模型使用说明

首先，在系统中安装 plink 和 Beagle 软件，对原始 vcf 文件（以 filter_269_indel.vcf 为例）进行清洗：

#1.vcf 转 plink 文件

plink --vcf filter_269_indel.vcf --double-id --allow-extra-chr --recode --out 269_indel1

#2.检查基因型和样本缺失情况

plink --file 269_indel1 --allow-extra-chr --missing

#3.先对基因型进行过滤

plink --file 269_indel1 --allow-extra-chr --geno 0.1 --recode --out 269_indel2

#4.接着对个体进行缺失控制

plink --file 269_indel2 --allow-extra-chr --mind 0.2 --recode --out 269_indel3

#5.次等位基因频率（MAF）控制，首先对基因进行检验

plink --file 269_indel3 --allow-extra-chr --freq

#6.过滤基因频率小于 0.05 的位点

plink --file 269_indel3 --allow-extra-chr --maf 0.05 --recode --out 269_indel4

#7.将 plink 文件转化为 vcf 后进行基因型填充

plink --file 269_indel4 --allow-extra-chr --recode vcf-iid --out 269_indel_GENO_I

#8.填充

Java -Xmx512g -jar beagle.22Jul22.46e.jar gt=269_indel_GENO_I.vcf out=imputed_269_indel nthreads=64

beagle.22Jul22.46e.jar 见：LGB\LGB-main\beagle.22Jul22.46e.jar

#8.解压 imputed_269_indel.vcf.gz

gunzip imputed_269_indel.vcf.gz

#9.对 vcf 文件转化为 012 文件(raw 格式)

plink --vcf imputed_269_indel.vcf --double-id --recodeA --allow-extra-chr --out 269_indel_geno0.1_maf0.05_mind0.2_imputed_01

清洗完成后，从得到的 raw 文件中筛选出 SVR\SVR-mian 中对应的位点，使用对应的 module 文件进行预测即可。（以下是范例展示）

import numpy as np

from sklearn.metrics import mean_squared_error

```
from scipy.stats import pearsonr
from joblib import load
import pandas as pd

# 定义计算相对误差的函数
def calculate_relative_error(y_true, y_pred):
    """
    计算相对误差
    :param y_true: 真实值
    :param y_pred: 预测值
    :return: 相对误差数组
    """
    return np.abs((y_true - y_pred) / y_true)

try:
    # 加载 SVR 模型

    svr_model=load("/vochome/ZTY/jupyter/work/tao_ML/SVR/model/tao_aparted_indel_m
    odel/best_svr_model_malic_acid_indel.joblib")
    # 加载特征归一化器

    feature_scaler=load("/vochome/ZTY/jupyter/work/tao_ML/SVR/model/tao_aparted_inde
    l_model/scaler_X_malic_acid_indel.joblib")
    # 加载目标值归一化器

    target_scaler=load("/vochome/ZTY/jupyter/work/tao_ML/SVR/model/tao_aparted_indel
    _model/scaler_y_malic_acid_indel.joblib")
except FileNotFoundError:
    print("未找到模型或归一化器文件，请检查文件路径。")
    raise
```

```
# 这里需要你替换为实际的测试数据
# X_test 是特征数据，y_test 是真实目标值数据
# 示例数据（请替换）
X_test=pd.read_csv("/vochome/ZTY/jupyter/work/tao_ML/indel/indel_train_data/ZJ_apart_from_tao_indel.csv")
Y_test = pd.read_csv("/vochome/ZTY/jupyter/work/tao_ML/pheno/ZJ_pheno.csv")
y_test = Y_test["malic_acid"]
X_test = X_test.iloc[:,2:]
# 对测试集特征进行归一化
X_test_normalized = feature_scaler.transform(X_test)

# 使用 SVR 模型进行预测
y_pred_normalized = svr_model.predict(X_test_normalized)

# 对预测值进行反归一化
y_pred = target_scaler.inverse_transform(y_pred_normalized.reshape(-1, 1)).flatten()
print(y_pred)

# 计算相对误差
relative_errors = calculate_relative_error(y_test, y_pred)
mre = np.mean(relative_errors)
print("malic_acid 的相对误差:", mre)

# 计算均方误差
mse = mean_squared_error(y_test, y_pred)
print("malic_acid 的均方误差（MSE）:", mse)

# 计算皮尔逊相关系数
pearson_corr, _ = pearsonr(y_test, y_pred)
print("malic_acid 的皮尔逊相关系数:", pearson_corr)
```