DNNGP 使用(包含 vcf2tsv/PCA)

DNNGP 的使用

> wheat1.tsv 制表符分割的表型文件，第一列是id,第2列是表型值
> wheat599_pc95.tsv 制表符分割的主成分矩阵文件，第一列是id,后面是pc1到pc251列
> wheat599_pc95.pkl是模型可读取的主成分矩阵文件

简 使用DNNGP进行基因组表型预测分析

```
plink2 --threads 30 --vcf test.vcf --pca 200 --out pca200 --allow-extra-chr
cat pca200.eigenvec|sed 's/#IID/ID/' >pca200.tsv
#这里是使用前300行作为训练的样本
head -301 pca200.tsv >pca200.train.tsv

#最后的20行作为要预测的样本
cat <(head -1 pca200.tsv) <(tail -20 pca200.tsv) >pca200.predict.tsv
tsv2pkl.py pca200.train.tsv pca200.train.pkl
tsv2pkl.py pca200.predict.tsv pca200.predict.pkl
```

进行主成分分析

~~plink --vcf combine_all.pass.recode.clear2.vcf --allow-extra-chr --chr-set 13 --out pca_dnngp~~

用全部 SNP 为 PCA 参数进行分析，计算每一个位点的贡献率，然后根据贡献率确定总贡献率在 80/90% 的 PCA 结果筛选出来。

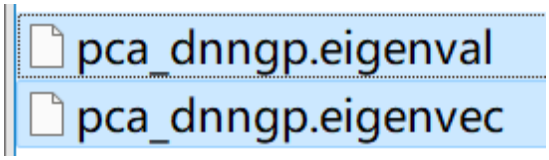
首先确定干净的 VCF 文件中的 SNP 数据信息：

bcftools +counts combine_all.pass.recode.clear2.vcf

```
Number of samples: 260
Number of SNPs:    5327033
Number of INDELs:  653986
Number of MNPs:    0
Number of others:  0
Number of sites:   5981019
```

plink --vcf combine_all.pass.recode.clear2.vcf --pca 260 --out pca_dnngp

```
PLINK v1.90b6.24 64-bit (6 Jun 2021)           www.cog-genomics.org/plink/1.9/
(C) 2005-2021 Shaun Purcell, Christopher Chang   GNU General Public License v3
Logging to pca_dnngp.log.
Options in effect:
  --out pca_dnngp
  --pca 260
  --vcf combine_all.pass.recode.clear2.vcf

2063877 MB RAM detected; reserving 1031938 MB for main workspace.
--vcf: pca_dnngp-temporary.bed + pca_dnngp-temporary.bim +
pca_dnngp-temporary.fam written.
5981019 variants loaded from .bim file.
260 people (0 males, 0 females, 260 ambiguous) loaded from .fam.
Ambiguous sex IDs written to pca_dnngp.nosex .
Using up to 95 threads (change this with --threads).
Before main variant filters, 260 founders and 0 nonfounders present.
Calculating allele frequencies... done.
Total genotyping rate is 0.978148.
5981019 variants and 260 people pass filters and QC.
Note: No phenotypes present.
Relationship matrix calculation complete.
--pca: Results saved to pca_dnngp.eigenval and pca_dnngp.eigenvec .
```



wc -l pca_dnngp.eigenval

wc -l pca_dnngp.eigenvec

```
(dnngp) (dnngp)[root@f41ac1b8282f GWAS]$wc -l  pca_dnngp.eigenval
260 pca_dnngp.eigenval
(dnngp) (dnngp)[root@f41ac1b8282f GWAS]$wc -l pca_dnngp.eigenvec
260 pca_dnngp.eigenvec
```

