

GEFormerV1.0 usermanual.pdf

1. Download the source code and sample data of GEFormer engineering project:

git clone <https://github.com/Deep-Breeding/GEFormer>

Or download from Baidu Netdisk:

Link: https://pan.baidu.com/s/1HrJSrV_tPrsqmlIEHEmYqA

Extraction code: 1234

2. GEFormer Environment configuration

First, install: Anaconda (<https://www.anaconda.com/>)

Install environment command:

```
conda create -n GEFormer python=3.8          // Create GEFormer virtual environment
```

```
conda activate GEFormer                      // Create GEFormer virtual environment
```

```
cd GEFormer
```

```
pip install -r requirements.txt              // The dependency packages required for installing
```

GEFormer

3. Input data file

(1) Genotype file: geno.csv

(2) Phenotype file: phe.csv

(3) Environment file: env.csv

(4) Data partitioning: CVF.csv

4. The model training of GEFormer

Parameters:

```
--geno_path    // Genotype file path
```

```
--pheno_path   // Phenotype file path
```

```
--pheno_name   // Phenotype name
```

```
--env_path     // Environment file path
```

```
--CVF_path     // Set up training and validation sets
```

```
--model_path   // Output file path
```

```
--device       //Runing device (CPU or GPU)
```

```
--optuna       // hyperparameter optimization
```

The following are optional parameters:

```
-- optuna_epoch // Number of tuning
```

```
--batch         //batch size
```

```
--lr            // learn reating
```

```
--drop_out      //drop put
```

```
--depth         //feature depth
```

```
--neurons1      // neurons number 1
```

```
--neurons2      // neurons number 2
```

The example of training model:

```
python run_train.py --geno_path ./data/geno.csv --pheno_path ./data/phe.csv --pheno_name  
PH --env_path ./data/env.csv --CVF_path ./data/CVF.csv --model_path ./model --device cpu  
--optuna True
```

Or

```
python run_train.py --geno_path ./data/geno.csv --pheno_path ./data/phe.csv --pheno_name  
PH --env_path ./data/env.csv --CVF_path ./data/CVF.csv --model_path ./model --device cpu  
--optuna False
```

Output:

After training, the Pearson correlation coefficient between the predicted values and the true values is written in the log file, as shown as the following:

Pearson = (0.71, 0.0030)

The value is the Pearson correlation coefficient (0.71) and the second number is the *P*-value (0.0030).