY FILE> --recode --transpose –out <OUTPUT TRANSPOSE FILE>
```

```
Ex: plink --bfile mydata --recode --transpose --out mydata
```

(ii) Convert PED/MAP(Ped plink files) to TPED/TFAM (Transpose plink files)

```
plink –file <INPUT PED FILE> --recode --transpose –out <OUTPUT TRANSPOSE FILE>
```

```
Ex: plink --file mydata --recode --transpose --out mydata
```

SAMPLE EXECUTION

```
#sample command, all one line
Plink --bfile /home/EzImputer_Sample_project/hapmap3_r3_b36_fwd.consensus.qc.poly –recode –
transpose –out /home/EzImputer_Sample_project/hapmap3_r3_b36_fwd.consensus.qc.poly
```

DATA QC

The following QC should be performed by the user prior to running our tools.

(1) Please provide unique samples ids in the tfam files (column 2).

(2) Convert your platform specific marker Id's to rsid's based on platform annotation file.

- a. Both NGS data and the 1000 genome should be on the + strand, so alleles should match. The position for indels follows the vcf convention.
- b. For microarrays, use annotation to figure out strand. This is mainly important for ambiguous SNPs as non-ambigous SNPs can be matched by impute2.

(3) Set mother and father id to missing (plink dataset: 3rd and 4th column in the tfam/fam file, i.e. the columns should be 0 in plink).

(4) No duplicate rsid 's in the input dataset(some dbSNP snps are mapped to multiple positions)

(5) Make sure your input dataset doesn't have duplicate positions for the same chromosome.

(6) Filter out SNPs with too much missing data (or for single sample, any missing SNP).

(7) TPED file should be sorted according to chromosome and position.

- a. Note that, at this time, vcftools is buggy and doesn't produce correct tped.
- b.

(8) Filter out indels prior to running the workflow

- a. plink (used during QC steps) do not accept indels).
- b. Shapeit uses the binary plink format as input, which does not support indels.
- c. You can always add back the indels after imputation (or compare to the imputed genotypes_
- d. It is possible to recode the indels as 'fake' SNPs for QC and shapeit phasing, but you must recode them properly prior to running impute2 (imputation) because it will try to match them to the 1000 genome reference and having them falsely coded will produce poorer results.

(9) Chromosome X should be coded as 23 and Y as 24. While plink will honor chromosome 25 (the PAR regions), the inpute2 reference uses the X chromosome only(e.g. 23).

(10)Sort the input genotype file(TPED) according to chromosome and position.

```
  Example for Sorting the tped file: sort -k1,1n -k4,4n sample.tped
```

Upgrade_inputmarkers_to_build37_by_DBSNP.pl

This script is used to convert input data which is on a previous human genome coordinate build (e.g. build 36) to the current build coordinates (e.g. build 37) using DBSNP files. This step can be skipped if the data is already mapped to the same build as the 1000 genomes reference files (currently build 37)

USAGE : perl $EZIMPUTER/Upgrade_inputmarkers_to_build37_by_DBSNP.pl -DBSNP_DIR <DBSNP DIR>

-DBSNP_DOWNLOADLINK <DBSNP VERSION>

-INPUT_FILE <INPUT TPED FILE>

- REMAPPED_CURRENT_BUILD <Filename with full path>

- NOTMAPPED_OLD_BUILD <Filename with full path>

INPUT ARGUMENTS

"-DBSNP_DIR"->Target DBSNP directory for downloading DBSNP file (Output directory will be created if not existing)

"-DBSNP_DOWNLOADLINK"->DBSNP download link

DBSNP download links

#for DBSNP version 141
ftp://ftp.ncbi.nlm.nih.gov/snp/organisms/human_9606_b141_GRCh37p13/database/organism_data/b141_SNPChrPosOnRef_GRCh37p13.bcp.gz

-INPUT_FILE -> Path to input TPED file

- REMAPPED_CURRENT_BUILD ->Full path and file name to current build output TPED file for successfully remapped SNPs.

- NOTMAPPED_OLD_BUILD -> Full path and file name to old build output TPED file for unsuccessfully remapped SNPs.

SAMPLE EXECUTION

```
#sample command, all one line
perl  $EZIMPUTER/Upgrade_inputmarkers_to_build37_by_DBSNP.pl   -DBSNP_DIR
```

```
/home/EzImputer_Sample_project/DBDIR/  -DBSNP_DOWNLOADLINK
ftp://ftp.ncbi.nlm.nih.gov/snp/organisms/human_9606_b141_GRCh37p13/database/organism_data/b1
41_SNPChrPosOnRef_GRCh37p13.bcp.gz
-INPUT_FILE  /home/EzImputer_Sample_project/hapmap3_r3_b36_fwd.consensus.qc.poly.tped
 -REMAPPED_CURRENT_BUILD
/home/EzImputer_Sample_project/hapmap3_r3_b36_fwd.consensus.qc.poly_build37.tped
- NOTMAPPED_OLD_BUILD
/home/EzImputer_Sample_project/hapmap3_r3_b36_fwd.consensus.qc.poly_build36.tped
```

Check_plink_data_ezimputer.pl

To prepare plink datasets for the QC and imputation scripts i.e. changing the names of the duplicate samples and duplicate markers.

USAGE: perl $EZIMPUTER/ Check_plink_data_ezimputer.pl -INPUT_FILE < INPUT PLINK TRANSPOSE FILES> -OUTPUT_FILE < OUTPUT PLINK TRANSPOSE FILES>

-INPUT_FILE -> Path to the input plink transpose files (if the input files are sample_input.tped and sample_input.tfam files then the parameter should be “sample_input”)

-OUTPUT_FILE -> Path to the output plink transpose files (if you are expecting the outpu files are sample_input.tped and sample_input.tfam files then the parameter should be “sample_output”)

SAMPLE EXECUTION

```
perl  $EZIMPUTER/Check_plink_data_ezimputer.pl -INPUT_FILE
/home/EzImputer_Sample_project/hapmap3_r3_b36_fwd.consensus.qc.poly_build37.tped -
OUTPUT_FILE
/home/EzImputer_Sample_project/hapmap3_r3_b36_fwd.consensus.qc.poly_build37_cleaned.tped
```

QC_fwd_structure.pl

This script does following (all below options are optional)

(1) Initial sample & marker level QC : Filter out samples and markers based on the sample level missing threshold and marker level missing threshold specified in the config file.

(2) Suggestion: remove SNPs and sample call rate < 90% for Affymetrix 6.0 and < 95% and for Illumina platforms, and remove monomorphic SNPs.

(3) Plink sex check : Gender information is imputed using plink. If the gender is included in the input FAM file, this script also reports if the input gender information is not concordant with imputed one.

(4) Reference strand mapping of input data: Maps the genotypes and SNP in the input data to the same strand as the reference panel in two steps:

   (i) The alleles and genotypes for non-ambiguous SNP will be made to match the alleles of the inpute2 reference strand, reverse-complementing the alleles and genotypes if needed.

   (ii) Ambiguous SNPs (A/T or C/G SNPs) will be flipped to match the impute2 reference strand using beagle utility tools. In brief, after converting the Impute2 reference files to beagle format, the beagle utility tool rapidly imputes the genotype at the ambigous sites using nearby markers. By comparing these imputed genotypes to the input genotypes (and the reverse complement of the input genotypes), the programs is able to determine the relative strand of the input genotype needed to match the reference alleles.

   (iii) If the input marker alleles don't match reference alleles (tri allelic with reference) then the marker is ignored (not included in the output).

(5) Ethnicity QC: Runs the structure program to get the population structure for the samples.

USAGE: perl $EZIMPUTER/QC_fwd_structure.pl -run_config <PATH TO THE RUN CONFIG FILE> -tool_config <PATH TO THE TOOLINFO CONFIG FILE>

(i) Create Run config file (see below)

(ii) You need a tool info file and helper tools configured (Appendix I)

SAMPLE EXECUTION

