IMPUTE 5

v1.1.5 - May 30th, 2021

Introduction

IMPUTE 5 is a genotype imputation method that can scale to reference panels with millions of samples. This method continues to refine the observation made in the IMPUTE2 method, that accuracy is optimized via use of a custom subset of haplotypes when imputing each individual. It achieves fast, accurate, and memory-efficient imputation by selecting haplotypes using the Positional Burrows Wheeler Transform (PBWT). By using the PBWT data structure at genotyped markers, IMPUTE 5 identifies locally best matching haplotypes and long identical by state segments. The method then uses the selected haplotypes as conditioning states within the IMPUTE model.

Imp5Chunker is a program to create imputation chunks from the target and reference panel.

imp5Converter is a program to convert VCF/BCF reference panels in imp5 file format. imp5 is a file format used by IMPUTE 5 to store reference panels and allows fast read of custom regions, without the need to use compression libraries like ZLIB, and faster imputation.

Citation:

If you use IMPUTE 5 in your research, please cite the following publication:

Rubinacci S, Delaneau O, Marchini J (2020) Genotype imputation using the Positional Burrows Wheeler Transform. PLOS Genetics 16(11): e1009049

Documentation

Example files to test IMPUTE 5 and imp5Converter can be found in the *test* directory. Examples shown below assume impute5 binary is called from the appropriate directory, when not explicitly noted.

1. Overview

1.1. Running software of the IMPUTE5 suite

IMPUTE5 is available as a static binary compiled in Ubuntu x64 system. To run the program, simply run:

```
./impute5_v1.1.5_static --help
```

Please note that the name might change between versions. Please use the latest version of the software as it contains the latest bug fixes.

In the case the binary does not work on your operating system, you can use docker to run IMPUTE5. For example you can run:

```
#pull an ubuntu image
[sudo] docker pull ubuntu
#run the docker
[sudo] docker run -it --rm --name impute5 -v <impute_folder>:/home/impute5 ubuntu
```

From the resulting terming (indicated as "/ #") you can run IMPUTE5 by simply using:

```
home/impute5/impute5_v1.1.5_static
```

And run the test example by using:

```
home/impute5/impute5_v1.1.5_static --h /home/impute5/test/reference.bcf --g /home/impute5/test/target.bcf --m /home/impute5/test/chr20.b37.gmap.gz --o home/impute5/test/imputed.bcf --r 20:1000000-4000000
```

1.2. Imputation workflow

IMPUTE5 does not perform automatic chunking and ligation. The reason is to allow each job to run completely independently on large clusters. The typical IMPUTE5 workflow is composed by considering each chromosome independently. The following are the steps of a typical IMPUTE5 run:

1. **Chunking step**: each chromosome is divided into a small set of chunks (overlapping in the buffer regions). (Section 2)

2. *Optional*: **convert the reference panel in IMP5** file format (section 3)

3. **Imputation** of each chunk of data: running IMPUTE5 to impute each chunk of data independently (section 4)

4. **Ligation**: ligate each chunk of imputed data together in order to create one file per chromosome. To perform this, it is possible to use *bcftools concat* program. (Section 5)

2. Chunking step

2.1. Simple run

Imp5Chunker is a software that takes two datasets (target and reference panel) to create a file containing the regions to be used for imputation.

Three main parameters are needed to run imp5Chunker:

```
imp5Chunker --h reference.vcf.gz --g target.vcf.gz --r 20 --o coordinates.txt
```

Where --h defines the reference panel (does not have to contain the GT field), --g defines the target and --r region of interested (usually a full chromosome) and --o the output file (a text file).

The output has the following fields:

```
Chunk ID / chromosome ID / Buffered region / Imputation region /
Length / Number of target markers / Number of reference markers
```

Other parameters define the minumum chunk and buffer size and counts. The parameter `--max-window-count` forces the minumum number of genotype markers to be fixed.

2.2. Option summary

The full list of options can be obtained by running the command:

```
imp5Chunker --help
```

This should output this list of options:

Input

Option		Default value	Description
--h	STRING	-	Haplotype reference panel in VCF/BCF format (must have .vcf[.gz]/.bcf/imp5 extension). The file must be indexed (tabix/imp5 index). For efficiency reasons, better if only positions are defined (no GT field)
--g	STRING	-	File containing target haplotypes for a study cohort that you want to impute in

			VCF/BCF format (must have .vcf[.gz]/.bcf extension). The file must be indexed (tabix index).
--r	STRING	-	Region containing the whole region (typically a whole chromosome). Example -r 20 (whole chromosome 20).

