03-701 Practical Computing for Biologists

Final Project

Harihara Prakash
hprakash@andrew.cmu.edu

Development of a linux-based pipeline for the analysis of RNA-Seq data from different biological experiments.

1. Introduction:

With the reduction in Next-Generation Sequencing costs and the improvement of sequencing experiments in terms of time, resources and quality, there has been a rise in the accumulation of biological data over the past decade. One of the most studied and carried out sequencing techniques is the RNA-Seq approach. RNA-Seq provides large amounts of information regarding transcriptomic data, gene expression and gene transcript counts.

This technique has redefined the way scientists understand the effects of various conditions on the same organism, by allowing analysis at the molecular level. The degree of gene expression of a population subjected to a certain condition can be compared to a population subjected to no conditions. On comparing the expression data of the two populations, valuable biological and genetic conclusions may be inferred.

Taking inspiration from the instructors pre-approved project ideas (Project 2.3), I aimed to build a linux-based pipeline capable of carrying out RNA-Seq analysis for my final project. My goal was to create a simplified RNA-Seq pipeline using publicly available bioinformatics tools. The workflow would follow the standard RNA-Seq pipeline, which involves data retrieval, quality check of sequences, alignment to the reference genome and gene quantification. For the illustration of the working of this pipeline, I have used yeast RNA-Seq data publicly available on the European Nucleotide Archive (ENA) (Project Accession: PRJEB35903).

2. Approach

The bash script of the RNA-Seq pipeline was compiled and edited using Notepad++. To create the RNA-Seq pipeline, I first identified the potential linux-based tools required to complete an RNA-Seq analysis. This includes the retrieval of datasets, quality check of the sequences, alignment of the sequences to a reference genome and finally quantification of the reads. Once I identified the tools, I downloaded them into my linux directory, and used them to carry out the RNA-Seq analysis of the yeast sample data.

The tools used in this linux-based pipeline include the following:

- FastQC: For the quality check of sequence files
- Trimmomatic: For the trimming of sequences with a poor quality score
- HISAT2: For the alignment of the sample dataset to the reference genome
- FeatureCounts: For the quantification of gene expression and read count

3. Commands

The RNA-Seq bash script contains the commands required for the entire pipeline in sequential order and is attached with the submission. (RNASeqPipeline.sh)

The bash script file contains a complete and detailed explanation of each command used in the pipeline, and the major commands used are highlighted below.

3.1 Quality Check:

$ fastqc Project/ena_files/ERR3772427_1.fq Project/ena_files/ERR3772427_2.fq -o Project/ena_files

//Reads two input files (forward and reverse sequence) and returns Quality Check reports in .html format. (Attached with the submission)

3.2 Trimming:

$ java -jar ~/Trimmomatic-0.39/Trimmomatic-0.39/trimmomatic-0.39.jar PE Project/ena_files/ERR3772427/ERR3772427_1.fastq Project/ena_files/ERR3772427/ERR3772427_2.fastq ERR3772427_1_forward_paired.fq.gz ERR3772427_1_forward_unpaired.fq.gz ERR3772427_2_reverse_paired.fq.gz ERR3772427_2_reverse_unpaired.fq.gz ILLUMINACLIP:Trimmomatic-0.39/Trimmomatic-0.39/adapters/TruSeq3-PE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:25

// Reads two input files (forward and reverse sequence) and returns four files (forward paired, forward unpaired, reverse paired and reverse unpaired)

3.3 Alignment:

$ hisat2 -q -x hisat2-2.2.1/r64/genome -1 Project/ena_files/ERR3772427_1_forward_paired.fq -2 Project/ena_files/ERR3772427_2_reverse_paired.fq -S hisat2-2.2.1/yeastaligned.sam

// Reads the reference genome index file, forward paired file and reverse paired file and returns a .SAM file called yeastaligned.sam

Converting SAM to sorted BAM file:

$ samtools view -S -b hisat2-2.2.1/yeastaligned.sam > hisat2-2.2.1/yeastaligned.bam

$ samtools sort hisat2-2.2.1/yeastaligned.bam -o hisat2-2.2.1/yeastaligned_sorted.bam

//Takes the SAM file and creates an unsorted BAM file, then takes that and creates a sorted BAM file

