Phasing and imputation

Скачаем все необходимые файлы

Code

```
import gdown

files = {
    '12ZYMOOmh_HWa_Kth3A1oF50f4O49o4u9': '1kg_subset.vcf.gz',
    '12d9JHXk1P5v3FvbL1GwDvPyA812Oxkgd': 'samples.vcf.gz',
    '12mawR58pm8s1GLHJ8A3K1F6YB_0zULaF': 'ground_truth.vcf.gz',
    '12nqGsxJBx424EcvF_Sgmt945BhlSpw8W': 'perform_imp.sh',
    '12oWRevd2ARJENOon-b9djLnZLo8tyrya': 'plink.GRCh38.map.zip'
}

for file_id, file_name in files.items():
  gdown.download(f'https://drive.google.com/uc?id={file_id}', file_name, quiet=False)

!apt-get install -y openjdk-11-jre-headless
```

И .map для GRCh38

Code

```
import zipfile
import os
import glob

!wget -O beagle.jar https://faculty.washington.edu/browning/beagle/beagle.29Oct24.c8e.jar
zip_file_path = 'plink.GRCh38.map.zip'
output_dir = 'files/Genetic_maps/hg38/'
os.makedirs(output_dir, exist_ok=True)

with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
    zip_ref.extractall(output_dir)

for file_path in glob.glob(os.path.join(output_dir, 'plink.chr*.GRCh38.map')):
    chr_number = file_path.split('chr')[1].split('.')[0]
    new_file_name = f'beagle_hg38_{chr_number}.map'
    new_file_path = os.path.join(output_dir, new_file_name)
    os.rename(file_path, new_file_path)
```

Установим BCFtools

Code

```
!wget https://github.com/samtools/bcftools/releases/download/1.21/bcftools-1.21.tar.bz2
!tar -xvjf bcftools-1.21.tar.bz2
%cd bcftools-1.21
!make
!sudo cp bcftools /usr/local/bin/
!bcftools --version
```

Исправим perform_imp.sh

Code

```
file_path = '/content/perform_imp.sh'

with open(file_path, 'r') as file:
    lines = file.readlines()

with open(file_path, 'w') as file:
    for line in lines:
        if 'beagle="/usr/bin/beagle.29Oct24.c8e.jar"' in line:
            line = 'beagle="/content/beagle.jar"\n'
        line.replace('map=/files/Genetic_maps/hg38/beagle_hg38_${4}.map \\',
                     'map=files/Genetic_maps/hg38/beagle_hg38_${4}.map \\')
        line = line.replace('              #### this is only prefix, don\'t put vcf.gz', '')

        file.write(line)
```

Проведем фильтрацию

Code

```
input_vcf = "samples.vcf.gz"
filtered_vcf = "filtered_samples"

# Filter input VCF
!bcftools view -i 'AF > 0.01' -Oz -o {filtered_vcf} {input_vcf}
!bcftools index {filtered_vcf}

# Count positions
positions_left = !bcftools query -f '%CHROM\t%POS\n' {filtered_vcf} | wc -l
positions_filtered_out = !bcftools query -f '%CHROM\t%POS\n' {input_vcf} | wc -l

print(f"Positions left: {positions_left}")
print(f"Positions filtered out: {positions_filtered_out}")
```

Получим:

- Позиции, оставшиеся после фильтрации: 10 666
- Позиции, отфильтрованные: 50 843

Фильтрация по частоте аллелей (AF) перед вменением необходима по нескольким причинам:

- Шум: позиции с низкой частотой аллелей (например, AF $\leqslant 1\%$) могут представлять собой редкие варианты, их удаление приводит к уменьшению 'шумов' в данных.
- Повышенная точность: удаляя варианты с низкой частотой, алгоритм рассматривает более распространенные варианты, лучше представленные в контрольных панелях, предоставляя более точные прогнозы генотипа.
- Уменьшение времени вычисления: отфильтровывание позиций с низкой AF уменьшает размер обрабатываемого набора данных, что может значительно сократить время вычислений и использование ресурсов в процессе импутации (и так как алгоритм импутации основан на Байесовских методах и Марковская цепи Монте-Карло, то он полиномиально(?) зависит от размера входных данных).
- Распространенные варианты часто представляют больший интерес и значимость в популяционных исследованиях и анализах ассоциаций заболеваний.

В этом случае фильтрация по AF $> 1\%$ привела к сохранению 10 666 позиций при отфильтровывании 50 843 позиций из исходного файла.

Проведем импутацию, используя скрипт perform_imputation.sh, параметры наглядно указаны в коде ниже

Code

```
import time

# Define chromosome and region for imputation
chrom = 'chr1'
start = 1  # Start position
end = 248956422  # End position
imputed_output_prefix = "imputed_samples_chr1"

# Measure time taken for imputation
start_time = time.time()

# Execute BEAGLE script with updated parameters
!bash perform_imp.sh {filtered_vcf} 1kg_subset {imputed_output_prefix}
    {chrom} {start} {end}

end_time = time.time()
print(f"Imputation for {chrom} took {end_time - start_time:.2f} seconds")
```

Воспользовавшись результатами, ответим на следующие вопросы:

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

6 Calculate genotype concordance between imputed and ground truth vcf.

Cumulative Statistics

Field	Details
File Name	beagle.29Oct24.c8e.jar (version 5.4)
Copyright	(C) 2014-2022 Brian L. Browning
Command to Run	java -jar beagle.29Oct24.c8e.jar
Start Time	10:46 AM UTC on 24 Nov 2024
Command Line	java -Xmx1024m -jar beagle.29Oct24.c8e.jar gt=/srv/common/filtered_samples.vcf.gz ref=/srv/common/imputation/reference_panel/1kg_subset.vcf.gz out=/srv/common/imputed_samples_chr1 chrom=chr1:1-248956422 map=/files/Genetic_maps/hg38/beagle_hg38_chr1.map nthreads=1 window=5 overlap=4 iterations=10 ne=20000 impute=true gp=true seed=-9999
Reference Samples	3,202
Study Samples	10

Window	Region	Reference Markers	Study Markers	Burnin Iteration Time	Estimated ne	Estimated err	Imputation Time
Window 1	chr1: 1000001 -3496623	88,118	5,620	1 second (x2)	725895	6.9e-04	1 second
Window 2	chr1: 1769849 -3720708	66,557	4,544	1 second (x2)	597390	5.8e-04	0 seconds
Window 3	chr1: 2284324 -4041024	60,004	4,663	1 second (x2)	611160	6.7e-04	0 seconds
Window 4	chr1: 2962745 -4320569	45,826	4,216	1 second (x2)	582106	3.4e-04	0 seconds
Window 5	chr1: 3233805 -4747085	48,916	4,955	1 second	585193	3.2e-04	0 seconds
Window 6	chr1: 3496623 -4999986	47,047	4,909	1 second	581756	3.4e-04	0 seconds

Field	Total
Reference Markers	135,164
Study Markers	10,529
Haplotype Phasing Time	34 seconds
Imputation Time	3 seconds
Total Time	48 seconds

End Information

End Time	Details
End Time	10:46 AM UTC on 24 Nov 2024
Final Message	beagle.29Oct24.c8e.jar finished
Total Duration for chr1	Imputation for chr1 took 48.15 seconds

Code

```
imputed_vcf_chr1 = f"{imputed_output_prefix}.vcf.gz"
ground_truth_vcf = "ground_truth.vcf.gz"

# Use bcftools to calculate genotype concordance
!bcftools gtcheck -g {ground_truth_vcf} {imputed_vcf_chr1} > concordance_chr1.txt

# Parse and display concordance output
with open("concordance_chr1.txt") as f:
    concordance_data = f.read()
print("Genotype Concordance for chr1:")
print(concordance_data)
```

Получим:

Query Sample EGAN0000	Genotyped Sample EGAN0000	Discordance	Average -log P(HWE)	Number of sites compared	Number of matching genotypes
1060849	1060849	$6.355135e{+}02$	$1.371852e{-}01$	54220	54151
1060849	1060184	$5.622913e{+}04$	$3.907153e{-}02$	54220	48720
1060849	1060185	$5.364102e{+}04$	$4.781371e{-}02$	54220	48949
1060849	1060850	$5.308840e{+}04$	$4.395425e{-}02$	54220	48971
1060849	1060187	$4.095838e{+}04$	$5.824279e{-}02$	54220	50010
1060849	1061072	$4.837271e{+}04$	$4.875914e{-}02$	54220	49768
1060849	1060190	$5.267394e{+}04$	$5.452548e{-}02$	54220	49132
1060849	1060852	$5.190948e{+}04$	$4.223769e{-}02$	54220	48990
1060849	1060194	$4.529645e{+}04$	$5.197073e{-}02$	54220	49566
1060849	1060197	$4.991083e{+}04$	$4.701610e{-}02$	54220	49318
1060184	1060849	$5.632123e{+}04$	$3.933541e{-}02$	54220	48712