进入 R-last 虚拟环境用 R 进行可视化处理

```
conda activate R-last
```

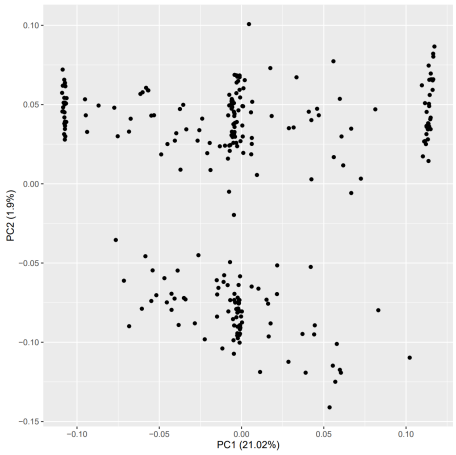
进入 R

```
install.packages("tidyverse")

library(tidyverse)

library(data.table)

library(ggplot2)

re1a = fread("pca_dnngp.eigenval")

re1b = fread("pca_dnngp.eigenvec")

re1a$por = re1a$V1/sum(re1a$V1)*100

#计算每一行的贡献率
```

```
head(re1a)
#查看前六行
pdf("pca_dnngp.pdf")
ggplot(re1b,aes(x = V3,y = V4)) + geom_point() +  xlab(paste0("PC1 (",round(re1a$por[1],2),"%")) +  ylab(paste0("PC2 (",round(re1a$por[2],2),"%"))
dev.off()
#ggplot作图；必须关闭图形设备
```

```
> re1a
             V1
          <num>
  1: 54.6971000
  2:  4.9409200
  3:  2.8676300
  4:  2.4996600
  5:  2.2846800
 ---
256:  0.2507860
257:  0.1876880
258:  0.1631740
259:  0.1576940
260: -0.0248097
```

```
> head(re1a)
          V1        por
       <num>      <num>
1: 54.69710 21.0216444
2:  4.94092  1.8989355
3:  2.86763  1.1021114
4:  2.49966  0.9606901
5:  2.28468  0.8780672
6:  2.11558  0.8130773
```




```
> re1b
         V1     V2          V3        V4          V5          V6          V7
     <char> <char>       <num>     <num>       <num>       <num>       <num>
  1: H-R100 H-R100 -0.00546927 0.0523900  0.02005600 -0.11061600 -0.00162475
  2: H-R101 H-R101  0.05975530 0.0535994 -0.09027310  0.10740500 -0.00757271
  3: H-R102 H-R102 -0.00115664 0.0683465 -0.00242934  0.00868491 -0.05120150
  4: H-R103 H-R103  0.03342610 0.0672050  0.00153493  0.03325410 -0.08707590
  5: H-R104 H-R104  0.01832650 0.0427649 -0.01303270  0.00827996 -0.08166250
 ---
256:  MY-R5  MY-R5 -0.10729800 0.0392175 -0.01609560 -0.00237430 -0.01191690
257:  MY-R6  MY-R6 -0.10632600 0.0540608 -0.01144950 -0.02093540 -0.01618490
258:  MY-R7  MY-R7 -0.10833400 0.0456683 -0.06694180  0.02094940 -0.03017740
259:  MY-R8  MY-R8 -0.09464060 0.0432072 -0.04783060  0.00936645 -0.03645020
260:  MY-R9  MY-R9 -0.10764700 0.0419035 -0.05834670  0.00390889 -0.03209220
```

