



群体进化分析培训

学习文档



目 录

一、群体分层分析

1.1 snp 数据预处理

1.2 系统发育树 (phylogenetic tree) 分析

1.3 主成分 (PCA) 分析

1.4 群体结构 (structure) 分析

二、计算遗传统计量

2.1 遗传多样性 (π) 计算

2.2 遗传距离 (f_{st}) 计算

三、选择性清除分析

四、连锁不平衡衰减 (LDdecay) 分析



准备的示例文件说明：

Filter.snp.vcf.gz：vcf 格式的 snp 数据；

group.tsv：样本分组信息；

genome.fa：参考基因组文件

Domestic.list：Domestic 亚群的样本列表；

Feral.list：Feral 亚群的样本列表；

Wild.list：Wild 亚群的样本列表；

chrom_map.txt：基因组文件中所有染色体序列对应编号文件

pca.ind：PCA 分析所需文件

一、群体分层分析

1.1 snp 数据预处理

- 使用软件：vcftools、plink(1.9)
- 输入文件：vcf 格式的 snp 文件 (Filter.snp.vcf.gz)

1) vcf 格式转换成 plink 格式：

```
vcftools \                # 软件程序
--gzvcf Filter.snp.vcf.gz \  # 指定输入文件(要求 vcf 格式的 gz 压缩文件)
--chrom-map chrom_map.txt \    # 指定染色体名的映射文件
--maf 0.05 \                # 指定次等位基因频率的过滤标准
--plink \                  # 转成 plink 格式
--out ramie_plink1          # 指定输出文件名的前缀
```



2) 过滤出连锁平衡的位点：

```
plink \                # 软件程序

--threads 4 \          # 设置最大线程数

--memory 5000 \        # 设置初始工作区大小（单位：MB）

--file ramie_plink1 \   # 指定输入文件名的前缀

--indep-pairwise 50 5 0.5 \ # 设置窗口大小(kb)、步长大小(ct)和  $r^2$  阈值

--out ramie_plink2      # 指定输出文件名的前缀
```

3) 根据.in 文件从总的 plink 文件中提取出连锁平衡的位点 ,用于后续的群体分层分析

```
plink \                # 软件程序

--threads 4 \          # 设置最大线程数

--memory 5000 \        # 设置初始工作区大小（单位：MB）

--file ramie_plink1 \   # 指定输入文件名的前缀（总的 plink 文件）

--extract ramie_plink2.prune.in \ # 指定第 2 步输出的包含连锁平衡位点的文件

--recode \              # 将二进制文件转换成可读的文本文件

--out ramie_plink3      # 指定输出文件名的前缀
```



1.2 系统发育树分析 (phylogenetic tree)

- 使用软件 : plink(1.9)、PHYLIP
- 输入文件 : plink 格式文件(ramie_plink3.ped 和 ramie_plink3.map)

1) 计算距离矩阵 :

```
plink \                                # 软件程序
--file ramie_plink3 \                  # 指定 plink 格式输入文件前缀
--distance square 1-ibs flat-missing \ # 计算距离矩阵
--out tree                             # 指定输出文件名前缀
```

2) 将距离矩阵文件转换成 phylip 格式的文件 :

```
perl mdist2phylip.pl \                # 执行 perl 脚本
tree.mdist \                           # 指定第 1 步输出的距离矩阵文件
tree.mdist.id \                        # 指定第 1 步输出的样本名文件
tree.phy.dist                          # 指定输出的文件名
```

3) 使用 PHYLIP 软件的 neighbor 程序进行 NJ 法建树 :

```
neighbor                               # 软件程序 (交互式程序)
tree.phy.dist                          # 指定第 2 步输出的 phylip 格式的距离矩阵文件
Y                                       # 确认以上信息
```



4) 用 R 包美化树文件：

```
Rscript tree.r \           # 运行 R 脚本
-t outtree \               # 指定树文件
-g group.tsv \             # 指定样本的分组文件
-m fan \                   # 指定树的形式 (None、fan、daylight)
-o tree                     # 输出文件前缀
```

1.3 主成分分析 (PCA)

- 使用软件：EIGENSOFT(7.2.1)软件的 smartpca 模块
- 输入文件：plink 格式文件(ramie_plink3.ped 和 ramie_plink3.map)

1) 生成 smartpca 的配置文件 (.par)：

```
perl smartpca.pl \         # 执行 perl 脚本
-i ramie_plink3.ped \      # 指定 plink 格式的 ped 文件
-a ramie_plink3.map \      # 指定 plink 格式的 map 文件
-b pca.ind \               # 指定样本分组文件
-o PCA \                   # 指定输出文件名前缀
-e PCA.egivals \           # 指定特征值文件名
-m 0 \                     # 指定离群值的数目
-t 8                       # 指定线程数
```



