Chapter 2. Quality Control: Determining the Quality of Your Data and Cleaning Your Data

Software Tools: FastQC, MulitQC, Trimmomatic

Input files: Your fastq files
Output files: Trimmed fastq files, Reports

RNA-seq Data (Illumina platform) - Fastq files

Four rows per read:
1. The read identifier
2. The sequence (4 bases plus "N" for undefined base)
3. The read position (+/-)
4. The sequencing read quality values

Example of a read in a fastq file:
@SEQ_ID (sequence identifier)
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+ (plus strand)
!"*((((***+))%%%++)(%%%%).1***-+*"))**55CCF>>>>>>CCCCCCC65 (quality values)

+ = **plus strand**. This means the read sequence is in the same orientation as the reference genome or the original DNA fragment (- = is reverse: reverse complement of the reference genome or the original DNA fragment).

@@@ 1. FastQC on Savio: @@@

Determine if you have paired-end reads or single reads:
We typically have Illumina paired-end reads. Recent reads are phred33, but best to ask the sequencing center; unlikely to have phred64.

Use fastQC on your fastq files before and after performing Trimmomatic (cleaning your reads).

Make sure that you have performed the following commands once (may have been done on a previous login):

Create a new conda environment and install fastQC:

$ conda create -n fastqc_env -c conda-forge -c bioconda fastqc

$ conda activate fastqc_env

To verify the installation:

$ fastqc --version

To run fastQC:

Be sure that you activated your conda environment:

$ conda activate fastqc_env

Because you installed fastQC with Conda, you can be in any directory to run it. Suppose you uploaded your fastq.gz files (typically gunzipped) onto Savio and then moved them into your scratch directory's "mouse_data" directory and then to your "fastq" directory (nested directories):

$ mv filename.fastq.gz /global/scratch/users/your-username/mouse_data/fastq

Move to this folder:
$ cd /global/scratch/users/your-username/mouse_data/fastq

Check what's there:
$ ls -la (you should see all your data - files such as sample1_1.fastq.gz or sample1_1.fq.gz; _1 is forward strand and _2 is reverse strand; if you see files with extension .fq.gz, then change the extension on your commands as appropriate)

Make an output directory for your fastQC results:
$ mkdir /global/scratch/users/your-username/mouse_data/fastqc_results

To run fastQC on a single sample (example: ctrl_rep1_1.fastq.gz):

$ fastqc ctrl_rep1_1.fastq.gz -o /global/scratch/users/your-username/mouse_data/fastqc_results (all one line)

If you look at the contents of your fastqc_results folder, you should see:
ctrl_rep1_forward_fastqc.html, ctrl_rep1_forward_fastqc.zip
where .html is your quality control report and the .zip file are the images in the report. Download the .html file to your local computer and click on it to see it.

To run fastQC on all samples:
$ fastqc *.fastq.gz -t 8 -o /global/scratch/users/your-username/mouse_data/fastqc_results
where -t 8 means use 8 threads instead of 1 since you have a lot of data and -o indicates the output folder.

Ideally, if you are running fastQC on all your samples at once, you should be using a SLURM file. See https://github.com/elinorv21/RNA-Seq_workshop/ for the SLURM file: fastqc.sh.

Download this file ("$" and ">" are cursors so you don't type them; be sure to type the "\"):

$ wget -O fastqc.sh \ (type this and then press return)
> https://raw.githubusercontent.com/elinorv21/RNA-Seq_workshop/main/fastqc.sh (then press return)

Edit it as appropriate, then submit it:

$ sbatch fastqc.sh

Savio replies with the job number which you can use to track your job's progress.

@@@ 2. FastQC on the Galaxy website: @@@

The Galaxy website: https://usegalaxy.org/ (Storage quota: 250GB)

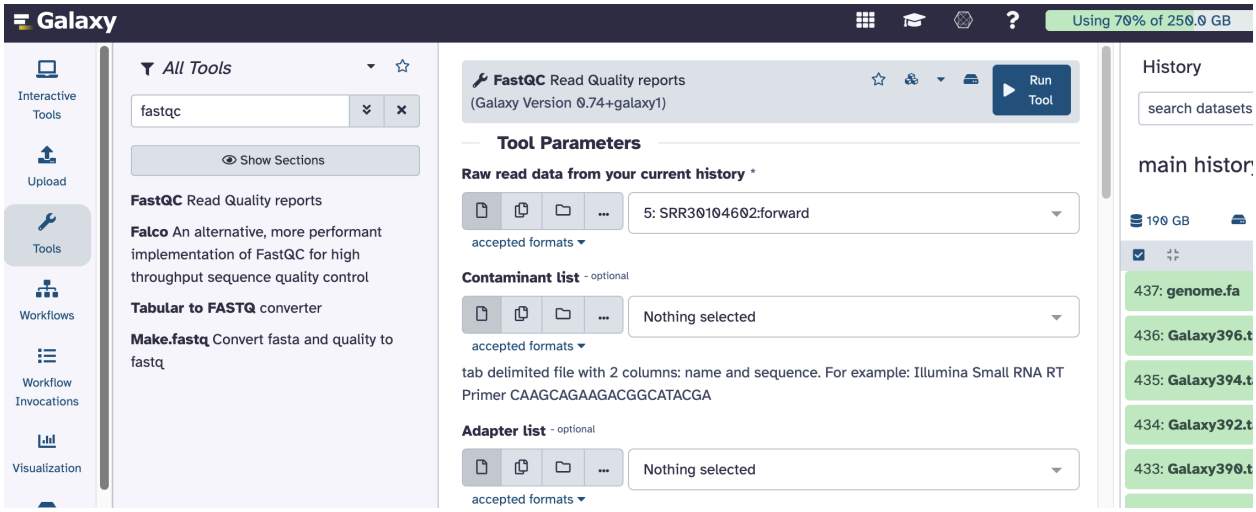


Figure 1. The Galaxy website set up to apply the tool fastQC to a single sample taken from online (SRR30104602_1.fastq).

