Improse Documentation

Release 1.0

Aziz Khan

January 24, 2017

Contents

Welcome to Improse - Integrated Methods for Prediction of Super-Enhancers

CHAPTER 1

Table of Contents

1.1 Introduction

Improse is a supervised machine learning approach to predict super-enhancers or constituents of super-enhancers for a list of candidate enhancers. Improse integrated diverse features including DNase I hypersensitivity (DNaseI), histone modifications (HMs), cofactors, transcription factors (TFs) and DNA sequence specific features.

Improse comes with six state-of-the-art machine learning models including Random Forest (RF), Support Vector Machines (SVM), K-Nearest Neighbor (kNN), AdaBoost (AB), Decision Tree (DT) and Naive Bayes (NB).

Random Forest is our optimal and default model but user can select any of the model and further test it using cross-validation, independent test data or to make predictions.

1.2 Installation

1.2.1 Prerequisites

Improse requires:

- Python (>= 2.6 or >= 3.3)
- NumPy (>= 1.6.1): http://www.numpy.org/
- SciPy (>= 0.9): http://www.scipy.org/
- Scikit-learn (>= 0.17): http://scikit-learn.org/
- Pandas (>= 0.16.2): http://pandas.pydata.org/

If you already have a working installation of numpy and scipy, the easiest way to install scikit-learn and pandas is using pip:

```
pip install -U scikit-learn

pip install -U pandas
```

or using conda:

```
conda install scikit-learn

conda install pandas
```

If you don't already have a python installation with numpy, scipy and pandas, we recommend to install either via your package manager or via a python bundles (Canopy, Anaconda). These come with numpy, scipy, scikit-learn, pandas and many other helpful scientific and data processing libraries and available for platforms including Windows, Mac OSX and Linux.

1.2.2 Install Improse

You can install Improse either from PyPi using pip and install it from the source. Please make sure you have already installed the above mentioned python libraries required to run Improse.

Install from PyPi:

```
pip install improse
```

Install from the source:

```
tar -zxvf improse-1.0.tar.gz
cd improse-1.0
python setup.py install
```

1.3 How to use Improse

Once you have installed Improse, you can type:

```
improse --help
```

to find the available commands and required parameters to run Improse.

1.3.1 Improse demo

To run a demo using Random Forest model and validate it using 10-fold cross-validation, you can type:

```
improse --demo
```

This will save the results in the current working directory with a folder named `Improse_results`. If you wish to save the results in a specific folder, you can type:

```
improse --demo --output ~/path/to/your/folder
```

1.3.2 Select model

Improse comes with six state-of-the-art machine learning models including Random Forest (RF), Support Vector Machines (SVM), K-Nearest Neighbor (kNN), AdaBoost (AB), Decision Tree (DT) and Naive Bayes (NB). Random Forest is the default model.

To select model you need to type:

```
improse --model MODEL_NAME
```

MODEL_NAME can be `rf`, `svm`, `knn`, `ab`, `dt`, `nb` or use `all` if you want use all models one by one.

1.3.3 Define features and feature subsets

To tell the model to use specific features you need to type:

```
improse --model svm --feature H3K27ac,Brd4,p300,pGC
```

Make sure the features names are coma separated.

If you want to compare the individual predictive power or combinatorial predictive power of different features, you need to pass the argument `--compare` with `--features`:

```
mprose --model svm --feature H3K27ac,Brd4,p300,pGC --compare
```

To check the combinatorial predictive power of features, you need to combine features with + symbol:

```
improse --model svm --feature H3K27ac+Brd4,p300,pGC+pAT --compare
```

Here model will test the combinatorial predictive power [H3K27ac,Brd4] and [pGC,pAT] along with p300.

1.3.4 Run model with cross-validation

By default all models use 10-fold cross-validation. If you want to set different fold lets say 5, set `--cv` parameter as:

```
improse --model rf --feature H3K27ac,Brd4,p300,pGC,pAT,phastCons --cv 5
```

1.3.5 Run model with test data

To run the model with a test data you need the feature data saved a CSV file. Next, you need to tell the model, features you have to make prediction with using `--feature` and also provide the CSV file to '–input' and next type `--test` to tell model it is test datasets:

```
improse --model rf --feature H3K27ac,Brd4,p300,pGC,pAT,phastCons --input ~/path/to/CSV/file.csv --tes
```

This will generate an ROC plot and save the performance evaluations [precision, recall, f1-score, AUC, PRC] to `Improse_tesults.txt`.

1.3.6 Make predictions

To make predictions should have computed available features and saved a CSV file. Next, you need to tell the model the features you have to make prediction with using `--feature` and also provide the CSV file to `--input` and next type `--pred` to make predictions:

```
improse --model rf --feature H3K27ac,Brd4,p300,pGC,pAT,phastCons --input ~/path/to/CSV/file.csv --pre
```

This will save the predictions results as CSV file `Improse_[MODEL_NAME]_predictions.csv`. In the CSV file the field Class is 1=SE and 0=TE. We also report probability score for each prediction to tell the user how good and bad a prediction is. This will help to decide which candidates to select for further analysis.

1.4 Support

If you have questions, or found any bug in the program, please write to us at `khana10[at]mails.tsinghua.edu.cn`

1.5 Cite us

If you use Improse please cite us: