QAA_report

Davin Marro

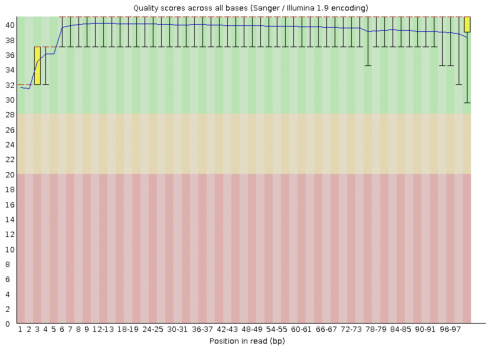
2022-09-08

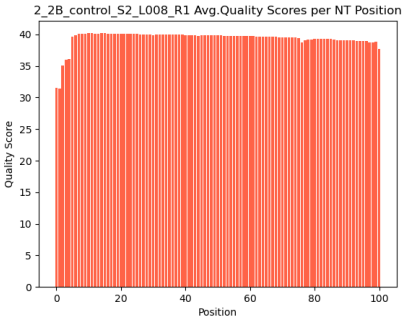
Introduction:

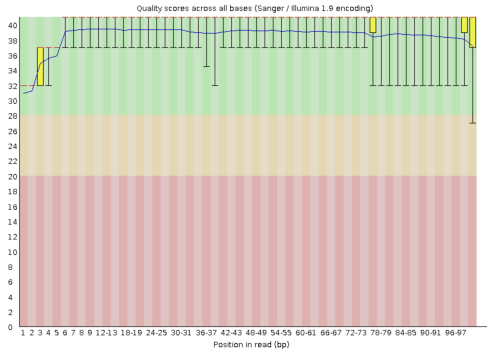
The goals of this QAA assignment is to assess RNA-seq data. This involves the utilization of FastQC, to assess the overall quality of the RNA-seq data. Afterwords, cutadapt is used to trim adapter sequence from reads, and trimmomatic to filter reads based on quality. This will produce an adapter-trimmed, quality-trimmed RNA-seq dataset for the two target files assigned. (Undetermined_S0_L008, 2_2B_control_S2_L008) These datasets are then alligned to a STAR-generated genome database, and their feature-mapping is finally quantified via htseq-count. The resulting information provides context for the strandedness of the RNA-seq data, quality of reads, and whether directionality has an effect on the quality distribution for reads.

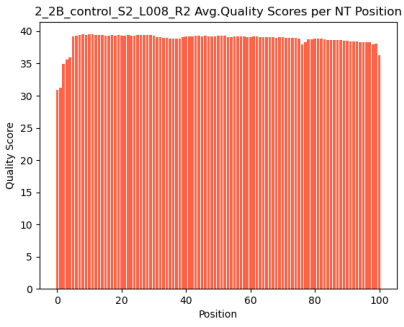
Part 1: Read Quality Score Distributions

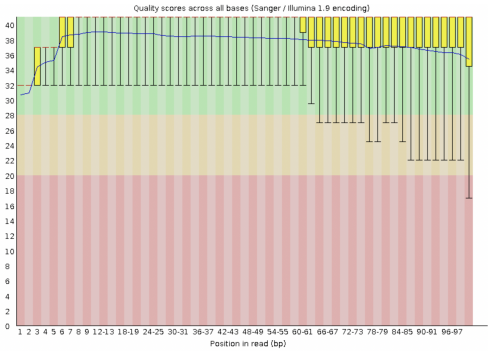
This portion of the assignment will involve comparing quality score distribution for the two target files. The generated distributions to be compared are sourced from either running FastQC , or dist.py. (a pre-written python script which plots the distribution of reads based on a running-sum method) Below, you will find the side-by-side comparison of fastqc quality distributions (left), and dist.py distributions (right).

