Bulk RNA-Seq Analysis and DeSeq2 Pipeline (BAM creation)

Tools: cutadapt 2.5, FastQC v0.11.6, STAR 2.7.3a, samtools 1.11

Ulimit -n (should be 65535)

General:

This pipeline is based on two-lane dual read fastq fasta files from bulk RNA-seq experiments. It performs the following steps.

- STAR index (using GRCh38.p13.genome.fa and gencode.v37.annotation.gtf)
- Fastqc concatenation from two lane (L001 and L002) samples for each read file
- Adapter trimming using cutadapt
- Fastqc for qc
- STAR alignment using 2 pass STAR mode with -–outSAMattributes XS for downstream differential analysis using StringTie2 and DeSeq2

How to run:

- Create a new directory, say, **new_data**
- Copy all samples (raw reads) from RNA-seq experiment into **new_data** directory.
- If creating star index from scratch (move on to next step if using previously generated STAR Index files)
    - copy GRCh38.p13.genome.fa and gencode.v37.annotation.gtf or genome fasta and annotation files of your choice. Please make sure that genome fasta and annotation files should be from the same source (bothe from gencode or UCSC or any other source)
    - Comment this line
        - star_index=../../RawData/star_index
    - Uncomment this line:
        - #STAR --runMode genomeGenerate --runThreadN 32 --limitBAMsortRAM=322122547200 --genomeDir star_index --genomeFastaFiles GRCh38.p13.genome.fa --sjdbGTFfile gencode.v38.annotation.gtf --sjdbOverhang 149
    - --limitBAMsortRAM=322122547200 currently allocates 300GB RAM, **please increase it if code crashes at this line.**
- Using previously generated star index files
    - Please update below line with the proper location of index files
        - star_index=../../RawData/star_index
- Use following command to run your job

    - bash `run_rna_seq.sh`
- Computational Resource
    - Please allocate sufficiently enough resources to make sure that job runs smoothly.
        - As each task has different resources requirements, I would allocate max(memory for different tasks) and max(cpu's for different tasks)
        - As star_index generation (if creating star index files from scratch) is memory and cpu extensive, It is better to run it separately. For this simply add following line at the end of this line: STAR --runMode genomeGenerate --runThreadN 32 --limitBAMsortRAM=322122547200 --genomeDir star_index --genomeFastaFiles GRCh38.p13.genome.fa --sjdbGTFfile gencode.v38.annotation.gtf --sjdbOverhang 149
        - exit 1 and once it is complete, comment above line and this line and rerun.

DESeq2 Pipeline for 150bp RBP date sets

Useful Link: http://ccb.jhu.edu/software/stringtie/index.shtml?t=manual, https://bioc.ism.ac.jp/packages/2.14/bioc/vignettes/DESeq2/inst/doc/beginner.pdf
DESeq2 installation: https://bioconductor.org/packages/release/bioc/html/DESeq2.html
Stringtie2 installation: https://anaconda.org/bioconda/stringtie

This pipeline is based on stringtie2 generated transcripts for each RBP sample (and controls) that were aligned using STAR aligner (all BAM/BAI from STAR and stringtie2 gtf files for each RBP and control samples are copied in indiproteomic folder).
Following steps are performed on stringite2 generated gtf files (for 6 TDP43 and 12 controls) to prepare input files for DESeq2 run

- Stringtie2 generated gtf files for 12 control and 6 TDP43 samples were merged using stringite2's merge option to generate stringtie_merged_nttdp.gtf (files merge_script.sh with samples_nt_tdp.txt, later file contains gtf file names generated from stringtie2).
- re-estimated abundances for each sample using stringtie2 -e options (merge_script_e.sh script, bam files and stringtie_merged_nttdp.gtf file from last step) which uses stringtie_merged_nttdp.gtf file (generated in last step) to generate "sample".merged.gtf, where "sample" is the sample name (control and tdp samples)
- Prepare read count information table (transcript_count_matrix.csv) for 12 control and 6 TDP43 samples using prepDE.py3 (requires input_prepDE.txt file containing sample file names and their location)

- Run DESeq2 (using run_deseq2.R file) with inputs:
    - Transcript_count_matrix.csv
    - Metadata file (new_samples.csv). It is a two column files, col1: sample names (same as first row of transcript_count_matrix.csv file) col2: condition (in current example we have 12 control and 6 TDP43).

Note:

- **A script called prepare_deseq2_input.sh files is also copied in the gbucket to run all steps at once and then use outputs to run DESeq2.**
- **DESeq2 can also be directly run (using run_deseq2.R) using provided gene_count_matrix.csv and new_samples.csv files which are generated using 12 control and 6 TDP43 samples from 150 bp RBP datasets.**
- **Please change file names accordingly in the scripts and input files.**