1. Upload your file
2. Search for fastQC in the toolbar
3. Click “Run Tool” - see report in “main history” on the right.

Example of a FastQC Report:

Plots and Tables for FASTQC for a given sample:

1) Basic Statistics - table

GC bias is a known issue with the Illumina sequencing workflow, primarily arising from PCR amplification.

Mean %GC content for mouse is about 42% (Reference: https://bionumbers.hms.harvard.edu/bionumber.aspx?id=102409&ver=6). (For human, mean %GC content is 41% (https://en.wikipedia.org/wiki/GC-content).

Measure	Value
Filename	SRR10207204_1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	11645096
Total Bases	1.1 Gbp
Sequences flagged as poor quality	0
Sequence length	101
%GC	45

Figure 2. Basic Statistics for a single sample

2) Per base sequence quality - plot of boxplots

A series of boxplots are plotted, with each boxplot showing the read quality (y-axis is phred score) at a given position in the read, computed over all reads. Higher scores are better. The phred score (Q) is defined: Q = -10 * log10(P), with P the probability of an incorrect base call. The sequencing instrument computes P and reports Q.

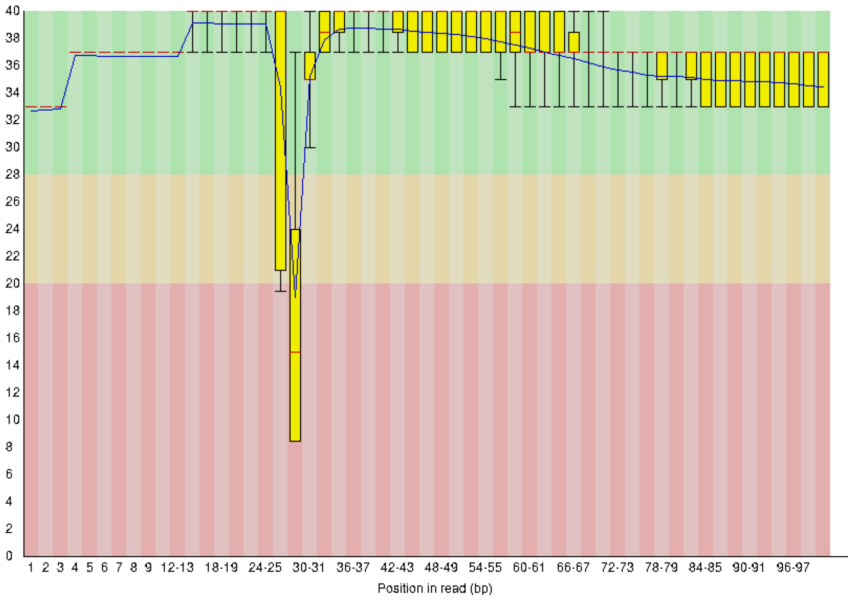


Figure 3. Per base sequence quality - plot of boxplots

3) Per sequence quality scores - plot

Shows the distribution of mean quality scores (phred) across all reads. Ideally, most of the reads should have high mean quality scores. No problematic issues in this example (see below).

A phred score of 30 is considered good (https://www.illumina.com/science/technology/next-generation-sequencing/plan-experiments/quality-scores.html).

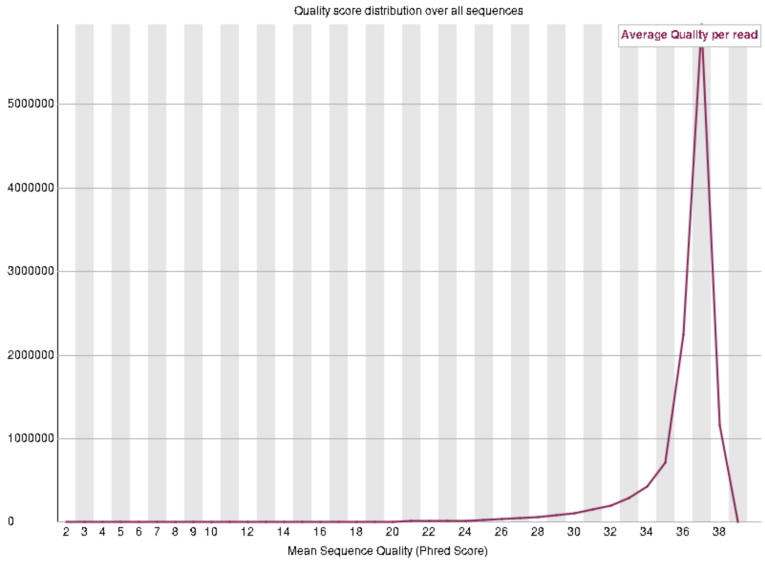


Figure 4. Per sequence quality scores - plot

4) Per base sequence content - plot

What the figure is supposed to show: For every read cycle it plots the proportion of A, T, G and C observed across all reads. Each nucleotide is expected to hover near 25% when the library covers a genome with roughly balanced base composition, and no position-specific bias is introduced during library preparation or sequencing. The figure seen below is created pre-Trimmomatic software application.

