BPSM ICA1 submission
B217754

Git repository details

Repository address: https://github.com/B217754-2022/ICA1.git

CCRYPT password for B217754-2022.ICA1.tar.gz.cpt: We_love_RNASeq

Code walkthrough

This code should enable the user to take RNAi pair-ended data through a workflow including quality checks using fastqc, alignment using bowtie2, and gene expression quantification using bedtools/samtools, to identify differentially expressed genes in experimental setups with multiple organism lineages and testing conditions. The programme could accommodate data from additional cell lines in the future (i.e. 'Clone 3').

#7 Copying the ICA data and Pair-ended RNAseq sequence data from the server to the present working directory

#12 we need to run the fastqc command on all of the files in the data directory, then output the resulting reports to a format we can analyse.

#14 -1

#18 Create a folder in the current directory for the fastqc output to be held

#22 Run fastqc on .fq.gz in the fastq folder of the downloaded directory, with -o specifying the newly created 'FASTQC_OUTPUT' directory as the destination.

#24 --noextract is used to prevent output uncompressing, -t6 specifies that 6 threads should be used to process fastqc, -q suppresses all progress messages to make the interface more appealing to the user.

#25 These are not required, deleting *.html

#31 For each fastqc, use grep to identify the quality categories with WARN or FAIL (except per base sequence content, which is typically FAIL for RNAseq because the read generation method contains anomalous base content at the start of the strands). The quality flag lines are then written to a new reads_quality_warnings_list.txt and print a warning message.

#53 -2

#57 The reference genome is fasta.gz, so needs to be unzipped into .fasta, then used by bowtie2-build to make an index for bowtie alignment

#68 Aligning reads to the indexed reference genome. --very-fast reduces the accuracy of the alignment process in exchange for speed of alignment, -local aligns the reads locally, potentially clipping the ends of the reads if this improves alignment score. -p16 specifies that 16 parallel threads should be used to generate alignments. -S is to generate a .sam output

#76 -3

#80 convert the output from sam to bam format using samtools to make the output readable for the machine. -S specifies that the input is a .sam format, -b specifies that the output should be in .bam format

#84 -4

#88 Using bedtools to count aligns for exons

#89 cd to bam location

#94 -a and -b specify the input files for the comparison, -c generates a count for the number of overlapping features from the -b file to the -a file

BPSM ICA1 submission
B217754

#107 Grouping read files based on their experimental classification (not elegant but creates the intended files!) These files could be used for crude comparison of alignment and count numbers between experimental groups if the user chooses. (See end of section)

#113 Filtering by Sample Type

#136 Filtering by Time elapsed

#159 Filtering by Treatment

#179 A list of the total align counts per category group can be read at ../groupCounts.txt. This is not informative with regards to the experimental investigation, however, is presented here to demonstrate that the programme has functioned up to this point.

#182 At this point, the use of --rg-id and -rg in bowtie2 might have allowed me to identify which groups had caused alignment (which conditions allowed for gene expression), however, I was not able to get this code and its parameters to function. The process would involve using a grep pipe to search through the output of bedtools with parameters -wa -wb (which would display the earlier cited read group identifiers from bowtie2's --rg-id parameters) to find the associated read information. Essentially, this would bridge the gap between the details file and the alignment/ hits files generated.

#183 Alternatively, I would have preferred to use SQL to link the Tco.fqfile, count.txt and TriTrypDB-46_TcongolenseIL3000_2019.bed files to associate the counts with the relevant experimental group.

#185 Were I to have reached the end of the coding workflow, the group-wise comparison of gene expression from experimental data would have been possible.

#186 I would have compared the gene expression levels of clones 1 and 2 (different T.congo. lines with RNAi knockout) to WT which is expected to have nominal gene expression. This would be (2 groups * 3 time points vs. WT time points) 6 comparisons.

The ability to use the crude count data might give users the ability to compare experimental groups. In the future, I would programme an interactive code to ask which groups are desired for comparison and display the count numbers for the user. This would have to take into account replicates, so some filtering would be required to eliminate error here.

Difficulties faced during work

The main challenge I encountered in this coursework was the correct application of syntax in bash, specifically in the use of for loops such as the loop beginning at line 37. The source of the issue was that I had referred to the wrong subject in the opening line of the for command, specifying too deep into a directory by using *fastqc and rendering the rest of loop useless. After some guidance from lecturers, I was able to identify the problem.

Similarly, when using bowtie2, I employed a for loop which created some issues. My solution to this was crude in my opinion, as I simply relocated working directory to the destination of the required .bam and .bed files.

Ultimately, I was not able to complete the programme to fulfil the desired functionality. The main hurdle to this was the steep learning curve presented by this coursework, and more specifically, the barrier to any further progress was my inability to find a way to attribute the align counts to the appropriate read files (thus identifying the treatment group responsible). This barrier would have been overcome given enough research and time.

I would have preferred that the final programme was more concise, however I am satisfied with the progress in programming skills brought on by this challenging coursework.