iCLIPro Documentation

Release 0.1.0

Tomaz Curk

October 30, 2014

CONTENTS

CHAPTER

ONE

OVERVIEW AND HOW TO CITE

iCLIPro is a Python package that can be used to control for systematic errors in iCLIP data.

If you use iCLIPro in your research, please cite this paper (submitted for review):

> Christian Hauer, Tomaz Curk, Simon Anders, Jana Lober, Ina Hollerer, Anne-Marie Alleaume, Madhuri Bhuvanagiri, Jernej Ule, Wolfgang Huber, Matthias W. Hentze and Andreas E. Kulozik
> *High-resolution mapping of RNA-protein interactions by iCLIP: a novel approach to data analysis that avoids potential systematic errors*

CHAPTER

TWO

USAGE

Usage: iCLIPro [options] <in1.bam> <in2.bam> ...

Options:

-o FOLDER	output folder (default is cwd - current working directory)
-b INT	genomic bin size (100..1000, default: 300)
-r INT	number of reads required in bin (20..500, default: 50)
-c LIST	max distances when clustering neighboring positions (0..20, default: 0,1,2,3)
-q INT	use only reads with minimum mapping quality (mapq) (0..100, default: 30)
-g LIST	read len groups (default: "A:16-39,A1:16-25,A2:26-32,A3:33-39,B:42")
-p LIST	generate read overlap maps based on these comparisons (default: "A1-A3,A2-A3,A1-B,A2-B,A3-B,A-B")
-f INT	flanking region for read overlap maps (default: 60)
-h	longer help

For each input BAM file [inX.bam] the script will generate a number of output files that can be used to check for and diagnose systematic errors in iCLIP data.

Main result is stored in file [inX_report.txt], for each BAM file.

Read (query template) names in BAM files should include a record of form expressed with this regular expression: `:rbc[ATCGN]+:`. The ending colon can be omitted if random barcode record is placed at the end of the name. Some valid examples:

```
D3FCO8P1:206:C2M53ACXX:8:1207:17086:80291:1:N:0:rbcTGTAC: 272     1       11861 ...
D3FCO8P1:206:C2M53ACXX:8:1101:6625:73240:1:N:0:rbcCCGCC   16      1       11976 ...
D3FCO8P1:206:C2M53ACXX:8:1203:17298:81179:rbcCCGCC:1:N:0  16      1       11976 ...
```

Random barcodes can be specified at the end of the name but must be preceded by colon, for example:

```
D3FCO8P1:206:C2M53ACXX:8:1207:17086:80291:1:N:0:TGTAC     272     1       11861 ...
D3FCO8P1:206:C2M53ACXX:8:1101:6625:73240:1:N:0   :CCGCC   16      1       11976 ...
D3FCO8P1:206:C2M53ACXX:8:1203:17298:81179:1:N:0 :CCGCC    16      1       11976 ...
```

If no random barcode information is available, then iCLIPro will most likely be able to work with the original read names. In such case, please check that the read names do not include any text that conforms to the rules for specifying random barcode as it may mislead iCLIPro.

The generated report file includes a list of random barcodes indentified by iCLIPro. You should check it first and make sure that proper random barcode information is being used.

Parameters -b and -r: Specifies the bin size to use when segmenting the genome. Only bins with enough reads (parameter -r) are then considered in the read overlap testing.

Parameter -c: Specifies the maximum distance (in nucleotides) that can still be intrepreted as to indicate the same cross-linked site. Neighboring cross-link sites are joined and considered as one site. Position of neighbor with highest cDNA is retained, cDNA count of its neighbors is summed and assigned to it.

Parameter -q: Consider only reads that pass the minimum mapping quality, ignore the rest.

Parameter -g: iCLIPro needs to group reads based on their length. Any number of (overlapping) groups can be specified. For each group, an interval (INT1-INT2) or single value (INT) of the read lengths in the group can be specified.

Parameter -p (read overlap maps): Specifies which groups of reads to compare. When performing a comparison (G1-G2), cross-linked sites identified based on group G2 are used as reference (zero position). The relative positioning of sites identified in G1 is the computed and shown in read overlap maps.

Parameter -f: Width of the flanking region relative to reference point shown in read overlap maps.

