How to Select the Number of Clusters for fastPHASE

Gennady Khvorykh, inzilico.com

2018-03-10

Motivation

Assume you have a Plink files and decided to impute the missing genotypes with fastPHASE 1.4.8 tool. The number of haplotype clusters (K) is required as an argument. One option is to leave the default value of 15. For the most cases it will probably work. But what if you have several populations in one file? Say, you search for the fingerprints of selection with hapFLK tool which exploits the fastPHASE algorithm.

How many clusters to choose for 4, 6, 8 or even more populations in a pool? Tests revealed that K influences the results of hapFLK output. If fastPHASE had an option to estimate the error of imputation, the life would be easier. We could apply the mask analysis.

The idea of the analysis is as follow. Hide some genotypes, impute them, then count the error of imputation. Repeat the trick several times with different K and choose that one corresponding to the minimal error of imputation.

Below is the implementation of the masked analysis on the base of imputeqc R package. Let's list 4 main steps and then consider them a little bit closer.

Workflow to select the best K

1. Convert Plink files to fastPHASE *.inp files, using Plink tool.
2. Generate a few test files with *make_test_files.R* which is enclosed to the package. I think five test files are enough to start with.
3. Impute the missing genotypes in each test file with fastPHASE. Apply different K for each set of files.
4. Estimate the imputation quality with *EstimateQuality()* function and chose the K that minimizes the imputation error.

How to convert Plink files to fastPHASE

Say, you have *chr1.{ped, map}* files in the current directory. They contain the genotypes of individuals from one or several populations.

Check that alleles are coded as letters or numbers. The missing ones should be *?*. If your data look differently, send me a chunk of them. I'll figure out what to do next.

To convert Plink files to fastPHASE, run from command line

```
plink --file chr1 --recode fastphase-1chr --out chr1
```

It will create *chr1.recode.phase.inp* ready for further manipulations.

The usage of make_test_files.R

1. Check that you have *imputeqc* R package installed. If not, install it from GitHub. Run in R

```
install.packages("devtools")
devtools::install_github("inzilico/imputeqc", build_vignettes = TRUE)
```

If you already have *devtools* package installed, skip the first command.

2. Get the path to *make_test_files.R* which comes with *imputeqc*. Run in R

```
system.file("extdata", package="imputeqc")
```

Save the returned path in the environment variable in BASH:

```
$ export IMPQC="/path/to/imputeqc/extdata"
```

3. Run *make_test_files.R* to generate *n* test files of fastPHASE format (*.inp), each having *p* proportion of genotypes randomly masked.

Let *chr1.inp* to contain the genotypes of a population from the chromosome one.

The following command will result in 5 test files with 10% percent of genotypes missed.

```
Rscript $IMPQC/make_test_files.R chr1.inp
```

The default parameters (-n 5 -p 0.1 -o test/test) are applied. Five test files named *test.m{1:5}.inp* are saved in */test* subdirectory that script created.

The following command will produce 3 test files with 5% of genotypes masked. The test files *chr1.m{1:3}.inp* are saved in */masked* subdirectory.

```
Rscript $IMPQC/make_test_files.R -n 3 -p 0.05 -o masked/chr1 chr1.inp
```

If there are missing genotypes before running the script, the actual proportion of missing genotypes will be higher, since mask adds missing genotypes to those that exist in original data set. You will see the total proportion of missing genotypes on screen.

The usage of fastPHASE tool

According to fastPHASE manual, we can adjust the following option arguments:

- -T, the number of seeds to launch EM cycles
- -C, the number of EM cycles
- -K, the number of haplotype clusters

There are some flags I advice to apply:

- -H-1, to turn off the phasing, since we need only imputation
- -n, to tell that we have simplified input, since *make_test_files.R* produces that sort of files
- -Z, to simplify the format of output files

I recommend to save the results of imputation in different folders for different K (*/k10*, */k15*, etc.)

An example of command for imputation with K = 10:

```
for i in $(seq 1 5); do fastPHASE -T10 -C25 -K10 -H-1 -n -Z -ok10/chr1.m$i masked/chr1.m$i.inp; done
```

We assume 5 test files (*chr1.m{1:5}.inp*) being in folder */masked*.

The code will produce 5 *chr1.m{1:5}_genotypes.out* files, where *chr1* is your identifier, *m{1:5}* is added by the above command, and *_genotypes.out* is given by fastPHASE. The imputed data sets are saved in */k10* subdirectory.

The above code helps us to agree input/output from different stages of workflow.

Be aware this step takes a lot of computational time depending on the number of populations under consideration, the number markers, and, of course, the K.

Upon accomplishing, we can estimate the quality of imputation.

The usage of EstimateQuality()

There are several metrics to estimate the imputation quality of genotypes (Chan et al, 2016). So far I applied the proportion of correctly imputed genotypes. To compute it, use *EstimateQuality()* function. It returns a data frame with three columns: “alleles”, “genotypes”, and “K”. The first two contains the errors, the third one - K.

The error is counted as 1 - accuracy, where accuracy is a proportion of correctly imputed genotypes or alleles. By correctly imputed genotype we mean that both alleles coincided with the original ones. The function returns the values for one set of test files.

```
# Count errors for one set of test files
errors <- EstimateQuality(origin = "chr1.inp",
                          mask = "masks.RDS",
                          imputed = imputed,
                          K = 15)
```

Here we assume that *chr1.inp* has original genotypes, *masks.RDS* contains the list of all masks (matrices) generated above by *make_test_files.R*, and *imputed* is a vector with the full paths to **_genotypes.out* produced by fastPHASE. The order of files in *imputed* is the same as the order of masks applied upstream. All test files in a set were treated with fastPHASE applying K = 15.

References

1. imputeqc R package github
2. Scheet P, Stephens M. A Fast and Flexible Statistical Model for Large-Scale Population Genotype Data: Applications to Inferring Missing Genotypes and Haplotypic Phase. American Journal of Human Genetics. 2006;78(4):629-644. PubMed
3. fastPHASE 1.4.8. link
4. fastPHASE 1.4 manual. download
5. hapFLK software. link
6. hapFLK tutorial. link
7. Plink tool. link
8. Chan AW, Hamblin MT, Jannink J-L. Evaluating Imputation Algorithms for Low-Depth Genotyping-By-Sequencing (GBS) Data. Feltus FA, ed. PLoS ONE. 2016;11(8):e0160733. PubMed

Citing

Gennady Khvorykh. inzilico/imputeqc v1.0.0 (2018). GitHub repository, https://github.com/inzilico/imputeqc.

Author

Gennady Khvorykh, a bioinformatician, inzilico.com

Suggestions, questions, and comments are open! Feel free to drop me the message.