```
perl  $EZIMPUTER/QC_fwd_structure.pl  -run_config $PWD/SAMPLEFILES/INPUTDATA/config_QC_FWD
-tool_config  $EZIMPUTER/tool_info.config
```

RUN Config file parameters :

EZImputer User Guide

You should specify all options. There should be no extra spaces in the config file.

TPED=/home/EzImputer_Sample_project/hapmap3_r3_b36_fwd.consensus.qc.poly_build37_cleaned.tped [PLINK F ORMAT TPED FILE FULL PATH]
TFAM=/home/EzImputer_Sample_project/hapmap3_r3_b36_fwd.consensus.qc.poly.tfam [PLINK F ORMAT TFAM FILE FULL PATH]
TEMP_FOLDER=/home/EzImputer_Sample_project/temp [The directory path where you want to create the temp files processing]
INNER_DIR=temp179807890 [enter unique sub directory name you want to create in 'TEMP_FOLDER' :all the temp processing will be done in this directory, if you don't want enter one the program will create one for you]
IMPUTE_REF=/home/EzImputer_Sample_project/impute_ref/ALL_1000G_phase1integrated_v3_impute [Impute reference path+"version" prefix]
OUTPUT_FOLDER=/home/EzImputer_Sample_project [Forward Stranded plink files, Indicator files,markers.ignored and QC table will be send to OUTPUT_FOLDER]
IMPUTEREF_VERSION=ALL_1000G_phase1integrated_v3 [Note: This field is the file prefix of the impute reference files. Look at the files in the impute reference file folder and enter the root prefix of the filename (bold in) **ALL_1000G_phase1integrated_v3**_chr9_impute.hap.gz]
GENOTYPE_PERCENT_CUTOFF=0.05 [Marker level missing Threshold Percentage (0-1 values) : any Marker which have more missing value percentage will be dropped]
SAMPLE_PERCENT_CUTOFF=0.05 [Sample level missing Threshold Percentage (0-1 values) : any Sample which have more missing value percentage will be dropped]
BEAGLE_REF_DB=/home/EzImputer_Sample_project/BEAGLE/ [Directory that will contain imputation reference reformatted in beagle format. For mapping ambiguous markers to impute reference strand, beagle utility program (check_strands_16May11) is used and for this impute reference should be converted to beagle reference format (these files are generated when your running for the first time and for later runs these files are reused)]
OPTION=forward_strand,missing,sexcheck,structure [Specify the options for the QC steps to be performed.Multiple options can be given as comma separated." forward_strand -> Forward Strand the input data", " missing-> Perform the missingness QC", "sexcheck-> Perform plink sex check", " structure-> Run structure on the input data"] {for all options "forward_strand,missing,sexcheck,structure"}

OUTPUTFILES

You can see output files in the output directory mentioned in the config parameter "OUTPUT_FOLDER".

OUTPUT FILE DESCRIPTIONS

(1) sample.qc : Sample QC file
(2) fwdStrandResults_input.tped: Input plink tped file in the forward strand.
(3) fwdStrandResults_input.tfam: Input plink tfam sample description file.
(4) fwdStrandResults_input.ind: Indicator file for each marker in the tped file.
(5) markers.ignored: Markers ignored in the process and not present in the fwdStrandResults_input.tped.

Output from sample qc file

QC FILE COLUMN DESCRIPTION (per sample)

- **MISSING_GENO_CLEAN** – Number of missing Genotypes after genotype level QC
- **NUM_GENO_CLEAN** - Total Number of Genotypes after genotype level QC
- **QC_SAMPLE_FLAG** – FLAG
  - ***1:*** Sample passed the QC and included
  - ***0****:* Sample failed the QC and removed
- **QC_SAMPLE_IND** – Warning FLAG [Sample Indicated as bad quality sample but not removed]
  - ***1:*** Sample passed the QC
  - ***0:*** Sample failed the QC
- **QC_SEX_IND** - FLAG for sex check QC
  - ***0:*** Plink not able to determine sex [chr 23 not available]
  - ***1:*** Provided sex information concordant with plink imputed one
  - ***2:*** Provided missing sex information and plink imputed the sex
  - ***3:*** Provided sex information is discordant with plink imputed one
  - ***NA:*** Samples are removed by QC
- **PLINK_SEX_IMP_ESTIMATE** - The actual X chromosome inbreeding (homozygosity) estimate Plink_Sex_Imp_Estimate
  - Usually float values or integers
  - 'nan' = if plink not able to determine sex[chr 23 not available]
- **EUR** **:** Percent sample to European population **(FROM STRUCTURE OUTPUT)**
- **ASN** **:** Percent sample to Asian population
- **YRI** **:** Percent sample to Yoruba population

FORWARD STRAND INDICATOR FILE GENERATED BY THE FORWARD STRAND PROGRAM

(1) Marker ID

(2) INDICATOR VALUES

  (1) ‘0’: Marker is not flipped as input alleles are on same strand as reference.

  (2) ‘1’: Marker is flipped to match input alleles to same strand reference.

  (3) ‘2’: Marker doesn’t exist on the reference.

Create_fwdstrd_ind_4ambiInputMarkers.pl

The QC module tries to correct the strand of ambiguous SNPs. For users who prefer to remove from the input data set any ambiguous SNPs, this script offers an alternative. It generates an indicator file that tells the imputation workflow to not use any ambiguous SNPs in the input data set. This indicator file is used by the imputation workflow to deal with ambiguous markers in the two step process (refer to Phase_Impute_by_parallel_proc manual). Note that there are historically very few ambiguous SNPs on Illumina genome-wide platforms (on the order of 0-3%), but this number can be much higher on Affymetrix Genotyping arrays (~ 10-15%).

```
perl  $EZIMPUTER/Create_fwdstrd_ind_4ambiInputMarkers.pl -INPUT_FILE <INPUT TPED FILE> -
OUTPUTFILE <OUTPUT FORWARD STRAND INDICATOR FILE>
```

SAMPLE EXECUTION

```
#sample command, all one line
perl  $EZIMPUTER/Create_fwdstrd_ind_4ambiInputMarkers.pl -INPUT_FILE
/home/EzImputer_Sample_project/processed_input.tped -OUTPUTFILE
/home/EzImputer_Sample_project/processed_input_ambi_strand.txt
```

Phase_Impute_by_parallel_proc.pl

This is the module that performs the imputation two steps

(i) STEP 1 : Phasing
  a. Tool Used: SHAPEIT program

(ii) STEP 2 : Imputation
  a. Tool Used: IMPUTE2 program

You can perform

(i) Phasing only

(ii) Imputation only, using data pre-phased from another phasing run. You would do that if you downloaded a more recent reference or were trying different filtering settings for the reference.

(iii) Phasing and Imputation

NOTE:

(1) The workflow assumes that you are starting with tfam/tped files. These can be exported directly from Plink.

(2) Please provide unique samples ids in the tfam files (column 2).

(3) Convert your platform specific marker Id's to rsid's based on platform annotation file.

(4) If your input data is build36 , upgrade it to build 37 because the workflow currently only supports build 37 reference samples.

(5) Set mother and father id to missing (plink dataset: 3rd and 4th column in the tfam/fam file, i.e. the columns should be 0 in plink).

(6) No duplicate rsid 's in the input dataset.

(7) While the imputation imputes insertion/deletion genotypes, the workflow cannot accept insertion/deletions as input genotype data.

(8) Make sure your input dataset doesn't have duplicate positions for the same chromosome.

(9) For imputing a single chromosome or small region, truncate the plink input data files to the desired region.

(10) The imputation process may take a few days to complete depending on the request, so it is best practice to submit the job to a compute cluster (cluster managed with SGE or PBS are supported).

(11) There should be no extra spaces in the config files.

STEPS

(After the installation of the workflow and the external tools)

(i) First step is to download the impute reference (here).

(ii) Create config file (see section on Config file at the end)

a. For Phasing only mode

    i. set config parameter SHAPEITONLY=YES

b. For Imputation only (If data is already phased),

    i. Set config parameter SHAPEITONLY=NO
    ii. Set config parameter HAPS=<PATH to the tar file> (see below)

c. For Phasing and Imputation

    i. Set config parameter SHAPEITONLY=NO
    ii. Set config parameter HAPS=NA (see below)

(iii) Downloading External tools & preparing tool info file

(iv) Invoke the script

SAMPLE EXECUTION

```
#perl perl_workflow_impute.pl -run_config <PATH TO THE RUN CONFIG FILE> -tool_config <PATH TO
THE TOOL CONFIG FILE>
#sample command, all one line
perl  $EZIMPUTER/Phase_Impute_by_parallel_proc.pl –run_config $PWD/SAMPLEFILES/INPUTDATA/
config_phase_impute -tool_config  $PWD/SAMPLEFILES/INPUTDATA/tool_info.config
```

Optional modules

Filter_inputdata_by_impute_ref.pl

This script filters the input data markers based on the different 1000 genome population or subpopulations minor allele frequency ranges. This script also automatically filters out markers not on the reference since they are not needed for imputation.

Filtering extremely rare SNPs can speed up imputation. Additionally, it has been shown by Hancock et al (2012) [see article for citation] that for samples of multi-ethnicity origin, it is best to eliminate SNPs monomorphic in subpopulations relevant to those samples. For example, for imputing genotypes for African-American samples, exclude SNPs monomorphic in European or African populations, e.g. –POP AFR,CEU –LMAF 0.0001,0.0001 . We use 0.0001 instead of 0, since the –LMAF limit is inclusive.

USAGE : perl $EZIMPUTER/Filter_inputdata_by_impute_ref.pl

-POPULATION <Different population : afr,amr,asn,eur>

-LMAF <Lower Limit Minor allele frequency, inclusive>

-UMAF <Upper Limit Minor allele frequency, inclusive>

-REF_GENOME_DIR <Reference Genome Directory>

-IMPUTEREF_VERSION <IMPUTE REFERENCE directory name (e.g. ALL_1000G_phase1integrated_v3)>

-INPUT_FILE <INPUT TPED FILE>

-OUTPUTFILE <OUTPUT TPED FILE>

INPUT ARGUMENTS

-POPULATION -> The user can specify populations or sub populations. Multiple populations can be separated by ‘,’. Available populations are AFR, AMR, ASN, EUR and available sub populations are ASW,CEU,CHB,CHS,CLM,FIN,GBR,IBS,JPT,LWK,MXL, PUR,TSI,YRI .

-LMAF -> Lower limit of minor allele frequency for each population

-UMAF -> Upper limit of minor allele frequency for each population

-REF_GENOME_DIR-> Path to the IMPUTE2 Reference genome directory

-IMPUTEREF_VERSION -> ALL_1000G_phase1integrated_v3

> [Note: This field is the file prefix of the impute reference files. Look at the files in the impute reference file folder and enter the root prefix (boldface part) of the filename **ALL_1000G_phase1integrated_v3**_chr9_impute.hap.gz]

-INPUT_FILE -> Full path and filename to the INPUT TPED FILE

-OUTPUTFILE -> Full path and filename to the OUTPUT TPED FILE

SAMPLE EXECUTION

