1. TEAM INFORMATION

Name: Lorenzo Madeddu
Email: madeddu@di.uniroma1.it
Affiliation:
Department of Translational and Precision Medicine, “La Sapienza” University of Rome, Rome, Italy
Contribution: Write the code; Analyze the results

2. CODE DESCRIPTION/USAGE

My source code includes two files written with Python 3.8.6

1) validation.py
   - **Description**: This code is used to calculate the performance metrics with 10-fold cross-validation from an edge list. Furthermore, it computes the top-500 new PPIs for the input interactome.
   - **Usage**: The script can then be called from the command line “python validation py”.
   - **Hyperparameters**: None.
2) params.py
   - **Description**: This file contains the parameters settings and a description of them. is used to predict the top-500 new PPIs and calculate their scores for the human interactome (HuRI).
   - **Usage**: If you need to change the input file, you have to edit only the parameter **edgelist** (the input interactome file). You can customize the number of workers for parallel computation by setting the parameter **workers**.
   - **Hyperparameters**: None.
3) rw2.py
   - **Description**:This file contains the code of the core algorithm called RW^2.
   - **Usage**: None
   - **Hyperparameters**: None
4) utils.py:
   - **Description**: This file contains utilities for validation.py.
   - **Usage**: None
   - **Hyperparameters**: None

3. COMPUTING ENVIRONMENT

Scripts were run on a server of the Computer Science Department in my university. The server has the following hardware configuration:

- No GPU
- 40 CPU cores
- 252 GB of RAM

The same code can be used with CPUs only without any modification. However, it works better by setting the **workers** parameters to 1.

4. EXTERNAL PACKAGES/LIBRARIES

The computing environment is based on the Docker image `tensorflow/tensorflow:latest-gpu`, pulled and run with Singularity. The libraries `pandas`, `sklearn`, `networkx`, `node2vec`, `ray`, `pickle`, `itertools`, `numpy`, `csv`, `argparse` were added using pip.
This resulted in the following environment:

```
aiohttp
aiohttp-cors==0.7.0
aioredis==1.3.1
async-timeout==3.0.1
attrs
blessings==1.7
boto==2.49.0
boto3
botocore
brotlipy==0.7.0
bz2file==0.98
cachetools
certifi==2020.12.5
cffi
chardet
click==7.1.2
colorama==0.4.4
colorful==0.5.4
cryptography==2.9.2
decorator==4.4.2
filelock==3.0.12
gensim
google-api-core
google-auth
google-cloud-core
google-cloud-storage==1.19.0
google-resumable-media
googleapis-common-protos
gpustat==0.6.0
grpcio
hiredis==1.1.0
idna
jmespath
joblib
jsonschema==3.2.0
mkl-fft==1.2.0
mkl-random==1.2.0
mkl-service==2.3.0
msgpack==1.0.2
multidict
networkx
node2vec==0.3.0
numpy
nvidia-ml-py3==7.352.0
opencensus==0.7.12
opencensus-context==0.1.2
pandas==1.2.1
prometheus-client==0.9.0
protobuf==3.14.0
psutil==5.8.0
py-spy==0.3.4
pyasn1==0.4.8
```

```
pyasn1-modules==0.2.7
pycparser
pyOpenSSL==19.1.0
pyrsistent==0.17.3
PySocks
python-dateutil==2.8.1
pytz
PyYAML==5.4.1
ray==1.1.0
redis==3.5.3
requests
rsa
s3transfer
scikit-learn
scipy
six
smart-open
texttable
threadpoolctl
tqdm
typing-extensions
urllib3
yarl
```

5. ADDITIONAL DATASET USED IN THE METHOD

6. huri_labels.pickle
    - **Description**: This file contains gene-GOterm associations for the gene/proteins in the human interactome (HuRI.tsv file). I collected the gene-GOterm associations from the UniProt platform (https://www.uniprot.org/).

7. METHOD DESCRIPTION

The prediction implemented is based on the node and edge embeddings computed by the *RW*^{2} algorithm [1], integrated with a logistic regression classifier.

First, the set of all edges (protein-protein interactions) provided was split into 10 subsets for cross-validation. For each cross-validation fold, a set of non-edges (pairs of nodes absent from the original edge list) was generated for the training set using (degree-) balanced sampling as previously described and studied [2,3].
Our methodology consists of three steps: network construction, network representation learning and classification. In the representation learning step, we apply *rw*^{2} to the labeled human training interactome to generate low-dimensional representations of nodes (e.g. genes/proteins) and go-terms. Then, we generate edge representations of both positive and negative training PPIs set by computing the Hadamard product of their endpoints. In the classification step, we train a Logistic Regression by passing the edge representations as input instances.
In the testing phase, the testing set for a fold contains all the potential edges not in the training set.
Notice that the testing set could appear nodes never seen in the training set by cross-validation splitting. To improve the performance of the proposed model, we use an inductive technique to generate node embeddings for those nodes not occurring in the training set. The inductive technique works as follows. For each node not in the training set, we average the learned

representations of its associated Go-Terms. However, some nodes could not have GO-term associations. To solve this issue, for any testing edge with an endpoint without a representation, our method outputs a zero probability to exist.
Finally, in the evaluation phase, I computed AUROC, AUPRC and NDCG using the *sklearn* implementation.

Top-500 scores:
The score for a prediction is the probability that interaction exists. The probability is computed by the Logistic regression.

tabella parametri
Algorithmic idea:
GO-terms describe the biological functional context of a protein. RW^2, using node IDs and Go-terms as node labels, generates node/protein embeddings that jointly integrate functional and structural patterns in the network.

Reference

[1] Madeddu, Lorenzo, Giovanni Stilo, and Paola Velardi. "A feature-learning-based method for the disease-gene prediction problem." *International Journal of Data Mining and Bioinformatics* 24.1 (2020): 16-37.
[2] Yu J, Guo M, Needham CJ, Huang Y, Cai L, Westhead DR. Simple sequence-based kernels do not predict protein-protein interactions. *Bioinformatics* 2010; **26**: 2610–4.
[3] Park Y, Marcotte EM. Revisiting the negative example sampling problem for predicting protein-protein interactions. *Bioinformatics* 2011; **27**: 3024–8.