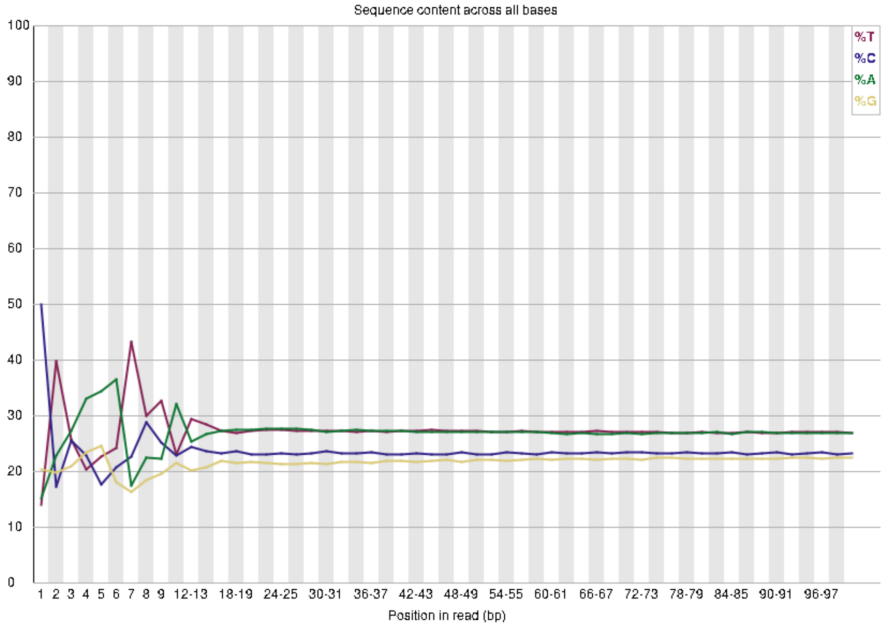


Figure 5. Per base sequence content - plot

5) Per Sequence GC Content - plot

Shows the distribution of GC content across all reads. The plot is expected to show a Gaussian (normal) distribution. This example looks like the expected outcome.

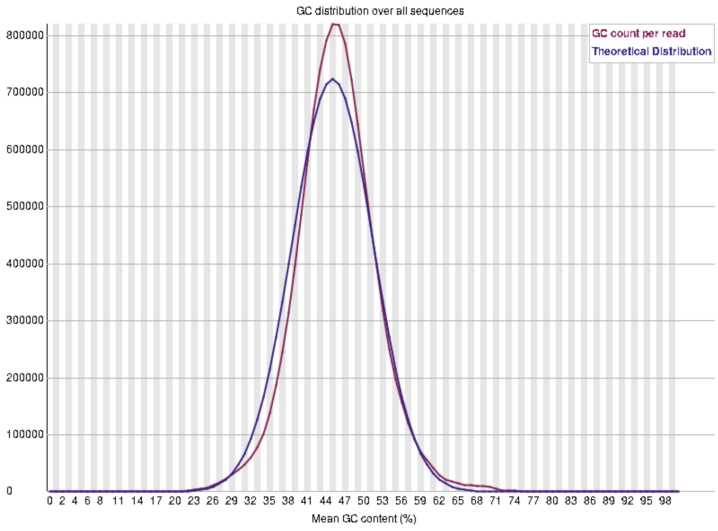


Figure 6. Per Sequence GC Content - plot

6) Per Base N Content - plot

Shows the percentage of N bases (unknown bases) at each read position. Ideally, the number of Ns should be zero at any position throughout the reads. Here, we see a slight problem/error in read calls: 25% N in the reads at positions 25-30bp.