```
#sample command, all one line

perl $EZIMPUTER/Filter_inputdata_by_impute_ref.pl
-POPULATION afr,eur
-LMAF 0.01,0.01
 -UMAF 0.5,0.5
-REF_GENOME_DIR
/home/EzImputer_Sample_project/impute_ref/ALL_1000G_phase1integrated_v3_impute
-IMPUTEREF_VERSION ALL_1000G_phase1integrated_v3
-INPUT_FILE  /home/EzImputer_Sample_project/processed_input.tped
 -OUTPUT_FILE  /home/EzImputer_Sample_project/Filter_processed_input_ambi_strand.txt
```

Filter_1000genome_reference_by_maf.pl

This script will create a new copy of the imputation reference after removing SNPs that fail to be within the specified limits of population specific Minor Allele Frequency. [An alternative, which is not supported by EZImputer, is to use the new reference is to extract the list of SNPs passing QC and use the 'include' option in the original impute2 software (this option can restrict the markers to be included in the imputation process), but this option will increase the running time when compared to using all the markers in the reference.]

USAGE : perl $EZIMPUTER/Filter_1000genome_reference_by_maf.pl

-POPULATION <Different population : AFR,AMR,ASN,EUR>

-LMAF <Lower Limit Minor allele frequency>

-UMAF <Upper Limit Minor allele frequency>

-REF_GENOME_DIR <Reference Genome Directory>

-OUTPUT_DIR <OUTPUT DIR>

-IMPUTEREF_VERSION <version of the impute reference files e.g. ALL_1000G_phase1integrated_v3>

-INCLUDE_POP < Populations to be included in the new reference>

INPUT ARGUMENTS

-POPULATION -> You can specify populations or sub populations. Multiple populations can be separated by ','. Available populations are AFR, AMR, ASN, EUR and available sub populations are ASW,CEU,CHB,CHS,CLM,FIN,GBR,IBS,JPT,LWK,MXL, PUR,TSI,YRI . When downloading the 1000 genome data, a file (e.g. ALL_1000G_phase1integrated_feb2012.sample) is downloaded containing the population definitions that can be used by this script. This file will be downloaded in the directory specified by the OUT_REF_DIR argument of Get_impute_reference.pl (optional)

-LMAF -> Lower limit of minor allele frequency for each population (optional)

-UMAF -> Upper limit of minor allele frequency for each population (optional)

-REF_GENOME_DIR-> Path to the IMPUTE2 Reference genome directory

-IMPUTEREF_VERSION -> ALL_1000G_phase1integrated_v3

> [Note: This field is the file prefix of the impute reference files. Look at the files in the impute reference file folder and enter the root prefix of the filename (bold in) **ALL_1000G_phase1integrated_v3**_chr9_impute.hap.gz]

-INPUT_FILE -> PATH to the INPUT TPED FILE

-OUTPUT_DIR -> PATH to the NEW REFERENCE FILES

-INCLUDE_POP -> Populations to be included in the new reference (other populations are removed)

SAMPLE EXECUTION

```
#sample command, all one line
perl  $EZIMPUTER/Filter_1000genome_reference_by_maf.pl
-POPULATION eur
-LMAF 0.005
-UMAF  0.5
-REF_GENOME_DIR
/home/EzImputer_Sample_project/impute_ref/ALL_1000G_phase1integrated_v3_impute
 -OUTPUT_DIR
/home/EzImputer_Sample_project/impute_ref/ALL_1000G_phase1integrated_v3_impute_filtered_eur
 -IMPUTEREF_VERSION ALL_1000G_phase1integrated_v3
-INCLUDE_POP EUR

#Note that after running this script, you have to update the config file with the new reference directory
(IMPUTE_REF) but  not change the IMPUTEREF_VERSION

IMPUTE_REF=/home/EzImputer_Sample_project/impute_ref/ALL_1000G_phase1integrated_v3_impute_eur
IMPUTEREF_VERSION= ALL_1000G_phase1integrated_v3
```

Wrapper.pl

This script is a wrapper to run all the above modules serially. The wrapper script takes input parameters for all the modules and executes each module accordingly.

USAGE : perl $EZIMPUTER/Wrapper.pl -wrapper_config <PATH TO THE RUN CONFIG FILE> -tool_config <PATH TO THE TOOL CONFIG FILE>

INPUT ARGUMENTS

-wrapper_config: Runinfo Config file with all the run parameters required to run each module

-tool_config : Tool info Config file with path to the tools used (Tool info file generated by the install_tools.sh. [Required Parameters : *]

RunInfo Config file parameters for imputation

- INPUT_PLINK=/home/EzImputer_Sample_project/processed_input[Path to the binary Plink file set]**[Required]**
- MODULES_NEEDED=UPGRADE_BUILD, QC_REQUIRED, REF_FILTER, IMPUTE [Keyword to invoke each module. Use 'UPGRADE_BUILD' keyword if you want to run the script 'Upgrade_inputmarkers_to_build37_by_DBSNP.pl'
  Use 'QC_REQUIRED' keyword if you want to run the script 'QC_fwd_structure.pl'
  Use 'REF_FILTER' keyword if you want to run the script 'Filter_1000genome_reference_by_maf.pl'
  Use 'IMPUTE' keyword if you want to run the script 'Phase_Impute_by_parallel_proc.pl']
  [Required]
- IMP2_OUT_DIR=/home/EzImputer_Sample_project/Impute_out_dir [Path to the process directory (temp & output directory)] **[Required]**
- DEAL_AMBIGUOUS=YES [Use this if you want to deal with ambiguous markers(For more information)] **[Required]**
- IMPUTE_REF=/home/EzImputer_Sample_project/impute_ref/ALL_1000G_phase1integrated_v3_impute [Impute reference path+"version" prefix] **[Required]**
- IMPUTEREF_VERSION=ALL_1000G_phase1integrated_v3 [Note: This field is the file prefix of the impute reference files. Look at the files in the impute reference file folder and enter the root prefix of the filename (bold in) **ALL_1000G_phase1integrated_v3**_chr9_impute.hap.gz] **[Required]**
- USERNAME=mXXXXX [To create unique directories in the local temp directory to identify the user] **[Required]**
- EMAIL=lastname.firstname@mayo.edu [Fill in user's full e-mail address] **[Required]**

Remaining unique parameters for each module
UNIQUE UPGRADE_BUILD PARAMETERS

- DBSNP_DOWNLOADLINK= ftp://ftp.ncbi.nlm.nih.gov/snp/organisms/human_9606_b141_GRCh37p13/database/organism_data/b141_SNPChrPosOnRef_GRCh37p13.bcp.gz(Target DBSNP directory for downloading DBSNP file (Output directory will be created if not existing)) **[Required]**

- DBSNP_DIR=/home/EzImputer_Sample_project/DBDIR(DBSNP download directory)

QC_REQUIRED PARAMETERS

- GENOTYPE_PERCENT_CUTOFF=0.05 [Marker level missing Threshold Percentage (0-1 values) : any Marker which have more missing value percentage will be dropped]
- SAMPLE_PERCENT_CUTOFF=0.05 [Sample level missing Threshold Percentage (0-1 values) : any Sample which have more missing value percentage will be dropped]
- BEAGLE_REF_DB=/home/EzImputer_Sample_project/BEAGLE/ [Directory that will contain imputation reference reformatted in beagle format. For mapping ambiguous markers to impute reference strand, beagle utility program (check_strands_16May11) is used and for this impute reference should be converted to beagle reference format (these files are generated when your running for the first time and for later runs these files are reused)]
- OPTION=forward_strand,missing,sexcheck,structure [Specify the options for the QC steps to be performed.Multiple options can be given as comma separated." forward_strand -> Forward Strand the input data", " missing-> Perform the missingness QC", "sexcheck-> Perform plink sex check", " structure-> Run structure on the input data"] {for all options "forward_strand,missing,sexcheck,structure"}
-

REF_FILTER PARAMETERS

- LMAF= 0.005,0.005,0.005,0.005(Lower limit of minor allele frequency for each population (optional))
- UMAF= 0.5,0.5,0.5,0.5(Upper limit of minor allele frequency for each population (optional))
- POPULATION= ALL,AFR,AMR,ASN,EUR (You can specify populations or sub populations. Multiple populations can be separated by ','. Available populations are AFR, AMR, ASN, EUR and available sub populations are ASW,CEU,CHB,CHS,CLM,FIN,GBR,IBS,JPT,LWK,MXL, PUR,TSI,YRI . When downloading the 1000 genome data, a file (e.g. ALL_1000G_phase1integrated_feb2012.sample) is downloaded containing the population definitions that can be used by this script.)
- INCLUDE_POP= ALL,AFR,AMR,ASN,EUR(Populations to be included in the new reference (other populations are removed))

IMPUTE PARAMETERS

- FORWARDSTRAND_IND=/home/EzImputer_Sample_project/processed_input_ambi_strand.txt [fwd strand indicator file if you have one or place 'NA' [no gzip files allowed]] [If you want to create a indicator file specially to deal with ambiguous snps you can use this script Create_fwdstrd_ind_4ambiInputMarkers*).
- IMPUTE_WINDOW=5000000 [Impute window size you want to chop, should be between 2mb and 5mb]
- IMPUTE_EDGE=125 [buffer region for the windows ,must be in KB]
- HAPS=/data4/bsi/RandD/Workflow/temp/shapeit_jobs.tar.gz [If you have pre-phased already done then give path to tar file or enter 'NA']
- SGE_SHAPEIT_MEM=15G [Expected memory taken by your shape it jobs] [even if your shapeit jobs are done just give some numbers][optional parameter but should present when PBS=NO]
- SGE_SHAPEIT_QUEUE=1-day [Queue you want submit for shapeit jobs] [even if your shapeit jobs are done just give some numbers] [optional parameter but should present when PBS=NO]
- SGE_IMPUTE_MEM=15G [Expected memory taken by your impute jobs] [optional parameter but should present when PBS=NO]

- SGE_IMPUTE_QUEUE=1-day [Queue you want submit for impute jobs] [optional parameter but should present when PBS=NO]
- IMPUTEREF_VERSION= ALL_1000G_phase1integrated_v3

  [Note: This field is the file prefix of the impute reference files. Look at the files in the impute reference file folder and enter the root prefix of the filename (bold in) **ALL_1000G_phase1integrated_v3**_chr9_impute.hap.gz]

- LOCALTEMP_SHAPEIT=4G [Local temp space required by the shapeit sungrid engine jobs to compress result files]
- LOCALTEMP_IMPUTE=1G [Local temp space required by the impute sungrid engine jobs to compress result files]
- LOCALTEMP= /home/EzImputer_Sample_project/temp/LOCALTEMP [Path to local temp space required by the impute sun grid engine jobs to compress result files. If your SGE doesn't have enough space on the computational nodes you can point to the TEMP_FOLDER]
- RESTART=NO [(1) 'NO' for initial run<no restart>. (2) 'SHAPEIT' to restart the workflow if some phasing (shapeit) jobs fail at phasing state. (3) 'IMPUTE' to restart the workflow if some impute jobs fail at imputation state. (4) 'POST' to restart the workflow if some post processing jobs fail at post processing state(generation of r2 files, indicator files). Please remember and enter the same 'INNER_DIR" name as of the initial run for options 'SHAPEIT', 'IMPUTE' and for 'POST'.]
- SHAPEITONLY=YES(IF you want just the prephasing set this option to 'YES" or else use 'NO')
- PBS=NO (This should be set to YES if your running on PBS system instead of SUNGRID engine (For BORA set this option to YES if you are running jobs on iforge))
- PBS_PARAM="-l walltime=10:05:00,-l nodes=1:ppn=1,-A normal,-A bf0"[Should be in double quotes "" and should include only '-l' (wall clock times and number of nodes, make sure you select optimum time for each shape it and impute job each. Some shapeit<Phasing> jobs may take few days to complete) and '–A' (account string and queue type) remaining parameters are filled by the workflow] [optional parameter but should present when PBS=YES]
- CHR_START_INPUT=YES (Set this option to 'YES' if you want impute to small regions instead of whole chromosomes. So the imputation start position and stop position will be based on input data but not on the reference start and stop)
- SMALL_REGION_EXTN_START=2000000 (If you want to extend the imputation on the start of the small region. Please specify bases in unit Bases) [optional parameter but should present when CHR_START_INPUT=YES] [optional parameter but should present when CHR_START_INPUT=YES]
- SMALL_REGION_EXTN_STOP=2500000 (If you want to extend the imputation on the end of the small region. Please specify bases in unit Bases) [optional parameter but should present when CHR_START_INPUT=YES] [optional parameter but should present when CHR_START_INPUT=YES]
- WINDOW_CUTOFF_NUM_MARKERS=200 (Minimum number of input data and reference markers to be present on a window for imputation. If the number of markers are less than cutoff then the window will be merge to next one to meet the cutoff).
- EDGE_CUTOFF_NUM_MARKERS =50 (Minimum number of input data markers to be present on a buffer region for imputing a window. If the numbers of markers are less than cutoff then the edges will be extended up to 2 MB to meet the cutoff).
- SHAPEIT_EXTRA_PARAM="--seed 123456789 --states 100 --thread 4" (This config file parameter is to pass parameters to the shapeit tool.Default value suggested by the SHAPEIT manual for

states parameter is 100. To specify the number of conditioning haplotypes to be used during the phasing process. The running time of the algorithm increases only linearly with this number).

- LESS_NUM_SAMP=”NO” (If the dataset to be imputed has only 1 sample then use LESS_NUM_SAMP=”YES”, SHAPEIT tool will combine the reference data with the input dataset and does the phasing. There will be no change in the imputation process).
- CHECK_POINT_STOP=SHAPEIT (If this workflow is not compatible with your job scheduler like SGE/PBS then you can run the workflow manually. use this option (here).)(optional parameter)

VCF_to_plink_transpose_TPED_TFAM.pl

This script will convert the vcf file to plink transpose files ‘tped’ & ‘tfam’ files and optionally ‘lowQual’ and ‘coverage’ filtering can be applied.(Failed SNVs will be set to missing). LowQual filtering will be not applied to reference homozygote calls.

USAGE : perl VCF_to_plink_transpose_TPED_TFAM.pl

-i <input vcf file (can be compressed gz file)>

-o <Path to plink output set tfam & tped files will be generated>

-d <YES/NO (Want to include indels or not)>

-l <YES/NO)

-c <cutoff coverage must be a number & greater than or equal to 0>>

INPUT ARGUMENTS

-i -> Input vcf file. Optionally input vcf file can be gzip compressed file.

-o -> Output file set name. Files ‘tped’ and’ tfam’ will be generate by concatenating the ‘.tped’ and ‘.tfam’ to the output file set name.

-l -> YES/NO Lowqual filter will be applied is the argument is yes.

-c ->Number greater than or equal to 0. Coverage cutoff will be applied according to column ‘INFO’ and field DP in the vcf file.

-d -> YES/NO. Indel will be included in the output if argument is YES.

SAMPLE EXECUTION

