Digenic Interaction Effect Predictor(DItEP)

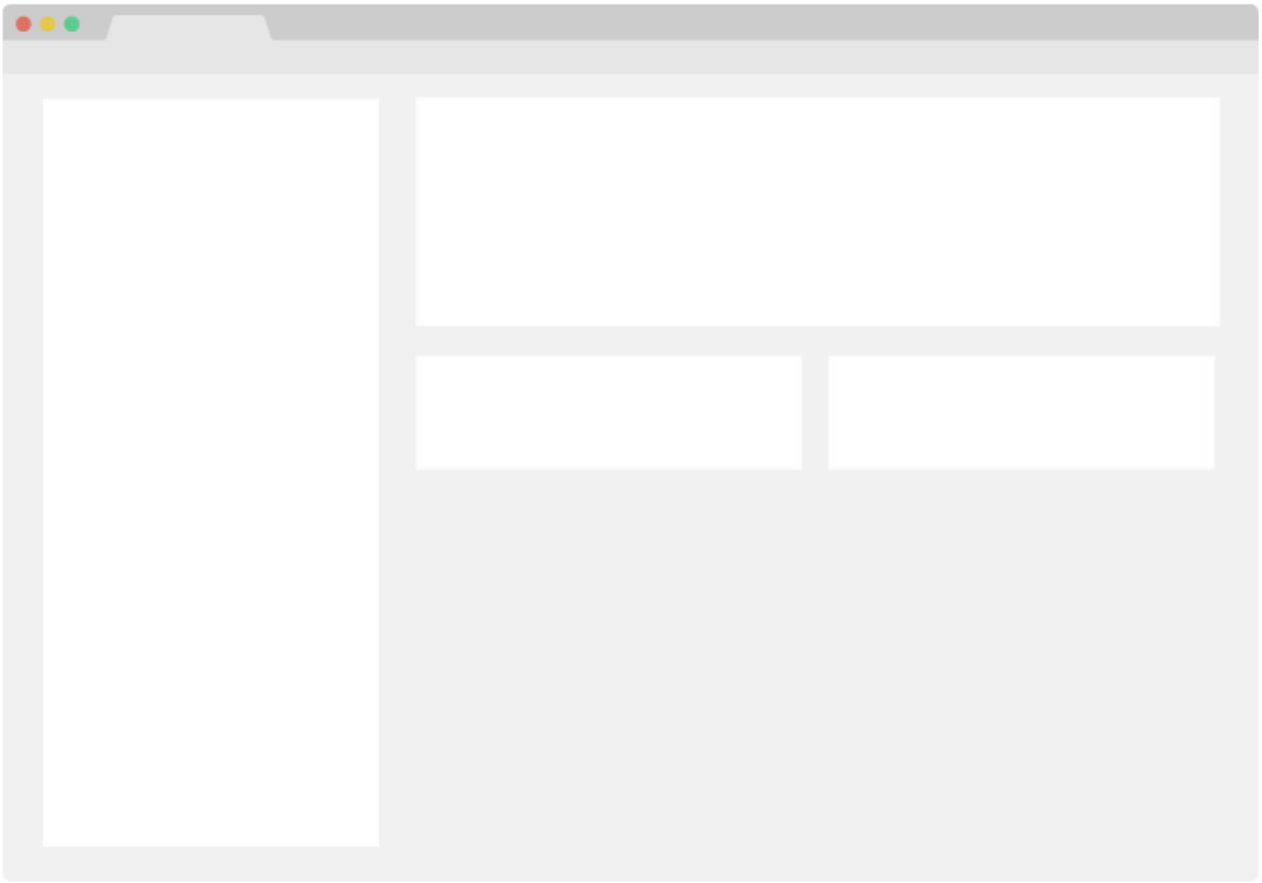
A framework to predict pathogenic gene pairs with digenic interaction effect based on the biological relatedness or similarity of the genes.

Explore the docs »

View Demo · Report Bug · Request Feature

Table of Contents

About The Project



Digenic Interaction is the simplest manifestation of genetic interaction, which means one mutated genes may alter (aggravate or relieve) the impact of the other gene on a phenotype. In-depth study of digenic interaction effect may enable us to better understand gene interaction networks and elucidate the potential relationship between genes and phenotypes, thereby making up the missing heritability to some extent.

We proposed a framework to predict pathogenic gene pairs with digenic interaction effect based on the biological relatedness or similarity of the genes, named Digenic Interaction Effect Predictor. **The digenic interaction scores of each of the two coding genes across the whole genome are also available.** DItEP may play a useful role in facilitating genetic mapping of interactive causal genes in human diseases.

Here's a blank template to get started:
To avoid retyping too much info. Do a search and replace with your text editor for the

following:

`github_username` , `repo` , `twitter_handle` , `email`

Built With

- Python
- pandas
- scikit-learn
- shap

Getting Started

To get a local copy up and running follow these simple steps.

Prerequisites

This is an example of how to list things you need to use the software and how to install them.

- Python download website (select based on your OS platfom)

Installation

1. Clone the repo

```
git clone https://github.com/github_username/repo.git
```

2. Install python packages

```
pip install scikit-learn
pip install pandas
pip install shap
```

Usage

Use this space to show useful examples of how a project can be used. Additional screenshots, code examples and demos work well in this space. You may also link to more resources.

1. Down_sampling_RF.py
   Use the down-sampling technique for training individual classifiers with different proportions.
2. Add_WeakRF_Classifier.py

Use the bagging-based method for adding individual weak classifiers to construct the final promising predictor with different weight according to the obb_scores of each classifier.

3. RF_Interpreter.py
   Use the SHAP for calculating the feature contributions.
4. Get_DigenicInteractionScores.py
   Manually extracting the digenic interaction scores for specified list of gene pairs. The input file should be named as “Genecom_xxx.txt” and the output file name will be “DigenicScore_xxx.txt”. The input file must containing two columns as the first two columns, and the first two columns are better be named as “GeneA” and “GeneB”.

GeneA	GeneB	Info1	Info2	Info3
ACE	CHP1			
MYH9	MTHFR			
OFD1	CEP131			

   Attention: The gene symbol should be provided as the HGNC official gene symbol, otherwise the digenic scores will be incorrect sometimes.
5. Results
   - 150Classifiers.zip
     The used 150 individual classifiers are compress into the package.
   - Weights.txt & Seeds.txt
     The obb_score, Accuracy, AUC, PR and sapling seeds for each individual classifier.

For more examples, please refer to the Documentation

Roadmap

See the open issues for a list of proposed features (and known issues).

Contributing

Contributions are what make the open source community such an amazing place to be learn, inspire, and create. Any contributions you make are **greatly appreciated**.

1. Fork the Project
2. Create your Feature Branch (`git checkout -b feature/AmazingFeature`)
3. Commit your Changes (`git commit -m 'Add some AmazingFeature'`)
4. Push to the Branch (`git push origin feature/AmazingFeature`)

5. Open a Pull Request

License

Distributed under the MIT License. See `LICENSE` for more information.

Contact

Your Name - @twitter_handle - email

Project Link: https://github.com/github_username/repo

Acknowledgements

-
-
-