Supplementary methods

Construction of RNA reference materials for improving the quality of transcriptomic data

Yu Y *et al.*

Materials

Software

- A Linux-based operating system
- Ballgown, version 2.20.0 (https://bioconductor.org/packages/release/bioc/html/ballgown.html)
- Fastp, version 0.19.6 (https://github.com/OpenGene/fastp)
- FastQ Screen, version 0.15.3 (https://www.bioinformatics.babraham.ac.uk/projects/fastq_screen/)
- FastQC, version 0.12.1 (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/)
- HISAT, version 2.1.0 (https://ccb.jhu.edu/software/hisat/index.shtml)
- MultiQC version 1.8 (https://multiqc.info/)
- Qualimap, version v2.3 (http://qualimap.conesalab.org/)
- R, version 3.6.5 or higher (https://www.r-project.org/) and Rstudio (https://posit.co/downloads/) or any other integrated development environment
- SAMtools, version 1.3.1 (https://github.com/samtools/samtools)
- StringTie, version 1.3.4 (https://ccb.jhu.edu/software/stringtie/)

Software environment setup

Command for RNA-seq upstream analysis

The commands for RNA-seq upstream analysis used in the protocol should all be run from the Linux shell prompt within a terminal window connected to a Linux server. We encourage the user to create a specific working directory (e.g., “RNA_ref_data_analysis/”) to store all the example data and files created by the analysis. All commands are described under the assumption that the user is working in this directory. The commands will be executed in the Linux shell environment and prefixed with a ‘$’ character in the following parts.

Download and organize example data

In this protocol, we take the raw fastq data of sample D5_1 from batch R_BGI_L3_B1 as an example to illustrate the RNA-seq upstream analysis. The raw fastq data can be downloaded from the Genome Sequence Archive (GSA) (accession number: HRA001859) and the Open Archive for Miscellaneous Data (OMIX) (accession number: OMIX002254) hosted by the National Genomic Data Center (https://ngdc.cncb.ac.cn/gsa/).

The user can download these files using the following commands:

$ wget -c ftp://download.big.ac.cn/gsa-human/HRA001859/HRR777511/HRR777511_f1.fastq.gz -o R_BGI_L3_B1_D5_1.R1.fastq.gz
$ wget -c ftp://download.big.ac.cn/gsa-human/HRA001859/HRR777511/HRR777511_r2.fastq.gz -o R_BGI_L3_B1_D5_1.R2.fastq.gz

The user should get these files in the current path:

- R_BGI_L3_B1_D5_1.R1.fastq.gz
- R_BGI_L3_B1_D5_1.R2.fastq.gz

▲**CRITICAL** The GSA, OMIX, and QDP provide the md5 value. The md5 value is usually stored in a txt file called ‘md5.txt’.

The user should check the integrity of the data files by using the following command:

$ md5sum -c md5.txt

▲**CRITICAL** It will give an ‘OK’ result if the data file is intact. Otherwise, contact the data provider to transfer the correct data again immediately.

Download the reference data

The user should download two reference files, including reference human genome build 38 (https://genome-idx.s3.amazonaws.com/hisat/grch38_snptran.tar.gz) and gene model from

Ensembl (http://ftp.ensembl.org/pub/release-93/gtf/homo_sapiens/Homo_sapiens.GRCh38.93.gtf.gz) into the reference folder by using the following command:

$ mkdir -p reference/genome

$ wget -c https://genome-idx.s3.amazonaws.com/hisat/grch38_snptran.tar.gz -P reference/genome

$ wget -c http://ftp.ensembl.org/pub/release-93/gtf/homo_sapiens/Homo_sapiens.GRCh38.93.gtf.gz -P reference/

Decompress the grch38_snptran.tar.gz and Homo_sapiens.GRCh38.93.gtf.gz files by using the following command:

$ gunzip reference/genome/rch38_snptran.tar.gz

$ gunzip reference/Homo_sapiens.GRCh38.93.gtf.gz

Download and install the software

Install all the executable software used in this protocol by using conda (if none already exists):

Create the new conda environment for software downloads and installs:

$ conda create -n rna-seq-env python=3.7

Activating the new environment:

$ conda activate rna-seq-env

Install all the executable software:

$ conda install -c bioconda fastqc=0.12.1

$ conda install -c bioconda fastp=0.19.6

$ conda install -c bioconda fastq-screen=0.15.3

$ conda install -c bioconda hisat2=2.1.0

$ conda install -c bioconda multiqc=1.8

$ conda install -c bioconda qualimap=2.3

$ conda install -c bioconda samtools=1.3.1

$ conda install -c bioconda stringtie=1.3.4

! CAUTION Python 2.7 should be pre-installed in the conda environment for the specific requirement of some software. However, it is important to note that Python 2.7 was discontinued. It is recommended to use Python version 3.x whenever possible.