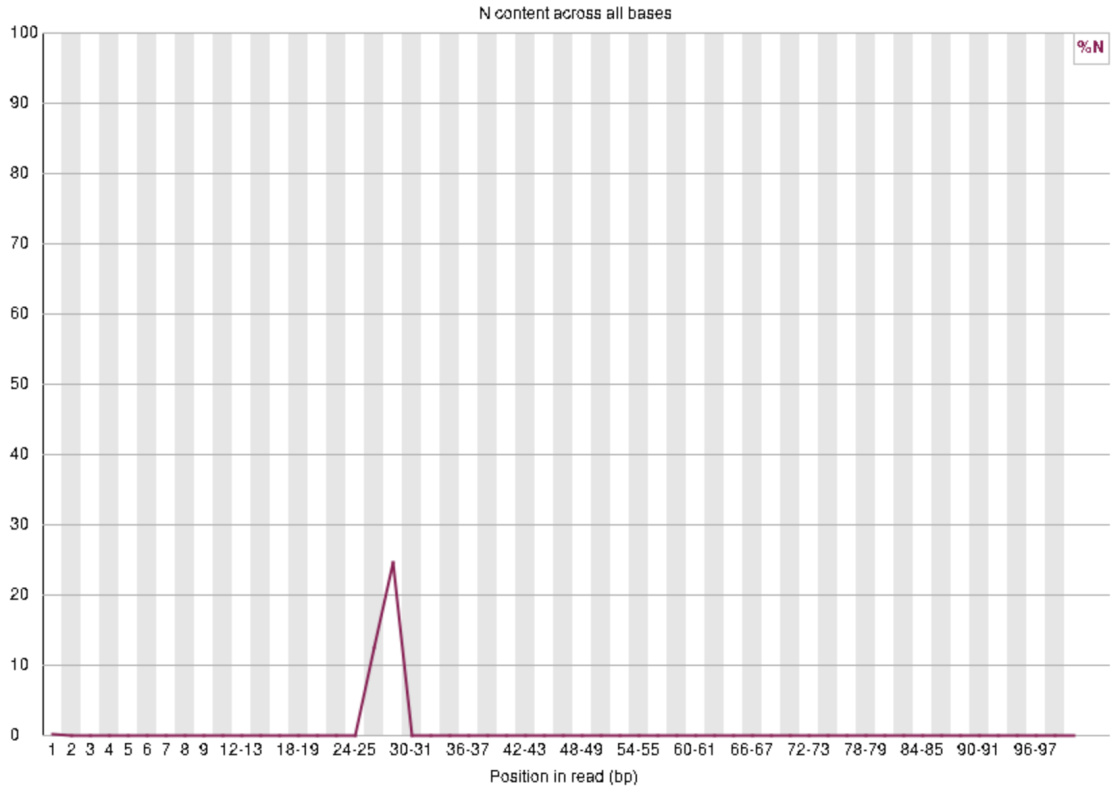


Figure 7. Per Base N Content - plot

7) Sequence Length Distribution - plot

Shows the distribution of read lengths. Expect to observe a peak at the expected single read length (100bp) and a few reads at other lengths. In this example, there is an extremely sharp peak at 101bp and none at other read positions. It could be that either the reads were highly trimmed prior to considering quality control or there is a problem with the sequencing.

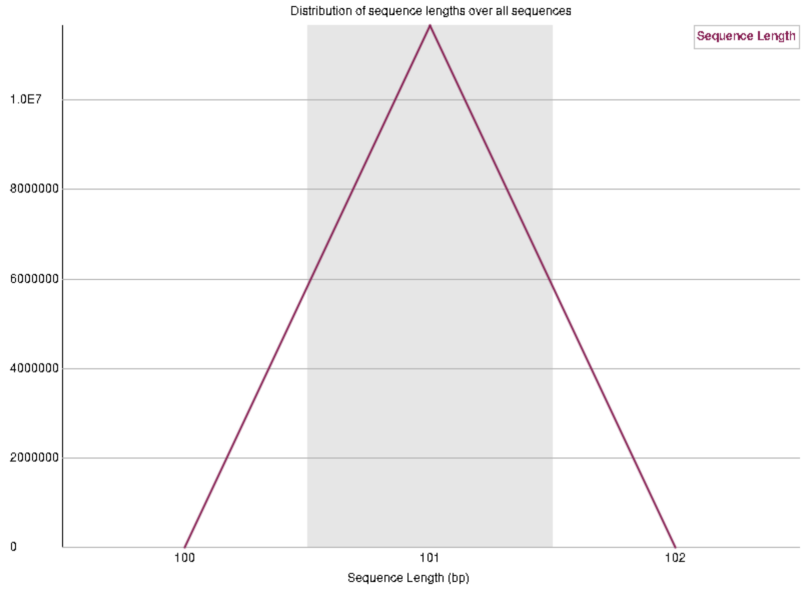


Figure 8. Sequence Length Distribution - plot

8) Sequence Duplication Levels

Shows the percentage of reads with different levels of duplication. Some duplication is expected. In the example (see below), “Percent of seq[uence]s remaining if deduplicated 61.01%" means that 38.99% of the sequences are duplicates in this sample. Thus, computations involving quantification (e.g. gene expression analysis) could be inaccurate since duplication is high. This may have arisen from the sequencing of this sample. Notice also, there is a bump around “>10”, which may indicate an unusual duplication pattern.