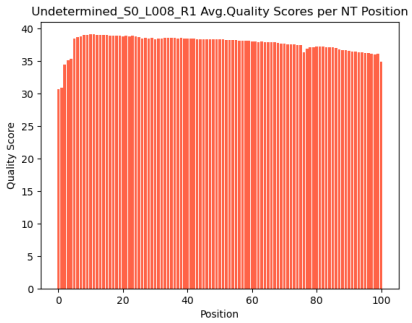


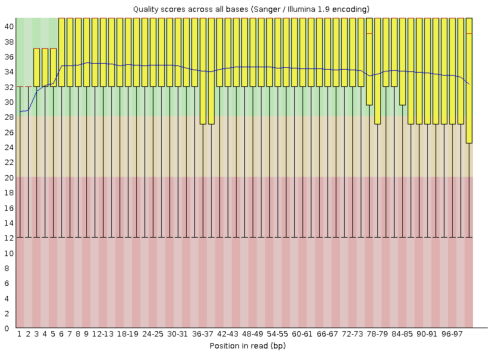


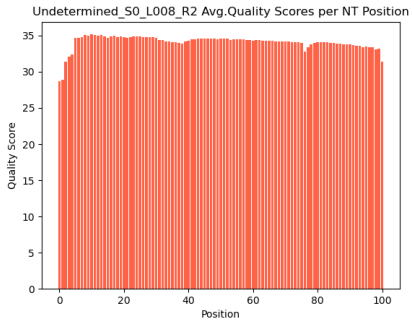












For both R1 and R2 distributions for target files, the quality distribution is mostly conserved. The trend of quality across the length of reads is also conserved. Overall, FastQC is faster, and contains more information - there seems to be underlying algorithms to detect data attributes such as sources of overexpressed sequences. This pre-fed information that allows it to detect these attributes likely reduces the need for stepwise calculations to encode phred scores whith dist.py , a process which takes up a large portion of runtime. Fastqc error bar attribute is useful when assessing the overall confidence given quality score per base. 2_2B_control_S2_L008 libraries have higher quality than Undetermined_S0_L008.

Part 2: Adaptor Trimming Comparison

This portion of the assignment involves trimming known adapters from the RNA-seq reads, and filtering reads based on quality using cutadapt(4.1), and trimmomatic(0.39). The adapters given were:

```
R1: AGATCGGAAGAGCACACGTCTGAACTCCAGTCA
R2: AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT
```

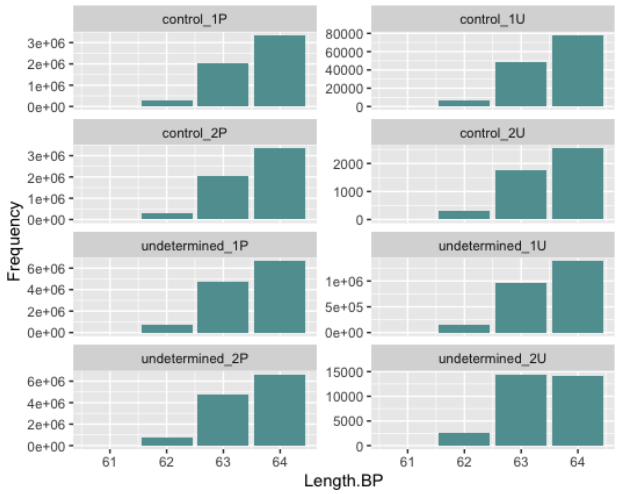
The presence of these adapters could potentially be confirmed in FastQC in part 1, when over expressed sequence is identified as illumina adapter. Using unix, however- the presence can be confirmed by the following command:

```
cat FILE | grep '^AGATCGGAAGAGCACACGTCTGAACTCCAGTCA' | wc -l
cat FILE | grep '^AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT' | wc -l
```

Cutadapt was utilized in order to trim these adapters from RNA-seq libraries. Running the sbatch script cutadapt.sh, the following summaries of trimming executed were produced:

```
**UNDETERMINED**
=== Summary ===

Total read pairs processed:         14,760,166
  Read 1 with adapter:                 543,021 (3.7%)
  Read 2 with adapter:                 607,660 (4.1%)
Pairs written (passing filters):    14,760,166 (100.0%)

Total basepairs processed: 2,981,553,532 bp
  Read 1: 1,490,776,766 bp
  Read 2: 1,490,776,766 bp
Total written (filtered):  2,968,880,881 bp (99.6%)
  Read 1: 1,484,746,362 bp
  Read 2: 1,484,134,519 bp
----------------
**CONTROL**
=== Summary ===

Total read pairs processed:          5,830,665
  Read 1 with adapter:                 423,128 (7.3%)
  Read 2 with adapter:                 473,368 (8.1%)
Pairs written (passing filters):     5,830,665 (100.0%)

Total basepairs processed: 1,177,794,330 bp
  Read 1:   588,897,165 bp
  Read 2:   588,897,165 bp
Total written (filtered):  1,160,435,631 bp (98.5%)
  Read 1:   580,298,948 bp
  Read 2:   580,136,683 bp
```