$ conda install -c bioconda bioconductor-ballgown=2.20.0

▲**CRITICAL** An additional process for installing the dependencies in the R session is important for the ballgown.

To install ballgown, start an R session:

$ R

R version 4.0.2 (2020-06-22) -- "Taking Off Again"

Copyright (C) 2020 The R Foundation for Statistical Computing

Platform: x86_64-conda_cos6-linux-gnu (64-bit)

The R introductory message will end with a ">" prompt. Install the Ballgown packages and other dependencies in R as follows:

> if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

> BiocManager::install("GenomeInfoDbData")

> BiocManager::install("ballgown")

> BiocManager::install("BiocGenerics",force = TRUE)

Download the customized R scripts and Python scripts used in this protocol from https://github.com/YingYu12345/Reference_standard_protocol/blob/main/ballgown and https://github.com/gpertea/stringtie/blob/master/prepDE.py3, which include one R script and one Python script:

$ wget -c https://github.com/YingYu12345/Reference_standard_protocol/blob/main/ballgown

$ wget -c https://github.com/gpertea/stringtie/blob/master/prepDE.py3

The user should get these two scripts in the current path:

- ballgown
- prepDE.py3

Download the other customized R scripts used in this protocol from https://github.com/YingYu12345/Reference_standard_protocol/blob/main/

Make a directory for output files

$ mkdir path-to/output

Procedure

▲**CRITICAL** All the command examples use the raw fastq data of sample D5_1 from batch R_BGI_L3_B1 in this protocol. For other samples, such as the data of sample D6_1 from R_ILM_L8_B1, change the name of the input and output files accordingly.

Alignment

- **TIMING 3 h**

1. Set the working directory in the Linux system, which contains the raw data.

   $ mkdir -p ${work_dir} && cd ${work_dir}

▲**CRITICAL** / should not be included at the end of ${work_dir}, and work_dir should be an absolute path on the system.

2. Make an output directory for the analysis results of fastp, which helps remove adapter sequences from fastq files.

   $ mkdir -p ${work_dir}/fastp

3. Use fastp v0.19.6 to perform quality filtering of raw fastq reads.

   $ fastp --thread 8 --trim_front1 ${trim_front1} --trim_tail1 ${trim_tail1} --max_len1 ${max_len1} --trim_front2 ${trim_front2} --trim_tail2 ${trim_tail2} --max_len2 ${max_len2} -i ${work_dir}/R_BGI_L3_B1_D5_1.R1.fastq.gz -I ${work_dir}/R_BGI_L3_B1_D5_1.R2.fastq.gz -o ${work_dir}/fastp/R_BGI_L3_B1_D5_1.R1.fastq.tmp1.gz -O ${work_dir}/fastp/R_BGI_L3_B1_D5_1.R2.fastq.tmp1.gz -j ${work_dir}/fastp/R_BGI_L3_B1_D5_1.qf.json -h ${work_dir}/fastp/R_BGI_L3_B1_D5_1.qf.html

4. Make an output directory for the trimmed fastq files.

   $ mkdir -p ${work_dir}/trim_fastq

5. Use fastp v0.19.6 to perform adapter trimming of raw fastq reads.

$ fastp --thread 8 -l ${length_required} -q ${qualified_quality_phred} -u ${length_required1} --adapter_sequence ${adapter_sequence} --adapter_sequence_r2 ${adapter_sequence_r2} --detect_adapter_for_pe --trim_front1 ${trim_front1} --trim_tail1 ${trim_tail1} --max_len1 ${max_len1} --trim_front2 ${trim_front2} --trim_tail2 ${trim_tail2} --max_len2 ${max_len2} -i ${work_dir}/fastp/R_BGI_L3_B1_D5_1.R1.fastq.tmp1.gz -I ${work_dir}/fastp/R_BGI_L3_B1_D5_1.R2.fastq.tmp1.gz -o ${work_dir}/trim_fastq/R_BGI_L3_B1_D5_1_clean_R1.fastq.gz -O ${work_dir}/trim_fastq/R_BGI_L3_B1_D5_1_clean_R2.fastq.gz -j ${work_dir}/fastp/R_BGI_L3_B1_D5_1.at.json -h ${work_dir}/fastp/R_BGI_L3_B1_D5_1.at.html