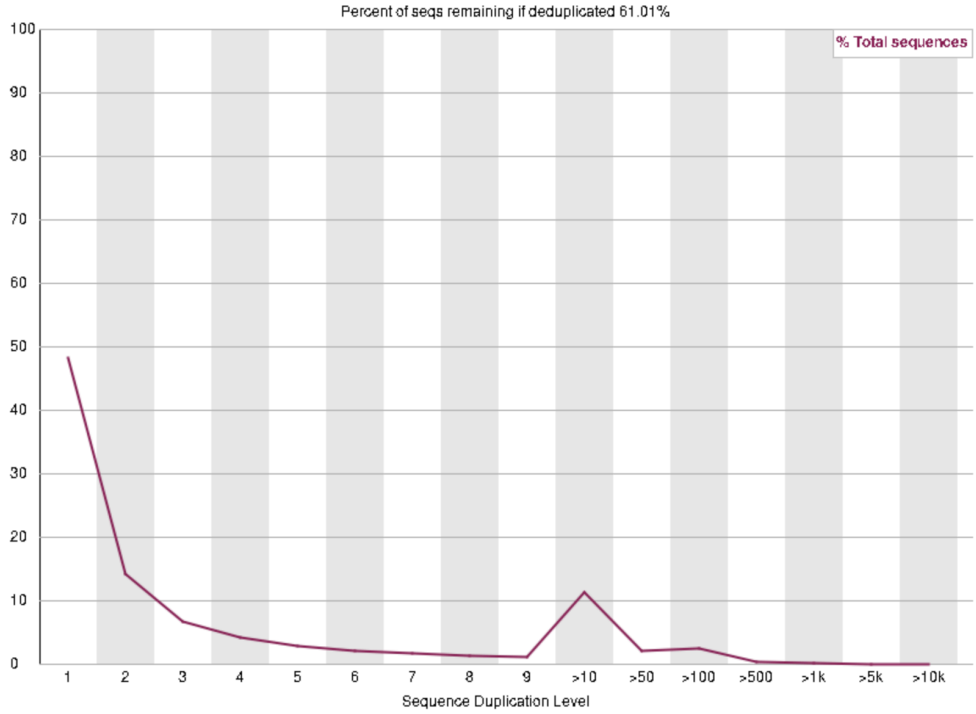


Figure 9. Sequence Duplication Levels

9) Overrepresented sequences - table

Lists sequences that occur more frequently than expected.

Sequence column: FastQC examines every read, but to save memory it only keeps strings that it has already seen in the first 200,000 reads; new strings encountered later are ignored. For any read longer than 75 bp it chops the read down to its first 50 bp and uses that slice as the key it counts. Reads that are ≤ 75 bp are counted at their full length. Once a particular sequence makes up ≥ 0.1% of all reads, FastQC writes it to the table, “Sequence” column.

Count = the number of times this specific sequence occurs in the whole .fastq file.

Percentage = Count / Total-reads × 100. So if Count = 2 and Percentage = 0.4%, the total number of reads FastQC processed was 2 / 0.004 ≈ 500. The portion of the sample that FastQC processed is therefore very small.

Possible Source = FastQC BLASTs each over-represented sequence against a small built-in database of common adapters, indexes, rRNA, PhiX, etc. If nothing meets the ≥20-bp, ≤1-mismatch cut-off, the entry is labelled “No Hit.”

Overrepresented sequences can indicate contamination (e.g., adapter sequences, vector sequences) or highly abundant biological sequences (e.g., rRNA). In the following example, there are many overrepresented sequences.

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
CTCGTTCGTTATCGGAATTAACCAGACAAATCGCTCCACCAACTAAGAAC	3	0.6	No Hit
CGGGCGGTGTGTACAAAGGGCAGGGACTTAATCAACGCAAGCTTATGACC	2	0.4	No Hit
CAACCATACTCCCCCCGGAACCCAAAGACTTTGGTTTCCCGGAAGCTGCC	2	0.4	No Hit
CTCCGACTTTCGTTCTTGATTAATGAAAACATTCTTGGCAAATGCTTTCG	2	0.4	No Hit
CCCGCACTTACTGGGAATTCCTCGTTCATGGGGAATAATTGCAATCCCCG	2	0.4	No Hit
CATTGGTCTTAACATGGTGAATGAAGAAAAGGAAGGGGTGGTCGGCACAG	2	0.4	No Hit
CCCTCTTAATCATGGCCTCAGTTCCGAAAACCAACAAAATAGAACCGCGG	2	0.4	No Hit
CGAGCTTTTTAACTGCAGCAACTTTAATATACGCTATTGGAGCTGGAATT	2	0.4	No Hit
TTTGTTTTTGTTTTTTTTTTTGTTTTTTTGGTCCACACGGGGAATTTCAC	2	0.4	No Hit
CACCAGACTTGCCCTCCAATGGATCCTCGTTAAAGGATTTAAAGTGGACT	2	0.4	No Hit
CCCCCGGCCGTCCCTCTTAATCATGGCCTCAGTTCCGAAAACCAACAAAA	2	0.4	No Hit
TGCCCGCTCTCCCAGTCATCACGGTCTGGTTTCTTTATATCCTGCAGGAA	2	0.4	No Hit
GTCTGTTGTCTGGGCGCATGACCGAGGTGACCAAGTGTGGACCGTCAACC	2	0.4	No Hit
CCTTCATTCGGCTTCACATGAATTCTCCATTCCCTAGGAGCTGTAGGCCC	2	0.4	No Hit
GCCTGGATTTTCTTGGTGATCTCATTTTTCAGGTTTTTAATCAGAAGTGA	2	0.4	No Hit
CTGATAAATGCACGCATCCCCCCCCGGGAAGGGGGGTCAGCGCCCGTCGG	2	0.4	No Hit
GTCCCTGAGCCCTTGTGGGGTTGGTGTCCCAGTAGGACTGGGTGACTTGT	2	0.4	No Hit

Figure 10. Overrepresented sequences - table: Subset of overrepresented sequences. There were many overrepresented sequences (more than was shown in the figure).

10) Adaptor Content - plot

Shows the cumulative percentage of reads where, if exists, adapter sequences are found at the different positions.

The presence of adapter sequences can affect downstream analyses and should be trimmed, as they do not represent the biology. Use of Trimmomatic accomplishes this trimming or discarding.

In this case, very few adapter sequences exist in this sample file (few contaminants).

