priUTR user manual

Min Tu, Yin Li, Yaping Feng, and Dibyendu Kumar

Dec. 31st. 2019

Contents:

1. Introduction

To identify alternative 3' untranslated regions (3'UTR) or those transcripts or genes with length variations in their 3'UTRs compared to the reference genome or transcriptome, we developed this pipeline, namely **P**ipeline for **R**NA-seq-based **I**dentification of Alternative 3'**UTR** (**priUTR**). Our priUTR pipeline is compatible with standard RNA-seq analysis pipelines and uses the output files from such a pipeline, for example the classic "Tophat-Cufflinks-Cuffdiff" pipeline (Trapnell et al. 2012, 2013; Tophat website; Cufflinks website and Cufflinks' github website).

Previously, high-throughput studies on variations of 3'UTRs required characterizing all potential 3'UTRs based on long-read sequencing technologies, for instance the single-molecule sequencing methods, for example PacBio Isoform-Sequencing (Iso-Seq). Alternatively, high-throughput 3'UTR study could be achieved by characterizing alternative poly(A) sites using specialized library preparation methods and bioinformatic pipelines, such as Poly(A)-tag RNA-seq (PAT-seq) (Wu et al. 2011). Both types of approaches are, however, technically challenging and more expensive than standard RNA-seq, especially for plant biologists working with non-model species, particularly those with large genomes. Computational methods for identifying alternative poly(A) sites using RNA-seq data have been reported recently albeit their applications in non-model plant species yet to be tested (Arefeen et al. 2018; Ye et al. 2018). Besides, the scripts for analyzing potential polyadenylation events from RNA-seq data in plants have been developed, they inevitably require the users to have significant bioinformatic coding skills (Hunt et al. 2016; Jafar et al. 2019). To make full use of standard RNA-seq data and to provide user-friendly 3'UTR analysis function for many non-model plant species and plant biologists who may have limited bioinformatic skills, the priUTR pipeline: 1) takes the output files from the "Tophat-Cufflinks-Cuffdiff" pipeline (transcripts expression data and reference-guided assembly of transcriptome); 2) calculates the length and position information of 3'UTRs annotation (fasta format assembled genome and gff3 format annotation); 3) calculates the alternative 3'UTRs compared to annotated 3'UTRs for each transcripts and each RNA-seq sample using RNA-seq reads; 4) filter for high-confidence alternative 3'UTRs by transcripts' expression levels and the number of reads mapped to 3'UTR regions. The workflow and its combination with standard RNA-seq analysis pipeline is shown in Figure 1 (also see our manuscript XXXXX).

The priUTR pipeline is available at Github website
(https://github.com/mint1234/3UTR-).

The priUTR pipeline and its example input and output files are also available at Waksman Institute of Microbiology, Rutgers, The State University of New Jersey (https://data.waksman.rutgers.edu/200106-3utr/).

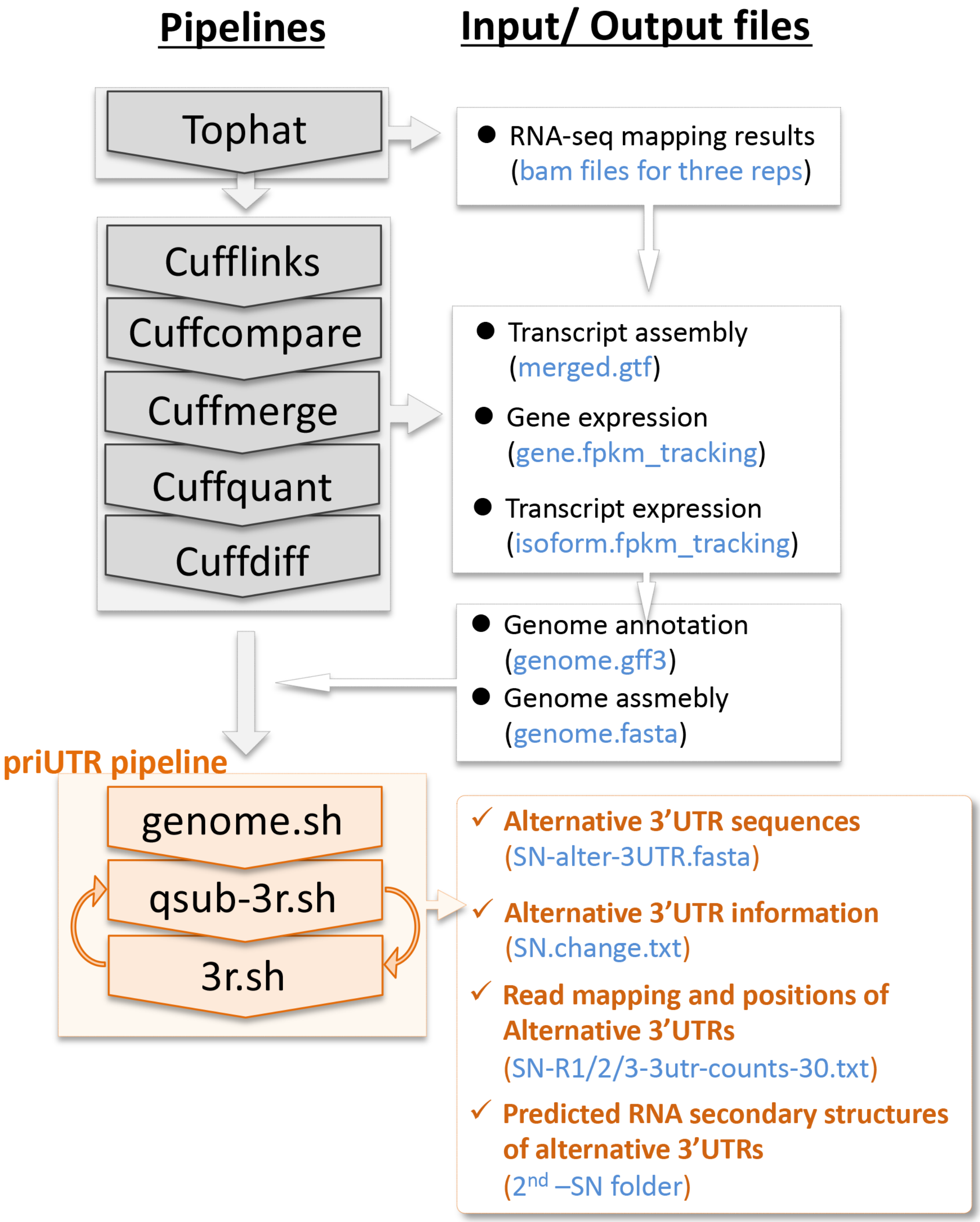
Pipelines
Input/ Output files
Tophat
● RNA-seq mapping results (bam files for three reps)
Cufflinks
Cuffcompare
Cuffmerge
Cuffquant
Cuffdiff
● Transcript assembly (merged.gtf)
● Gene expression (gene.fpkm_tracking)
● Transcript expression (isoform.fpkm_tracking)
● Genome annotation (genome.gff3)
● Genome assmebly (genome.fasta)
priUTR pipeline
genome.sh
qsub-3r.sh
3r.sh
✓ Alternative 3'UTR sequences (SN-alter-3UTR.fasta)
✓ Alternative 3'UTR information (SN.change.txt)
✓ Read mapping and positions of Alternative 3'UTRs (SN-R1/2/3-3utr-counts-30.txt)
✓ Predicted RNA secondary structures of alternative 3'UTRs (2nd –SN folder)

Figure 1. Workflow of the priUTR and upstream RNA-seq analysis pipeline.

The priUTR pipeline is compatible with HiSAT aligner, but we use Tophat-Cufflinks pipeline as an example in this workflow chart so as to be consistent with the user manual and the example input and output data provided.

