scikit-ribo Documentation

Release 0.2.3b1

Han Fang

Jun 27, 2017

Contents


scikit-ribo

CHAPTER 1

Contents:

Getting Started

This document will show you how to install and run Scikit-ribo.

What is Scikit-ribo

Scikit-ribo is an open-source software for accurate genome-wide A-site prediction and translation efficiency inference from Riboseq and RNAseq data.

Source Code: https://github.com/hanfang/scikit-ribo

Introduction

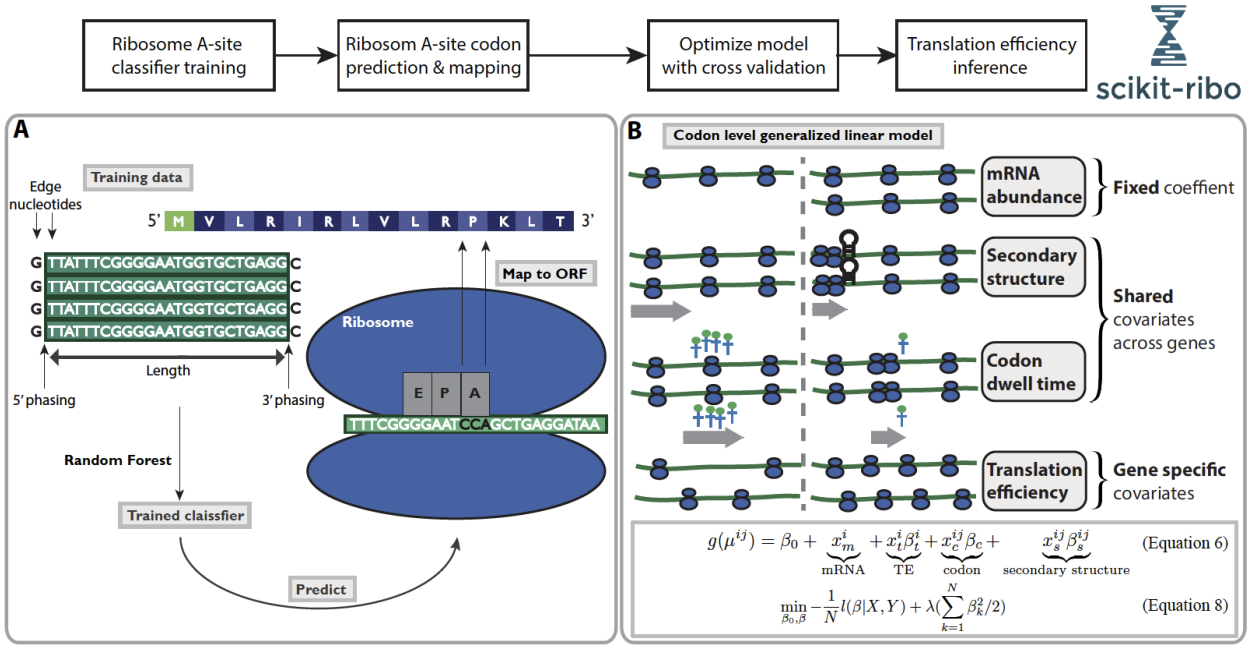
Scikit-ribo has two major modules:

- **Ribosome A-site location prediction** using random forest with recursive feature selection
- **Translation efficiency inference** using a codon-lvel generalized linear model with ridge penalty

A complete analysis with scikit-ribo has two major procedures:

- The data pre-processing step to prepare the ORFs, codons for a genome: `scikit-ribo-build.py`
- The actual model training and fitting: `scikit-ribo-run.py`

Detailed workflow



Inputs

- The alignment of Riboseq reads (bam)
- Gene-level quantification of RNA-seq reads (from either Salmon or Kallisto)
- A gene annotation file (gtf)
- A reference genome for the model organism of interest (fasta)

Output

- Translation efficiency estimates for the genes
- Translation elongation rate for 61 sense codons
- Ribosome profile plots for each gene
- Diagnostic plots of the models

Cite

Fang et al, "Scikit-ribo: Accurate inference and robust modelling of translation dynamics at codon resolution" (Preprint coming up)

Contact

Han Fang

Stony Brook University & Cold Spring Harbor Laboratory

Email: hanfang.cshl@gmail.com

Requirement

Environment

- Python3
- Linux
- Recommend setting up your environment with Conda

Dependencies

- Command-line pacakges:

Python package	Version >=
bedtools	2.26.0

- Python package:

Python package	Version >=
colorama	0.3.7
glmnet_py	0.1.0b
gffutils	0.8.7.1
matplotlib	1.5.1
numpy	1.11.2
pandas	0.19.2
pybedtools	0.7.8
pyfiglet	0.7.5
pysam	0.9.1.4
scikit_learn	0.18
scipy	0.18.1
seaborn	0.7.0
termcolor	1.1.0

Note: When using pip install scikit-ribo, all the following dependencies will be pulled and installed automatically.