2) 运行 EIGENSOFT 软件的 smartpca 模块：

```
smartpca \           # 执行 smartpca 程序
-p PCA.par \         # 指定配置文件
> PCA.log            # 将标准输出写入 log 日志文件
```

3) 使用 R 包 ggplot2 和 scatterplot3d 将分析结果可视化：

```
Rscript PCA_plot.evec.r \   # 运行 R 脚本
-e PCA.evec \              # 指定 PCA 分析输出的特征向量结果文件
-i pca.ind \               # 指定样本分组文件
-o PCA                     # 指定输出文件名前缀
```

1.4 群体结构分析 (structure)

- 使用软件：plink(1.9)、admixture(1.3.0)、python 包 pong
- 输入文件：plink 格式文件(ramie_plink3.ped 和 ramie_plink3.map)

1) 生成的二进制的 plink 文件：

```
plink \               # 软件程序
--threads 4 \         # 设置最大线程数
--memory 5000 \       # 设置初始工作区大小（单位：MB）
--file ramie_plink3 \  # 指定输入文件名的前缀（前期过滤
后的 plink 文件）
```



```
--make-bed \           # 转换成二进制的 plink 文件  
--out structure        # 指定输出文件名的前缀
```

2) 使用 admixture 软件进行群体结构分析：

```
admixture \            # 软件程序  
--cv \                # 进行交叉错误验证  
-j8 \                 # 设置最大线程数  
-seed=20 \            # 设置随机种子数（需要重复时  
指定不同的随机整数）  
ramie_structure.bed \  # 输入文件（bed 格式）  
K \                   # 设置 K 值（只能是整数）  
> ramie_structure.$K.log # 标准输出写入 log 日志文件
```

3) 提取每个 K 的交叉验证错误统计值：

```
perl CV_error.pl \    # 运行 perl 脚本  
3 \                   # 指定最大重复数  
./ \                  # 指定统计目录  
CV_error.tsv          # 指定输出文件名
```

4) 将交叉验证错误统计值结果进行可视化绘图：

```
Rscript CV_plot.r \   # 运行 R 脚本  
CV_error.tsv \        # 指定输入文件
```




CV_error # 指定输出文件名前缀

5) 使用 pong 进行结果可视化：

pong \ # 程序名称
-m pong_filemap \ # 该文件包含 admixture 分析结果
文件（所有重复所有 k 值的结果）
-i ind2pop.txt \ # 该文件包含每个样本所属的分组名
-n pop_order.txt \ # 该文件包含分组的顺序
-p 8976 # 打开网页的端口

网页打开：[http:// localhost:8976/](http://localhost:8976/) 即可看到 structure 图，并下载保存

二、计算遗传统计量

2.1 遗传多样性 (pi) 计算

- 使用软件：vcftools
- 输入文件：vcf 格式的 snp 文件 (Filter.snp.vcf.gz)

1) 计算每个亚群单点的 pi 值：

vcftools \ # 软件程序
--gzvcf Filter.snp.vcf.gz \ # 指定输入文件名
--keep Domestic.list \ # 指定亚群包含的样本列表
--site-pi \ # 计算单点 pi 值
--out Domestic # 指定输出文件名前缀



2) 滑窗计算每个亚群窗口区域的 π 值 :

```
vcftools \                # 软件程序
--gzvcf Filter.snp.vcf.gz \    # 指定输入文件名
--keep Domestic.list \        # 指定亚群包含的样本列表
--window-pi 100000 \          # 指定窗口的长度 (单位 : bp)
--window-pi-step 20000 \      # 指定步长的长度 (单位 : bp)
--out Domestic                # 指定输出文件名前缀
```

3) 基于单点的 π 值计算全基因组的 π 均值 :

(1) 统计全基因组每条染色体的非 N 长度 :

```
iTools \                  # 软件程序
Fatools stat \            # 子命令 : 对 fasta 格式序列进行信息统计
-InPut genome.fa \        # 指定要统计的 fasta 格式的文件
-OutPut chrom_info.txt    # 指定输出文件名
```

(2) 统计全基因组的 π 均值 :

```
python calculate_WG_pi.py \    # 运行 python 脚本
-c chrom_info.txt \            # 指定 (1) 的输出文件 (包含染色体统计信息)
-d ./ \                        # 指定运行目录 (fst 和 pi 文件夹所在目录)
-p Domestic \                  # 指定要统计的亚群名
-o Domestic_wide_genome.pi     # 指定输出文件名
```



2.2 遗传距离 (fst) 计算

- 使用软件 : vcftools
- 输入文件 : vcf 格式的 snp 文件 (Filter.snp.vcf.gz)

