Analyzing Trac-looping data

Binbin Lai
updated 08/22/2018

This document describes the codes and commands we used to analyze Trac-looping data using published software and custom codes in the Trac-looping paper. The custom methods were co-designed by Gangqing Hu and Binbin Lai, and were implemented by Binbin Lai.

Citation

Please cite the following Trac-looping paper if you use the methods:

Binbin Lai*, Qingsong Tang*, Wenfei Jin*, Gangqing Hu*, Darawalee Wangsa, Kairong Cui, Benjamin Z. Stanton, Gang Ren, Yi Ding, Ming Zhao, Shuai Liu, Jiuzhou Song, Thomas Ried, and Keji Zhao. (2018) Trac-looping measures genome structure and chromatin accessibility. ***Nature Methods***, in press (*equal contribution).

Prerequisite

- Python
- R
- gcc

Installation

Python scripts and R scripts are in the ***Trac-looping-methods/bin*** directory and ready for use. C++ codes are in ***Trac-looping-methods/src*** directory and need to be compiled by simply "***cd src"*** and "***make***". All programs will be installed in ***Trac-looping-methods/bin*** directory.

Procedures

1 Alignment

We used Bowtie 2 (v2.2.5 or later) (Langmead and Salzberg 2012) for sequence mapping and samtools (Li et al. 2009) and bedtools (Quinlan and Hall 2010) for generating .bedpe files that are needed for downstream analysis.

Trac-looping technology generates paired-end reads. We use local alignment mode in Bowtie 2 to allow linker sequences in both end of the reads, which increases mappability. An example:

```
bowtie2 -p [threads] -t -x [genome-bt2-idx] -q --local -1 [example.R1.fastq] -2 \
        [example.R2.fastq] | samtools view -bS - > [example.bam]
samtools view -bf -1 -F 12 -q 10 [example.bam] | bamToBed -i stdin -bedpe \
        | cut -f 1-6,8,9,10 > [example.bedpe]
```

2 .bedpe file format

A whitespace/tab separated file that contains, on each line,

```
<chr1> <start1> <end1> <chr2> <start2> <end2> <mapq> <str1> <str2>
```

- chr = chromosome
- start = read coordinate start
- end = read coordinate end
- mapq = mapping quality score
- str = strand ('+' for forward, '-' for reverse)

3. Filtering non-redundant PETs/reads

Redundant PETs/reads can be removed using following command:

```
Trac-looping-methods/bin/filter_NR_bedpe [input_bedpe] [output_bedpe]
```

4. Separate PETs into different categories

Trac-looping PETs include PETs with different chromatin structure information. In our study, we did not consider the inter-chromosomal PETs. We used PETs <150 bp to identify accessibility region and measure the accessibility levels. We used PETs > 1 Kb to identify the significant chromatin interactions. We include a python script to obtain PETs < 150 bp and PETs with length ranging from 1 Kb to 2 Mb. We set an upper bound 2 Mb as most interactions are shorter than 2 Mb. The command is as below:

```
python Trac-looping-methods/bin/seperate_PETs_len_statistics.py [input_bedpe] \
    [<150_bed] [1K2M_bedpe] [length_statistics]
```

- input_bedpe Input file in bedpe format.
- <150_bed Output file in normal bed format. PETs <150 bp.
- 1K2M_bedpe Output file in bedpe format. PETs from 1Kb to 2Mb.
- length_statistics Output distribution for PETs with different length ranges.

5. Identify accessibility regions using MACS2 or SICER

Both MACS2 (Zhang et al. 2008) and SICER (Zang et al. 2009) can be used to identify accessibility regions. In the Trac-looping paper, we used MACS2 to identify accessibility regions. We used following command:

```
macs2 callpeak -t [bedfile] -n [nameprefix] -g 2.7e9 -q 1e-6 -f BEDPE
```

- bedfile Input bed file. We used bed file with PETs <150 bp.
- nameprefix Output name prefix.