CHAPTER

THREE

METHOD

A typical (i)CLIP experiment may result in the detection of RNA fragments of different lengths. Depending on the sequencing read length (usually 50 nt), shorter fragments may get fully sequenced and will include the adapter sequence. Longer fragments will include only the beginning (50 nt or so) bases and will not include the adapter sequence.

Before mapping, the adapter sequence must be removed from reads. The presence or absence of the adapter sequence is reflected in the read length. Short, trimmed reads indicate fully sequenced fragments, while reads of maximum length indicate fragments that are longer than the achievable sequencing read length.

iCLIPro works with groups of mapped reads. Its main function is to compare the cross-linked sites identified using reads of different lengths.

If no systematic error is present in iCLIP data, then the read length should not have an influence on the indentification of cross-linked sites.

iCLIPro first identifies regions (bins in genome, parameter -b) with a sufficient number of reads (parameter -r) for an read overlap test. Reads from each selected bin are processed separately.

Reads get grouped based on their length (parameter -g). Each group of reads is used to identify cross-linked sites.

Cross-linked sites from different groups are compared and read overlap maps are generated.

With iCLIPro you can test two ways to interpret iCLIP data:

- cross-link is one nucleotide before first mapped nucleotide
- cross-link is somewhere in middle of the read

The first intrepretation may not hold when systematic errors are present in iCLIP data (*e.g.*, substantial read-through, differences in chemistry of binding of the RBP to the RNA, protein footprint, etc). For details, see associated paper by Hauer and coauthors.

3.1 Read overlap map

Read overlap maps are generated by comparing cross-linked sites identified in the *test* and *reference* groups.

Cross-linked sites identified based on a reference group are used to define the reference (zero) position in the map. The regions (-60 to +60, x-axis on plots) relative to the reference positions are then scanned and number of co-occuring sites in test group is recorded.

The x-axis shows the offset of the cross-linked sites of the test group (shorter reads) relative to the cross-linked sites of the reference group (usually longer reads). The y-axis shows the number of test cross-linked sites that co-occur at a given offset relative to the reference sites.

Read overlap maps for the two interpretations (cross-link at start, cross-link at center of read) get generated.

CHAPTER

FOUR

INSTALLATION

iCLIPro will be made available from the Python Package Index (PyPI), iCLIPro package on PyPI. For now, please download this source file.

You need Python version 2.6 or later (Python 3 was not tested yet).

Please, install first matplotlib (plotting) and pysam (reading BAM files).

iCLIPro is a typical Python package. To install it from source follow these steps:

- download source
- unpack the tarball (`tar -xvzf iCLIPro-0.1.0.tar.gz`)
- go into the unpacked folder (`cd iCLIPro-0.1.0`)
- type to install for current user:

```
python setup.py install --user
```

A system-wide installation (requires admin rights) can be performed instead:

```
python setup.py build
sudo python setup.py install
```

To test the installation, in command-line:

1. change working folder to other than the build folder (*e.g.*, `cd ~`)
2. type `python` to enter Python
3. then type `import iCLIPro` to see if the package can be imported
4. exit Python and in command-line type `iCLIPro`

If you get an error message when importing iCLIPro in Python (step three above), then please make sure that the environment variable `PYTHONPATH` points to the iCLIPro package.

If you get an error message when trying to run the iCLIPro script (step four above), then please make sure that the environment variable `PATH` points to the script (also found in source `scripts/iCLIPro`).

CHAPTER

FIVE

EXAMPLES

Scripts used to generate some of the figures in paper are available in folder *examples*:

- 00_get_raw_BAMs.sh: downloads 7 raw BAM and 4 fastq files [4.8 GB] needed for the analysis, files are stored in folder `raw_data`,
- 01_prep_BAMs.py: modifies the read (query) names in BAM files (stored in `raw_data`) to include properly formatted information on random barcodes, files are stored in folder `data`,
- 02_run_iCLIPro.sh: runs the overlap analysis on each of the 7 BAM files in folder `data`, results are stored in folder `results`,
- 03_figures_for_paper.py: generates fig4.pdf, fig5.pdf, figS2.pdf and figS3.pdf that are used in the paper, figures are stored in folder `results`.

To learn more about each step, please check the individual scripts and modify them according to your needs.