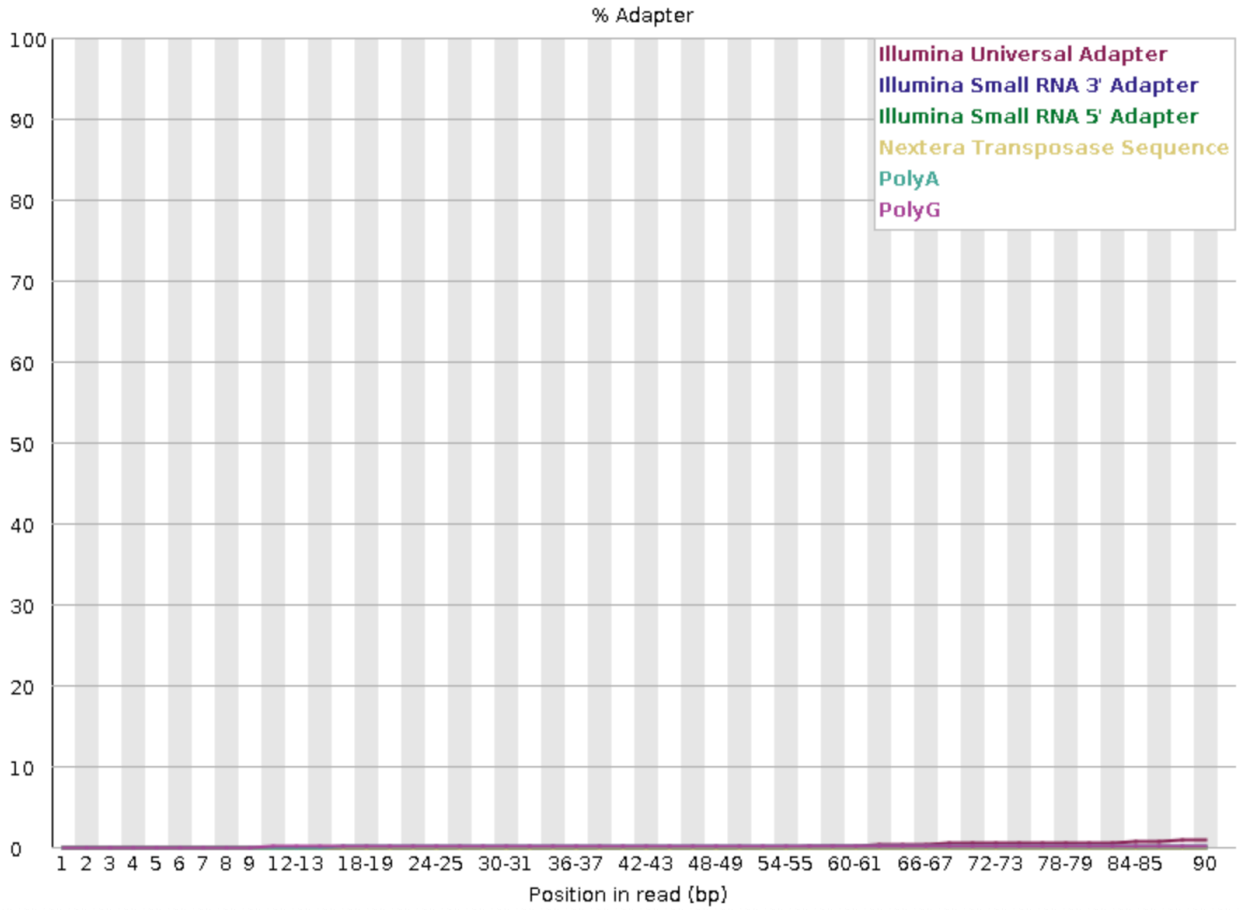


Figure 11. Adaptor Content - plot

Remark: Additionally, Trimmomatic is used for the removal of specific adaptor sequences.

Here is the code for that:

ILLUMINACLIP:adapter_file.fa:2:30:10 \

adapter_file.fa lists the adaptor sequences. This is a FASTA file. Illumina provides lists of common adapter sequences on their website*, but the downloaded Trimmomatic folder also contains a folder full of adapter files.*

Example of a custom adapter_file.fa
(plain text file, which you can create with either the vim or nano editor:

> Adapter1
AGATCGGAAGAGCACGTCTGAA
> Adapter2
GATCGGAAGAGCGTCGTGTAGG
>IndexedAdapter_Read1_barcode_sequence_within_adapter
GATCGGAAGAGCACGTCTGAA
> IndexedAdapter_Read2_barcode_sequence_within_adapter
AGATCGGAAGAGCGTCGTGTAGAT

2:30:10:
2: Maximum number of mismatches allowed in the seed alignment = short adapter sequence used for initial matching
30: Minimum score to keep the read/pair for “palindrome mode” (for paired-end reads where adapters might ligate to each other)
10: Minimum score to keep the read/pair for “simple mode” (for finding adapter at the 3' end)

@@@ 3. MultiQC on Savio @@@

$ cd /global/scratch/users/your-username

Create a new Conda environment:

$ conda create -n multiqc_env python=3.10 -y

$ conda activate multiqc_env

Install multiqc via pip:

$ pip install multiqc

MultiQC needs an alternative polars:
pip uninstall polars
pip install polars-lts-cpu

To verify the installation:

$ multiqc --version

Savio replies:
multiqc, version 1.29 (or similar version)

When to run MultiQC:

$ cd /global/scratch/users/username/mouse_data/
$ ls (lowercase “L”)

Savio replies:
fastqc_results trim_results star_results featurecounts_results (and other unrelated files)

In other words, ideally, you will run MultiQC after you've run fastQC, Trimmomatic, STAR, and featureCounts (and have created "results" directories for each of these programs). Suppose this is the case.

To run MultiQC:

$ pwd

Savio replies:
/global/scratch/users/username/mouse_data/

Type simply:
$ multiqc . (Type the ".")