```
#sample command, all one line
perl  $EZIMPUTER/VCF_to_plink_transpose_TPED_TFAM.pl
-i /home/EzImputer_Sample_project/input.vcf
-o /home/EzImputer_Sample_project/input
-l yes
 -c 7
-d no
```

Appendices

Appendix A: Specifying the files with the pre-phased data.

PATH to the tar file:

When data is already phased, the user needs to provide the previously phased data in a tar file. Here is how the workflow expects the tar-file of existing haplotype. This directory structure will be created by the workflow, it is only documented for the user who has already run the workflow once, and wants to use pre-phased data with a new release of the imputation reference.

Create a directory and subdirectories with names chromosome names from 1-23 (if chr X exist in your dataset, it is named 23)

For chromosomes 1-22, files needed

(1) Haps file (Output from shapeit program)
(2) Sample file (Output from shapeit program)

For chromosome 23, files needed

(1) Haps file (Output from shapeit program)
(2) TFAM file with SEX code non-missing (column 5)
(3) Sample file (Output from shapeit program)

Tar & gzip compress directory : tar -zcf <PATH TO TAR.GZ FILE > -C <TARGET DIRECTORY> .

Ex: tar -zcf <OUTPUT DIRECTORY>/shapeit_jobs.tar.gz -C <INPUT FOLDER>/SHAPEIT/ .

#To uncompress phased output tar file(example) :

# To uncompress tar –C <PATH TO THE TARGET DIRECTORY > -xzf <PATH TO THE TAR.GZ FILE>

Ex: tar -C <PATH TO THE TARGET DIRECTORY > / -xzf <PATH TO THE TAR.GZ FILE>/shapeit_jobs.tar.gz

Appendix B: Ambiguous Markers

FORWARD STRAND INDICATOR FILE

The imputation workflow will automatically flip alleles to match the alleles of the reference genotypes. However, for A/T and C/G SNPs, this is not possible (ambiguous SNPs). In this case, the user can provide a "forward strand indicator file" to the workflow to indicate how the alleles in the data should be flipped (for any SNP).

(1) Tab delimited file
(2) First column should be the marker id
(3) Second column should be indicator values ('0' or '1')
    a. values>0->Data for this marker is on the forward strand,
    b. values > 1 -> Data for this marker is not on the forward strand.
(4) Ambiguous SNPs that do not have indicator values '0' or '1' are removed at the beginning of the imputation process , and re-imputed genome-wide. After imputation, the normalization process will compare the imputed genotypes with two alternative genotypes: original orientation and reverse-complemented orientation. It will replace the imputed values with original genotypes that match best the imputed genotypes (and keep missing values).

p.s. The program "QC_fwd_structure.pl" can pre-correct SNPs to the forward strand (this program will flip the markers to forward strand according to 1000 genome markers and platform specific probes) . This program generates the forward strand indicator file.

DEALING WITH AMBIGUOUS MARKERS

If you're not sure about the strand information for ambiguous markers then create a FORWARD STRAND INDICATOR file for markers and give indicator value greater than 1. This is done with the script

Appendix C : Small region imputation

SINGLE CHROMSOME OR SMALL REGION IMPUTATION

First, subset the plink input data files to the region you want to impute.

SMALL REGION IMPUTATION

For small region imputations, supply a truncated input dataset to the imputation workflow. The parameters you need to set in the config file are [do not enter the comments in brackets]

- CHR_START_INPUT=YES [Set this option to 'YES' if you want impute to small regions instead of whole chromosomes. So the imputation start position and stop position will be based on input data but not on the reference start and stop]
- SMALL_REGION_EXTN_START=2000000[If you want to extend the imputation region on start end for small region imputation, you can specify here in unit bases. If you do not want any extension just set this option to '0')]
- SMALL_REGION_EXTN_STOP=3000000[If you want to extend the imputation region on stop end for small region imputation, you can specify here in unit bases. If you do not want any extension just set this option to '0']

Note: The region will be extended by the buffer region as needed, but no result will be generated for the buffer region.

Appendix D: Config File Parameters

There are two configuration files. The first one is used by the main imputation workflow, while the second one(tool info) is used by many scripts to know the location of tools.

Config file parameters for imputation (You should specify all options, but omit the comments within the square brackets)

- TPED=/home/EzImputer_Sample_project/fwdStrandResults_input.tped.gz [gzipped TPED FILE FULL PATH [Strictly gzip file is required]]
- TFAM=/home/EzImputer_Sample_project/fwdStrandResults_input.tfam [TFAM FILE FULL PATH]
- FORWARDSTRAND_IND=/home/EzImputer_Sample_project /fwdStrandResults_input.ind [fwd strand indicator file if you have one or place 'NA' [no gzip files allowed]] [If you want to create a indicator file specially to deal with ambiguous snps you can use this script Create_fwdstrd_ind_4ambiInputMarkers*).
- TEMP_FOLDER=/home/EzImputer_Sample_project/temp [The directory path where you want to create the temp files processing and output files]
- IMPUTE_REF=/home/EzImputer_Sample_project/impute_ref/ALL_1000G_phase1integrated_v3_impute[Impute reference path you want to use]
- IMPUTE_WINDOW=5000000 [Impute window size you want to chop, should be between 2mb and 5mb]
- IMPUTE_EDGE=125 [buffer region for the windows ,must be in KB]
- HAPS=/data4/bsi/RandD/Workflow/temp/shapeit_jobs.tar.gz [If you have pre-phased already done then give path to tar file or enter 'NA']
- EMAIL=lastname.firstname@mayo.edu [Fill in user's full e-mail address]
- SGE_SHAPEIT_MEM=15G [Expected memory taken by your shape it jobs] [even if your shapeit jobs are done just give some numbers][optional parameter but should present when PBS=NO]
- SGE_SHAPEIT_QUEUE=1-day [Queue you want submit for shapeit jobs] [even if your shapeit jobs are done just give some numbers] [optional parameter but should present when PBS=NO]
- SGE_IMPUTE_MEM=15G [Expected memory taken by your impute jobs] [optional parameter but should present when PBS=NO]
- SGE_IMPUTE_QUEUE=1-day [Queue you want submit for impute jobs] [optional parameter but should present when PBS=NO]
- IMPUTEREF_VERSION= ALL_1000G_phase1integrated_v3

  [Note: This field is the file prefix of the impute reference files. Look at the files in the impute reference file folder and enter the root prefix of the filename (bold in) **ALL_1000G_phase1integrated_v3**_chr9_impute.hap.gz]

- LOCALTEMP_SHAPEIT=4G [Local temp space required by the shapeit sungrid engine jobs to compress result files]
- LOCALTEMP_IMPUTE=1G [Local temp space required by the impute sungrid engine jobs to compress result files]
- LOCALTEMP= /home/EzImputer_Sample_project/temp/LOCALTEMP [Path to local temp space required by the impute sun grid engine jobs to compress result files. If your SGE doesn't have enough space on the computational nodes you can point to the TEMP_FOLDER]

- INNER_DIR=temp179807890 [enter unique sub directory name you want to create in 'TEMP_FOLDER' :all the temp processing will be done in this directory, if you don't enter one the program will create one for you]
- RESTART=NO [(1) 'NO' for initial run<no restart>. (2) 'SHAPEIT' to restart the workflow if some phasing (shapeit) jobs fail at phasing state. (3) 'IMPUTE' to restart the workflow if some impute jobs fail at imputation state. (4) 'POST' to restart the workflow if some post processing jobs fail at post processing state(generation of r2 files, indicator files). Please remember and enter the same 'INNER_DIR" name as of the initial run for options 'SHAPEIT', 'IMPUTE' and for 'POST'.]
- USERNAME=mXXXXX [To create unique directories in the local temp directory to identify the user]
- SHAPEITONLY=YES(IF you want just the prephasing set this option to 'YES" or else use 'NO')
- PBS=NO (This should be set to YES if your running on PBS system instead of SUNGRID engine (For BORA set this option to YES if you are running jobs on iforge))
- PBS_PARAM="-l walltime=10:05:00,-l nodes=1:ppn=1,-A normal,-A bf0"[Should be in double quotes "" and should include only '-l' (wall clock times and number of nodes, make sure you select optimum time for each shape it and impute job each. Some shapeit<Phasing> jobs may take few days to complete) and '–A' (account string and queue type) remaining parameters are filled by the workflow] [optional parameter but should present when PBS=YES]
- CHR_START_INPUT=YES (Set this option to 'YES' if you want impute to small regions instead of whole chromosomes. So the imputation start position and stop position will be based on input data but not on the reference start and stop)
- SMALL_REGION_EXTN_START=2000000 (If you want to extend the imputation on the start of the small region. Please specify bases in unit Bases) [optional parameter but should present when CHR_START_INPUT=YES] [optional parameter but should present when CHR_START_INPUT=YES]
- SMALL_REGION_EXTN_STOP=2500000 (If you want to extend the imputation on the end of the small region. Please specify bases in unit Bases) [optional parameter but should present when CHR_START_INPUT=YES] [optional parameter but should present when CHR_START_INPUT=YES]
- WINDOW_CUTOFF_NUM_MARKERS=200 (Minimum number of input data and reference markers to be present on a window for imputation. If the number of markers are less than cutoff then the window will be merge to next one to meet the cutoff).
- EDGE_CUTOFF_NUM_MARKERS =50 (Minimum number of input data markers to be present on a buffer region for imputing a window. If the numbers of markers are less than cutoff then the edges will be extended up to 2 MB to meet the cutoff).
- SHAPEIT_EXTRA_PARAM="--seed 123456789 --states 100 --thread 4" (This config file parameter is to pass parameters to the shapeit tool.Default value suggested by the SHAPEIT manual for states parameter is 100. To specify the number of conditioning haplotypes to be used during the phasing process. The running time of the algorithm increases only linearly with this number).
- LESS_NUM_SAMP="NO" (If the dataset to be imputed has only 1 sample then use LESS_NUM_SAMP="YES", SHAPEIT tool will combine the reference data with the input dataset and does the phasing. There will be no change in the imputation process).
- CHECK_POINT_STOP=SHAPEIT (If this workflow is not compatible with your job scheduler like SGE/PBS then you can run the workflow manually. use this option (here).)(optional parameter)

Appendix E [Output Files]

RESULTS

The TEMPFOLDER/ INNER_DIR (You specified this in the config file) will be your final output folder. All the below files are located in the individual chromosome directories (restricted to chromosomes present in the input data).For whole genome imputation you can see folders from 1-23 (From chromosome 1 to chromosome 23) and for small region imputation you can only see the chromosomes present in the input data.

(i) **final.2prob.gz (Two probability file)** : Individual window two probability output files will be merged at the chromosome level. (One file per chromosome) . The two likelihood format reports two numbers (F1 I1) per genotype (F1=Likelyhood of Hom of first Allele, and I1=Likelyhood of Het).

(ii) **dosage.gz :** Dosage file per chromosome. (One file per chromosome).

(iii) **Combined_impute_results_3_prob_ambi_out.gz :** Three probability file for SNVs only (not indels).
Columns are '---', markerid, position, AlleleA, AlleleB, and then 3 columns for each samples representing 3 genotype likelihoods per samples for the AA, AB, and BB genotypes.

(iv) **beagle_r2.gz :** Output from beagle utility (gprobsmetrics.jar for calculating various R2 values) . This file does not contain r2 values for indel polymorphisms. See below section for header description.

(v) **final.map.gz :** Plink map file is generated for all the markers. (COLUMN DESCRIPTION: chromosome, rsid, distance, position).

(vi) **Combined_impute_results_3_prob_indels_out.gz :** Imputed indels are separated.

(vii) **Combined_impute_results_3_prob_ind.gz** : Indicator file(matrix) for SNVs to differentiate observed and imputed marker(rows) and sample(columns).(Column 1 : RSID, Column2 : Dosage R2 value, From column 3 : sample 1, Column 4 : sample2 ……Rows are markers and Columns are samples. Sample order matches tfam file. Indicator "1"is imputed and Indicator "0" is observed).

(viii) Phasing output ".haps" files from all the chromosomes will be combined and tar compressed "shapeit_jobs.tar.gz" and can be found in the SHAPEIT_OUTPUT directory.

Copy the necessary output file and delete the temp folder. The only reason we retained all the temp files even after the run is successful to trace the unexpected errors easily.

HEADERS FOR BEAGLE R2 FILES

1) marker identifier

2) minor allele

3) minor allele frequency

4) allelic r-squared

5) dosage r-squared

6) HWE qdosage r-squared

7) accuracy

8) missing score

Notes:

a) Allelic r-squared is the estimated squared correlation between the most likely allele dosage (rounded to the nearest integer) and the true allele dosage.

b) Dosage r-squared is the estimated squared correlation between the estimated allele dosage (0*P(Hom Ref/first) + 1*P(AB) + 2*P(Hom Alt/second)) and the true allele dosage.

c) HWE dosage r-squared, is the estimated squared correlation between the estimated allele dosage and the true allele dosage when the variance of the true allele dosage is calculated from the estimated allele frequency.

d) Accuracy is the estimated genotype accuracy based on the posterior probability of the most likely genotype.

e) Missing score is the infinum of the C > 0 such that the proportion of genotype with probability < C is greater than or equal to (1.0 - C).

f) Monomorphic markers are assigned r-squared values of 0, and accuracy and missing score values of 1.

g) The squared correlation metrics can be derived using arguments found in Appendix 1 of "Browning BL and Browning SR, Am J Hum Genet 2009;84(2):210-23".

h) For comparison purposes, we report all r-squared metrics. The allelic r-squared metric is the default reported by BEAGLE and the dosage r-squared metric is the default reported by MACH.

Appendix F: Restart workflow at different stages [Job failed?]

If you want to know at what stage the work flow got failed just check for the log file produced by the main job and log file directory present in the temp directory.

'shapeit_logfiles_sungrid' -> Workflow got failed at phasing stage.

'impute_logfiles_sungrid' -> Workflow got failed at imputation stage.

'post_logfiles_sungrid' -> Workflow got failed at postprocessing stage.

Restart Workflow at Phasing Phase [SHAPEIT]

If one of your phasing jobs got failed, first check for the error log file in the 'shapeit_logfiles_sungrid' directory check for the exact reason, correct the input files if required and rerun the workflow by changing the parameter "RESTART=SHAPEIT"

Restart Workflow at Imputation Phase

If one of your imputation jobs got failed, first check for the error log file in the 'impute_logfiles_sungrid' directory check for the exact reason, correct the input files if required and rerun the workflow by changing the parameter "RESTART=IMPUTE"

Restart Workflow at Post Processing Phase

If one of your post processing jobs got failed, first check for the error log file in the 'post_logfiles_sungrid' directory check for the exact reason, correct the input files if required and rerun the workflow by changing the parameter "RESTART=POST"

Appendix G: Links (Reference Files)

1000 GENOME REFERENCE

1000 genome imputation reference is available at

http://mathgen.stats.ox.ac.uk/impute/data_download_1000G_phase1_integrated.html

DBSNP download links

#for DBSNP version 141
ftp://ftp.ncbi.nlm.nih.gov/snp/organisms/human_9606_b141_GRCh37p13/database/organism_data/b141_SNPChrPosOnRef_GRCh37p13.bcp.gz

Impute2 Reference links

http://mathgen.stats.ox.ac.uk/impute/data_download_1000G_phase1_integrated.html

http://mathgen.stats.ox.ac.uk/impute/ALL_1000G_phase1integrated_v3_impute.tgz

Appendix H: WORKFLOW NOT SUPPORTING JOB SCHEDULER PARAMETERS SGE or PBS (RUNNING WORKFLOW MANUALLY)

RUNNING WORKFLOW MANUALLY

If your job scheduler parameters are different from config parameters and so you cannot run the workflow automatically. Then you can still use the imputation work flow manually as follows

(1) Phasing

(i) Create the config files ad mentioned in the manual.
(ii) select the config file parameter “CHECK_POINT_STOP=SHAPEIT” & “RESTART=NO”.
(iii) Run the workflow and the array job script “ArrayJob_shapeit.csh” in the temp folder will be generated.
(iv) Edit the SGE or PBS parameters accordingly in this script “ArrayJob_shapeit.csh” and submit the script to job scheduler.
(v) Check for the files “.haps” and “.sample” in each chromosome directory and check the log files in the directory “shapeit_logfiles_sungrid” to make sure all jobs got successful without any error.
(vi) Remove the config parameter “CHECK_POINT_STOP=SHAPEIT” and select option “RESTART=SHAPEIT” and SHAPEITONLY=YES .
(vii) Resubmit the workflow and you can see “shapeit_jobs.tar.gz” output file in the directory “SHAPEIT_OUTPUT” in the temp directory.

(2) Imputation

(i) Copy the above shapeit tar file “shapeit_jobs.tar.gz” to some direcoty.
(ii) Create the config files ad mentioned in the manual.
(iii) select the config file parameter “HAPS=<PATH TO FILE/ shapeit_jobs.tar.gz”,“CHECK_POINT_STOP=IMPUTE” & “RESTART=NO”.
(iv) Run the workflow and the array job script “ArrayJob_shapeit.csh” in the temp folder will be generated.
(v) Edit the SGE or PBS parameters accordingly in this script “ArrayJob_impute.csh” and submit the script to job scheduler.
(vi) Check the log files in the directory “impute_logfiles_sungrid” to make sure all jobs got successful without any error.

(3) Post Processing

(i) Create the config files as mentioned in the manual.

(ii) select the config file parameter "HAPS=<PATH TO FILE/ shapeit_jobs.tar.gz", & "RESTART=POST". Remove the parameter "CHECK_POINT_STOP=IMPUTE" from the config file.

(iii) Run the workflow and the array job script "ArrayJob_post.csh" in the temp folder will be generated.

(iv) Edit the SGE or PBS parameters accordingly in this script "ArrayJob_post.csh" and submit the script to job scheduler.

(v) Check the log files in the directory "post_logfiles_sungrid" to make sure all jobs got successful without any error.

Appendix I: Single sample Imputation

If the input dataset has only one sample then SHAPEIT2 allows phasing the input data with combined with the reference data. There will be no change in the imputation process. Use the run info config parameter ‘LESS_NUM_SAMP=”NO” ‘to enable this process.

Appendix J: Downloading External Tools (and TOOLINFO FILE)

We provide script to install the required tools and create the tool info file. If you need to install the tools separately (perhaps some of the tools are already installed in your environment), the instructions are below. However, you can also just run the install_tools.sh (this tool only require the path to the ezimputer directory and path to new tool info file).