3.4 Quantification:

$ featureCounts -S 2 -a Saccharomyces_cerevisiae.R64-1-1.107.gtf -o Project/yeast_featurecounts.txt hisat2-2.2.1/yeastaligned_sorted.bam

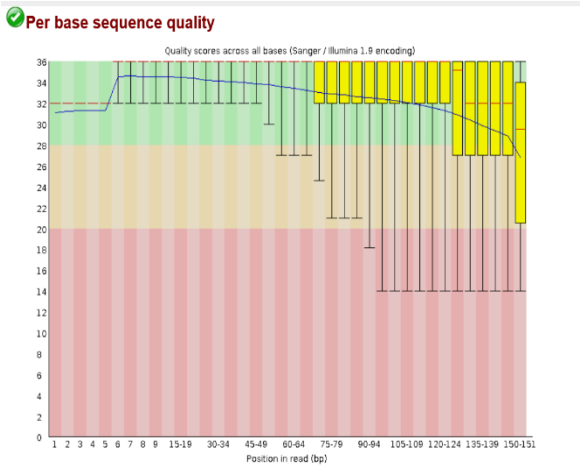
// Takes a .gtf file (genomic feature information) and the sorted .BAM file, completes the quantification analysis and returns a .txt file with the complete quantification information.

4. Results

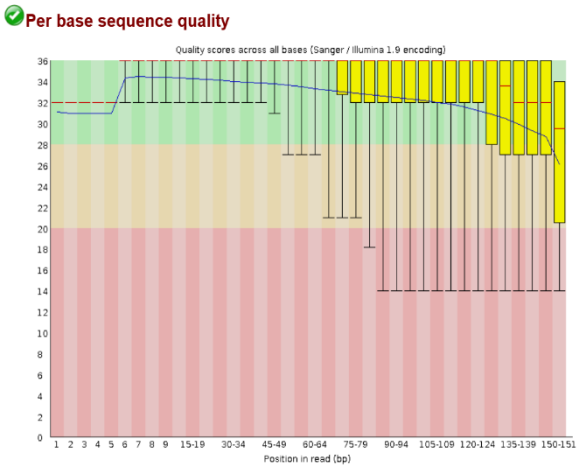
The linux-based RNA-Seq pipeline was run for a sample yeast dataset and the results can be represented as shown.

4.1 Initial quality check (FastQC)

The quality check of the forward and reverse sequence files indicated that some reads present in the sequence were in the yellow zone, which implies that trimming of the poor reads must be carried out. The results were presented as a .html file (ERR3772427_1_fastqc.html and ERR3772427_2_fastqc.html).



ERR3772427_1.fastq



ERR3772427_2.fastq

These sequence files were sent to Trimmomatic for the trimming of the poor reads and to get the paired forward and reverse sequence files.

4.2 Trimming of poor reads (Trimmomatic)

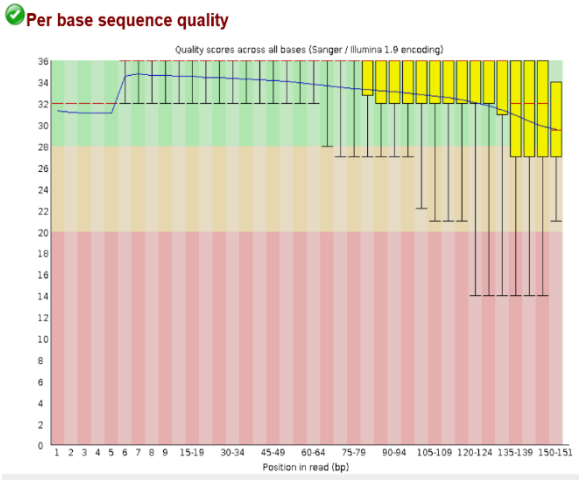
The forward and reverse sequence files were input to Trimmomatic and the Unpaired Forward, Paired Forward, Unpaired Reverse and Paired Reverse sequence files were obtained as the output from trimming.

The results of the trimming can be seen below.