Savio replies:

```
/// MultiQC 🔍 v1.29

       file_search | Search path: /global/scratch/users/e
         searching | ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━
     featurecounts | Found 20 reports
              star | Found 23 reports
            fastqc | Found 1 reports
     write_results | Data        : multiqc_data
     write_results | Report      : multiqc_report.html
           multiqc | MultiQC complete
```

If you type "ls" again, you'll find the new directory, "multiqc_data" and the new file, "multiqc_report.html" listed in the mouse_data folder.

To inspect "multiqc_report.html", you may want to download this file to a local computer and open it there (click on it):

1) FastQC Section (Raw Read Quality) summarizes individual FastQC reports

Key Metrics:

- **Per base sequence quality:**
    - High-quality reads should have mean Phred scores >30.
    - A drop in quality at the 3′ end suggests adapter contamination or low-complexity tails(repetitive or biased base composition near the 3′ end of the read-could be Poly-A tails or a technical issue occurring during sequencing).
- **Per sequence GC content:**
    - Should resemble the expected distribution for mouse: GC content = percentage of guanine and cytosine in each read; The plot is the number of reads against their GC content and should approximate a normal distribution with a mean of 40-55% GC (mouse).
    - A weird shape might indicate contamination.

- **Overrepresented sequences:**
    - May indicate leftover adapters, primers, or contamination.
- **Sequence length distribution:**
    - Especially relevant if you're trimming reads—if a lot of very short reads seen in the histogram, increase the minimum allowed length during trimming (when applying Trimmomatic); Sharp peak at expected length (e.g., 100 bp) means a good quality sample.

  Using fastQC before applying STAR: Helps you catch problems before alignment (e.g., bad libraries, adapter contamination).

2) Trimming Tool Section (e.g., Trimmomatic)

Key Metrics:

- % of reads surviving: How many reads passed filtering.
- Adapters removed: Number of reads trimmed due to adapters.
- Length distributions after trimming
- Confirms that adapters and low-quality bases were trimmed effectively — or alerts you to problems such as if too much data was removed.

3) Alignment (e.g., STAR)

Key Metrics:

- % uniquely mapped reads:
    - Typically 70–90% for good RNA-seq libraries.
- % multi-mapped or unmapped reads:
    - High unmapped rates may indicate contamination, wrong genome, or poor library prep.
- Total reads: Confirms sequencing depth.

Tells you if alignment worked well and if your genome index and annotation are appropriate.

4) Quantification (e.g., featureCounts)

Key Metrics:

- Assigned reads: % of reads assigned to genes.

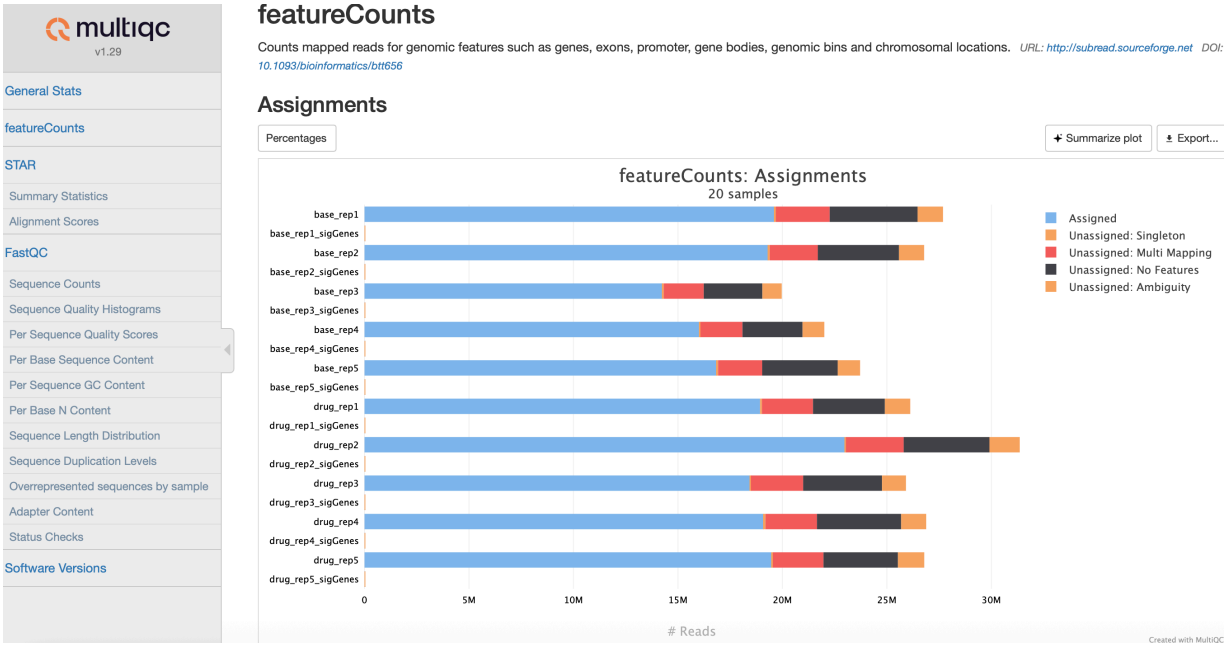
- Unassigned due to ambiguity or no feature: Can indicate issues with annotation or library prep. Poor assignment can impact downstream differential expression.