```
export EZIMPUTER=~/EZIMPUTER
```

Manually downloading tool and tool_info.txt config file.

Download the following the tools, unzip them and place them in a common directory. Create a tool info config file for the programs to execute.

(1) **PLINK(Version v1.07):** Plink is a free, open-source whole genome association analysis toolset. You can download plink from http://pngu.mgh.harvard.edu/~purcell/plink/download.shtml .

(2) **STRUCTURE(Version 2.3.2.1):** The program *structure* is a free software package for using multi-locus genotype data to investigate population structure. You can download structure from http://pritch.bsd.uchicago.edu/structure_software/release_versions/v2.3.4/html/structure.html .

(3) **SHAPEIT(Version V2):** SHAPEIT is a fast and accurate haplotype inference software. You can download SHAPEIT from https://mathgen.stats.ox.ac.uk/genetics_software/shapeit/shapeit.html.

(4) **IMPUTE: IMPUTE (version 2.3)** (also known as **IMPUTE2**) is a genotype imputation and haplotype phasing program. You can download IMPUTE from http://mathgen.stats.ox.ac.uk/impute/impute_v2.html#download .

(5) **CHECK_STRAND_UTILITY(Beagle Utilities):** This is a collection of Beagle utility python scripts to perform strand-checking of input files from different genotyping platforms. You can download from http://faculty.washington.edu/sguy/beagle/strand_switching/strand_switching.html .

(6) **GPROBSMETRICS(Beagle Utilities):** The gprobsmetrics utility calculates per-marker statistics from Beagle format genotype probabilities. You can download from file.**http://faculty.washington.edu/browning/beagle_utilities/utilities.html#gprobsmetrics .**

SAMPLE TOOL INFO CONFIG FILE

Here is the sample tool info config file (see below detailed description) [a copy is included as tool_info.txt in package. and you can use the install_tools.sh script to create a copy with your own path for the tools.]