Parameters

Option	Type	Default value	Description
--window-size	INT	5000000	Minimum Window size in bp
--buffer-size	INT	250000	Minimum buffer size in bp
--window-count	INT	-1	Minimal number of genotyped markers in the chunk (shared with the reference panel). Default means that the parameter is set to the expected number of chip markers in a chunk (#variants_shared / (length_region_shared / window_size))
--buffer-count	INT	-1	Minimal number of genotyped markers in the chunk (shared with the reference panel). Default means that the parameter is set to the expected number of chip markers in a buffer region (#variants_shared / (length_region_shared / buffer_size))
--max-window-count	INT	-1	Minimal number of genotyped markers in the chunk (shared with the reference panel). Default is set to 2*window-count

Output

Option	Type	Default value	Description
--o	STRING	-	Specifies output file name.

Other parameters

Option	Type	Default value	Description
--help	NA	-	Produces help message, listing all the accepted arguments
--l	STRING	-	Location of the log file to be written. If not specified, only console output will be generated.

2.3. Alternative: bcftools scatter

The Mocha imputation pipeline (https://github.com/freeseek/mocha/blob/master/wdl/impute.wdl) adopts an alternative to perform the chunking step by using the bcftools scatter plugin.
More information is in the wdl file provided by the Mocha pipeline (https://github.com/freeseek/mocha/blob/master/wdl/impute.wdl), more details about the bcftools scatter can be found by running:

```
bcftools +scatter
```

Running bcftools plugins requires BCFTOOLS_PLUGINS variable to be in PATH. More details here: `https://samtools.github.io/bcftools/howtos/plugins.html`

3. Reference panel format: imp5Converter

imp5Converter converts a reference panel in VCF/BCF format to the imp5 file format. An imp5 file contains a region within a chromosome. Typically we want to create a IMP5 file for each chromosome in order to perform imputation on different chunks of the chromosome. An imp5 file is also complemented by an index (.imp5.idx), that allows IMPUTE 5 to have random access to the region of the chromosome.

IMPUTE5 greatly benefits of this file format to speed-up the process of imputation, by performing lazy imputation when the variant is very rare and no computation would be needed (monomorphic variants).

New versions of the format are updated with the versioning of IMPUTE5: in the case an old imp5 cannot be read by a newer version of IMPUTE5, please create a new imp5 file using the latest imp5Converter version.

3.1. Simple run

To convert the full reference panel chromosome 20 to an imp5 file you can simply use:

```
imp5Converter --h reference.vcf.gz --r 20 --o reference.imp5
```

The output is a file named reference.imp5 and an index file named reference.imp5.idx. We can now call IMPUTE 5 and pass the imp5 file as a reference panel (IMPUTE 5’s --h option).

Similarly a file can be converted from imp5 to VCF/BCF format using:

```
imp5Converter --h reference.imp5 --r 20 --o reference.vcf.gz
```

3.2. Option summary

The full list of options can be obtained by running the command:

```
imp5Converter --help
```

This should output this list of options:

Input

Option		Default value	Description
--h	STRING	-	Haplotype reference panel in VCF/BCF format (must have .vcf[.gz]/.bcf/imp5 extension). The file

			must be indexed (tabix/imp5 index).
--r	STRING	-	Region containing the whole imputation region (typically a whole chromosome). Example -r 20 (whole chromosome 20).

Output

Option	Type	Default value	Description
--o	STRING	out.ref.imp5	Specifies output file name (must have .vcf[.gz]/.bcf/imp5 extension).