2. System Requirements and Software Installation

Our pipeline is developed in Scientific Linux release 6.10 (Carbon). This pipeline is based on Linux shell scripts, therefore it requires a 64-bit CPU computer running on Linux system. 5-50GB running RAM is recommended depending on data size. And this pipeline was tested with the following version of the programs:

-- samtools Version 1.3.1,

http://www.htslib.org

-- bedtools v2.24.0,

https://bedtools.readthedocs.io/en/latest/

-- TransDecoder Version 2.0.1,

https://github.com/TransDecoder/TransDecoder/releases

-- gffread Version 0.9.8c,

https://github.com/gpertea/gffread

-- ViennaRNA Version 2.4.14,

https://www.tbi.univie.ac.at/RNA/RNAfold.1.html

-- ToGenePred:

https://bioconda-recipes-demo.readthedocs.io/en/docs/recipes/ucsc-gtftogenepred/README.html

-- exteractFromFasta.pl:

https://github.com/jonbra/NGS-Abel/blob/master/scripts/extractFromFasta.pl

-- qsub Version 3.0.4:

http://docs.adaptivecomputing.com/torque/2-5-12/help.htm#topics/commands/qsub.htm

3. Instructions for the priUTR pipeline

3.1 Notes for standard RNA-seq analysis pipeline

We recommend to use the “Tophat-Cufflinks-Cuffdiff” pipeline for the upstream RNA-seq data analysis, although replacement of Tophat or Cufflinks for RNA-seq read mapping and transcriptome assembly, respectively, using more updated software, such as HiSAT2, would be theoretically compatible. For a splice-aware RNA-seq read mapper, we suggest mapping the reads for each RNA-seq sample to the reference genome (add the option “-G genome.gtf” to TopHat). For the transcriptome assembly step, we suggest building the transcriptome assembly using a reference-guided mode (add the option “-g GTF” rather than “--GTF”, as the “--GTF” option only assemblies known transcripts annotations according to GTF file). We also recommend to use Cuffmerge to merge into one reference-guided assembly from multiple RNA-seq samples.

3.2 priUTR Instructions

Execute the priUTR pipeline as follow. The priUTR pipeline can calculate alternative 3’UTRs for multiple RNA-seq samples in one run. (Note: file names are colored in blue; commands are shaded in grey.)

1. Download priUTR pipeline from github address: XXX.
2. Create three folders, namely “scripts”, “inputs” and “softwares”. Install all of the above-mentioned software (see “**2. System Requirements and Software Installation**”) in the “softwares” folder.
3. Then move the following files into the “inputs” folder: (1) genome.gff3 (genome annotation); (2) genome.fa (assembled genome sequence); (3) the output files from RNA-seq read aligner and transcriptome assembler (Tophat and Cufflinks, *etc.*), including three “.bam” files corresponding to the triplicated RNA-seq samples respectively generated by Tophat, isoforms_fpkm.tracking (Cuffdiff output file to show isoform expression); merged.gtf (Cuffmerge output file of a transcript assembly merged from multiple RNA-seq samples).
4. Please first set local path to “$PRE” in the shell file, and then execute the “genome.sh” file using command “sh genome.sh”; The script “genome.sh” is to extract 3’UTR information of all annotated transcripts in the reference genome.

5. Please set local path to “$PRE” in the ”qsub-3r.sh“ and ”3r.sh“ files. Then set condition names to “${SN}”, and set how many RNA-seq samples (or experiment treatments or conditions) to “i” in the ”qsub-3r.sh“ file. The priUTR pipeline can automatically loop execute for “i” times to proceed multiple RNA-seq samples (“i” samples, in this case) in one run. In the “3r.sh” file, the default setting of RPKM filter is “A=1”, and the default setting of 3’UTR-reads filter is “B=30”, they can be set or adjusted in the pipeline manually. Then execute the “qsub-3r.sh” using command “sh qsub-3r.sh” (which loop executes “3r.sh”).