Useful output file in the downstream steps will be *prefix.narrowPeak*, which contains identified significant accessible region.

6. Identify significant interactions

We adopted the model described in Mango paper to identify the significant Trac-looping interactions. We used following command:

```
sh Trac-looping-methods/bin/TrACLoop_pipeline.sh [bedpe] [peaks] [nameprefix] [chr_len] \
[bin_size] [min_length] [max_length] [min_PET_c] [FDR_thr]
```

- bedpe String. Input bedpe file, e.g. 1K2M bedpe file
- peaks String. Input accessibility region file identified by MACS2 or SICER [first three columns: chr start end]
- nameprefix String. Output prefix
- chr_len String. chromosome length file. The chrlen files for mm9 and hg19 can be found in Trac-looping-methods/chrlen/. Files for other genome builds can be downloaded from UCSC genome browser.
- bin_size Integer. The genomic bin size used for loops. We used 2000 in our paper.
- min_length Integer. Minimal length of PETs considered. We used 4000 in our paper.
- max_length Integer. Maximal length of PETs considered. We used 2000000 in our paper.
- min_PET_c Integer. Minimal PET count for a significant loop. We used 3 in our paper.
- FDR_thr Float. Threshold of FDR for a significant loop. We used 0.05 in our paper.

Output files are listed as below:

- *prefix*.merged_sig_interaction.table.txt

  Tab delimited text file with header line. Each line represents a called significant interaction.

- *prefix*.merged_sig_interaction.hammock

  hammock format (http://wiki.wubrowse.org/Hammock) of significant interaction for displaying on WashU EpiGenome Browser.

7. Identify significant differential interactions between two Trac-looping libraries

The methods used for identifying differential interactions were described in the Trac-looping paper. We used the command as below:

```
sh Trac-looping-methods/bin/TrACLoop-diff-pipeline.sh [acc_bedfile_A] [acc_bedfile_B] \
[bedpe_A] [bedpe_B] [Loop_A] [Loop_B] [FDR_thr]
```

- acc_bedfile_A String. Input bed file containing tags used for measuring accessibility level in condition A. We used bed file for Trac-looping PETs<150 bp.
- acc_bedfile_B String. Input bed file containing tags used for measuring accessibility level in condition B.
- bedpe_A String. Input bedpe file containing Trac-looping PETs used for measuring interaction level in condition A.
- bedpe_B String. Input bedpe file containing Trac-looping PETs used for measuring interaction level in condition B.
- Loop_A String. Input interaction file in condition A.
- Loop_B String. Input interaction file in condition B.
- FDR_thr Float. Threshold of FDR.

Output files are listed as below:

- merged_loops_comp_summary2

  Tab delimited text file with header line containing measure information for each interaction union.

- merged_loops_increased_FDR*${FDR_thr}*

  Increased interactions in A condition relative to B.

- merged_loops_decreased_FDR*${FDR_thr}*

  Decreased interactions in B condition relative to A.

- merged_loops_nochange_FDR*${FDR_thr}*

  Not significantly changed interactions.

Acknowledgement

We thank Xiaobin Zheng's (Department of Embryology, Carnegie Institution for Science, Baltimore, MD 21218, USA) useful discussion on the algorithm design.

References

Langmead B, Salzberg S. 2012. Fast gapped-read alignment with Bowtie 2. *Nat Methods* **9**: 357-359.

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, Subgroup. GPDP. 2009. The sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**: 2078-2079.

Quinlan AR, Hall IM. 2010. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**: 841-842.

Zang C, Schones DE, Zeng C, Cui K, Zhao K, Peng W. 2009. A clustering approach for identification of enriched domains from histone modification ChIP-seq data. *Bioinformatics* **25**: 1952-1958.

Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W et al. 2008. Model-based analysis of ChIP-Seq (MACS). *Genome Biol* **9**: R137.