#版本: PGG.Selection.Burden

1.前期准备

在进行分析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。

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

此外,我们还需要对数据进行 phasing,需分染色体进行,如用 shapeit5 处理,可参考: https://github.com/Shuhua-Group/PGG.Panel/blob/main/PGG.Panel 数据操作文档.pdf step4

如果数据已经经过相应处理,请忽略上述步骤。

2. 正式运算

一、准备工作

基础软件安装:

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

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 beftools veftools plink -y

conda install bioconda::plink2

二、流程步骤

1. prepare

创建项目目录并进入

mkdir PGG_selection_burden_pipeline cd PGG selection burden pipeline

准备输入文件

在父目录 PGG_selection_burden_pipeline 下建立输入文件的存放目录并进入

mkdir 00.input; cd 00.input

将以下文件放入'00.input/':

- 样本信息文件

准备**汉族**样本信息文件 group.txt, 按地区分组, 第一列为地区, 第二列为样本名, 以制表符分隔, 格式如下:

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

- 原始 VCF 文件 (.vef.gz)、按地区分组提取子 vef 文件(.vef.gz 和.vef)及,根据原始 vef 按地区分组提取子 vef 文件的脚本如下。

```
info='group.txt' #替换为你的样本信息文件
for group in `awk '{print $1}' $info|sort -u`
do

mkdir -p $group
for k in {1..22}
do

vcf="your.chr$k.phased.vcf.gz" #替换为你的 vcf 文件
awk '$1==""$group""{print $2}' group.txt > $group/$group.list
bcftools view -S $group/${group}.list $vcf -Oz -o $group/${group}.chr${k}.vcf.gz
bgzip -d -c $group/${group}.chr${k}.vcf.gz > $group/${group}.chr${k}.vcf.gz
done

done
```

注: 如样本间存在亲缘关系,请在除了 genetic burden 以外的分析去除二级及以上亲缘关系后进行分析,并提供去除样本的 list。可参考:

https://github.com/Shuhua-Group/PGG.Panel/blob/main/PGG.Panel 数据操作文档.pdf step2

2. Frequency

在父目录 PGG_selection_burden_pipeline 下建立 frq 的工作目录

```
mkdir 01.frq; cd 01.frq
```

将以下脚本下载到工作目录中

https://github.com/Shuhua-Group/PGG.Selection.Burden/blob/main/frq.sh

对 chr1-22 执行脚本

```
for k in {1..22}
do
sh frq.sh $k
done
```

检查各个子目录中的结果文件 (*.chr*.afreq), 示例如下:

#CHRC	M ID	REF	ALT	ALT_FR	EQS	OBS_CT	
22	22:105192	276 C	C	0	206		
22	22:105193	25 C	A	0	206		
22	22:105193	889 T	C	0	206		

3. Theta D H

在父目录 PGG selection burden pipeline 下建立 Theta D H 的工作目录

mkdir 02.Theta_D_H; cd 02.Theta_D_H

将以下脚本下载到工作目录中

 $https://github.com/Shuhua-Group/Theta_D_H.Est/blob/master/Theta_D_H.Est \\ https://github.com/Shuhua-Group/PGG.Selection.Burden/blob/main/Theta_D_H.sh$

对 chr1-22 执行脚本

for k in {1..22}
do
sh Theta_D_H.sh \$k
done

检查各个子目录中的结果文件 (*.chr*.gz) , 示例如下:

regionID chr start end #sequence #marker #singleton ThetaPI ThetaK

#segregating #haplotype Hap_diversity Hfaywu norm_Hfaywu Ffuli Dfuli

Dtajima Dtajima_P Dtajima_adj.P

22 22 1 50000 206 0 0 0.0 0.0 0 0 0.0 NA NA

4. IHS

软件安装

selscan 下载链接:

https://github.com/szpiech/selscan/blob/master/releases/selscan-linux-2.0.0.tar.gz selscan 安装教程:

https://github.com/szpiech/selscan/blob/master/INSTALL

predictGMAP 下载链接:

https://github.com/szpiech/predictGMAP/tree/master/src

predictGMAP 安装教程:

https://github.com/szpiech/predictGMAP/blob/master/README

plink map 文件下载:

https://bochet.gcc.biostat.washington.edu/beagle/genetic maps/plink.GRCh38.map.zip