抽取数据制成tsv文件

```
cat pca_dnngp.eigenvec|sed 's/#IID/ID/' >pca_dnngp.tsv
less pca_dnngp.tsv
```

```
H-R100 H-R100 -0.00546927 0.05239 0.020056 -0.110616 -0.00162475 -0.0263933 0.0820683 0.035385 0.0539373 -0.0212531 -0.0567311 -0.00988083 -0.0620311 0.0903727 0.0539427 0.0498481 -0.00393468 -0.0295718 0.0174028 0.0229249 -0.0315181 -0.0366846 0.0796967 -0.0220976 -0.0551961 0.0306123 0.06468 0.0933791 0.0674141 0.0649669 0.146775 0.052638 -0.111696 -0.124465 -0.104119 0.0616419 -0.0683974 0.0706793 0.0245782 0.0299775 0.0671308 0.0372301 0.0136201 -0.100509 -0.0120269 -0.046847 0.0206612 0.0522251 -0.00786859 -0.0310126 0.0375088 -0.0672275 0.0513175 0.0153436 0.0138052 0.0742878 0.177839 -0.048776 -0.00552404 0.0577788 -0.0459991 -0.0207598 -0.0871087 -0.154001 0.0477101 0.0704692 0.0557432 -0.128608 -0.0217283 -0.0107278 0.0020665 0.0171241 -0.0463296 -0.0919532 -0.0665905 -0.00646526 0.000653644 0.00354732 -0.0448597 0.0720311 0.126872 -0.0117231 -0.0211258 -0.0336532 0.0141432 0.000407891 -0.0239794 -0.0733524 -0.018303 -0.104168 -0.0163526 -0.0947239 -0.0519043 0.000597752 0.0538603 0.010424 0.00640357 -0.0398819 0.113613 0.0318976 0.0440178 0.0240458 -0.105269 0.050112 0.0557961 0.0261419 0.0293915 -0.0993053 0.0626324 0.0760204 0.0102893 -0.0744077 -0.039905 -0.0737287 0.087997 0.0951761 -0.0432655 0.00238211 0.0330712 -0.0132257 0.0709834 0.0218239 0.0230894 0.0720939 -0.101133 0.0500217 0.0951352 -0.0314451 -0.0107883 -0.156247 -0.106164 0.00613543 0.0167033 0.158141 -0.0290284 0.0222601 -0.0225459 0.0884473 -0.025584 -0.0956615 0.123229 0.0813912 -0.0742897 0.051974 -0.0437978 -0.113849 0.00342678 -0.0551446 -0.143586 -0.0077099 0.0343758 -0.0279421 -0.0605051 0.00619933 -0.0372315 -0.0504483 0.0249574 0.0021252 0.170075 -0.0809298 0.115544 -0.0543537 -0.23033 0.157029 0.0587271 0.0104475 -0.0130047 0.136598 0.0333755 -0.180194 -0.0079871 -0.0330858 -0.148336 -0.0242358 -0.0860595 0.0233698 -0.104595 -0.0679935 0.0269196 -0.0102524 0.0751548 0.129824 0.0066044 0.111779 -0.0773644 0.0535369 0.0710476 0.0390964 0.0226425 -0.119333 -0.0500715 -0.0235118 0.00606679 -0.0228306 0.0436823 0.0143106 0.0168647 -0.0293317 0.0304925 0.0496196 -0.015199 -0.0117011 -0.0255508 0.0603663 0.00725689 0.057939 0.0232422 -0.0105211 -0.0028136 -0.00265957 0.0359195 0.0286256 -0.0227329 0.0146969 -0.027572 -0.0267077 0.03572 0.00379841 0.0110926 0.0101492 0.00435793 -0.0295048 -0.0011516 -0.0407509 -0.00208391 0.0373101 -0.021417 0.0790198 -0.0217412 -0.00268063 -0.0279456 -0.0333386 -0.0233709 -0.00663744 -0.009199 -0.0127107 -0.00650691 0.019943 0.00265419 0.0173112 0.0176964 -0.0285574 0.0173847 0.00642093 -0.0122785 0.00200965 -0.00964988 -0.0380955 0.00922971 -0.0121921 0.0288268 -0.00404203 0.000262964 0.0123913 0.0074386 -0.00212834 0.00135588 0.000448566 0.000646794 -0.0619167
H-R101 H-R101 0.0597553 0.0535994 -0.0902731 0.107405 -0.00757271 -0.127841 0.0340962 -0.105407 -0.0732273 -0.0428362 -0.108927 0.0192387 -0.0118615 0.072171 -0.0945872 -0.103545 -0.104656 -0.0311859 -0.0772965 0.110029 0.0464254 0.0405156 -0.00549833 0.000191647 0.0294838 0.0159456 0.00741809 -0.0229731 -0.0239779 -0.303096 0.151677 0.0282908 0.00670953 -0.0438111 -0.0706941 0.142123 0.0632168 0.125668 0.085497 -0.102009 0.0231456 0.120642 0.000141054 -0.0225045 0.0927876 0.0322805 0.000752152 -0.21452 0.255127 -0.0549338 0.0378468 0.0932217 -0.0476844 0.0136841 -0.0466187 -0.0306451 -0.0400359 -0.0337052 -0.185104 0.0542809 0.0488451 0.00573296 0.00418005 -0.0104062 -0.0893387 -0.154111 0.0938218 0.0406279 0.0174657 0.096191 0.0264369 0.062207 0.105747 0.133846 0.100404 -0.188072 -0.0143771 0.0759345 -0.0750815 0.0951382 -0.054224 -0.00856849 -0.116513 -0.15174 -0.0647149 -0.121136 -0.0979221 -0.0783738 -0.00958545 0.0292831 -0.0405611 -0.171673 0.0677091 -0.0801312 -0.00206662 -0.100682 -0.0427579 0.0324311 0.0298683 0.0452759 0.0107154 0.110942 -0.0527914 -0.00405612 0.012405 0.0286988 -0.0190918 0.0971028 -0.00902566 0.0103736 -0.0777372 -0.054914 0.064201 -0.0579918 -0.0658733 0.0462254 -0.0119974 0.047853 -0.0615116 -0.091875 0.0312393 0.0223418 0.0362007 0.0411937 -0.0516359 0.00575992 0.0579689 -0.0709757 0.045679 -0.0651314 0.0610249 0.0864241 0.0317271 0.0165375 0.0616904 -0.0376084 -0.0216814 -0.080043 -0.00441022 -0.0394766 -0.000568311 -0.0451406 0.0270694 -0.0185446 0.059476 0.0696011 -0.0115881 0.0508356 0.00246265 -0.00514935 0.0439965 -0.0180392 -0.0478026 -0.00144356 -0.028122 0.0283998 0.0256769 0.0299288 -0.0524052 -0.00445141 -0.0454471 0.0428371 0.00398686 -0.0330011 -0.0112651 0.0331011 0.0170471 0.000560791 0.0485495 -0.0373718 -0.0306116 0.0289646 -0.0956063 -0.0258882 0.00274616 0.046324 -0.00974111 -0.0553271 0.0221742 0.0352813 0.0228651 -0.0155118 0.00686651 -0.0264111 -0.0035613 -0.00255281 0.0177995 -0.0319534 -0.0
```


