

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 3.0

编码: UTF-8

2023-05-12

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](https://doi.org/10.1016/j.molp.2022.11.004), PMID: [36366781](https://pubmed.ncbi.nlm.nih.gov/36366781/)

联系我们

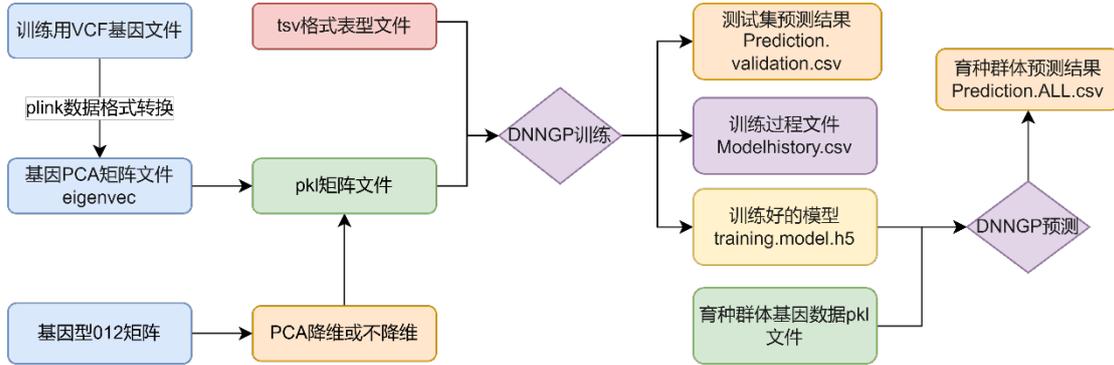
李慧慧: lihuihui@caas.cn

目录

1. DNNGP 项目概述	- 1 -
1.1 项目地址	- 1 -
1.2 文件目录结构	- 1 -
2. 数据准备	- 2 -
3. DNNGP 环境搭建	- 3 -
4. 输入数据文件	- 3 -
5. DNNGP 模型训练	- 5 -
6. 使用训练好的模型对待测数据进行预测	- 6 -
7. 特别说明	- 7 -

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 3.9.16 和 TensorFlow 2.6.0 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:

```
| double_click_me_first.bat
| Tsv2Pkl.exe
| Start_DNNGP.bat
| CN 使用说明.pdf
| EN usermanual.pdf
| requirements.txt
|
|---Input_files
|     wheat1.tsv
|     wheat599_pc95.pkl
|     wheat599_pc95.tsv
|
|---Output_files
|     Modelhistory.csv
|     Prediction.ALL.csv
|     Prediction.validation.csv
|     training.model.h5
|
|---Scripts
|     config_dnngp.cp39-win_amd64.pyd
|     D.ico
|     dnngp.cp39-win_amd64.pyd
|     DNNGP.exe
```

dnngp_runner.py
environment.yaml
Pre-Batch_run.py
Pre_config_dnngp.cp39-win_amd64.pyd
Pre_dnngp.cp39-win_amd64.pyd
Pre_runner.py
Train-Batch_run.py

文件主要包含以下五部分：

(1) GUI 文件

该部分文件主要包括三个文件，分别为：[double_click_me_first.bat](#)、[Tsv2Pkl.exe](#)、[Start_DNNGP.bat](#)。在 Windows 平台下，双击 [double_click_me_first.bat](#) 即可进行环境的搭建。双击 [Tsv2Pkl.exe](#) 根据 GUI 提示可以将 tsv 文件转换为 pkl 文件。而后双击 [Start_DNNGP.bat](#) 可以进行模型的训练和预测。

(2) requirements.txt / environment.yaml

用于环境搭建，环境配置所需的包及其版本。

(3) Input_files

该目录下为输入数据的示例文件。

(4) Scripts

该目录下包含训练模型及预测需要的脚本。

训练模型：完成后终端显示模型预测结果用于评估训练效果。同时输出三个文件：训练好的模型（training.model.h5）和验证集预测值（Prediction.validation.csv）以及训练过程中每一个 epoch 的各项数值（Modelhistory.csv）。

模型预测：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 \leq 样本个数 \leq 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>)，并将其添加进系统环境。(安装 miniconda 时可以勾选，使用 GUI 所必须的。)

Windows 平台下可以直接双击 **Double_click_me_first.bat** 文件实现一键搭建环境。

若一键搭建环境失败，则使用以下命令搭建运行环境：

```
conda create -n DNNGP3 python=3.9.16
```

```
conda activate DNNGP3
```

```
cd dnngp
```

```
conda install --yes --file requirements.txt
```

```
conda install -c nvidia cuda-nvcc
```

```
pip install framework-reproducibility==0.4.0
```

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 **tsv2pkl.py** (**tsv2pkl.exe**) 转换而来。也可由 plink2 生成的 **eigenval** 文件通过 **tsv2pkl.py** 直接转换。

转换方式如下

打开 **tsv2pkl.py** 后修改第五行 **inpath**、以及第六行 **outpath** 内的文件路径为自己的文件路径即可。然后在 conda 创建的 DNNGP3 环境中运行 **tsv2pkl.py** 即可。

`python tsv2pkl.py`

Windows 平台下, 可以双击 DNNGP 目录下的 **tsv2Pkl.exe** 程序使用 GUI 界面完成文件格式的转换。

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（Learning rate）

--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 无提升则减小学习率阈值（当模型在无提升次数抵达阈值时，自动降低学习率）

--seed 随机种子（Random seed）

--cv 设定交叉验证折数

--part 设定选取第几折数据为验证集

--earlystopping 无提升则停止训练阈值（当模型在无提升次数抵达阈值时，自动停止训练，并保存最佳参数）

--snp 主成分矩阵文件路径

--pheno 表型数据文件路径

--output 输出目录

以上参数除--snp、--pheno 以及--output 外，均为数值型参数。

进入 Scripts 目录示例命令：

```
cd Scripts
```

训练模型示例命令：

```
python dnnngp_runner.py --batch_size 28 --lr 0.001 --epoch 100 --dropout1 0.5  
--dropout2 0.3 --patience 5 --seed 123 --cv10 --part 1 --earlystopping 10 --snp  
"./input_files/wheat599_pc95.pkl" --pheno  "./input_files/wheat1.tsv" --output  
"/Your_path/"
```

Windows 平台下可以直接双击 DNNGP 目录下的 **Start_DNNGP.bat** 启动 GUI 后

根据 GUI 提示进行操作。GUI 右侧会输出标准命令以及训练过程。

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 模型对验证集的预测结果（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。

Modelhistory.csv: 记录了训练过程中，各项数值的变化情况。

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，我们要对待测数据集（即育种群体）表型性状进行预测。该部分需要两个输入文件，一个是上一步训练生成的模型文件即 **training.model.h5**，第二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。

预测育种群体表型性状示例命令：

```
python Pre_runner.py --Model "/Your_path/training.model.h5" --SNP "/Your_path/wheat599_pc95.pkl" --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

--output: 预测结果文件的生成目录

Windows 平台下可以通过双击 DNNGP 目录下的 **Start_DNNGP.bat** 启动 GUI 界面，然后根据 GUI 提示进行操作。

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:

Script 目录下含有名为 Train-Batch_run.py 和 Pre-Batch_run.py 的 Python 脚本可以进行批量测试, 其中前者为训练模型批量脚本后者为模型预测批量脚本。

运行示例命令:

```
python Train-Batch_run.py 或 python Pre-Batch_run.py
```