Next, trimmomatic was run to quality filter the adapter-trimmed reads produced by cutadapt. This was done using trimmomatic.sh. This resulted in the production of four output files per dataset. One file for paired reades, and one for unpaired reads - per direction (forward and reverse). The files were read into lendist.py in order to output the distribution of lengths. This data was then used to produce the following plot in Rstudio using ggplot:



Based on the prior observation of quality given directionality of reads from part 1. The assumption was

that reverse reads would be trimmed more extensively than forward reads, due to a noticeably lower overall quality score in the reverse reads of the undetermined file.However, the above figure seems to show that reverse reads have lower distributions of trimmed reads when compared to forward reads, despite having lower quality. Perhaps this is due to the range of quality scores being small, relative to the cutoff settings enforced by timmomatic.

Part 3: Alignment and Strand-Specificity

This section of the assignment involves generating a mouse genome database via STAR, and aligning the adapter-trimmed, quality-filtered reads from part 2, to said genome (also using STAR). After alignment, htseq-count will be run to determine the distribution of reads mapped to features across the generated mouse genome database. (assembly via ensembl)

For generating the genome databse, mouse genome assembly and gtf annotation files were downloaded from ensemble (build 107). These files were put through ./star.gen.sh to generate the database, and then the filtered reads from part 2 were put through ./star.align.sh to generate the alignment output SAM file.

The alignment output SAM file is then put into ./map.py which will read the bitflag per record and tally mapped vs. unmapped reads out of the total alignments. The result is as follows:

```
./map.py -f control.out.sam
Reads Mapped 11022029
Reads unmapped 283047
Reads 12247712

./map.py -f undetermined.out.sam
Reads Mapped 15497385
Reads unmapped 8822757
Reads 25438002
```

The distribution of mapped features are further examined using htseq-count, which takes the alignment output SAM files and references each alignment to the gtf genome annotation file downloaded from ensembl. This will allow for a list of mapped features, and IDs, as well as a final sumarry of mapped,unmapped,and ambiguous reads. The following are

```
Control: Stranded = YES
__no_feature    4863143
__ambiguous 8153
__too_low_aQual 4645
__not_aligned   138926
__alignment_not_unique  420119

Control: Stranded = Reverse
__no_feature    294864
__ambiguous 82105
__too_low_aQual 4645
__not_aligned   138926
__alignment_not_unique  420119

Undetermined: Stranded = YES
__no_feature    6925808
__ambiguous 5545
__too_low_aQual 76139
__not_aligned   4370378
```

```
__alignment_not_unique  480992

Undetermined: Stranded = Reverse
__no_feature    631595
__ambiguous 118913
__too_low_aQual 76139
__not_aligned   4370378
__alignment_not_unique  480992
```

Based on the information provided by htseq-count, testing for strandedness, it can be assumed that the RNA-seq libraries are stranded, as when the condition for strandedness is true in Htseq, the number of ambiguous reads is significantly lower than when the reverse condition is applied. 8153 ambiguous reads for stranded control library, vs. 82105 ambigious for reverse, and 5545 ambiguous for stranded undetermined library vs. 118913 for reverse. This implies that the number of ambiguous reads decreases, when the strandedness conditional property accurately represents the RNA-seq strandedness.

Conclusion

In conclusion this assignment examines the relationship between directionality and quality, and propensity to be trimmed. In addition to this, the alignment of trimmed and filtered reads was carried out and underwent quantification analysis which allowed for determining the strandedness of the RNA-seq libraries chosen for this particular assignment. Overall, an assessment of initial quality, improved quality, and the effects on alignment- and how accurately the experimental data represents the canonical genome.

Note: supplementary information can be found at https://github.com/Tripfantasy/QAA .