1060184	1060184	7.091962e+02	1.399885e−01	54220	54143
1060184	1060185	5.283051e+04	4.950644e−02	54220	49232
1060184	1060850	5.354892e+04	4.323204e−02	54220	49100
1060184	1060187	4.561882e+04	4.772631e−02	54220	49573
1060184	1061072	5.528046e+04	4.336541e−02	54220	49677
1060184	1060190	5.797909e+04	4.419833e−02	54220	48589
1060184	1060852	4.590434e+04	5.242713e−02	54220	49663
1060184	1060194	4.463331e+04	5.154403e−02	54220	49670
1060184	1060197	5.167922e+04	4.729439e−02	54220	49524
1060185	1060849	5.357655e+04	4.792267e−02	54220	48952
1060185	1060184	5.248973e+04	4.978510e−02	54220	49270
1060185	1060185	7.736686e+02	1.447151e−01	54220	54136
1060185	1060850	4.870428e+04	5.204791e−02	54220	49302
1060185	1060187	5.228710e+04	4.205820e−02	54220	49103
1060185	1061072	5.548309e+04	4.193267e−02	54220	49390
1060185	1060190	5.412917e+04	5.416229e−02	54220	48864
1060185	1060852	5.163317e+04	4.583735e−02	54220	49043
1060185	1060194	4.910953e+04	4.739416e−02	54220	49282
1060185	1060197	4.641091e+04	5.440545e−02	54220	49628
1060850	1060849	5.331866e+04	4.380369e−02	54220	48949
1060850	1060184	5.346603e+04	4.317193e−02	54220	49106
1060850	1060185	4.883322e+04	5.210418e−02	54220	49294
1060850	1060850	6.631445e+02	1.335761e−01	54220	54153
1060850	1060187	4.744246e+04	4.564801e−02	54220	49535
1060850	1061072	5.526204e+04	3.959488e−02	54220	49475
1060850	1060190	4.933979e+04	5.558317e−02	54220	49214
1060850	1060852	4.387806e+04	5.309785e−02	54220	49844
1060850	1060194	4.341754e+04	5.327710e−02	54220	49840
1060850	1060197	4.764509e+04	4.634027e−02	54220	49462
1060187	1060849	4.106891e+04	5.828453e−02	54220	49998
1060187	1060184	4.548987e+04	4.782492e−02	54220	49586
1060187	1060185	5.246210e+04	4.195164e−02	54220	49086
1060187	1060850	4.745167e+04	4.561972e−02	54220	49534
1060187	1060187	4.881480e+02	1.212511e−01	54220	54167
1060187	1061072	4.936742e+04	4.369375e−02	54220	50022
1060187	1060190	5.510547e+04	4.341191e−02	54220	48904
1060187	1060852	4.559118e+04	4.645929e−02	54220	49716
1060187	1060194	3.824133e+04	5.524512e−02	54220	50264
1060187	1060197	4.233993e+04	5.294607e−02	54220	50105
1061072	1060849	4.830824e+04	4.880424e−02	54220	49770
1061072	1060184	5.501336e+04	4.338162e−02	54220	49700
1061072	1060185	5.544625e+04	4.194558e−02	54220	49393
1061072	1060850	5.507784e+04	3.959843e−02	54220	49489
1061072	1060187	4.919243e+04	4.369673e−02	54220	50035
1061072	1061072	3.868343e+02	1.373300e−01	54220	54193
1061072	1060190	6.667365e+04	3.208866e−02	54220	48736
1061072	1060852	5.242526e+04	3.994004e−02	54220	49692
1061072	1060194	5.389891e+04	3.503694e−02	54220	49412
1061072	1060197	5.309761e+04	4.116014e−02	54220	49826
1060190	1060849	5.271078e+04	5.450962e−02	54220	49129