You can just download the complete examples folder http://www.biolab.si/iCLIPro/examples/ and explore the outputs of scripts.

The scripts should generate these images:

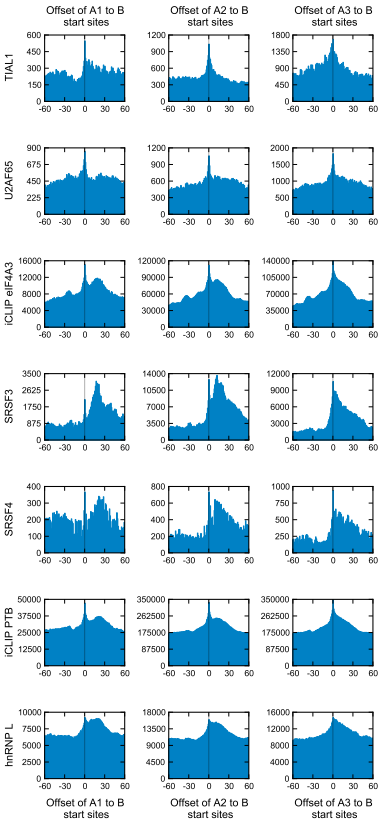

Figure 5.1: Figure 4 in paper.

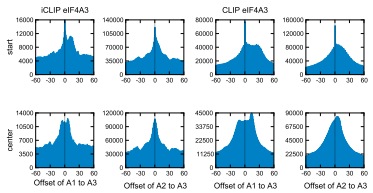

Figure 5.2: Figure 5 in paper.

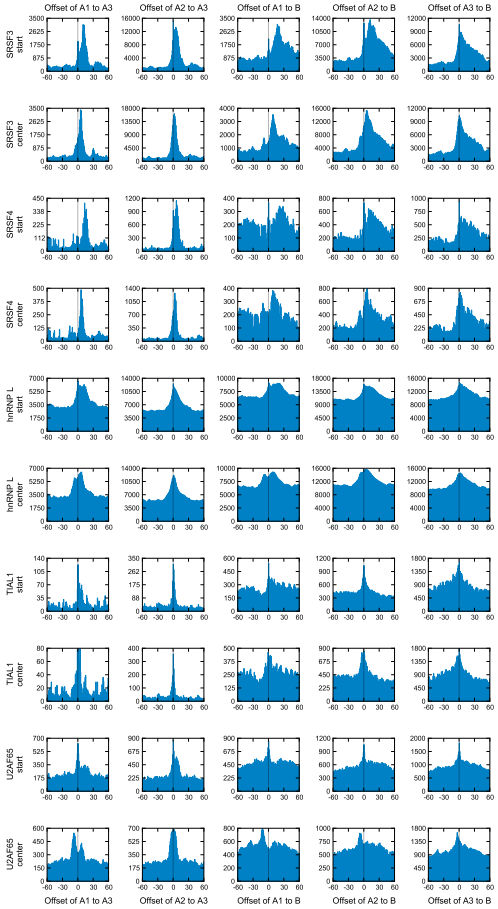

Figure 5.3: Figure S2 in paper.

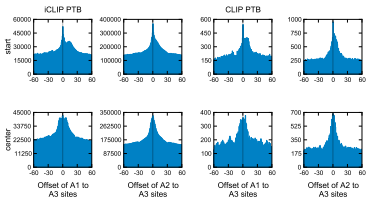

Figure 5.4: Figure S3 in paper.

CHAPTER

SIX

AUTHOR

iCLIPro is developed by Tomaz Curk at University of Ljubljana, Faculty of Computer and Information Science, Bioinformatics Laboratory.

This software is the result of a collaboration with the groups of prof. dr. Andreas E. Kulozik, MD, prof. dr. Matthias W. Hentze, MD, dr. Wolfgang Huber and prof. dr. Jernej Ule.

Special thanks to Christian Hauer who was instrumental during the inception and development of this tool.

You can contact me at tomaz.curk@fri.uni-lj.si.

CHAPTER

SEVEN

LICENSE

iCLIPro is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

The full text of the GNU General Public License, version 3, can be found here: http://www.gnu.org/licenses/gpl-3.0-standalone.html