DNNGP 使用手册

DNNGP, a deep neural network-based method for genomic prediction using multi-omics data in plants

Authors: Kelin Wang, Muhammad Ali Abid, Awais Rasheed, Jose Crossa, Sarah Hearne, **Huihui Li***

版本 2.0

编码: UTF-8

2023-02-04

许可协议: GUN, GPLv3

引用: Wang K, Abid MA, Rasheed A, Crossa J, Hearne S, Li H. DNNGP, a deep neural network-based method for genomic prediction using multi-omics data in plants. Mol Plant. 2023 Jan 2;16:279-293.

Doi: 10.1016/j.molp.2022.11.004, PMID: 36366781

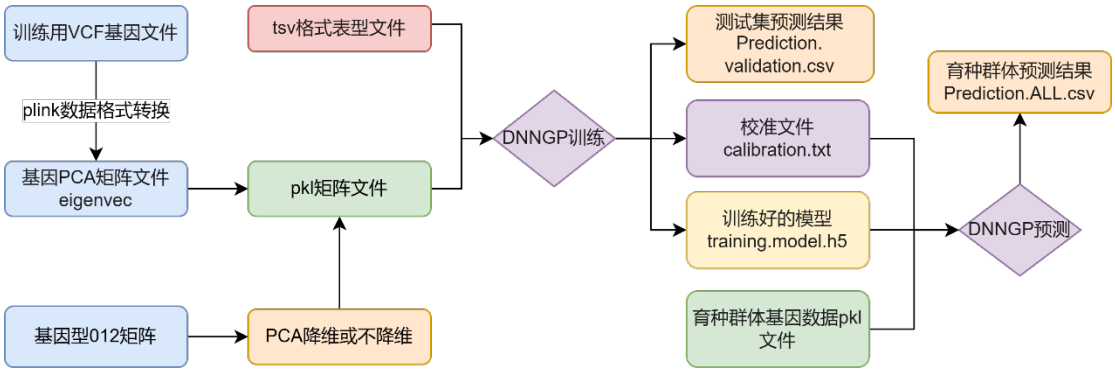
联系我们

李慧慧: lihuihui@caas.cn

目录

1. DNNGP 项目概述

DNNGP 是一个基于深度学习理论建立的全基因组预测模型，旨在利用全基因组标记预测植物和动物表型。此外，DNNGP 还可用于植物和动物的多组学数据预测。该模型主要使用 Python 3.6 和 TensorFlow 1.15 编写。DNNGP 的训练和预测过程如下所示:



1.1 项目地址：https://github.com/AIBreeding/DNNGP

1.2 文件目录结构

```
DNNGP:
│   CN 使用说明.pdf
│   DNNGP-usermanual.pdf
│   requirements.txt
│
├─Input_files
│       tsv2pkl.py
│       wheat1.tsv
│       wheat599_pc95.tsv
│       wheat599_pc95.pkl
│
├─Output_files
│       calibration.txt
│       Prediction.ALL.csv
│       Prediction.validation.csv
│       training.model.h5
│
├─Scripts_for_generating_training_model
│       config_dnngp.py
│       dnngp.cp36-win_amd64.pyd
│       dnngp.cpython-36m-darwin.so
│       dnngp.cpython-36m-x86_64-linux-gnu.so
│       dnngp_runner.py
│       run.py
```

```
│
└─Scripts_for_prediction
        config_dnngp.py
        dnngp.cp36-win_amd64.pyd
        dnngp.cpython-36m-darwin.so
        dnngp.cpython-36m-x86_64-linux-gnu.so
        dnngp_runner.py
        run.py
```

文件主要包含以下五部分：

(1) requirements.txt

用于环境搭建，环境配置所需的包及其版本。

(2) Input_files

该目录下为输入数据的示例文件。

(3) Scripts_for_generating training model

该目录下包含训练模型需要的脚本，训练完成后终端显示模型预测结果用于评估训练效果。同时输出两个文件：训练好的模型（training.model.h5）和测试集预测值（Prediction.validation.csv）。

(4) Scripts_for_prediction

该目录下包含模型预测需要的脚本。通过读取上一步训练好的模型，对育种群体表型进行预测，并输出所有个体的预测值（Prediction.ALL.csv）。

(5) Output_files

该目录下包括 DNNGP 方法执行输入示例文件后的输出文件。

模型的使用需遵照以下顺序进行：①搭建运行环境②准备数据③训练模型④使用训练模型进行预测

2. 数据准备

基于 plink2 软件的基因型数据处理

```
./plink2 --threads 30 --vcf *.vcf --pca 10 --out pca10
```

--threads 30 使用 30 个线程

--vcf *.vcf 读取 vcf 文件

--pca 10 取 PC1-PC10(可设定值≤样本个数≤8000)

--out pca10 输出文件名为 pca10

若存在非数字染色体编号则需添加--allow-extra-chr 参数

```
./plink2 --allow-extra-chr --threads 30 --vcf *.vcf --pca 10 --out pca10
```

结果会生成两个文件，后缀名分别为.eigenval 和.eigenvec，eigenval 显示了每个 PC 所占的比重，各个 PC 的比重/比重和为每个 PC 的解释度。eigenval 为我们需要使用的 PCA 矩阵。

提别提示：

(1) 以上命令适用于 windows 平台下的 Powershell 终端以及 Linux、Mac 终端。若在 windows 平台下的 cmd 终端使用请将./plink2 更换为 plink2。

(2) 若使用 PCA 手段对基因数据进行转换，需要首先将育种群体与训练群体的数据合并再进行 PCA 分析，得到 PCA 矩阵后再将二者分开。

3. DNNGP 环境搭建

(1) 下载项目地址：https://github.com/AIBreeding/DNNGP

(2) 运行 DNNGP 首先需要搭建运行环境:

首先安装：Miniconda（https://docs.conda.io/en/latest/miniconda.html）或 Anaconda（https://www.anaconda.com/）。

然后在安装后使用以下命令搭建运行环境：

- `conda create -n dnngp python=3.6.5`
- `conda activate dnngp`
- `cd dnngp`
- `conda install --yes --file requirements.txt`
- `pip install tensorflow-determinism==0.3.0`

4. 输入数据文件

在环境搭建后，需要按照示例数据格式准备各项数据文件，示例数据文件位于以下目录：

```
../dnngp/input_file/
```

其中包含以下四个文件:

- **wheat1.tsv**：以制表符分隔的表型数据文件。
- **wheat599_pc95.tsv**：以制表符分隔的主成分矩阵文件。
- **wheat599_pc95.pkl**：模型可读取的主成分矩阵文件。
- **tsv2pkl.py**：由 tsv 转为 pkl 文件的格式转换脚本。

其中，**wheat599_pc95.pkl** 文件可由 **wheat599_pc95.tsv** 文件通过运行脚本 **tsv2pkl.py** 转换而来。也可由 plink2 生成的 eigenval 文件通过 **tsv2pkl.py** 直接转换。

表型数据文本文件格式如下：

ID env1

M1 1.67162948

M2 -0.25270276

M3 0.341815127

M4 0.785439489

M5 0.998317613

M6 2.336096876

M7 0.617410817

主成分矩阵 tsv 文件格式如下：

ID	PC1	PC2	PC3	…
M1	7.0408269	2.053877771	-6.161150675	…
M2	5.924749016	1.137903031	1.132296531	…
M3	5.953045926	1.082444715	1.139961515	…

5. DNNGP 模型训练

该部分需要输入两个文件，即上一步准备完成的主成分矩阵文件以及表型数据文件。具体格式请参照上一部分说明。

参数说明：

--batch_size 训练模型所调用的样本量

--lr 初始学习率

--epoch 迭代次数

--dropout1 第一次特征抛弃（防止过拟合）

--dropout2 第二次特征抛弃（防止过拟合）

--patience 无提升阈值

--Seed 随机种子

--SNP 主成分矩阵文件路径

--pheno 表型数据文件路径

--output 输出目录

进入 Scripts_for_generating_training_model 目录示例命令：

cd Scripts_for_generating_training_model

训练模型示例命令：

python dnngp_runner.py --batch_size 28 --lr 0.001 --epoch 100 --dropout1 0.5 --dropout2 0.3 --patience 25 --seed 123 --SNP "../input_files/wheat599_pc95.pkl" --pheno "../input_files/wheat1.tsv" --output /Your_path/

训练模型输出文件

训练完成后会在指定输出目录下生成 3 个输出文件，分别是：

Prediction.validation.csv：DNNGP 模型对测试集的预测结果（第一列的序号代表预测值个体在原数据集中的名称）。

training.model.h5：训练好的模型文件，用于下一步对育种群体表型性状预测。

calibration.txt：与 h5 模型文件配套文件，内含模型训练集均值及标准差，用于下一步对育种群体表型性状预测。

训练完成后，终端显示预测值与真实值之间的 Pearson 相关系数，如下所示:

'Corrobs vs pred =', (0.582, 0.001)

第一个数字是相关系数（0.582），第二个数字是 *p* 值（0.001）。

6. 使用训练好的模型对待测数据进行预测

在得到训练好的模型文件后，我们要对预测模型文件夹内的待测数据集（即育种群体）表型性状进行预测。该部分需要三个输入文件，其中两个是上一步训练生成的模型文件即 **calibration.txt** 和 **training.model.h5**，第三个是育种群体主成分矩阵文件（*.pkl），格式与上一步训练模型时所用文件格式相同。

进入 Scripts_for_prediction 目录示例命令：

cd Scripts_for_prediction

预测育种群体表型性状示例命令：

python dnngp_runner.py --Model "/Your_path/training.model.h5" --SNP "/Your_path/wheat599_pc95.pkl" --cal "/Your_path/calibration.txt" --output

/Your_path/

DNNGP 预测参数说明：

--Model：训练模型时生成的.h5 模型文件路径

--SNP：待预测数据集的基因数据文件路径

--cal：训练时生成的校准文件

--output：预测结果文件的生成目录

模型预测输出文件

DNNGP 模型完成预测后将在指定目录下生成结果文件 **Prediction.ALL.csv**，该文件即是对育种群体所有个体的表型性状预测结果。

7. 特别说明：

训练和预测的两个文件夹内均有名为 run.py 的脚本进行批量测试。

运行示例命令: python run.py

在线版使用说明：https://www.wolai.com/xgq9jknGjCApQqs7xQiaTd