```
PLINK=/home/EzImputer_Sample_project/EXTERNALTOOLS/PLINK/ plink-1.07-x86_64/plink
STRUCTURE=/home/EzImputer_Sample_project/EXTERNALTOOLS/STRUCTURE/console/structure
STRUCTURE_PARAM=/home/EzImputer_Sample_project/EXTERNALTOOLS/STRUCTURE/console/extraparams
SHAPEIT=/home/EzImputer_Sample_project/EXTERNALTOOLS/SHAPEIT/shapeit
```

```
IMPUTE=/home/EzImputer_Sample_project/EXTERNALTOOLS/IMPUTE/impute_v2.3.0_x86_64_static/impute2
CHECK_STRAND=/home/EzImputer_Sample_project/EXTERNALTOOLS/CHECK_STRAND/check_strands_16May11/check_strands.py
GPROBS=/home/EzImputer_Sample_project/EXTERNALTOOLS/GPROBS/gprobsmetrics.jar
PERL=/usr/local/biotools/perl/5.10.0/bin/perl
PYTHON=/usr/local/biotools/python/2.7/bin/python
JAVA=/usr/java/latest/bin/java
QSUB=/home/sge6_2/bin/lx24-amd64/qsub
SH=/bin/bash
```

DETAILED DESCRIPTION TO DOWNLOAD THE TOOLS

```
export EZIMPUTER= "/home/EzImputer_Sample_project"
#Create main tools directory
mkdir $EZIMPUTER/EXTERNALTOOLS
#Change directory
cd  $EZIMPUTER/EXTERNALTOOLS
```

#PLINK

```
#Create main plink directory

mkdir $EZIMPUTER/EXTERNALTOOLS/PLINK
#Change to plink main directory
cd $EZIMPUTER/EXTERNALTOOLS/PLINK
#Download the plink package
wget http://pngu.mgh.harvard.edu/~purcell/plink/dist/plink-1.07-x86_64.zip
#Uncompresszip files
unzip plink-1.07-x86_64.zip
#The plink binary is now in $EZIMPUTER/EXTERNALTOOLS/PLINK/plink-1.07-x86_64/plink
# when you enter the path in the too info file, you must replace the value of
# $EZIMPUTER with the actual path, e.g.

testp=`$EZIMPUTER/EXTERNALTOOLS/PLINK/plink-1.07-x86_64/plink | grep -c 'Purcell'`

if [ $testp -eq 0 ]

        then
        echo "Error during plink installation"
        tools_installed_ind=0
            tools_installed=$tools_installed" PLINK"
fi
```

#STRUCTURE

```
#Create main STRUCTURE directory
mkdir $EZIMPUTER/EXTERNALTOOLS/STRUCTURE
#Change to STRUCTURE directory
cd $EZIMPUTER/EXTERNALTOOLS/STRUCTURE
#Download the STRUCTURE tool
wget http://pritch.bsd.uchicago.edu/structure_software/release_versions/v2.3.4/release/structure_linux_console.tar.gz
tar -zxvf structure_linux_console.tar.gz
cd console
#test the structure executable with absolute path (change the permission if necessary)
$EZIMPUTER/EXTERNALTOOLS/STRUCTURE/console/structure
```

#Once you run the executable you should be able to see below

```
----------------------------------------------------
STRUCTURE by Pritchard, Stephens and Donnelly (2000)
     and Falush, Stephens and Pritchard (2003)
       Code by Pritchard, Falush and Hubisz
             Version 2.3.4 (Jul 2012)
----------------------------------------------------



Reading file "mainparams".
datafile is
infile
Reading file "extraparams".
Note: RANDOMIZE is set to 1. The random number generator will be initialized using the system clock,
ignoring any specified value of SEED.
Unable to open the file infile.

