DeepAffinity Usage Manual

Di Wu, wudi930320@tamu.edu

April 7, 2020

Contents

1 DeepAffinity download link and its prerequisites

Source code of DeepAffinity is available at `https://github.com/Shen-Lab/DeepAffinity`.

DeepAffinity has been developed and tested in:

- Tensorflow-gpu v1.1
- Python 3.6
- TFlearn v0.3
- Scikit-learn v0.19
- Anaconda 3/5.0.0.1

To install Anaconda 3: Please refer to `https://www.anaconda.com/distribution/` to install Anaconda 3 according to your operating system.

We have already provided our environment list as environment.yml at `https://github.com/Shen-Lab/DeepAffinity/environment.yml`. After installing Anaconda, you can replace "envname" with any environment name you like and create your own environment by:

```
$ conda env create -n envname -f environment.yml
```

Then please activate your environment by:

```
$ source activate envname
```

2 Input Data Preparation

- Protein: SPS format.

  `e.g "CETM,ANGS,CETL,BETM,AEKM,CETL,AEKM,CEKL,AEDM,CEDL.."`

  We have provided the SPS dictionary we have generated for now. We will keep updating it and you are more than welcome to send us the new SPS sequence you generate. The dictionary can be found at:

  `https://github.com/Shen-Lab/DeepAffinity/data/dataset/protein_grouped_finalPresentation`

  For each two lines in this dictionary, the first line is original protein sequence from Uniprot and the second line is corresponding SPS format.

  To generate SPS format for your own protein, please refer to

  `DeepAffinity/data/script/split_data_script/README`

- Compound: Canonical SMILE from Pubchem.

  `e.g CCC(C(=O)C)(C(=O)O)O`

  Please make sure your SMILE format is the same as Pubchem Canonical SMILE. To generate canonical SMILE based on CID, please refer to

  `DeepAffinity/data/script/split_data_script/README`

- Label:

  `e.g 5.2`

  Here we use pIC50 = -$\log_{10}$(IC50), in which the unit of IC50 is in Mol. Please notice that you can calculate other labels, like EC50, Ki or Kd, in the same way.

When those three data files are ready, you may move IC50 data to the model you want and uncompress it. Here we use joint attention warm-start model as an example and we assume you are under the root folder "DeepAffinity". Then you can move IC50 dataset and replace the test data with your own data by following steps:

```
$ mv data/dataset/IC50.tar.xz Joint_models/joint_attention/
    joint_warm_start/data/
$ cd Joint_models/joint_attention/joint_warm_start/data/
$ tar -xf IC50.tar.xz
$ cd ..
$ ./replace_test.sh -p PROTEIN_FILE -c COMPOUND_FILE -l LABEL_FILE
```

3 Run the DeepAffinity and Output Analysis

We will introduce how to train our model and load checkpoints to test your data directly. Then we will give an example of result.

3.1 Run the DeepAffinity by script

We have provided automatic script for users to run our code in one command. The options of "DeepAffinity_inference.sh" are introduced below:

- **-h**: display this help and exit
- **-o|--output OUTPUT_FILE_NAME**:
  specify output filename
- **-p|--prot PROTEIN_SPS_FILE**:
  replace test_sps with user's protein sps format file
- **-c|--comp COMPOUND_SMILE_FILE**:
  replace test_smile with user's compound SIMLE format file
- **-l|--label LABEL_FILE**:
  replace test_ic50 with user's label file
- **-t|--labeltype LABEL_TYPE**:
  specify the label type you are using, it can be either IC50(default), EC50, Ki or Kd

3.1.1 Run the sample code

To simply run our model with IC50 dataset and test case, you can directly run

```
$ ./DeepAffinity_inference.sh
```

The default output will be at "./Joint_models/joint_attention/joint_warm_start/output". You can also specify the output name by

```
$ ./DeepAffinity_inference.sh -o OUTPUT_FILE_NAME
```

3.1.2 Test your own data from scratch

Also, our script will allow you to test your own data with our training set by:

```
$ ./DeepAffinity_inference.sh -o OUTPUT_FILE_NAME -p PROTEIN_SPS_FILE
    -c COMPOUND_SMILE_FILE -l LABEL_FILE
```

If your label is not IC50, you can also choose your label type by adding -t LABEL_TYPE:

```
$ ./DeepAffinity_inference.sh -o OUTPUT_FILE_NAME -p PROTEIN_SPS_FILE
    -c COMPOUND_SMILE_FILE -l LABEL_FILE -t LABEL_TYPE
```

3.1.3 Test your own data with our checkpoints

If you would like to use our checkpoints, we have provided two versions as follows:

- Download the checkpoints trained based on training set of IC50 from

  ```
  https://drive.google.com/drive/folders/1
      Pwn8uTyHNig4G2JDy0TErzH9hVacSadt?usp=sharing
  ```

- Download the checkpoints trained based on the whole dataset of IC50 from

  ```
  https://drive.google.com/drive/folders/1
      XAnXHSRnrO8DGA1drW3YnmaBaCihdiP5?usp=sharing
  ```

Please note that both checkpoints we provided are for the IC50 label. To use our checkpoint to test your data, you can run it by:

```
$ ./DeepAffinity_inference.sh -o OUTPUT_FILE_NAME -p PROTEIN_SPS_FILE
    -c COMPOUND_SMILE_FILE -l LABEL_FILE -m CHECKPOINT_FILE
```

3.2 Output

You should be able to see your result at the end below "error on test" line. Here the "error" means the root mean squared error(RMSE). One example is stated as below. 48928 should be the number of cases in test set and 0.6411142 is the RMSE. There are other results from OLS regression for your reference:

```
error on test
48928
[0.6411142]
                            OLS Regression Results
==============================================================================
Dep. Variable:                          y   R-squared:                       0.698
Model:                                OLS   Adj. R-squared:                  0.698
Method:                     Least Squares   F-statistic:                 1.129e+05
Date:                    Thu, 23 Jan 2020   Prob (F-statistic):               0.00
Time:                            16:12:46   Log-Likelihood:                -51554.
```

No. Observations:	48928	AIC:	1.031e+05
Df Residuals:	48926	BIC:	1.031e+05
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	1.8087	0.014	126.433	0.000	1.781	1.837
x1	0.7264	0.002	336.061	0.000	0.722	0.731

Omnibus:	1354.900	Durbin-Watson:	1.750
Prob(Omnibus):	0.000	Jarque-Bera (JB):	3473.420
Skew:	-0.017	Prob(JB):	0.00
Kurtosis:	4.305	Cond. No.	30.8