GAN-ac4C

Authors: Fei Li, Fengfeng Zhou (FengfengZhou@gmail.com)
Update: 2023-09-26

Description:

GAN-ac4C is an ac4C predictor combining transfer learning and GAN.

Develop environment:

System: WIN10
computer memory: 32G
GPU: 3060 12G
CUDA Version: 11.5

Installation:

Package	Version
Python	3.6.13
gensim	3.8.3
numpy	1.19.2
pandas	1.1.5
scikit-learn	0.24.2
scipy	1.5.4
tensorflow-gpu	2.4.1
xgboost	1.4.2

Software code structure

Folder of file name	description
config	Config.py contains parameters that control the training process
data	Original data files and tools which can read and format them.
dl	The DL-part of iDNA-TE

ensemble1	The main body of iDNA-TE
fs	Feature selection tool
prepare	Process original data
tools	Evaluation metrics
generate_positive_noise_BN.py	Generated RNA produce by it.
main.py	This is the entrance to the program

Config:

You can change parameters in config/config.py to train models.

Parameter name	description	Default value
device	the device used to train model, 'cpu' or 'gpu'	'gpu'
is_feature_selection	whether to perform feature selection for ML-based part model	'True'
load_global_pretrain_model	whether to load a pre-trained model for the DL-based model	'True
global_model_save_path	the path of the pre-trained model	None
model_save_path	The path to save the model	None
batch_size	Number of samples send to DL-based model each batch	256
learning_rate	Learning rate of DL-based model	1e-2
num_epochs	epochs	500
patience	Early stopping	50 (epochs not_improvement)

Format of input data

The training set and test set are pandas.DataFrame with 2 columns (label, seq). The optional value of column 'label' is 1(ac4C) or 0(non- ac4C). and the column 'seq' is a 201bp sequence containing 4 bases 'ACGU'.

```
>uc001old.3_879_1764_1609
GCCGUGGGCGUCCAGGGCGCUAGGAAGCCGGGUGGGGGUGAGGGUAGCCCUUGAGCCCUGUCCCUGCGGCUGUGAGAGCAGCAGGACCCUGGGCCA
>uc003alz.3_33_428_212
CGGCCGGCCGGCGAGCCAGUGCGCGUGCGCGGCGGCGGCCUCCGCAGCGACCGGGGAGCGGACUGACCGGCGGGAGGGCUAGCGAGCCAGCGGUGU
>uc002jvl.2_1053_1388_1260
UGGCACAAGCACAAGAAGCCAACCAGGAGAGAGUCCUGUAGCUCUGGGGGGAAAGAGGGCGGACAGGCCCCUCCCUCUGCCCUCUCCCUGCAGAAU
>uc001dyx.3_29_520_317
CAGGCGGGCGGGCGCCAGAGGGGGAAAGAGGCGGGGGCGGCGGGUCAGCCGCUGGCCGGGCCGGCCGGGGAAUGUCGAUGCCUGACGCGAUGCCGC
>uc002fzo.2_253_527_362
GCGACGGCGAGUGCCGGGGCCGGCGGGAUAGACGGGAAGCCCCGUACCUCCCCUAAGUCCGUCAAGUUCCUGUUUGGGGGCCUGGCCGGGAUGGGA
>uc001okj.3_115_799_229
GGGAAGGGGCCGUGCCCGGUGCCAGCCCAGGUGCUCGCGGCCUGGCUCCAUGGCCCUGGUCACAGUGAGCCGUUCGCCCCCGGGCAGCGGCGCCUC
>uc010enw.4_253_1251_963
AGAUGGACGCUUUCAUGCGGCGCAAGUGCCAGCAGUGCCGGCUGCGCAAGUGCAAGGAGGCAGGGAUGAGGGAGCAGUGCGUCCUUUCUGAAGAAC
>uc003xik.3_3_575_136
AAGUCGGGAGAGGCCGGUAGGCGUGAGGCGGGCCUGAAGCGGCAGCGGGCGGCCUUCGUCCGGCGAGAGCUAGGCCGAGGACCCGCGCCGCGCUCC
>uc003xbv.3_1154_2753_1932
GCACGGGGCGCCAAGGCAUAUUCCCUGCCAGCUACGUGCAGGUGUCUCGUGAACCCCGGCUCCGGCUCUGUGACGACGGCCCCCAGCUCCCCACGU
```

Train and test model

Before running main_1.py. If you want to use transfer learning, set config.load_global_pretrain_model=True. You can change the pre-trained model to yours. By running main.py, it will output the prediction metrics including ACC, SN, SP, MCC, AUC, and F1-score.