6. Make an output directory for HISAT, which will perform RNA-seq alignment.

   $ mkdir -p ${work_dir}/hisat

7. Set the working directory for HISAT.

   $ cd ${work_dir}/hisat

8. Use HISAT v2.1 to map the trimmed reads to the reference genome.

   $ hisat2 -t -p 8 -x ${idx}/${idx_prefix} --pen-cansplice ${pen_cansplice} --pen-noncansplice ${pen_noncansplice} --pen-canintronlen ${pen_intronlen} --min-intronlen ${min_intronlen} --max-intronlen ${max_intronlen} --maxins ${maxins} --minins ${minins} --rna-strandness ${strandness} --un-conc-gz ${work_dir}/hisat/R_BGI_L3_B1_D5_1_un.fq.gz -1 ${work_dir}/trim_fastq /R_BGI_L3_B1_D5_1_clean_R1.fastq.gz -2 ${work_dir}/trim_fastq /R_BGI_L3_B1_D5_1_clean_R2.fastq.gz -S ${work_dir}/hisat /R_BGI_L3_B1_D5_1.sam

▲**CRITICAL** ${idx}/${idx_prefix} should be prepared in advance, with idx being the absolute path where the hisat-specific reference genome index is located, and idx_prefix being the index starvation prefix.

9. Make an output directory for Samtools, which helps convert SAM files to BAM files for to save storage space.

   $ mkdir -p ${work_dir}/samtools

10. Set the working directory for Samtools.

    $ cd ${work_dir}/samtools

11. Use Samtools v1.3.1 to convert SAM files to BAM files, sort and index the BAM files before read counting:

    $ samtools view -bS ${work_dir}/hisat/R_BGI_L3_B1_D5_1.sam > ${work_dir}/ samtools /R_BGI_L3_B1_D5_1.bam

    $ samtools sort -m 1000000000 ${work_dir}/samtools /R_BGI_L3_B1_D5_1.bam -o ${work_dir}/samtools /R_BGI_L3_B1_D5_1.sorted.bam

    $ samtools index ${work_dir}/samtools /R_BGI_L3_B1_D5_1.sorted.bam

Gene quantification

● TIMING 3 h

12. Make an output directory for the output files of StringTie and ballgown.

    $ mkdir -p ${work_dir}/ballgown/R_BGI_L3_B1_D5_1

13. Assemble transcripts for each sample. Use StringTie v1.3.4 to estimate transcript abundances and create table counts for ballgown.

    $ stringtie -e -B -p 8 -f ${minimum_isoform_abundance} -m ${minimum_length_allowed_for_the_predicted_transcripts} -a ${Junctions_no_spliced_reads} -M ${maximum_fraction_of_muliplelocationmapped_reads} -G ${gtf} –rf -o ${work_dir}/ballgown/R_BGI_L3_B1_D5_1/R_BGI_L3_B1_D5_1.gtf -C ${work_dir}/ballgown/R_BGI_L3_B1_D5_1/R_BGI_L3_B1_D5_1.cov.ref.gtf -A ${work_dir}/ballgown/R_BGI_L3_B1_D5_1/R_BGI_L3_B1_D5_1.gene.abundance.txt ${work_dir}/samtools /R_BGI_L3_B1_D5_1.sorted.bam

14. Ballgown readable expression output. Use the ballgown v2.20.0 package to process CTAB files calculated from StringTie for gene expression profiling (FPKM).

    $ ballgown ${work_dir}/ballgown/R_BGI_L3_B1_D5_1 R_BGI_L3_B1_D5_1.txt

▲**CRITICAL** Ballgown can process a single sample or merge the expression profiles of multiple samples.