6. The output results will be generated in two folders: “results” and “2nd-SN” folder. The “results” folder will contain three result files for RNA-seq sample: ***(1)*** the “SN-alter-3UTR.fasta” file is a fasta format file containing all the alternative 3’UTR sequences identified;

   (2) the “SN.change.txt” file contains the following information in each column:

   1) TCONS ID: assembled transcript ID from Cufflinks;
   2) transcripts ID: corresponding transcript ID in the annotation;
   3) gene ID: corresponding gene ID in the annotation;
   4) class code: the type of match between the Cufflinks transcripts and the reference transcript. A full description of class code meaning can be obtained from the link: http://cole-trapnell-lab.github.io/cufflinks/cuffcompare/
   5) alternative 3UTR length;
   6) annotated 3UTR length;
   7) transdecoder;
   8)+/- transdecoder;
   9) 3utr length transdecoder: columns 7, 8, and 9 represent the source, strand and length of predicted 3’UTR using transdecoder;
   10) FPKM-replicate-1;
   11) FPKM-replicate-2;
   12) FPKM-replicate-3: columns 10, 11, 12 represent the expression levels of the alternative 3’UTR for RNA-seq replicate1, 2, and 3, respectively.

   (3) the “SN-R1/2/3-3utr-counts-30.txt” file contains the following information for each column:

   1) chromosome: chromosomal location of the alternative 3’UTR;
   2) alternative 3utr_start;

3) alternative 3utr_end: columns 2 and 3 represent the start and end positions of the alternative 3'UTR.
4) transcript ID: the corresponding transcript ID in the annotation;
5) addition information;
6) +/-;
7) number of reads mapped to the 3'UTR region, those transcripts with less than 30 reads have been filtered out;

7. The "2nd-SN" folder contains all the secondary structure plots that are predicted for the priUTR- identified alternative 3'UTR sequences by RNAfold from the ViennaRNA package with minimum free energy. For the predicted secondary structures of alternative 3'UTRs, please also see the example output files available at Waksman Institute (folder namely "2nd-Endosperm_16DAP" and "2nd-Endosperm_20DAP" at link: https://data.waksman.rutgers.edu/200106-3utr/)

References:

Trapnell C, Roberts A, Goff L, Pertea G, Kim D, Kelley DR, Pimentel H, Salzberg SL, Rinn JL, Pachter L. (2012) Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. Nature Protocol. 7(3): 562-578.

Trapnell C, Hendrickson DG, Sauvageau M, Goff L, Rinn JL, Pachter L. (2013) Differential analysis of gene regulation at transcript resolution with RNA-seq. Nature Biotechnology. 31, 46–53.

Tophat website: http://ccb.jhu.edu/software/tophat/index.shtml

Cufflinks website: http://cole-trapnell-lab.github.io/cufflinks/manual/

Cufflinks github website: https://github.com/cole-trapnell-lab/cufflinks

Wu X, Liu M, Downie B, Liang C, Ji G, Li QQ, Hunt AG. (2011) Genome-wide landscape of polyadenylation in Arabidopsis provides evidence for extensive alternative polyadenylation. Proc. Natl. Acad. Sci. USA, 108, 12533- 12538.

Jafar Z, Tarig S, Sadiq I, Nawaz T, Akhtar MN. (2019) Genome-Wide Profiling of Polyadenylation Events in Maize Using High-Throughput Transcriptomic Sequences. G3: Genes, Genomes, Genetics. 9, 2749-2760.

Ye C, Long Y, Ji G, Li QQ, Wu X. (2018) APAtrap: identification and quantification of alternative polyadenylation sites from RNA-seq data. Bioinformatics, 34, 1841-1849.

Arefeen A, Liu J, Xiao X, Jiang T. (2018) TAPAS: tool for alternative polyadenylation site analysis. Bioinformatics, 34, 2521- 2529.