ID 生成了两个，具体生成的步骤不确定尚不确定（eigenvec 文件中就已经存在）


```
H-R100 H-R100 -0.00546927 0.0
H-R101 H-R101 0.0597553 0.053
H-R102 H-R102 -0.00115664 0.0
H-R103 H-R103 0.0334261 0.067
H-R104 H-R104 0.0183265 0.042
H-R105 H-R105 0.0666951 -0.00
H-R106 H-R106 0.0813193 0.046
H-R107 H-R107 0.0316808 0.035
H-R108 H-R108 -0.0188515 0.00
H-R109 H-R109 -0.0256464 0.03
H-R110 H-R110 -0.00607717 0.0
```


在 R 中进行修改：

```
pca<-read.table("pca_dnngp.tsv")
row.names<-pca[,1]
pca<-pca[,-2]
row.names(pca)<-row.names
write.table(pca,file="pca_dnngp2.tsv")
```

手动修改后（**制表符分隔！**）：

ID	pca1	pac2	pac3	pac4	pac5	pac6	pac7
H-R100	-0.00546927	0.05239	0.020056		-0.110616	-0.0016	
H-R101	0.0597553		0.0535994	-0.0902731	0.107405		-0.0
H-R102	-0.00115664	0.0683465		-0.00242934	0.00868491		-0.0
H-R103	0.0334261		0.067205	0.00153493	0.0332541		-0.0
H-R104	0.0183265		0.0427649	-0.0130327	0.00827996		-0.0
H-R105	0.0666951		-0.00583012	0.0411324	0.0630466		-0.1
H-R106	0.0813193		0.0469828	-0.0210424	-0.0592962		-0.1
H-R107	0.0316808		0.0355317	-0.03183	-0.00736309		-0.0
H-R108	-0.0188515		0.0086684	-0.0704498	0.0284902		-0.0
H-R109	-0.0256464		0.0338273	-0.102189	0.00398593		-0.0
H-R110	-0.00607717	0.0306336		-0.0316994	-0.0204654		-0.0
H-R111	-0.00681625	0.0244502		0.0361667	-0.0139179		-0.0
H-R112	-0.0058513		0.0328701	0.0535765	-0.00551242		0.00
H-R113	-0.00187538	0.046336		0.0396821	-0.0357737		0.00
H-R114	-0.0100598		0.0240641	0.0301196	-0.0586852		0.01
H-R115	-0.00453019	0.0315687		0.0314879	-0.0253707		0.00
H-R116	-0.0046328		0.0454764	0.0429094	-0.046908		0.01
H-R117	-0.00825434	0.0158801		0.0772168	-0.000427118		
H-R118	-0.00205514	0.0648565		0.015648	-0.128262		0.04
H-R119	-0.00543946	0.029527		0.0440194	0.0105223		-0.0
H-R120	-0.00494213	0.025853		0.0647279	0.0112097		-0.0
H-R121	-0.00321217	0.0258412		-0.0269234	-0.00845639		-0.0
H-R122	0.0607816		0.0298604	0.102678	0.122007		0.01
H-R123	-0.00745967	0.0241223		-0.0208237	-0.00477264		0.00

以前 20 个为预测样本，后 240 个(包含三个群体)为训练样本：

~~head -20 pca_dnngp.tsv >pca_dnngp.predict.tsv~~

~~cat <(tail -240 pca_dnngp.tsv) >pca_dnngp.train.tsv~~

~~python3 tsv2pkl.py pca_dnngp.predict.tsv pca_dnngp.predict.pkl~~

~~python3 tsv2pkl.py pca_dnngp.train.tsv pca_dnngp.train.pkl~~

(格式修改后的)

```
head -21 pca_dnngp2.tsv >pca_dnngp2.predict.tsv
```

```
cat  <(head -1 pca_dnngp2.tsv) <(tail -240 pca_dnngp2.tsv) >pca_dnngp2.train.tsv
```

```
python3 tsv2pkl.py pca_dnngp2.predict.tsv pca_dnngp2.predict.pkl
```

```
python3 tsv2pkl.py pca_dnngp2.train.tsv pca_dnngp2.train.pkl
```