Installation

Options

There are three options to install Scikit-ribo.

1. Install Scikit-ribo with pip:

```
pip install scikit-ribo
```

2. Install Scikit-ribo with conda/biocodon:

```
Coming up
```

3. Compile from source:

```
git clone https://github.com/hanfang/scikit-ribo.git
cd scikit-ribo
python setup.py install
```

Test whether the installation is successful

Once the installation is successful, you should expect the below if you type:

```
scikit-ribo-run.py
```

```
(py345) hfang@bnbdev2:~$ scikit-ribo-run.py
          _ _    _ _             _ _
 ___  ___(_) | _(_) |_      _ __(_) |__   ___
/ __|/ __| | |/ / | __|____| '__| | '_ \ / _ \
\__ \ (__| |   <| | ||_____| |  | | |_) | (_) |
|___/\___|_|_|\_\_|\__|    |_|  |_|_.__/ \___/


usage: scikit-ribo-run.py [-h] [-i I] [-f F] [-p P] [-q Q] [-s S] [-l L] [-c]
                          [-r] [-u U] [-o O]

optional arguments:
  -h, --help  show this help message and exit
  -i I        Input bam file
  -f F        path to the Folder of BED/index files generated by the pre-
              processing module
  -p P        Prefix for BED/index files
  -q Q        minimum mapQ allowed, Default: 20
  -s S        Shortest read length allowed, Default: 10
  -l L        Longest read length allowed, Default: 35
  -c          enable cross validation for glmnet
  -r          setting this flag will enable the RelE mode
  -u U        Un-mappable regions
  -o O        Output path, recommend using the sample id
```

Usage

How-to-use

- Pre-process: `scikit-ribo-build.py`
- Fit model: `scikit-ribo-run.py`

Detailed usage

- Pre-process: `scikit-ribo-build.py`:

```
scikit-ribo-build.py -g gtf-file -f fasta-file -p prefix -r rna-fold-file -t TPM-
↪file -o index-path
```

required arguments:

```
-g G          Gtf file, required
-f F          Fasta file, required
-p P          Prefix to use, required
-r R          Rnafold file, required
-t T          TPM of RNAseq sample, required
-o O          Output path of the built indexes, required
```

- Fit model: `scikit-ribo-run.py`:

```
scikit-ribo-run.py -i bam-file -f index-path -p prefix -o output-path
```

required arguments:

```
-i I          Input bam file, required
-f F          path to the Folder of BED/index files generated by the pre-processing␣
↪module, required
-p P          Prefix for BED/index files, required
-o O          Output path, recommend using the sample id, required
```

optional arguments:

```
-h, --help    show this help message and exit
-q Q          minimum mapQ allowed, Default: 20
-s S          Shortest read length allowed, Default: 10
-l L          Longest read length allowed, Default: 35
-c            enable cross validation for glmnet
-r            setting this flag will enable the RelE mode
-u U          Un-mappable regions
```

For more information, please refer to the template shell script about details of executing the two modules.

Preparing Input Data

- RNAfold: `call_rnafold.py`:

```
python call_rnafold.py  -f <prefix>.expandCDS.fasta -r rnafold-binary -p prefix -
↪n num-processes -o output-folder
```

Resulting file: <output-folder>/<prefix>.rnafold_lbox.txt Pre-built files can be also downloaded from here: https://github.com/hanfang/scikit-ribo/tree/master/data/prebuilt_rnafold

- TPM: Gene-level quatification from Salmon or Kallisto

Output format

- Translation efficiency estimates for the genes: **genesTE.csv**

gene	log2_TE
YAL001C	-0.7444
YAL002W	-0.9811
YAL003W	2.0833
...	...

- Translation elongation rate for 61 sense codons: **codons.csv**

codon	codon_dwell_time
AAA	0.9795
AAC	-0.9811
...	...
TTT	2.0833

- Diagnostic plots of the models

1. `asite_feature_importances.pdf`: Feature importance plot for the random forest classifier.
2. `asite_roc.pdf`: ROC curve for the random forest classifier.
3. `asite_3offset.pdf`: Distribution of A-site by read length and 3' phase
4. `asite_5offset.pdf`: Distribution of A-site by read length and 5' pahse
5. (optional) `riboseq.lambda_cv.pdf`: If cross-validation is enabled, the cross-validation curve is plotted.

Contact

Contribute

- Issue Tracker: https://github.com/hanfang/scikit-ribo/issues
- Source Code: https://github.com/hanfang/scikit-ribo

Support

If you are having issues, please let us know: hanfang.cshl@gmail.com

License

The project is licensed under the GPLv2 license.

Developer

Han Fang

Stony Brook University & Cold Spring Harbor Laboratory

Email: hanfang.cshl@gmail.com

Cite

Fang et al, "Scikit-ribo: Accurate inference and robust modelling of translation dynamics at codon resolution" (Preprint coming up)

CHAPTER 2

Indices and tables

- genindex
- modindex
- search