Exiting the program due to error(s) listed above.
```

#If you the program is giving the error “/lib/libc.so.6: version `GLIBC_2.7' not found”. You need to
#download the source and recompile the program

```
mkdir $EZIMPUTER/EXTERNALTOOLS/STRUCTURE/console/source/
cd $EZIMPUTER/EXTERNALTOOLS/STRUCTURE/console/source/
```

Get the source package

```
wget http://pritch.bsd.uchicago.edu/structure_software/release_versions/v2.3.4/structure_kernel_source.tar.gz
```

#uncompress the package

```
tar -zxvf structure_kernel_source.tar.gz
cd structure_kernel_src/
```

#compile

```
make
```

Test the new executable

```
$EZIMPUTER/EXTERNALTOOLS/STRUCTURE/console/source/structure_kernel_src/structure
```

#SHAPEIT

#Create main SHAPEIT directory

```
mkdir $EZIMPUTER/EXTERNALTOOLS/SHAPEIT
```

#Change to SHAPEIT directory

```
cd $EZIMPUTER/EXTERNALTOOLS/SHAPEIT
```

#Download SHAPEIT tool package

```
https://mathgen.stats.ox.ac.uk/genetics_software/shapeit/shapeit.v2.r790.RHELS_5.4.static.tar.gz
```

#Untar the downloaded package

```
tar -zxvf shapeit.*.tar.gz
```

#Try SHAPEIT

```
$EZIMPUTER/EXTERNALTOOLS/SHAPEIT/shapeit
```

You should be able to see

```
Phaser options:
  --help                      Produce licence message
  --licence                   Produce licence message
  -v [ --version ]            Produce version details
  -L [ --output-log ] arg (=shapeit_05032013_14h55m40s_b4340e74-e467-4ed8-9bc9-4dee02807b9a.log)
                              Log file
  --exclude-snp arg           File containing all the positions of
                              the SNPs to exclude in input files
  --include-snp arg           File containing all the positions of
#…(and many more lines)
```

#IMPUTE

```
#Create main IMPUTE directory
mkdir $EZIMPUTER/EXTERNALTOOLS/IMPUTE
#Change to IMPUTE directory
cd $EZIMPUTER/EXTERNALTOOLS/IMPUTE
#Download IMPUTE tool package
wget http://mathgen.stats.ox.ac.uk/impute/impute_v2.3.1_x86_64_static.tgz
#Untar the downloaded package
tar -zxvf impute_v2.3.1_x86_64_static.tgz
#change directory
cd impute_v2.3.1_x86_64_static
#Try Impute
impute2
#You should be able to see
======================
 IMPUTE version 2.3.1
======================

Copyright 2008 Bryan Howie, Peter Donnelly, and Jonathan Marchini
Please see the LICENCE file included with this program for conditions of use.

The seed for the random number generator is 1997316289.

Command-line input: impute2

ERROR: You must specify a valid interval for imputation using the -int argument.
```

#CHECK_STRAND

```
#Create main CHECK_STRAND directory
mkdir $EZIMPUTER/EXTERNALTOOLS/CHECK_STRAND
#Change to CHECK_STRAND directory
cd $EZIMPUTER/EXTERNALTOOLS/CHECK_STRAND
#Download CHECK_STRAND package
wget
http://faculty.washington.edu/sguy/beagle/strand_switching/check_strands_16May11.tar.gz
#uncompress the package
tar -zxvf check_strands_16May11.tar.gz
#change directory
cd check_strands_16May11
```

```
# test program
python check_strands.py
# You should be able to see
usage: python check_strands.py infileprefixes outfileprefix
outfiles: check_strands.py.markers check_strands.py.log
completed combining marker files
done checking frequencies
```

(1) GPROBS

```
#Create main GPROBS directory
mkdir $EZIMPUTER/EXTERNALTOOLS/GPROBS
#Change to GPROBS directory
cd $EZIMPUTER/EXTERNALTOOLS/GPROBS
#Download GPROBS package
wget http://faculty.washington.edu/browning/beagle_utilities/gprobsmetrics.jar
```

#Once you are done with downloading next step is to create the tool info config file

TOOL Config file parameters (You should specify all options)

PLINK=/PATH_TO/plink [Full Path and filename to plink executable]

STRUCTURE=/PATH_TO/STRUCTURE/structure [Full Path and filename to structure executable]

STRUCTURE_PARAM=/PATH_TO/STRUCTURE/structure.extraparams [Full path and filename to structure param file (you will get when you download structure)]

CHECK_STRAND=/PATH_TO/check_strands_16May11/check_strands.py [Full Path and filename to CHECK_STRANDS PYTHON SCRIPT]

PERL=/PATH_TO/perl [Full Path and filename to PERL]

PYTHON=/PATH_TO/python [Full Path and filename to PYTHON]

SH=/PATH_TO/bash [Full Path and filename to bash, usually /bin/bash]

SH=/PATH_TO/bash[Path to SH/CSH/TSH]

EXAMPLES

The examples below are included as complete scripts in the downloadable packages. You just have to edit the EZIMPUTER variable to the location of your own installation.. and alternatively substitute your own datasets, location for your own temp space, and your own email.

- First step is to install the tools
  # Skip this if you have already installed the tools.. and set TOOLINFO to the path/name of your tool_info.config file
  $EZIMPUTER/install_tools.sh $EZIMPUTER <PATH TO THE TARGET DIR> <Tool info file>

WHOLE GENOME IMPUTATION EXAMPLE (CHR 22)

SAMPLE DATA: You can download build 36 HAPMAP data from
http://hapmap.ncbi.nlm.nih.gov/downloads/genotypes/hapmap3_r3/plink_format/

#Define a directory for EZimputer

```
export EZIMPUTER=”full path of your own directory”
```

# note for non-bash shell, can try set EZIMPUTER=”full path of your own directory”

#Create a project directory

```
mkdir $EZIMPUTER

cd  $EZIMPUTER

wget http://hapmap.ncbi.nlm.nih.gov/downloads/genotypes/hapmap3_r3/plink_format/hapmap3_r3_b36_fwd.consensus.qc.poly.map.gz

wget http://hapmap.ncbi.nlm.nih.gov/downloads/genotypes/hapmap3_r3/plink_format/hapmap3_r3_b36_fwd.consensus.qc.poly.ped.gz
```

#Uncompress(gunzip) them and convert the plink files to plink transpose files (this may take ~1 hour)

```
gunzip hapmap3_*.gz
```

#getting the path to the SH

```
SH=`which sh`
```

#download & prepare external tools

```
PERL=`grep 'PERL=' $TOOLINFO | cut -d '=' -f2`
```

```
#convert the plink file format to transpose

PLINK=`grep 'PLINK='  $TOOLINFO | cut -d '=' -f2 `
```

```
#Extract the top 50 samples

cut -f1-6 -d ' ' $EXAMPLES/hapmap3_r3_b36_fwd.consensus.qc.poly.ped|head -50   > $EXAMPLES/hapmap3_r3_b36_fwd.consensus.qc.poly_50.keep

$PLINK --file $EXAMPLES/hapmap3_r3_b36_fwd.consensus.qc.poly  --keep $EXAMPLES/hapmap3_r3_b36_fwd.consensus.qc.poly_50.keep --chr 22 --make-bed --out $EXAMPLES/hapmap3_r3_b36_fwd.consensus_subset50_chr22.qc.poly
```

# Download and prepare external tools (here)

#Get the imputation reference by executing the script Get_impute_reference.pl.

```
#get imputataion reference

mkdir $EXAMPLES/impute_ref

$PERL  $EZIMPUTER/Get_impute_reference.pl  -OUT_REF_DIR  $EXAMPLES/impute_ref  -DOWNLOAD_LINK   http://mathgen.stats.ox.ac.uk/impute/ALL_1000G_phase1integrated_v3_impute.tgz

#preparing run config file for the Wrapper script

echo "INPUT_PLINK=$EXAMPLES/hapmap3_r3_b36_fwd.consensus_subset50_chr22.qc.poly"  > $EXAMPLES/Wrapper_run_info.config

echo "IMP2_OUT_DIR=$EXAMPLES/WHOLECHR"  >> $EXAMPLES/Wrapper_run_info.config

echo "MODULES_NEEDED=UPGRADE_BUILD,IMPUTE"  >> $EXAMPLES/Wrapper_run_info.config
echo "DBSNP_DOWNLOADLINK= ftp://ftp.ncbi.nlm.nih.gov/snp/organisms/human_9606_b141_GRCh37p13/database/organism_data/b141_SNPChrPosOnRef_GRCh37p13.bcp.gz "  >> $EXAMPLES/Wrapper_run_info.config

echo "DBSNP_DIR=/data2/bsi/RandD/Arraybased_RND/Easy_imputer_test/DBDIR"  >> $EXAMPLES/Wrapper_run_info.config

echo "IMPUTE_REF=$EXAMPLES/impute_ref/ALL_1000G_phase1integrated_v3_impute"  >> $EXAMPLES/Wrapper_run_info.config

echo "IMPUTEREF_VERSION=ALL_1000G_phase1integrated_v3_impute"  >> $EXAMPLES/Wrapper_run_info.config

echo "BEAGLE_REF_DB=/data2/bsi/RandD/Arraybased_RND/Easy_imputer_test/DBDIR/BEAGLE/" >> $EXAMPLES/Wrapper_run_info.config

echo "EMAIL=prodduturi.naresh@mayo.edu"  >> $EXAMPLES/Wrapper_run_info.config

echo "USERNAME=m081429"  >> $EXAMPLES/Wrapper_run_info.config

echo "DEAL_AMBIGUOUS=YES"  >> $EXAMPLES/Wrapper_run_info.config

echo "ENVR=MANUAL"  >> $EXAMPLES/Wrapper_run_info.config
```

HOW TO EXECUTE

```
perl  $EZIMPUTER/Wrapper.pl  -wrapper_config $EXAMPLES/Wrapper_run_info.config -tool_config $TOOLINFO
```

OUTPUTFILES

Please check temp folder ($EXAMPLES /WHOLECHR/Impute_tmp/impute/for the results once the job gets completed.

Out file description can be found here.

SAMPLE REGION IMPUTATION EXAMPLE

To run this example we need files from the above WHOLE CHROMOSOME example, So run that example first.