表型文件的准备

（图为表型 tsv 文件的格式，第一列是 ID，第 2 列是表型值，空格间隔）

可以直接提取每个表型单列 csv 后手动改为 tsv，并将其中的","批量替换为空格

注意：需要提取与基因型数据相对应的样本作为训练表型！(最右边为正确的分隔)

~~cat <(tail -240 260sample_flowerNUM.tsv) >260sample_flowerNUM.train.tsv~~

```
cat  <(head -1 260sample_flowerNUM.tsv) <(tail -240 260sample_flowerNUM.tsv) >260sample_flowerNUM.train.tsv
```

1	H-R100 13
2	H-R101 22
3	H-R102 10
4	H-R103 9
5	H-R104 10
6	H-R105 16
7	H-R106 8
8	H-R107 4
9	H-R108 15
10	H-R109 12
11	H-R110 8
12	H-R111 12
13	H-R112 10
14	H-R113 7
15	H-R114 11
16	H-R115 9

1	H-R100 39
2	H-R101 42
3	H-R102 37.46
4	H-R103 36.3
5	H-R104 28.8
6	H-R105 31
7	H-R106 41
8	H-R107 44
9	H-R108 46.5
10	H-R109 37.3
11	H-R110 39.2
12	H-R111 44.6
13	H-R112 40
14	H-R113 38
15	H-R114 37
16	H-R115 40.6

ID	flowerNUM
H-R100	13
H-R101	22
H-R102	10
H-R103	9
H-R104	10
H-R105	16
H-R106	8
H-R107	4
H-R108	15
H-R109	12
H-R110	8
H-R111	12
H-R112	10
H-R113	7
H-R114	11
H-R115	9

DNNGP 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

--dropout2 第 2 次特征抛弃

--patience 无提升则减小学习率阈值

--seed 随机种子

--cv 交叉验证折数

--part 设定选取第几折数据作为验证集

--earlystopping 无提升则停止训练阈值，

--snp 训练集的基因型文件，格式是 pkl

--pheno 表型文件，第一列是 id, 第 2 列是表型值

--output 输出文件夹

训练数据集

需要启动 conda 的 DNNGP 所在虚拟环境！

mkdir DNNGP_TEST

cd /root/home/liuqirui/SNPdata/DNNGP-main

~~python3 Scripts/dnngp_runner.py --batch_size 20 --lr 0.001 --epoch 100 --dropout1 0.5 --dropout2 0.3 --patience 5 --seed 2024 --cv 10 --part 1 --earlystopping 10 --snp "/root/home/liuqirui/SNPdata/GWAS/pca_dnngp.train.pkl" --pheno~

~~"/root/home/liuqirui/SNPdata/260sample_flowerNUM.train.tsv" --output /root/home/liuqirui/SNPdata/DNNGP_TEST~~

python3 Scripts/dnngp_runner.py --batch_size 20 --lr 0.001 --epoch 100 --dropout1 0.5 --dropout2 0.3 --patience 5 --seed 2024 --cv 10 --part 1 --earlystopping 10 --snp "/root/home/liuqirui/SNPdata/GWAS/pca_dnngp2.train.pkl" --pheno "/root/home/liuqirui/SNPdata/GWAS/260sample_flowerNUM.train.tsv" --output /root/home/liuqirui/SNPdata/DNNGP_TEST

```
------------- Options information ---------------
Batch_size: 20
Epochs (niters) : 100
Learning rate : 0.001
Patience : 5
Dropout1 : 0.5
Dropout2 : 0.3
Earlystopping : 10
CV : 10
Part : 1
Random seed : 2024
SNP : /root/home/liuqirui/SNPdata/GWAS/pca_dnngp.train.pkl
Pheno : /root/home/liuqirui/SNPdata/260sample_flowerNUM.train.tsv
Output : /root/home/liuqirui/SNPdata/DNNGP_TEST


---------- Operating device information ----------
('Num CPUs Available: ', 1)
Use CPU
-------------------------------------------------
```