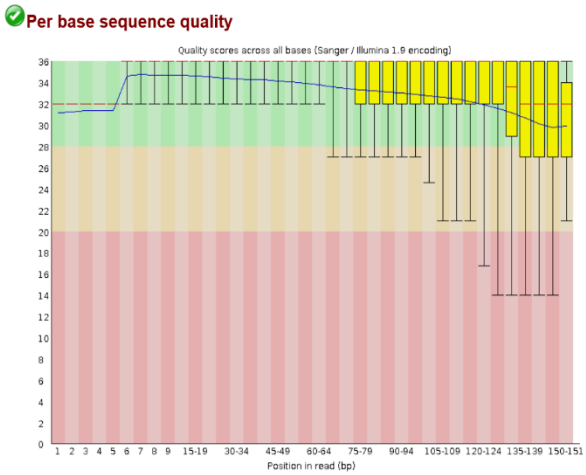
```
Input Read Pairs: 8755573 Both Surviving: 8422160 (96.19%) Forward Only Surviving: 274560 (3.14%) Reverse Only Surviving: 41084 (0.47%) Dropped: 17769 (0.20%)
TrimmomaticPE: Completed successfully
```

4.3 Quality check of paired forward and reverse sequences (FastQC)

The quality check of the new paired forward and reverse sequence files now indicates that all reads in the sequence file are in the green zone, i.e, the reads are of good quality and can be sent to alignment.



ERR3772427_1_forward_paired.fq



ERR3772427_2_reverse_paired.fq

4.4 Alignment of sequences to reference genome (HISAT2)

The paired forward and reverse sequences are now aligned to the yeast reference genome index file, which has been downloaded from the HISAT2 website using wget. The complete alignment of the sequence files will result in an output .SAM file (yeastaligned.sam)

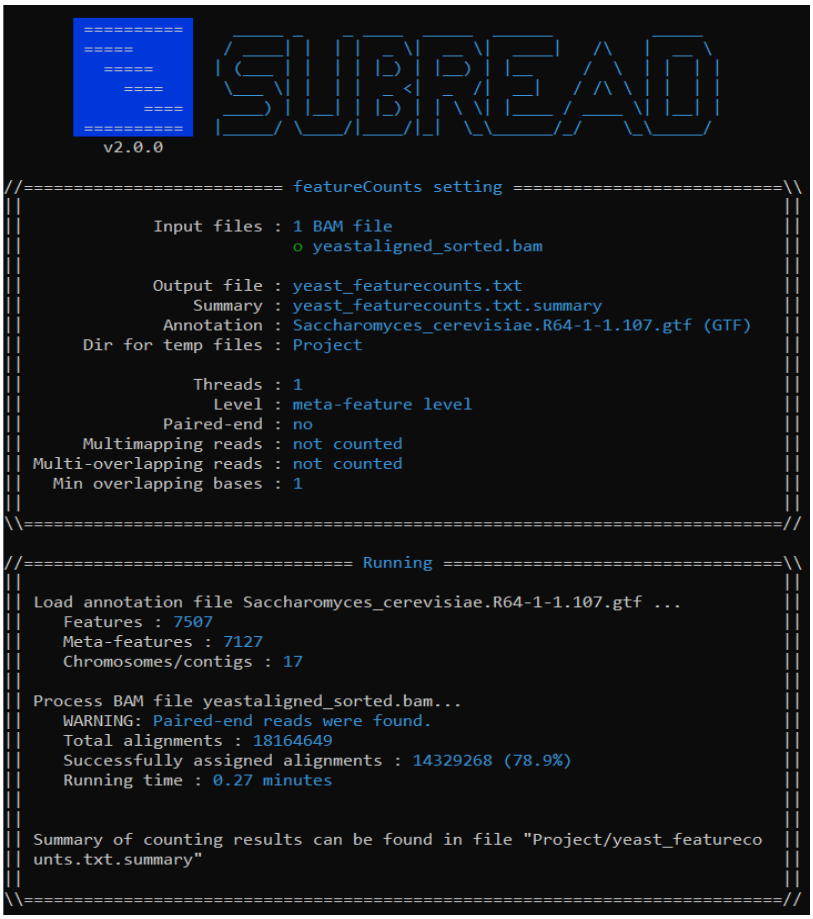
The SAM file is then converted to an unsorted BAM file (yeastaligned.bam) using the samtools command (See RNASeq.sh). The BAM is then converted to a sorted BAM file (yeastaligned_sorted.bam). The sorted BAM file is now ready for quantification.

```
8422160 reads; of these:
  8422160 (100.00%) were paired; of these:
    450349 (5.35%) aligned concordantly 0 times
    7650769 (90.84%) aligned concordantly exactly 1 time
    321042 (3.81%) aligned concordantly >1 times
    ----
    450349 pairs aligned concordantly 0 times; of these:
      24965 (5.54%) aligned discordantly 1 time
    ----
    425384 pairs aligned 0 times concordantly or discordantly; of these:
      850768 mates make up the pairs; of these:
        688601 (80.94%) aligned 0 times
        153107 (18.00%) aligned exactly 1 time
        9060 (1.06%) aligned >1 times
95.91% overall alignment rate
```