Other parameters

Option	Type	Default value	Description
--help	NA	-	Produces help message, listing all the accepted arguments
--threads	INT	1	Number of threads used for decompression of the input file

4. IMPUTE5

4.1. Simple run

To run IMPUTE5 with default parameters, use the following command line:

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --o imputed.vcf.gz
```

All five options are mandatory and their descriptions are:

- `--h` specifies the haplotype reference panel in VCF/BCF/IMP5 format (must have .vcf[.gz]/.bcf/.imp5 extension). The file must be indexed (tabix/imp5 index). The reference panel should be phased and non-missing at every position.
- `--m` specifies the fine-scale recombination map for the region to be analysed. Maps for humans can be found HERE. In the case this parameter is not defined, a constant recombination rate is assumed.
- `--g` specifies the file containing target haplotypes for a study cohort that you want to impute in VCF/BCF format (must have .vcf[.gz]/.bcf extension). The file must be indexed (tabix index). The target dataset should be phased and non-missing in the set of markers specified. Markers that are only present in the reference panel and not in the target set, are imputed.
- `--r` specifies the target region or chromosome to be imputed . Buffer parameters will expand this region, if specified.
- `--o` specifies the output filename. A proper extension is mandatory.

IMPUTE5 considers as genotype markers the markers in the intersection between `--g` and `--h`. In practice, it considers as genotype markers only the variants with chromosome ID, position, REF and ALT alleles that perfectly match between the two panels. Markers only in the reference panel are considered imputed markers and markers present only in the target panel are simply reported in output.

4.2. Log file

To record all the verbose that appear on the screen, use the `--l` option as follows:

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --o imputed.vcf.gz --l imputed.log
```

The use of this parameter is strongly recommended.

4.3. Imputing a chunk of data

To impute the 5Mb region located in the genomic interval 2Mb-7Mb, use:

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --o imputed.vcf.gz
```

`--r` option is mandatory. Double check that the chromosome ID matches one of those specified in the VCF file. A common mistake is to use other specifications for the chromosome different from the one specified in the VCF/BCF file. A quick way to check it would be running:

```
bcftools view -H -G target.vcf.gz | head -n 1 | awk  '{print $1}'
```

Also, please verify that your reference and target panel present the same notation for the chromosome.

The definition of the imputation regions depends of the dataset. Typically, imputation regions ~5 Mb should be enough for the majority of applications. Increasing the imputation region could bring additional benefits at rare variants, at the cost of an increased running time.

Each chunk of imputed data is expanded by a buffer region, in order to help preventing imputation quality from deteriorating near the edges of the region. Markers in the buffer region will help the inference but do not appear in the output files. Larger buffers can improve accuracy. Value of the buffer regions can be expressed in two ways:

- using the **--b** option, defining a fixed buffer in kb

  ```
  impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz
  --r 20:2000000-7000000 --o imputed.vcf.gz --b 500
  ```

- using the **--buffer-region**, defining a region to be used as buffer, that expands the region defined with the --r parameter:

  ```
  impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz
  --r 20:2000000-7000000 --o imputed.vcf.gz --buffer-region
  20:1500000-7500000
  ```

When no buffer option is defined, IMPUTE5 assumes **--b 250** as default. When both **--b** and **--buffer-region** are specified in the same command line, **--buffer-region** is used, and **--b** discarded.

4.4. Output file format

IMPUTE5 file automatically detects the format of the input and output file by the extension. Input can be specified in three different file format: VCF[.gz]/BCF/IMP5. Output can be specified in three file formats: .vcf[.gz]/.bcf/.bgen.

4.5.1 BGEN output

You can choose the compression used by BGEN for the output file format using --bgen-compr parameter (values accepted: no,zlib,zstd).

For example, to output a BGEN file compressed using ZSTD you will use:

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --bgen-compr zstd --o imputed.bgen
```

to output a BGEN file compressed using ZLIB you will use:

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --bgen-compr zlib --o imputed.bgen
```

to output a BGEN file compressed with no compression you will use:

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --bgen-compr no --o imputed.bgen
```

From IMPUTE5 v1.1.5 it is now possible to output a VCF file containing only phased haplotypes (FORMAT/GT field), together with bgen file format. The name of the VCF gzipped file shares the prefix of the output bgen file, but instead of bgen will have ".phased.vcf.gz". To enable the option it is necessary the output is in bgen file format.

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --o imputed.bgen --bgen-vcf
```

The previous option will produce both a bgen file containing genotype posteriors and a file named "imputed.phased.vcf.gz" containing phased haplotypes.

4.5.2 VCF/BCF output

VCF and BCF file format contains phased genotypes in the GT field. At imputed markers the VCF/INFO field will contain:

- IMP flag, denoting that the marker is imputed;
- INFO field: containing the IMPUTE INFO score at the variant
- AF field: containing the estimated allele frequency

The VCF/FORMAT by default has:

- GT, containing the most likely genotype;
- DS, containing the output genotype dosage;

The option `--out-gp-field` can be used to output the genotype probabilities in the GP field, while `--out-ap-field` can be used to output ALT haplotype probabilities in the FORMAT/AP1 and FORMAT/AP2 fields.

To output a file vcf.gz file, use the following option:

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r 20:2000000-7000000 --o imputed.vcf.gz
```

The option `--out-gp-field` can be used in combination with VCF/BCF output to add GP (genotype probability) format in the VCF INFO field of imputed markers.