结果（**注意，这部分结果是用了双 ID 的错误格式 TSV 的结果**）：

```
Traceback (most recent call last):
  File "/root/home/liuqirui/SNPdata/DNNGP-main/Scripts/dnngp_runner.py", line 24, in <module>
    dnngp.main(SNP, pheno, batch_size, lr, epoch, patience, dropout1, dropout2, output, SEED, CV, part, NMearlystopping)
  File "dnngp.pyx", line 138, in dnngp.main
  File "dnngp.pyx", line 108, in dnngp.dnngp_model
  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/utils/traceback_utils.py", line 70, in error_handler
    raise e.with_traceback(filtered_tb) from None
  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/tensorflow/python/eager/execute.py", line 53, in quick_execute
    tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name,
tensorflow.python.framework.errors_impl.InvalidArgumentError: Graph execution error:
```

```
Detected at node sub defined at (most recent call last):
  File "/root/home/liuqirui/SNPdata/DNNGP-main/Scripts/dnngp_runner.py", line 24, in <module>

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/utils/traceback_utils.py", line 65, in error_handler

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/engine/training.py", line 1807, in fit

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/engine/training.py", line 1401, in train_function

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/engine/training.py", line 1384, in step_function

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/engine/training.py", line 1373, in run_step

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/engine/training.py", line 1155, in train_step

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/engine/training.py", line 1249, in compute_metrics

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/engine/compile_utils.py", line 620, in update_state

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/utils/metrics_utils.py", line 77, in decorated

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/metrics/base_metric.py", line 140, in update_state_fn

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/metrics/base_metric.py", line 723, in update_state

  File "/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/losses.py", line 1848, in mean_absolute_error

Incompatible shapes: [1280,1] vs. [20,0]
         [[{{node sub}}]] [Op:__inference_train_function_2936]
```

存在错误，并没有输出。

修改后TSV数据格式以后的执行：

```
  import config_dnngp, dnngp
2024-03-11 03:53:26.656183: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register cuDNN factory: Attemp
ting to register factory for plugin cuDNN when one has already been registered
2024-03-11 03:53:26.656382: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register cuFFT factory: Attempt
ing to register factory for plugin cuFFT when one has already been registered
2024-03-11 03:53:26.660612: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuBLAS factory: Atte
mpting to register factory for plugin cuBLAS when one has already been registered
2024-03-11 03:53:29.324309: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT


------------- Options information ---------------
Batch_size: 20
Epochs (niters) : 100
Learning rate : 0.001
Patience : 5
Dropout1 : 0.5
Dropout2 : 0.3
Earlystopping : 10
CV : 10
Part : 1
Random seed : 2024
SNP : /root/home/liuqirui/SNPdata/GWAS/pca_dnngp2.train.pkl
Pheno : /root/home/liuqirui/SNPdata/GWAS/260sample_flowerNUM.train.tsv
Output : /root/home/liuqirui/SNPdata/DNNGP_TEST


---------- Operating device information ----------
('Num CPUs Available: ', 1)
Use CPU
-------------------------------------------------
Epoch 1/100
```

这一次成功了：

```
Epoch 38: ReduceLROnPlateau reducing learning rate to 1e-05.
Restoring model weights from the end of the best epoch: 28.
11/11 - 1s - loss: 26.7696 - mae: 4.2019 - mse: 26.6933 - val_loss: 67.8131 - val_mae: 7.3359 - val_mse: 67.7367 - lr: 1.0000e-05 -
540ms/epoch - 49ms/step
Epoch 38: early stopping
1/1 [==============================] - 1s 1s/step
/root/anaconda3/envs/dnngp/lib/python3.9/site-packages/keras/src/engine/training.py:3103: UserWarning: You are saving your model as
an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `mo
del.save('my_model.keras')`.
  saving_api.save_model(


-------------------- Result --------------------
('Corr obs vs pred =', PearsonRResult(statistic=-0.5191351108598727, pvalue=0.009335346882844852))
('Prediction validation save in:', '/root/home/liuqirui/SNPdata/DNNGP_TEST/Prediction.validation.csv')
('Model save in:', '/root/home/liuqirui/SNPdata/DNNGP_TEST/training.model.h5')
('Model history save in:', '/root/home/liuqirui/SNPdata/DNNGP_TEST/Modelhistory.csv')
```

('Corr obs vs pred =', PearsonRResult(statistic=-0.5191351108598727, pvalue=0.009335346882844852))

Pearson 相关系数 statistic=-0.5191351108598727，
pvaluepvalue=0.0093353468828448 52。

('Prediction validation save in:',
'/root/home/liuqirui/SNPdata/DNNGP_TEST/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）

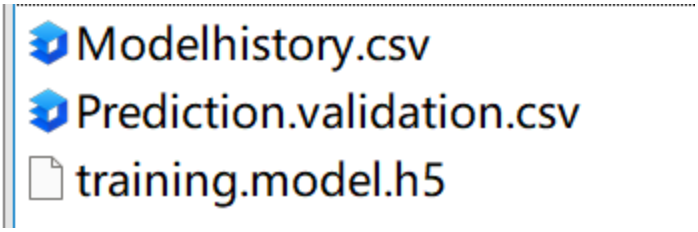
('Model save in:', '/root/home/liuqirui/SNPdata/DNNGP_TEST/training.model.h5')