15. Make and change the working directory for count expression output:

    $ mkdir -p ${work_dir}/count && cd ${work_dir}/count

16. Copy the file R_BGI_L3_B1_D5_1.gtf into the working directory:

    $ cp ${work_dir}/ballgown/R_BGI_L3_B1_D5_1/R_BGI_L3_B1_D5_1.gtf ${work_dir}/count

17. Prepare the file samplelist.txt for the prepDE.py3, the first column is the sample id, and the second column is the path to the gtf file:

    $ echo -e "R_BGI_L3_B1_D5_1\t${work_dir}/count/R_BGI_L3_B1_D5_1.gtf" > samplelist.txt

18. Count expression output. Use prepDE.py3 to process GTF files calculated from StringTie for gene expression profiling (count).

    $ python3 ${work_dir}/prepDE.py3 -i samplelist.txt -l 150

▲**CRITICAL** prepDE.py3 can also process a single sample or merge the expression profiles of multiple samples.

Quality control

●TIMING 2-8 h (Depending on the size of the fastq file)

19. Make an output directory for the analysis results of FastQC.

    $ mkdir -p ${work_dir}/fastqc

20. Use FastQC v0.11.5 to perform quality control analysis of raw FASTQ files.

    $ fastqc -t 8 -o ${work_dir}/fastqc ${work_dir}/R_BGI_L3_B1_D5_1.R1.fastq.gz

    $ fastqc -t 8 -o ${work_dir}/fastqc ${work_dir}/R_BGI_L3_B1_D5_1.R2.fastq.gz

21. Check the quality of the clean fastq files after filtering and trimming.

    $ fastqc -t 8 -o ${work_dir}/fastqc ${work_dir}/trim_fastq /R_BGI_L3_B1_D5_1_clean_R1.fastq.gz

    $ fastqc -t 8 -o ${work_dir}/fastqc ${work_dir}/trim_fastq /R_BGI_L3_B1_D5_1_clean_R2.fastq.gz

22. Make an output directory for FastQ Screen.

    $ mkdir -p ${work_dir}/fastq_screen

23. Set the working directory for FastQ Screen.

    $ cd ${work_dir}/ fastq_screen

24. Use FastQ Screen to extract the first 10,000 reads from the clean fastq files to detect whether the raw data are contaminated with other species, junction primers, etc.

    $ fastq_screen --aligner bowtie2 --conf ${fastq_screen_conf} --top 100000 --threads 8 ${work_dir}/trim_fastq /R_BGI_L3_B1_D5_1_clean_R1.fastq.gz

    $ fastq_screen --aligner bowtie2 --conf ${fastq_screen_conf} --top 100000 --threads 8 ${work_dir}/trim_fastq /R_BGI_L3_B1_D5_1_clean_R2.fastq.gz

▲**CRITICAL** ${fastq_screen_conf} should be prepared in advance, with fastq_screen_conf being the config files where the reference genome index of different species is located.

25. Make an output directory for Qualimap2.

    $ mkdir -p ${work_dir}/qualimap

26. Set the working directory for Qualimap2.

    $ cd ${work_dir}/ qualimap

27. Use Qualimap2 v2.3 to calculate the quality of the mapping process and quantify the reads mapped to different genomic regions:

    $ qualimap bamqc -bam ${work_dir}/samtools /R_BGI_L3_B1_D5_1.sorted.bam -outformat PDF:HTML -nt 8 -outdir ${work_dir}/qualimap /R_BGI_L3_B1_D5_1_bamqc --java-mem-size=32G

    $ qualimap rnaseq -bam ${work_dir}/samtools /R_BGI_L3_B1_D5_1.sorted.bam -outformat PDF:HTML -nt 8 -outdir ${work_dir}/qualimap /R_BGI_L3_B1_D5_1_RNAseq -gtf ${gtf} -pe --java-mem-size=32G

▲**CRITICAL** Qualimap takes longer to run and requires more computational resources, whereas the rnaseq QC module may not run successfully due to version issues.

28. Make a working directory for MultiQC.

    $ mkdir -p ${work_dir}/multiqc

29. Copying the QC results from FastQC, FastQ Screen, and Qualimap into the working directory of MultiQC.

    $ cp -r ${work_dir}/fastqc ${work_dir}/fastq_screen ${work_dir}/qualimap ${work_dir}/multiqc

30. Set the working directory for MultiQC.

    $ cd ${work_dir}/multiqc

31. Use MultiQC v1.8 to merge the QC results from FastQC, FastQ Screen, and Qualimap

    $ multiqc .

▲**CRITICAL** All files in ${work_dir}/multiqc should be unzipped.