#版本：PGG.AF1M 频率数据产出

1.前期准备

在进行频率数据产出之前需要对数据进行预处理和质控，推荐使用 GATK 进行 jointcalling 和 VQSR 的过滤。

一般我们在下游遗传结构的分析中，大多采用经过质控的 biallelic SNP，因此还需一步操作来提取这些位点。

```
$ bcftools view -f PASS -m 2 -M 2 -v snps NGS2144.jointcalling.VQSR.variants.vcf.gz |bgzip -@
16 -c > NGS2144.jointcalling.VQSR.variants.PASS.biallelic.SNPs.vcf.gz
$ tabix -p vcf NGS2144.jointcalling.VQSR.variants.PASS.biallelic.SNPs.vcf.gz
```

如果数据已经经过过滤处理，请忽略上述步骤。

```
1. 准备工作目录
# 创建项目目录并进入
mkdir PGG_allele_frequency_pipeline
cd PGG_allele_frequency_pipeline

2. 准备输入文件
将以下文件放入相应目录：
- 原始 VCF 文件（gzip 压缩格式）放入`/PGG_allele_frequency_pipeline`
```

2.计算频率信息

本流程将指导您如何从 VCF 文件中筛选特定基因组区域，并为不同群体计算等位基因频率，全程仅使用命令行工具。

一、准备工作

在开始前，请先安装以下工具： bcftools, vcftools,plink

```
# 安装 miniconda（用户目录）
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
bash Miniconda3-latest-Linux-x86_64.sh -b -p $HOME/miniconda
source $HOME/miniconda/bin/activate
# 通过 conda 安装
conda install -c bioconda bcftools vcftools plink -y
```

或者使用 wget 源码安装（见 README.md）

二、流程步骤

1. 准备输入文件

将以下文件放入相应目录：

- 原始 VCF 文件（gzip 压缩格式）放入`input/`

2. 计算等位基因频率

```
# 使用 vcftools 直接计算
vcftools --vcf your_file.vcf --freq --out output_frequency
```

3.性能优化建议
内存消耗：vcftools 处理大型文件会占用大量内存

```
# 先压缩 VCF 文件
bgzip your_file.vcf
tabix -p vcf your_file.vcf.gz

# 然后使用压缩文件运行
vcftools --gzvcf your_file.vcf.gz --freq --out output_frequency

# 对于极大文件，可以：
# 1. 先按染色体拆分处理
for chr in {1..22} X Y; do
   vcftools --vcf your_file.vcf --chr $chr --freq --out output_chr${chr}
done

# 2. 使用并行处理(需安装 GNU parallel)
parallel -j 4 "vcftools --vcf your_file.vcf --chr {} --freq --out output_chr{}" ::: {1..22} X Y
```

3. 频率数据处理

使用 vcftools 或者 bcftools 计算频率信息后，我们需要更改.frq 文件的格式，计算 genotype frequency 并输入 sample size、dataset 等基本信息，获取所需的数据落库的文件格式。

大样本 jointcalling 文件直接使用 linux 内置的 awk 工具处理频率数据

```
# AF-based Frequency Processing Pipeline
# 使用方法：bash af_processing_pipeline.sh [输入 VCF] [输出目录]
```

可以直接 bash 这个 shell 脚本，在 shell 内 vim 更改输入和输出路径

4. 多族群/跨省份样本集需要分开额外计算一次频率

因为 vcftools 计算频率会忽略 indels、多等位基因位点之外还会去掉很多的 singleton，同时考虑到数据集内的族群和省份信息是非常复杂的，所以我们需要使用 plink 额外计算一次频率。
准备两份样本基本信息的 txt 文件，格式如下

Population.pop

```
Han HG00123
Han HG00124
Han HG00125
Han HG00126
```

Province.pop

```
Shanghai HG00123
Shanghai HG00124
Beijing  HG00125
Xizang   HG00126
```

另外，如果样本内有不同的地区和 population，请使用 pop_freq_pipeline.sh 脚本内第二部分，独立计算每个省份对应的民族频率信息。准备一个 samplelist。

samples_province_pop.txt

```
WGC022072D    Xizang  Sherpa
WGC022073D    Xizang  Sherpa
WGC022074D    Xizang  Sherpa
```

```
# Simplified Population-specific Frequency Calculation
# 使用方法：bash pop_freq_pipeline.sh
可以直接 bash 这个 shell 脚本，在 shell 内 vim 更改输入和输出路径
```

5. 验证 pipeline 计算成果并计算 genotype

Af_freq_pipeline 当中的 genotype 为占位符，需要额外计算 genotype frequency。使用 plink/vcftools 计算 genotype frequency。

Pipeline：基于 genotype 字段

```
# Genotype Frequency Processing Pipeline
# 使用方法：bash genotype_freq_pipeline.sh [输入 VCF] [输出目录]
可以直接 bash 这个 shell 脚本，在 shell 内 vim 更改输入和输出路径
```

5. 可能遇到的问题

Shell 脚本在 github 端下载 code 之后如果经过 windows 端的 vscode，可能会产生 windows 的换行符导致报错，需要进行手动切换至 linux 端 shell 读取适配的换行符。

```
vim pipeline.sh
#执行转换命令
:set ff=unix
:wq
```

7. 输出结果检查

完成频率数据处理后，查看该文件样式

less result.tsv

输出文件示例：

chr rs_id pos ref alt ref_allele_freq alt_allele_freq dataset sample_size homozygous_reference homozygous_reference_freq heterozygous heterozygous_freq homozygous_alternative homozygous_alternative_freq variant population

chr1 rs12345 100 A G 0.75 0.25 MyDataset 100 homozygous_reference 0.5625 heterozygous 0.375 homozygous_alternative 0.0625 chr1:100-A-G global

chr1 rs56789 200 C T 0.10 0.90 MyDataset 100 homozygous_reference 0.01 heterozygous 0.18 homozygous_alternative 0.81 chr1:200-C-T global

完成群体 population 频率数据之后，检查该文件样式

less population.frq

最终需求文件：

final_result.tsv

Plink 产生的 output 文件：

.fam/.bam/.bed/.bim

.frq