训练好的模型文件，用于下一步对育种群体表型性状预测。

('Model history save in:',
'/root/home/liuqirui/SNPdata/DNNGP_TEST/Modelhistory.csv')

记录了训练过程中，各项数值的变化情况，打开后可以看到各项指标的变化情况，从这里可以选择决断，什么时间终止训练，随着训练层数的增加，各项指标不再继续变优就是决断层数

自己的数据要根据实际情况调整参数，尽量获得更高的相关系数和更低的 Pvalue 值。



DNNGP 算法的问题：

当拥有某一个物种的新的群体的基因型时，想预测该群体的表型，你需要把预测基因型和已知表型的基因型合并到一起进行 PCA, 然后再分开进行建模预测。因为**预测使用的是降维之后的 PCA, 所以每次有新的基因型都得重复这么做一遍**，特别是如果你训练的数据集的样本数量比较多时，每次都需要重新训练模型。

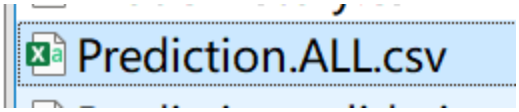
理想的模型应该是，你只要是使用的同一个参考基因组比对获得的基因型，那就直接调用模型进行预测即可，即不需要在重新建模。但是这种是需要你拥有基因型覆盖足够全面，而且样本数量足够大，材料代表性足够强，这种群体的规模至少是要数千个材料才行，否则群体不具有代表性，预测出的准确性很低。

使用模型预测

```
cd /root/home/liuqirui/SNPdata/DNNGP-main

python3 Scripts/Pre_runner.py --Model
"/root/home/liuqirui/SNPdata/DNNGP_TEST/training.model.h5" --SNP
"/root/home/liuqirui/SNPdata/GWAS/pca_dnngp2.predict.pkl" --output
"/root/home/liuqirui/SNPdata/DNNGP_TEST/"
```

```
/root/home/liuqirui/SNPdata/DNNGP-main/Scripts/Pre_runner.py:2: DeprecationWarning:
Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
(to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
but was not found to be installed on your system.
If this would cause problems for you,
please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

  import Pre_config_dnngp, Pre_dnngp #导入的设置文件和dnngp.pyx文件
2024-03-11 10:25:53.088856: E external/local_xla/xla/stream_executor/cuda/cuda_dnn.cc:9261] Unable to register cuDNN factory: Attemp
ting to register factory for plugin cuDNN when one has already been registered
2024-03-11 10:25:53.095417: E external/local_xla/xla/stream_executor/cuda/cuda_fft.cc:607] Unable to register cuFFT factory: Attempt
ing to register factory for plugin cuFFT when one has already been registered
2024-03-11 10:25:53.098994: E external/local_xla/xla/stream_executor/cuda/cuda_blas.cc:1515] Unable to register cuBLAS factory: Atte
mpting to register factory for plugin cuBLAS when one has already been registered
2024-03-11 10:25:56.511321: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
Model : /root/home/liuqirui/SNPdata/DNNGP_TEST/training.model.h5
SNP : /root/home/liuqirui/SNPdata/GWAS/pca_dnngp2.predict.pkl
Output : /root/home/liuqirui/SNPdata/DNNGP_TEST/
1/1 [==============================] - 2s 2s/step
Running time: 4.194602966308594 Seconds
```

结果：



ID	Prediction
H-R100	25.17047
H-R101	24.69003
H-R102	24.83401
H-R103	24.80449
H-R104	25.01106
H-R105	25.1227
H-R106	25.27595
H-R107	24.94314
H-R108	24.76139
H-R109	24.45903
H-R110	24.84194
H-R111	24.99985
H-R112	24.97273
H-R113	24.78787
H-R114	24.58461
H-R115	24.93557
H-R116	24.86774
H-R117	24.99695
H-R118	25.19738
H-R119	25.06429

ID	flowerNUM
H-R100	13
H-R101	22
H-R102	10
H-R103	9
H-R104	10
H-R105	16
H-R106	8
H-R107	4
H-R108	15
H-R109	12
H-R110	8
H-R111	12
H-R112	10
H-R113	7
H-R114	11
H-R115	9
H-R116	11
H-R117	11
H-R118	13
H-R119	12