1) 计算每个亚群单点的 fst 值 :

```
vcftools \                                # 软件程序
--gzvcf Filter.snp.vcf.gz \               # 指定输入文件名
--weir-fst-pop Domestic.list \            # 指定亚群 1 包含的样本列表
--weir-fst-pop Feral.list \               # 指定亚群 2 包含的样本列表
--out Domestic_Feral                      # 指定输出文件名前缀
```

2) 滑窗计算每个亚群窗口区域的 fst 值 :

```
vcftools \                                # 软件程序
--gzvcf Filter.snp.vcf.gz \               # 指定输入文件名
--weir-fst-pop Domestic.list \            # 指定亚群 1 包含的样本列表
--weir-fst-pop Feral.list \               # 指定亚群 2 包含的样本列表
--fst-window-size 100000 \                # 指定窗口的长度 (单位 : bp)
--fst-window-step 20000 \                 # 指定步长的长度 (单位 : bp)
--out Domestic_Feral                      # 指定输出文件名前缀
```

3) 基于单点的 fst 值计算全基因组的 fst 均值 :

```
python calculate_WG_fst.py \              # 运行 python 脚本
```



```
-d ./\                # 指定运行目录(fst 和 pi 文件夹所在目录)

-p1 Domestic \        # 指定亚群 1 名称

-p2 Feral \           # 指定亚群 2 名称

-o Domestic_Feral_wide_genome.fst    # 指定输出文件名
```

三、选择性清除分析

1. 对滑窗区域的 pi 和 fst 文件进行过滤 (过滤掉位点数目过少的窗口区域) :

```
python filter_pi_fst.py \    # 运行 python 脚本

-i ../6.Genetic_statistics \  # 指定输入文件目录(fst 和 pi 文件夹所在目录)

-o ./filter \                # 指定输出文件目录

-t 5 \                       # 指定过滤的 snp 数目阈值

-p1 Domestic \               # 指定亚群 1 名称

-p2 Feral                    # 指定亚群 2 名称
```

2. 筛选受选择的区域并绘图 :

```
Rscript selection_comb.r \    # 运行 R 脚本

-i ./filter \                 # 指定输入文件目录(fst 和 pi 文件夹所在目录)

-f Domestic \                 # 指定第 1 个亚群名称

-s Feral \                    # 指定第 2 个亚群名称

-t 0.05 \                     # 指定阈值 (常用 0.05 或者 0.01)

-o ./                          # 指定输出目录
```



3. pi ratio 绘制曼哈顿图：

3.1 pi 数据整理绘图文件：

```
python manhattan_data_pi.py \           # 运行 python 脚本
-i filter/pi \                          # 指定 pi 文件所在目录
-p1 Domestic \                          # 指定第 1 个亚群名称
-p2 Wild \                              # 指定第 2 个亚群名称
-o Domestic_Wild_common.pi              # 指定输出文件名
```

3.2 绘图：

```
Rscript population_Pi_Fst_manhattan.r \   # 运行 R 脚本
-i Domestic_Wild_common.pi \              # 指定输入文件名
-m pi \                                  # 指定绘图的统计量
-t 0.05 \                                # 指定阈值
-n 14 \                                  # 指定染色体数目
-w Wild \                                # 指定 wild-like 的亚群名称
-c Domestic \                             # 指定 cultivar-like 的亚群名称
-o ./                                     # 指定输出文件名目录
```

4. fst 绘制曼哈顿图：

```
Rscript population_Pi_Fst_manhattan.r \   # 运行 R 脚本
-i Domestic_Wild.windowed.weir.fst \      # 指定输入文件名
```



```
-m fst \ # 指定绘图的统计量
-t 0.05 \ # 指定阈值
-n 14 \ # 指定染色体数目
-w Wild \ # 指定 wild-like 的亚群名称
-c Domestic \ # 指定 cultivar-like 的亚群名称
-o ./ # 指定输出文件名目录
```

四、连锁不平衡衰减分析 (LDdecay)

- 使用软件：PopLDdecay
- 输入文件：vcf 格式的 snp 文件 (Filter.snp.vcf.gz)

1. 用 PopLDdecay 软件计算亚群的 r2 值：

```
PopLDdecay \ # 软件程序
-InVCF Filter.snp.vcf.gz \ # 指定输入文件名
-SubPop Domestic.list \ # 指定亚群包含的样本列表
-OutStat Domestic \ # 指定输出文件名的前缀
-MaxDist 300 # 指定计算 r2 的最长距离
```

2. 使用 Plot_*Pop.pl 脚本对 LDdecay 结果进行绘图：

```
perl Plot_MultiPop.pl \ # 运行 perl 脚本
-inList allpop.stats.list \ # 指定每个亚群对应的 LDdecay 结果文件
-output LDdecay \ # 指定输出文件名的前缀
```



```
-MaxDist 300 \           # 指定计算 r2 的最长距离  
  
-keepR                   # 输出绘图的 R 脚本
```