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r 20:2000000-7000000 --out-gp-field --o imputed.vcf.gz
```

4.6. Parallelization

4.6.1 Parallelization by chunk

IMPUTE5 parallelise by chunks so that different imputation regions can be imputed at the same time on a different process.

To do this, you just need to run a IMPUTE5 job per imputation region, by exploiting the `--r` parameter, for example running the following tow commands in parallel:

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r 20:2000001-7000000 --o imputed.00.vcf.gz
```

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:7000001-12000000 --o imputed.01.vcf.gz
```

4.6.2 Multi-threaded parallelization

A single chunk can also be multi-threaded. Multi-threading is only performed on parts of the algorithm (e.g. HMM calculations), therefore is not as efficient as parallelization by chunk.

Multi-threaded parallelization imputes in parallel several individuals, therefore is useful is the number of target samples is large.

To run impute5 on a chunk in parallel, run:

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --o imputed.vcf.gz --threads 4
```

This will run imputation for the chunk using four threads.

4.7. Tuning the PBWT based selection

Reducing the number of conditioning neighbours in the PBWT can be achieved using the `--pbwt-depth` option (called L in the paper). The default value is 4.

Decreasing it results in faster runs at the cost of some accuracy.

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --o imputed.vcf.gz --pbwt-depth 8
```

To change how frequently the PBWT selection is performed you can use the option `--pbwt-cm`. The default value is 0.02 (cM), meaning that the selection is performed once every 0.02 cM. Decreasing this value will result in more states selected and a possible increase in accuracy. However, the following steps will require more time and memory.

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --o imputed.vcf.gz --pbwt-cm 0.005
```

You can also change the selection algorithm by using –neigh-select, turning on the neighbour select algorithm:

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --o imputed.vcf.gz –neigh-select
```

4.8 Chromosome X imputation

Since IMPUTE5 v1.1.4 it is possible to perform haploid imputation. For chromosome X imputation in the non-PAR region, it is required to split the target samples by sex, as females are diploids and males are haploid. For females standard diploid imputation is performed. For males, haploid imputation must be perfomed, therefore it is possible to inform IMPUTE5 using the --haploid option:

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.haploids.vcf.gz --r 20:2000000-7000000 --o imputed.haploids.vcf.gz --haploid
```

Please note that the imp5 file format can only handle diploid reference panels, therefore for chromosome X it is required to use VCF/BCF file format. IMPUTE will infer the ploidy of the reference samples automatically, from the first marker of the reference panel in the region.

Additionally, as in other versions of IMPUTE, when the chromosome name of the imputation region is "X" or "chrX", the scaling parameter of the genetic map is scaled by ¾.

4.9. Other options

The parameter **--ohapcopy** outputs a CSV file containing the expected amount of sequence (in cM) copied from each reference haplotype in the list of copying states of the target haplotype. The output is in CSV file format, tab separated (*gzipped* – can be read using *zcat*).

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r 20:2000000-7000000 --o imputed.vcf.gz --ohapcopy
```

The first three columns of the output CSV file indicate (i) the ID of the reference haplotype, from 0 to N-1 (ii) the name of the reference sample (iii) a value indicating which of the two reference haplotypes. Other columns of the file output the value of the hapcopy (in cM) for each target haplotype. The file has a header, describing each column of the CSV file and the target samples/haplotype.

As the parameter **--ohapcopy** can be used to identify shared segments, it might be appropriate to use a dataset sharing the same samples as target and reference panel in specific situations. For this reason it is possible to ban from the copy list the reference haplotypes sharing the same sample ID (string) to the target, using the option –**ban-repeated-sample-names**.

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r 20:2000000-7000000 --o imputed.vcf.gz --ohapcopy –ban-repeated-sample-names
```

One of the parameters in the IMPUTE model is *ne*, the effective population size. To change the effective population size value you can use **`--ne`** option.

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --o imputed.vcf.gz --ohapcopy --ne 11000
```

The parameter **`--nothreshold`** allows to skip the thresholding step at the end of the HMM, and keep all the states for imputation. If this parameter is specified imputation might be more accurate, however, the Li and Stephens state probabilities are typically very sparse and this option might make imputation a lot slower and much more expensive in RAM.

```
impute5 --h reference.bcf --m chr20.b37.gmap.gz --g target.vcf.gz --r
20:2000000-7000000 --o imputed.vcf.gz --ohapcopy --no-threshold
```

4.10. Option summary

The full list of options can be obtained by running the command:

```
impute5 --help
```

This should output this list of options:

Input

Option		Default value	Description
--h	STRING	-	Haplotype reference panel in VCF/BCF/IMP5 format (must have .vcf[.gz]/.bcf/.imp5 extension). The file must be indexed (tabix/imp5 index).
--m	STRING	-	Fine-scale recombination map for the region to be analyzed. If not specified, a constant recombination rate of 1cM per Mb is used.
--g	STRING	-	File containing target haplotypes for a

			study cohort that you want to impute in VCF/BCF format (must have .vcf[.gz]/.bcf extension). The file must be indexed (tabix index).
--r	STRING	-	Region to be imputed (replaces IMPUTE4's -int parameter). Example -r 20:1000000-5000000 (region within chromosome 20). Buffer parameters will expand this region, if specified.
--buffer-region	STRING	-	Length of buffer region (in kb) to include on each side of the analysis window specified by the -r option. Variants in the buffer regions inform the inference but do not appear in output files. If both the buffer options are used (--b and --buffer-region) only --buffer-region is used.
--b	INT	250	Length of buffer region (in kb) to include on each side of the analysis window specified by the -r option. Variants in the buffer regions inform the inference but do not appear in output files

State selection

Option	Type	Default value	Description
--pbwt-depth	INT	4	Depth of PBWT indexes to condition on
--pbwt-cm	FLOAT	0.02	Distance in cM where the selection is performed
--ohapcopy	STRING	hapcopy.list	Output a file for each target haplotype.

			Each file contains the expected amount of sequence (in cM) copied from each reference haplotype in the list of copying states. The output is in CSV file format.
--ban-hapid	NA	NA	Ban reference haplotypes having the same name as the target from the state selection. with the same sample ID from the state selection. To be used when the target and reference panel share (even partially) the same set of individuals.
--neigh-select	NA	NA	Use neighbour selection algorithm

Output

Option	Type	Default value	Description
--l	STRING	-	Location of the log file to be written. If not specified, only console output will be generated.
--o	STRING	impute5.out.bgen	Specifies output file name.
--bgen-compr	STRING	zstd	Specifies the compression of the output file for BGEN file format (to be used with --o *.bgen). Accepted values: [no, zlib, zstd]
--bgen-bits	INT	8	Specifies the number of bits to be used for the encoding probabilites of the output BGEN file (to be used with --o *.bgen). Accepted values: 0< x <=32.
--bgen-vcf	NA	-	Specifies to output a vcf containing haplotype data other in addition to bgen out-

			put. The file has the same prefix as the output file, and will have ".phased.vcf.gz" as suffix.
--out-gp-field	NA	-	Print FORMAT/GP field (Genotype probabilities) if output is in VCF/BCF format.
--out-ap-field	NA	-	Print FORMAT/AP field (ALT haplotype probabilities) if output is in VCF/BCF format.

Other parameters

Option	Type	Default value	Description
--help	NA	-	Produces help message, listing all the accepted arguments
--threads	INT	1	Number of threads
--no-threshold	NA	-	Specifies if need to use all forward backward states and not apply a threshold
--ne	FLOAT	20000	Effective population size.

5. Ligation step

5.1 Simple run

The simplest way to ligate imputed chunks back is using bcftools concat providing the list of files in the right order:

```
bcftools concat -n -f list.txt -Ob -o ligated.bcf
```

More details about bcftools concat can be found in the here: http://samtools.github.io/bcftools/bcftools.html#concat

In the case your output is in BGEN file format, you can use the cat-bgen format: https://enkre.net/cgi-bin/code/bgen/wiki/cat-bgen

Contact

Please join the OXSTATGEN mailing list and then post any questions there

https://www.jiscmail.ac.uk/cgi-bin/webadmin?A0=OXSTATGEN