4.5 Quantification of gene expression (featureCounts)

The sorted BAM file is sent to featureCounts for the gene expression quantification. The file is quantified against a .gtf file which contains the coordinates of genomic features of the yeast data. This file is required for quantification and is downloaded from the Ensembl server using wget. (See RNASeq.sh)

The quantification results in an output .txt file (yeast_featurecounts.txt) and a yeast_featurecounts.txt summary file. (Files attached)



4.6 Most expressed genes and genes with zero expression.

The top 5 genes with highest expression can be identified by opening the yeast_featurecounts.txt file and sorting it based on the gene count using the following command. (See RNASeq.sh)

```
(base) hara@HaraXi:~$ cat Project/yeast_featurecounts.txt |cut -f 1,7 | sort -k 2 -r -g | head -5
YER065C 227627
YKR097W 169503
YAL054C 126319
YOR374W 123019
YER024W 115222
```

As we can see, gene YER065C has the highest expression followed by gene YKR097W.

The number of genes with no expression can be identified using the following command. (See RNASeq.sh)

```
(base) hara@HaraXi:~$ cat Project/yeast_featurecounts.txt | cut -f 1,7 | tail -n +2 | grep -w 0 | wc -l
497
```

The number of genes in the dataset with zero expression is 497 genes.

This RNA-Seq pipeline can be translated to larger sets of data which have been derived from not just yeast studies, but of various organisms. It is a relatively fast pipeline capable of generating invaluable quantification data for considerably large datasets in less than 30-40 minutes.

Using this pipeline for a sample yeast RNA-Seq dataset, I was able to identify the genes which are the most expressed and find the number of genes which have no expression at all.

A detailed explanation of the pipeline script can be found in RNASeq.sh.

5. Further expansions

Due to system and time constraints, it was not possible to carry out an integral part of RNA-Seq analysis, which is the visualization of the quantified data using DESeq2. This step of the RNA-Seq pipeline will be added and will allow the comparative analysis of organism gene expression using graphs and heatmaps.

In addition, the above pipeline was tested against a single sample of yeast data retrieved from ENA. For practical RNA-Seq analysis, we would be required to download the sample data under treatment and control conditions and run the RNA-Seq analysis for both the datasets, followed by comparison of gene expression between the two datasets.

While many quantification and alignment tools exist, I used the most popular tools. However, there is the likelihood that faster and simpler tools exist and I intend to improve the pipeline by substituting these tools where possible to improve the speed and efficiency of the pipeline.

6. References

https://useast.ensembl.org/Saccharomyces_cerevisiae/Info/Index

http://sgd-archive.yeastgenome.org/sequence/S288C_reference/genome_releases/

https://www.ebi.ac.uk/ena/browser/view/PRJEB35903?show=reads

http://daehwankimlab.github.io/hisat2/download/#s-cerevisiae

Andrews, S. (2010). FastQC: A Quality Control Tool for High Throughput Sequence Data [Online]

Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: A flexible trimmer for Illumina Sequence Data.

Yang Liao, Gordon K. Smyth, Wei Shi, featureCounts: an efficient general purpose program for assigning sequence reads to genomic features, Bioinformatics, Volume 30, Issue 7, 1 April 2014, Pages 923–930, https://doi.org/10.1093/bioinformatics/btt656

Zhao, S., Zhang, B., Zhang, Y., Gordon, W., Du, S., Paradis, T., Vincent, M., & Schack, D. v. (2016). Bioinformatics for RNA-Seq Data Analysis. In (Ed.), Bioinformatics - Updated Features and Applications. IntechOpen. https://doi.org/10.5772/6326