```
FILE=$EXAMPLES/WHOLECHR/hapmap3_r3_b36_fwd.consensus_subset50_chr22.qc.poly_PREPDATA.newbuild.tped

#TOOL INFO CONFIG FILE (tool_info.config)

#should have been created when installing ezimputer.

TOOLINFO=$EXAMPLES/tool_info.config

cd $EXAMPLES/WHOLECHR/

cp hapmap3_r3_b36_fwd.consensus_subset50_chr22.qc.poly_PREPDATA.tfam hapmap3_r3_b36_fwd.consensus_subset50_chr22.qc.poly_PREPDATA.newbuild.tfam

#extracting smalle region from whole genome

# Before running this script, you must run the whole genome workflow example up to the QC Step to generate the fwdStrandResults_input file

#convert the plink file format to transpose

PLINK=`grep 'PLINK='  $TOOLINFO | cut -d '=' -f2 `

#extracting a small region from the whole chromosome file

$PLINK --tfile hapmap3_r3_b36_fwd.consensus_subset50_chr22.qc.poly_PREPDATA.newbuild --chr 22 --from-kb 43500 --to-kb 46000 --make-bed --out small_region_fwdStrandResults_input

#preparing run config file

echo "INPUT_PLINK=$EXAMPLES/WHOLECHR/small_region_fwdStrandResults_input"  > $EXAMPLES/small_region_Wrapper_run_info.config

echo "IMP2_OUT_DIR=$EXAMPLES/SMALL_REGION"  >> $EXAMPLES/small_region_Wrapper_run_info.config

echo "MODULES_NEEDED=IMPUTE"  >> $EXAMPLES/small_region_Wrapper_run_info.config

echo "IMPUTE_REF=$EXAMPLES/impute_ref/ALL_1000G_phase1integrated_v3_impute"  >> $EXAMPLES/small_region_Wrapper_run_info.config
echo "IMPUTEREF_VERSION=ALL_1000G_phase1integrated_v3_impute"  >>
$EXAMPLES/small_region_Wrapper_run_info.config

echo "EMAIL=prodduturi.naresh@mayo.edu"  >> $EXAMPLES/small_region_Wrapper_run_info.config

echo "USERNAME=m081429"  >> $EXAMPLES/small_region_Wrapper_run_info.config

echo "DEAL_AMBIGUOUS=YES"  >> $EXAMPLES/small_region_Wrapper_run_info.config

echo "ENVR=MANUAL"  >> $EXAMPLES/small_region_Wrapper_run_info.config

echo "CHR_START_INPUT=YES"  >> $EXAMPLES/small_region_Wrapper_run_info.config

echo "SMALL_REGION_EXTN_START=2000000"  >> $EXAMPLES/small_region_Wrapper_run_info.config

echo "SMALL_REGION_EXTN_STOP=2000000"  >> $EXAMPLES/small_region_Wrapper_run_info.config
```

HOW TO EXECUTE

```
perl  $EZIMPUTER/Wrapper.pl  -wrapper_config
$EXAMPLES/small_region_Wrapper_run_info.config -tool_config $TOOLINFO
```

OUTPUTFILES

Please check temp folder ($EXAMPLES/SMALL_REGION/Impute_tmp/impute)for the results once the job gets completed. As the data contains only of chr 2 so all the impute dosage files can be found in the folder ‘2’.

Out file description can be found here.

SINGLE SAMPLE IMPUTATION

This example describes how to do the single sample imputation.

```
#First you need to run the WHOLE_GENOME_IMPUTATION.sh to generate the WHOLE_GENOME PLINK dataset

FILE=$EXAMPLES/WHOLECHR/hapmap3_r3_b36_fwd.consensus_subset50_chr22.qc.poly_PREPDATA.newbuild.tped


#TOOL INFO CONFIG FILE (tool_info.config)

#Should have been created when installing ezimputer.
TOOLINFO=$EXAMPLES/tool_info.config

#Extract the top sample

cd $EXAMPLES/WHOLECHR/

cp hapmap3_r3_b36_fwd.consensus_subset50_chr22.qc.poly_PREPDATA.tfam hapmap3_r3_b36_fwd.consensus_subset50_chr22.qc.poly_PREPDATA.newbuild.tfam

head -1 hapmap3_r3_b36_fwd.consensus_subset50_chr22.qc.poly_PREPDATA.newbuild.tfam > single_samp.txt

#create the plink dataset

PLINK=`grep 'PLINK='  $TOOLINFO | cut -d '=' -f2 `


$PLINK --tfile hapmap3_r3_b36_fwd.consensus_subset50_chr22.qc.poly_PREPDATA.newbuild --keep $EXAMPLES/WHOLECHR/single_samp.txt --make-bed --out single_sample_fwdStrandResults_input

#preparing run config file

echo "INPUT_PLINK=$EXAMPLES/WHOLECHR/single_sample_fwdStrandResults_input"  > $EXAMPLES/single_sample_Wrapper_run_info.config

echo "IMP2_OUT_DIR=$EXAMPLES/SINGLE_SAMPLE"  >> $EXAMPLES/single_sample_Wrapper_run_info.config

echo "MODULES_NEEDED=IMPUTE"  >> $EXAMPLES/single_sample_Wrapper_run_info.config
echo "IMPUTE_REF=$EXAMPLES/impute_ref/ALL_1000G_phase1integrated_v3_impute"  >> $EXAMPLES/single_sample_Wrapper_run_info.config
```

```
echo "IMPUTEREF_VERSION=ALL_1000G_phase1integrated_v3_impute"  >> $EXAMPLES/single_sample_Wrapper_run_info.config
```

```
echo "EMAIL=prodduturi.naresh@mayo.edu"  >> $EXAMPLES/single_sample_Wrapper_run_info.config
```

```
echo "USERNAME=m081429"  >> $EXAMPLES/single_sample_Wrapper_run_info.config
```

```
echo "DEAL_AMBIGUOUS=YES"  >> $EXAMPLES/single_sample_Wrapper_run_info.config
```

```
echo "ENVR=MANUAL"  >> $EXAMPLES/single_sample_Wrapper_run_info.config
```

```
echo 'LESS_NUM_SAMP="YES"'  >> $EXAMPLES/single_sample_Wrapper_run_info.config
```

HOW TO EXECUTE

```
perl  $EZIMPUTER/Wrapper.pl  -wrapper_config $EXAMPLES/single_sample_Wrapper_run_info.config -tool_config $TOOLINFO
```

OUTPUTFILES

Please check temp folder ($EXAMPLES/SINGLE_SAMPLE/Impute_tmp/impute)for the results once the job gets completed. As the data contains only of chr 2 so all the impute dosage files can be found in the folder ‘2’.

Out file description can be found here.

RUN ALL EXAMPLES

This script executes all the installation and run the all the examples

usage:sh RUN_ALL_EXAMPLES.csh <path_to_ezimputer_install> <path_to_ezimputer_directory_example_scripts> <path to the tool info file>

First requirement is to SHELL environment to execute shell script

SH=`which sh`

#running the WHOLECHROMSOME EXAMPLE

echo "running the whole chromosome example in the directory $EZIMPUTER/test "

$SH $EXAMPLES_SCRIPTS_DIR/WHOLE_GENOME_CHROMOSOME_IMPUTATION_SGE_WRAPPER.sh $EZIMPUTER $EZIMPUTER/test $TOOLINFO

#running the SMALL REGION IMPUTATION EXAMPLE

echo "running the small region example in the directory $EZIMPUTER/test "

$SH $EXAMPLES_SCRIPTS_DIR/SMALL_REGION_IMPUTATION_SGE_WRAPPER.sh $EZIMPUTER $EZIMPUTER/test $TOOLINFO

#running the SINGLE SAMPLE EXAMPLE

echo "running the single sample example in the directory $EZIMPUTER/test "

$SH $EXAMPLES_SCRIPTS_DIR/SINGLE_SAMPLE_IMPUTATION_SGE_WRAPPER.sh $EZIMPUTER $EZIMPUTER/test $TOOLINFO

TWO PLATFORM DATA

If your data is made of two (or more) different platform (for Example: AFFY & ILLUMINA), you can impute from the merged datasets, but

(i) This will only work if imputing a small region (a few 100 SNPs) otherwise the phasing step will lead inaccurate results.

(ii) but you must QC and forward map the alleles separately first. Follow the below steps

a. Separate the dataset (plink transpose files) in to individual platform specific datasets.

b. If the datasets are on build 36 you can convert them to build 37 using the script Upgrade_inputmarkers_to_build37_by_DBSNP*.

c. Run the QC step separately on each platform specific data (QC_fwd_structure*).

d. Combine the datasets. If you have same samples in two different platform you can keep the samples with high quality using the sample qc file generated by the QC step. (i.e sample with low missing values)

e. Run the imputation on the combined dataset (Phase_Impute_by_parallel_proc).

Citations:

If you like this package, the manuscript has been submitted.

EZimputer: A workflow for parallelized genome wide imputation.

Hugues Sicotte,Naresh Prodduturi, Julie A. Johnson,Yaxiong Lin, Paul A. Decker,Jeannette Eckel-Passow, Martha E. Matsumoto, Robert B. Jenkins, Shannon K. McDonnell, Mariza de Andrade, and Jean-Pierre A. Kocher

https://code.google.com/p/ezimputer/