iCLIPit

Version

1.0

Authors

Anastassiya Zidkova and Martin Zidek

Description

Code in python and test data for RNA iCLIP analysis. This repository contains 4 scripts that you can run:

- iclipBarplot.py
- iclipBoxplot.py
- iclipFdr.py
- iclipRealFreq.py
- iclipZscores.py

Example data and results are in ToyExample folder.

Installation

Clone git repository by following command `git clone https://github.com/anastazie/iCLIP`

Dependencies

To use **iCLIPit** you need to have following packages: numPy, pandas

You can install them on terminal by running following commands:

```
sudo pip install numpy
sudo pip install pandas
sudo pip install python-dateutil --upgrade
sudo pip install pytz --upgrade
```

Getting help

In order to get full list of parameters for function type: `python <function_name> -h`

Data preprocessing

SAM files

This package use only positions in sam file. Before running scripts extract first 4 columns without header using following command: `grep -v "^@" test.sam |cut -f1,2,3,4 > test1_sub.sam`

GTF files

Preprocess GFT file by removing header using following command: `grep -v "^##" test.gtf > test_nohead.gtf`

Get real frequencies

Description

Compute number of mapped reads starting at each genome position

Parameters

-f File containing first 4 columns of .sam files without header

Output

CSV file real_freq.csv containing two columns:

Command

```
python iclipRealFreq.py -f test1_sub.sam
```

Compute FDR (False Discovery Rate) threshold

Description

Compute FDR threshold using following formula: (mean number of positions with n reads across i iterations of randomized datasets - standard deviation of positions with n reads across i iterations of randomized datasets) / number of positions with n reads in real dataset

Parameters

-f File containing first 4 columns of .sam files without header

-g Corresponding annotation file in GTF format

-r Range of nucleotides in one direction from the position to calculate number of reads starting in the range: position +- range_number, default value: 15

-i Number of iterations to generate randomized datasets from input file, default value: 100

Output

CSV file containing two columns:

Command

```
python iclipFdr.py -f test1_sub.sam -g test_nohead.gtf -r 20 -i 10
```

Compute z-scores for k-mers

Description

Compute k-mer z-scores using following formula: (number k-mer in real dataset - mean number of k-mers across i iterations of randomized datasets)/standard deviation of k-mers number across i iterations of randomized datasets

Only positions (both in real and randomized datasets) above specified threshold are considered for this computation.

Parameters

-f File containing first containing two columns:

-fa Fasta file containing reference nucleotide sequence in fa.tab format with two columns:

-g Corresponding annotation file in GTF format

-t FDR threshold value

-r Range of nucleotides in one direction from the position to calculate number of reads starting in the range: position +- range_number, default value: 15

-i Number of iterations to generate randomized datasets from input file, default value: 100

-k k-mer type: pentamer, hexamer, etc.

-l Length of sequence to output into seq.txt file: position +- length, default value: 10

Output

Two files are produced:

_seq.txt contains sequence for positions above threshold +- l

_zscores.csv contains three columns:

Command

```
iclipZscores.py -f test1_sub.sam -fa test.fa.tab -g test_nohead.gtf -t 100 -r 15 -i 10 -k
4 -l 5
```

Create barplot with read fraction per each gene

Description

Create barplot by computing reads fraction started at each gene

Parameters

-f File containing first containing two columns:

-g Corresponding annotation file in GTF format

-n number of genes, default value: all genes detected in GTF file

-t plot title, default value: “Barplot”

Output

PNG file with barplot, x - gene name, y - reads fraction, %

Command

```
iclipBarplot.py -f test1_sub_real_freq.csv -g test_nohead.gtf -n 5 -t testBar
```

Get boxplot with reads fraction per each gene interval

Description

Create boxplot by computing dividing each gene into p parts and counting number or reads starting in each interval

Parameters

-f File containing first containing two columns:

-g Corresponding annotation file in GTF format

-n number of genes, default value: all genes detected in GTF file

-t plot title, default value: “Boxplot”

-p number of parts, into which all genes will be divide, default value: 100

Output

PNG file containing boxplot per each gene (distribution of reads number in ech interval)

Command

```
iclipBoxplot.py -f test1_sub_real_freq.csv -g test_nohead.gtf -n 5 -t testBox -p 100
```

References

1. Wagnon, Jacy L., et al. "CELF4 regulates translation and local abundance of a vast set of mRNAs, including genes associated with regulation of synaptic function." PLoS genetics 8.11 (2012): e1003067.
2. König, Julian, et al. "iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution." Nature structural & molecular biology 17.7 (2010): 909-915.
3. Wang, Zhen, et al. "iCLIP predicts the dual splicing effects of TIA-RNA interactions." PLoS biology 8.10 (2010): e1000530.