左：预测结果，右：真实表型（flowerNUM）

日志附录：VCF 修改

bcftools reheader -s /root/home/liuqirui/SNPdata/GWAS/sample.txt /root/home/liuqirui/SNPdata/GWAS/combine_all.pass.recode.clear.vcf -o combine_all.pass.recode.clear2.vcf

sample.txt 第一列代表 VCF 文件中原始的样本名称，第二列代表替换后的样本名称，两类之间用空格 / 分隔符分隔，需要注意的是，样本名不允许有空格。

```
##fileformat=VCFv4.2
##fileDate=20240201
##source=PLINKv1.90
##contig=<ID=1,length=63172938>
##contig=<ID=2,length=57473253>
##contig=<ID=3,length=55446044>
##contig=<ID=4,length=63879566>
##contig=<ID=5,length=58409368>
##contig=<ID=6,length=54514479>
##contig=<ID=7,length=48320675>
##contig=<ID=8,length=42829657>
##contig=<ID=9,length=47297829>
##contig=<ID=10,length=45275338>
##contig=<ID=11,length=50286464>
##contig=<ID=12,length=33658696>
##contig=<ID=13,length=38755729>
##INFO=<ID=PR,Number=0,Type=Flag,Description="Provisional reference allele, may not be based on real reference genome">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  H-R100  H-R101  H-R102  H-R103  H-R104  H-R105  H-R106  H-R1
07  H-R108  H-R109  H-R110  H-R111  H-R112  H-R113  H-R114  H-R115  H-R116  H-R117  H-R118  H-R119  H-R120  H-R121  H-R122  H-R123
H-R124  H-R125  H-R126  H-R127  H-R128  H-R129  H-R130  H-R131  H-R132  H-R133  H-R134  H-R135  H-R136  H-R137  H-R138  H-R139  H-R1
40  H-R141  H-R142  H-R143  H-R144  H-R145  H-R146  H-R147  H-R148  H-R149  H-R150  H-R151  H-R152  H-R153  H-R154  H-R155  H-R156
H-R157  H-R158  H-R159  H-R160  H-R161  H-R162  H-R163  H-R164  H-R165  H-R166  H-R167  H-R168  H-R169  H-R170  H-R171  H-R172  H-R1
73  H-R174  H-R175  H-R176  H-R177  H-R178  H-R179  H-R180  H-R181  H-R182  H-R183  H-R184  H-R185  H-R186  H-R187  H-R188  H-R189
H-R190  H-R191  H-R192  H-R193  H-R194  H-R195  H-R196  H-R197  H-R198  H-R199  H-R200  H-R201  H-R202  H-R203  H-R204  H-R205  H-R2
06  H-R207  H-R208  H-R209  H-R210  H-R211  H-R212  H-R213  H-R214  H-R215  H-R216  H-R217  H-R218  H-R219  H-R220  H-R221  H-R222
H-R223  H-R224  H-R225  H-R226  H-R227  H-R228  H-R229  H-R230  H-R231  H-R232  H-R233  H-R234  H-R235  H-R236  H-R237  H-R238  H-R2
39  H-R240  H-R241  H-R242  H-R243  H-R244  H-R245  H-R246  H-R247  H-R248  H-R249  H-R250  H-R251  H-R252  H-R253  H-R254  H-R255
H-R256  H-R257  H-R258  H-R259  H-R260  H-R61   H-R62   H-R63   H-R64   H-R65   H-R66   H-R67   H-R68   H-R69   H-R70   H-R71   H-R7
2   H-R73   H-R74   H-R75   H-R76   H-R77   H-R78   H-R79   H-R80   H-R81   H-R82   H-R83   H-R84   H-R85   H-R86   H-R87   H-R88
H-R89   H-R90   H-R91   H-R92   H-R93   H-R94   H-R95   H-R96   H-R97   H-R98   H-R99   LZ-R31  LZ-R32  LZ-R33  LZ-R34  LZ-R35  LZ-R
36  LZ-R37  LZ-R38  LZ-R39  LZ-R40  LZ-R41  LZ-R42  LZ-R43  LZ-R44  LZ-R45  LZ-R46  LZ-R47  LZ-R48  LZ-R49  LZ-R50  LZ-R51  LZ-R52
LZ-R53  LZ-R54  LZ-R55  LZ-R56  LZ-R57  LZ-R58  LZ-R59  LZ-R60  MY-R1   MY-R10  MY-R11  MY-R12  MY-R13  MY-R14  MY-R15  MY-R16  MY-R
17  MY-R18  MY-R19  MY-R2   MY-R20  MY-R21  MY-R22  MY-R23  MY-R24  MY-R25  MY-R26  MY-R27  MY-R28  MY-R29  MY-R3   MY-R30  MY-R4
MY-R5   MY-R6   MY-R7   MY-R8   MY-R9
1       31089   .       C       G       .       .       PR      GT      0/0     ./.     0/1     0/1     0/0     0/0     0/0     0/0
```