БИ ДЗ-5. Фазирование и импутация

Теория:

> **Фазирование** – это процесс статистической оценки гаплотипов.
> **Импутация** - это замещение ошибочных, противоречивых или отсутствующих данных другими данными.
> Известно, что **фазирование приводит к повышению точности импутации**.

1. Perform filtration of vcf that will be imputed. Use AF > 1%. How many positions are left? How many positions are filtered out? Why do we do filtration by AF before imputation?

AF - это алельная частота

```
cp -i common/imputation/data/samples.vcf.gz common/K

gunzip common/K/samples.vcf.gz

cat common/K/samples.vcf.gz
```

разархивировала, чтобы посмотреть, что там в файле

```
cp -i common/imputation/data/samples.vcf.gz common/K

bcftools filter -O z -o filtered.vcf.gz -i '%QUAL>50' in.vcf.gz

cd ./common/K

bcftools filter samples.vcf.gz 'AF <= 0.01' -e -o filtered.vcf.gz
```

-e делает filter-out
не сработало, требует индекс файл

```
bcftools index samples.vcf.gz
```

создается файл samples.vcf.gz.sci

```
bcftools filter -e 'AF <= 0.01' samples.vcf.gz -Oz -o filtered.vcf.gz
```

сработало

```
gunzip filtered.vcf.gz


bcftools view filtered.vcf | grep -v '^#' | wc -l
```

считает число строк, которые начинаются не с диеза (#)

💬 10666 (столько позиций осталось)

```
bcftools view samples.vcf | grep -v '^#' | wc -l
```

💬 50843 - 10666 = 40177 (столько позиций убралось)

💅 Фильтрация по аллельной частоте (AF) позволяет улучшить качество данных (исключаем ошибки импутации):

> Если AF > 1%, то этот вариант аллея достаточно широко представлен в популяции, и это не редкий аллель, то есть исключаем ошибки импутации

2. Use script /srv/common/imputation/perform_imputation.sh to impute vcf. Figure out what values should be passed to the parameters.

```
cd
```

перехожу в корневую папку

```
nano ./common/imputation/perform_imp.sh
```

смотрю на скрипт

```
java -Xmx1g \
  -jar ${beagle} \
    gt=${1}.vcf.gz \
    ref=${2}.vcf.gz \
    out=${3} \
    chrom=${4}:${5}-${6} \
```

для выполнения скрипта понадобится файл input.vcf, reference.vcf, а так же координаты первой и последней позиций

```
cd


cp -i ./common/imputation/reference_panel/1kg_subset.vcf.gz ./common/K
```

скопировала себе референсный файл

```
bcftools view filtered.vcf.gz |grep -v "^#"| head -n 1
```

```
bcftools view filtered.vcf.gz |grep -v "^#"| tail -n 1
```

```
(base) jupyter-gorovenko-e@hse-students:~/common/K$ bcfto
ols view filtered.vcf.gz |grep -v "^#"| head -n 1
chr1    1000112 rs3121571       G       T       .       P
ASS     NS=22691;AN=20;AF=0.779075;AC=13        GT      0
/1      0/1     1/1     1/1     0/1     0/0     1/1     0
/1      1/1     0/1
(base) jupyter-gorovenko-e@hse-students:~/common/K$ bcfto
ols view filtered.vcf.gz |grep -v "^#"| tail -n 1
chr1    4999725 rs75747187      T       C       .       P
ASS     NS=22691;AN=20;AF=0.0124058;AC=0        GT      0
/0      0/0     0/0     0/0     0/0     0/0     0/0     0
/0      0/0     0/0
```

нашла координаты: 1000112, 4999725

```
cd ./common/K

chmod +x perform_imp.sh
# даю права на выполнение скрипта

gzip filtered.vcf
# сжимаю файл

bash perform_imp.sh filtered 1kg_subset res_imp chr1 1000112 4999725
```

типа сначала подаем инпут файл, потом референсный файл

как-то тяжело последняя команда далась 👾

3. How long did imputation take? What parameters of BEAGLE can be adjusted to speed up the process of imputation?

```
Haplotype phasing time:         34 seconds
Imputation time:                3 seconds
Total time:                     51 seconds
```

```
nano ./common/K/perform_imp.sh
```