在父目录 PGG_selection_burden_pipeline 下建立 ihs 的工作目录并进入

mkdir 03.IHS; cd 03.IHS

将以下脚本下载到工作目录中

https://github.com/Shuhua-Group/PGG.Selection.Burden/blob/main/ihs.sh

对 ihs.sh 进行编辑,将以下三行替换为你的 predictGMAP 路径、selscan 路径、plink map 路径

predictGMAP_d='/home/sunyumeng/software/predictGMAP' selscan_d='/home/sunyumeng/software/selscan-linux-2.0.0' map_d='/home/sunyumeng/data/PLINK_format_genetic_map'

对 chr1-22 执行脚本

for k in {1..22} do sh ihs.sh \$k done

检查各个子目录中的结果文件 (*.chr*.ihs.out.100bins.norm) ,示例如下:

chr22:10634429 10634429 0.140777 0.00360501 0.00388136 -0.0320774

检查各个子目录中的结果文件 (*.chr*.ihs.out.100bins.norm), 示例如下:

chr22:10634429 10634429 0.140777 0.00360501 0.00388136 -0.0320774 -1.27343 0

检查各个子目录中的结果文件 (*.chr*.ihs.out.100bins.norm.50kb.windows) , 示例如下:

1 50001 0 -1 -1 NA

5. CLR

软件安装

wget http://sco.h-its.org/exelixis/resource/download/software/SweeD_v3.2.1_Linux.tar.gz tar -xzf SweeD_v3.2.1_Linux.tar.gz

在父目录 PGG_selection_burden_pipeline 下建立 CLR 的工作目录并进入

mkdir 04.CLR; cd 04.CLR

将以下脚本下载到工作目录中

https://github.com/Shuhua-Group/PGG.Selection.Burden/blob/main/clr.sh

对 clr.sh 进行编辑,将下行替换为你的 SweeD 下载路径

SweeD d='/home/sunyumeng/software/SweeD v3.2.1 Linux'

对 chr1-22 执行脚本

```
for k in {1..22}
do
sh clr.sh $k
done
```

检查各个子目录中的结果文件(SweeD_Report.*.chr*.50kb), 示例如下:

//1

Position Likelihood Alpha

10522570 7.786579e-01 1.635287e-03

6. Genetic Burden

在父目录 PGG_selection_burden_pipeline 下建立 Genetic Burden 的工作目录并进入

mkdir 05.burden; cd 05.burden

将 22 个 DamageSnp 文件下载到工作目录中

百度网盘链接: https://pan.baidu.com/s/1HZY9A6aSbyvGypayl0cRWw 提取码: ky8i

将 pathway.input.txt 文件下载到工作目录中

https://github.com/Shuhua-Group/PGG.Selection.Burden/blob/main/pathway.input.txt

将以下脚本下载到工作目录中

https://github.com/Shuhua-Group/PGG.Selection.Burden/blob/main/burden.sh https://github.com/Shuhua-Group/PGG.Selection.Burden/blob/main/burden.py

对 chr1-22 执行脚本

```
for k in {1..22}
do
sh burden.sh $k
done
```

检查各个子目录中的结果文件 (*.chr*.whole.burden.txt) , 示例如下:

```
##Total number of deleterious SNVs: 1
##Total number of loss of function SNVs: 0
#sample geo sum_Het ....... CADD_weigted_Aloft_Dominant_Hom
NA18525 test 1 ....... 1
.......
```

检查各个子目录中的结果文件 (*.chr*.pathway.burden.txt) , 示例如下:

```
##Total number of deleterious SNVs of hsa00020: 1
##Total number of deleterious SNVs of hsa00010: 1
##Total number of deleterious SNVs of hsa00030: 0
... ...
##Total number of deleterious SNVs of hsa05418: 0
#sample geo hsa00020_Het ... ... hsa05418_Het hsa05418_Hom
NA18525 test 1 ... ... 1 0
... ...
```

检查各个子目录中的结果文件 (*.chr*.gene.burden.txt) , 示例如下:

```
##Total number of deleterious SNVs of 10327: 0
##Total number of deleterious SNVs of 124: 0
##Total number of deleterious SNVs of 125: 0
... ...
##Total number of deleterious SNVs of 57534: 0
#sample geo 10327_Het ... ... 57534_Het 57534_Hom
NA18525 test 1 ... ... 1 0
... ...
```

3.输出结果文件

将整个工作目录,除了输入文件 (00.input) 外,全部打包

```
rm -rf PGG_selection_burden_pipeline/00.input
tar -czvf PGG_selection_burden_results.tar PGG_selection_burden_pipeline
```