1060190	1060184	$5.777647e{+}04$	$4.418045e{-}02$	54220	48612
1060190	1060185	$5.417522e{+}04$	$5.405382e{-}02$	54220	48863
1060190	1060850	$4.925690e{+}04$	$5.554564e{-}02$	54220	49223
1060190	1060187	$5.499494e{+}04$	$4.341138e{-}02$	54220	48916
1060190	1061072	$6.664602e{+}04$	$3.222080e{-}02$	54220	48741
1060190	1060190	$4.513067e{+}02$	$1.595239e{-}01$	54220	54171
1060190	1060852	$5.294104e{+}04$	$4.781406e{-}02$	54220	48992
1060190	1060194	$5.202000e{+}04$	$4.736277e{-}02$	54220	48984
1060190	1060197	$5.153185e{+}04$	$5.017325e{-}02$	54220	49145
1060852	1060849	$5.177132e{+}04$	$4.242639e{-}02$	54220	49012
1060852	1060184	$4.552671e{+}04$	$5.247846e{-}02$	54220	49699
1060852	1060185	$5.148580e{+}04$	$4.594014e{-}02$	54220	49061
1060852	1060850	$4.369385e{+}04$	$5.300136e{-}02$	54220	49859
1060852	1060187	$4.537935e{+}04$	$4.644170e{-}02$	54220	49737
1060852	1061072	$5.238842e{+}04$	$3.991867e{-}02$	54220	49700
1060852	1060190	$5.291341e{+}04$	$4.770275e{-}02$	54220	48998
1060852	1060852	$5.894618e{+}02$	$1.272806e{-}01$	54220	54156
1060852	1060194	$3.899658e{+}04$	$5.593188e{-}02$	54220	50109
1060852	1060197	$4.600565e{+}04$	$4.738571e{-}02$	54220	49651
1060194	1060849	$4.548066e{+}04$	$5.179953e{-}02$	54220	49552
1060194	1060184	$4.452279e{+}04$	$5.154284e{-}02$	54220	49687
1060194	1060185	$4.913717e{+}04$	$4.745409e{-}02$	54220	49282
1060194	1060850	$4.338991e{+}04$	$5.325256e{-}02$	54220	49847
1060194	1060187	$3.814923e{+}04$	$5.534229e{-}02$	54220	50276
1060194	1061072	$5.398180e{+}04$	$3.509152e{-}02$	54220	49409
1060194	1060190	$5.203842e{+}04$	$4.743470e{-}02$	54220	48983
1060194	1060852	$3.919000e{+}04$	$5.593109e{-}02$	54220	50094
1060194	1060194	$6.170928e{+}02$	$1.215049e{-}01$	54220	54153
1060194	1060197	$4.507541e{+}04$	$4.847326e{-}02$	54220	49724
1060197	1060849	$5.002136e{+}04$	$4.695528e{-}02$	54220	49311
1060197	1060184	$5.162396e{+}04$	$4.712501e{-}02$	54220	49533
1060197	1060185	$4.654906e{+}04$	$5.425562e{-}02$	54220	49617
1060197	1060850	$4.765430e{+}04$	$4.617662e{-}02$	54220	49459
1060197	1060187	$4.232151e{+}04$	$5.288030e{-}02$	54220	50107
1060197	1061072	$5.318051e{+}04$	$4.112927e{-}02$	54220	49821
1060197	1060190	$5.158712e{+}04$	$5.021407e{-}02$	54220	49141
1060197	1060852	$4.616223e{+}04$	$4.738822e{-}02$	54220	49636
1060197	1060194	$4.511225e{+}04$	$4.839797e{-}02$	54220	49720
1060197	1060197	$3.223619e{+}02$	$1.306648e{-}01$	54220	54187

Таблица 5: Discordance and Genotype Comparison

Где информацию о каждом столбце можно получить из результата скрипта:

- Genotyped sample
- Discordance, given either as an abstract score or number of mismatches, see the options -E/-u in man page for details. Note that samples with high missingness have fewer sites compared, which results in lower overall discordance. Therefore it is advisable to use the average score per site rather than the absolute value, i.e. divide the value by the number of sites compared (smaller value = better match)

- Average negative log of HWE probability at matching sites, attempts to quantify the following intuition: rare genotype matches are more informative than common genotype matches, hence two samples with similar discordance can be further stratified by the HWE score (bigger value = better match, the observed concordance was less likely to occur by chance)
- Number of sites compared for this pair of samples (bigger = more informative)
- Number of matching genotypes

Осталось посчитать позиции чтобы ответить на следующие вопросы

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?

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

Code

```
reference_panel = "1kg_subset.vcf.gz"
ref_positions = !bcftools view -r {chrom}:{start}-{end} {reference_panel} | wc -l

imputed_positions = !bcftools view -r {chrom}:{start}-{end} {imputed_vcf_chr1} | wc -l

print(f"Number of positions in reference panel for chr1: {ref_positions}")
print(f"Number of positions in imputed VCF for chr1: {imputed_positions}")
```

Получим:

- Number of positions in reference panel for chr1: 135283
- Number of positions in imputed VCF for chr1: 135179

Ссылки на .ipynb и concordance_chr1.txt