Чтобы код работал быстрее, сократила число итераций с 10 до 5 через редактор nano.

```
cd ./common/K

bash perform_imp.sh filtered 1kg_subset res2_imp chr1 1000112 4999725
```

```
Haplotype phasing time:         21 seconds
Imputation time:                3 seconds
Total time:                     36 seconds
```

4. How many positions are in the reference panel? Does the size (number of positions, samples) of input vcf and reference panel matter for the speed of imputation process?

```
bcftools stats 1kg_subset.vcf.gz | grep "number of records"
```

ищу число записей в референсном файле

```
(base) jupyter-gorovenko-e@hse-students:~/common/K$ bcfto
ols stats 1kg_subset.vcf.gz | grep "number of records"
#   number of records   .. number of data rows in the VCF
SN      0       number of records:      135164
```

💬 135164

Естественно, число позиций и образцов влияет на скорость импутации. Чем больше данных, тем больше времени занимают вычисления (calculations, i mean).

5. How many positions in the imputed vcf? What do you notice?

```
bcftools stats filtered.vcf.gz | grep "number of records"
```

ищу число записей в референсном файле

```
(base) jupyter-gorovenko-e@hse-students:~/common/K$ bcfto
ols stats filtered.vcf.gz | grep "number of records"
#   number of records   .. number of data rows in the VCF
SN      0       number of records:      10666
```

💬 10666

Число записей в референсном файле почти в 13 раз больше, чем в инпут файле.

6. Calculate genotype concordance between imputed and ground truth vcf.

Теория:

> **Генотипическая конкордантность** - мера сходства генотипов.

```
cd
```

gtcheck хочет расширение .gz, дам его ему, потому что ругается. Еще хочет индекс файл - без проблем...

```
gunzip ./common/K/filtered.vcf.gz


bgzip ./common/K/filtered.vcf


bcftools index ./common/K/filtered.vcf.gz
```

теперь это сработало:

```
bcftools gtcheck -g ./common/imputation/data/ground_truth.vcf.gz ./common/K/
filtered.vcf.gz > ./common/K/concordance_report.txt
```

выплюнулся файл с таким содержимым:

```
31 #     - Number of sites compared for this pair of samples (bigger = more informative)
32 #     - Number of matching genotypes
33 #DCv2   [2]Query Sample [3]Genotyped Sample [4]Discordance  [5]Average -log P(HWE)
   [6]Number of sites compared [6]Number of matching genotypes
34 DCv2    EGAN00001060849 EGAN00001060849 0.000000e+00    4.905550e-01    10666   10666
35 DCv2    EGAN00001060849 EGAN00001060184 3.978867e+04    1.247976e-01    10666   10666
36 DCv2    EGAN00001060849 EGAN00001060185 3.767950e+04    1.537148e-01    10666   10666
37 DCv2    EGAN00001060849 EGAN00001060850 3.766108e+04    1.402561e-01    10666   10666
38 DCv2    EGAN00001060849 EGAN00001060187 2.870863e+04    1.940440e-01    10666   10666
39 DCv2    EGAN00001060849 EGAN00001061072 3.377432e+04    1.620788e-01    10666   10666
40 DCv2    EGAN00001060849 EGAN00001060190 3.724662e+04    1.784268e-01    10666   10666
41 DCv2    EGAN00001060849 EGAN00001060852 3.627953e+04    1.392484e-01    10666   10666
42 DCv2    EGAN00001060849 EGAN00001060194 3.162831e+04    1.727875e-01    10666   10666
43 DCv2    EGAN00001060849 EGAN00001060197 3.531244e+04    1.524132e-01    10666   10666
```

```
cd ./common/K


grep "^DCv2" concordance_report.txt | awk '$4 < 0.05 {count++} END {print
count}'
```

нахожу образцы со значением Discordance < 0.05

```
(base) jupyter-gorovenko-e@hse-students:~/common/K$ grep
"^DCv2" concordance_report.txt | awk '$4 < 0.05 {count++}
 END {print count}'
10
```

💬 10

Из 10666 образцов соответствие наблюдается только в 10 из них. Следовательно, можно сказать, что инпут файл и референсный файл обладают низким уровнем сходства.