ProtDet-CCH

As one of the most challenging tasks in sequence analysis, protein remote homology detection has attracted a great deal of interest. Methods based on discriminative models and ranking approaches have achieved the-state-of-the-art performance, and these two kinds of methods are complementary. However, the integration framework of combining the discriminative and ranking methods has never been explored.

In this study, three LSTM models have been used to construct the predictors for protein remote homology detection, including ULSTM, BLSTM, and CNN-BLSTM. They are able to automatically extract the local and global sequence order information. Combined with PSSMs, the CNN-BLSTM achieved the best performance among the three LSTM-based models. We named this method as CNN-BLSTM-PSSM. Finally, a new method called ProtDet-CCH was proposed by combining CNN-BLSTM-PSSM and a ranking method HHblits. Tested on a widely used SCOP benchmark dataset, ProtDet-CCH achieved an ROC score of 0.998, and an ROC50 score of 0.982, significantly outperformed other existing state-of-the-art methods. Experimental results on two updated SCOPe independent datasets showed that ProtDet-CCH can achieve stable performance. Furthermore, our method can provide useful insights for studying the features and motifs of protein families and superfamilies. It is anticipated that ProtDet-CCH will become a very useful tool for protein remote homology detection.

Dependency

HHsuite-2.0.16 http://wwwuser.gwdg.de/~compbiol/data/hhsuite/releases/all/
NCBI-BLAST-2.4.0 ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/2.4.0/
Keras 2.0.6
Theano 0.9.0
Numpy 1.11.2
Biopython 1.68

Content

./data: the training dataset and testing dataset in FASTA format.
./PSSM: the PSSM files of training data and testing datasets generated by using PSI-BLAST.
./result: the prediction results of ProDet-CCH.
./db: the customized database of training samples for HHblits searching against.

Usage

```
python ProDet-CCH.py [-e] [<0.0005>] [-family_index <family index>] [-train
FALSE] [-test FALSE] [-pos_train_dir] [<dir>] [-neg_train_dir] [<dir>] [-pos_test_dir]
[<dir>] [-neg_test_dir] [<dir>] [-model_dir] [<dir>] [-weights_dir] [<dir>]
```

-e: the threshold of HHblits

-family_index: family index

-train: train a CNN-BLSTM-PSSM model

-test: load the trained CNN-BLSTM-PSSM model

-model_dir: the directory of the trained model json file of CNN-BLSTM-PSSM. If test is false, this argument can be empty.

-weights_dir: the directory of the trained model weight file of CNN-BLSTM-PSSM. If test is false, this argument can be empty.

-pos_train_dir: the directory of positive training dataset

-neg_train_dir: the directory of negative training dataset

-pos_test_dir: the directory of positive testing dataset

-neg_test_dir: the directory of negative testing dataset

-h: show this usage

The parameters of CNN-BLSTM-PSSM can be set in Paramters.py.

Example

1. Please install NCBI-BLAST to generate the PSSMs of training and testing samples. The PSSMs are generated by running PSI-BLAST searching against the database of Uniref50 with E-value threshold of 0.001 and 3 iterations.
2. The HHsuite-2.0.16 should be installed. The customized database for HHsuite should also be constructed, which can be downloaded at https://drive.google.com/open?id=0B3DNKjG0B3ZfSFgzTnBmN0ZHVTQ and it should be placed in the directory of “db”.
3. An example is provided for detecting the proteins in the target family a.138.1.3 (SCOP ID) by using only one command line “./run_and_train.sh”. The trained model of CNN-BLSTM-PSSM in the GPU environment is also provided. This example using the trained model can be implemented by using “./run_and_test.sh”.