Summary of Items in MultiQC:

Software Tool	What to Inspect	Key Considerations
FastQC	Per base quality, overrepresented seqs, GC content	Library integrity & contamination check
Trimmomatic	% trimmed, adapters removed	Adapter contamination; check if reads are too short
STAR	% uniquely mapped, total mapped	Confirms good match to reference genome
featureCounts	% assigned reads	Ensures reads are mapping to known genes

Possible Issues:

Problem	Software Tool	Possible Reasons
Low quality scores at ends	FastQC	Adapter contamination or poor library
Low % of mapped reads	STAR	Contamination, wrong reference genome
High % of unassigned reads	featureCounts	Wrong GTF file or poor annotation match
GC content peaks shifted	FastQC	Contaminants or biased libraries



@@@ 4. Trimmomatic on Savio @@@

Purpose: To trim adapters from reads and remove low quality fragments.

Details: To remove (trim) technical sequences (Illumina adapters and other contaminants (such as primer dimers, adapter dimers, etc) from the reads. Note that “insert” equals cDNA material; “fragment” equals the insert with ligated adapters; “read” equals the digital output from a sequencer modeling the fragment.

Website: https://github.com/usadellab/Trimmomatic
Manual: http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf (all one line)

Step 1. Create Conda Environment:

$ conda create -n trim_env -c conda-forge -c bioconda trimmomatic

$ conda activate trim_env

Check installation:
$ trimmomatic -version

Savio replies:
0.39

Step 2. Locate the correct adapters file, filename.fa:

Locate Trimmomatic:
$ which trimmomatic

Savio replies:
/global/scratch/users/your-username/conda/envs/trim_env/bin/trimmomatic

$ cd /global/scratch/users/your-username/conda/envs/trim_env/bin

$ ls -la

You see (among other files):
“trimmomatic -> ../share/trimmomatic-0.39-2/trimmomatic”
This arrow is a symlink. To get its full path (instead of “..”), type the following:

$ readlink -f trimmomatic (you’re in the bin directory when you type this)

Savio replies:
/global/scratch/users/your-username/conda/envs/trim_env/share/trimmomatic-0.39-2/trimmomatic (all one line)

“trimmomatic” is the executable file - what you use when trimming the reads. But we want the adapters folder. Move to the trimmomatic-0.39-2 (or similar version) folder:

$ cd /global/scratch/users/your-username/conda/envs/trim_env/share/trimmomatic-0.39-2

and list the contents by typing:

$ ls -la

Savio replies:
adapters build_env_setup.sh conda_build.sh LICENSE metadata_conda_debug.yaml trimmomatic trimmomatic.jar

Move into the adapters folder:

$ cd adapters

And list the contents. Savio replies:
NexteraPE-PE.fa TruSeq2-PE.fa TruSeq2-SE.fa TruSeq3-PE-2.fa TruSeq3-PE.fa TruSeq3-SE.fa
Type

$ pwd

to get the absolute path to the desired adapter file:

/global/scratch/users/your-username/conda/envs/trim_env/share/trimmomatic-0.39-2/adapters (all one line)

If you're curious about the contents of a specific adapters file, such as TruSeq3-PE.fa:

$ cat TruSeq3-PE.fa
Savio replies:

>PrefixPE/1
TACACTCTTTCCCTACACGACGCTCTTCCGATCT
>PrefixPE/2
GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT

Adapter sequences/files:

1. TruSeq2-SE.fa (single-ended, for Illumina GAII) (Obsolete)
2. TruSeq3-SE.fa (single-ended, for MiSeq or HiSeq)
3. TruSeq2-PE.fa (paired-ended, for Illumina GAII) (Obsolete)
4. TruSeq3-PE.fa (paired ended, for MiSeq or HiSeq) (Most common for recent sequencing)
5. TruSeq3-PE-2.fa (additional seqs, paired-ended, for MiSeq or HiSeq) (Use only if standard TruSeq3 misses adapters)
6. NexteraPE-PE.fa (paired-ended) Use only if "grep 'CTGTCTCTTATACACATCT' fastq_filename" is not empty; Nextera XT or Nextera RNA-Seq kit is used

For ease of adapter retrieval, make a folder called "software" and copy the adapters directory into the folder:

$ cd .. (this puts you back into the trimmomatic-0.39-2 folder)
$ mkdir /global/scratch/users/your-username/software
$ cp -r adapters /global/scratch/users/your-username/software/adapters
(You use "-r" because you're copying a directory, not a file)

The path to the desired adapters file in the software folder is:

/global/scratch/users/elinorvelasquez/software/adapters/TruSeq3-PE.fa

Now you're ready to run Trimmomatic. You can use the following command:

Step 3. Run Trimmomatic on command line (for paired-ended reads)**:**

```
$ cd /global/scratch/users/your-username/mouse_data

$ mkdir trim_results

$ trimmomatic PE -threads 8 \
> fastq/ctrl_rep1_1.fastq.gz \   (forward)
> fastq/ctrl_rep1_2.fastq.gz \  (reverse)
> trim_results/trim_paired_ctrl_rep1_1.fastq.gz \
> trim_results/trim_unpaired_ctrl_rep1_1.fastq.gz \
> trim_results/trim_paired_ctrl_rep1_2.fastq.gz \
> trim_results/trim_unpaired_ctrl_rep1_2.fastq.gz \
> ILLUMINACLIP:/global/scratch/users/your-username/software/adapters/TruSeq3-
PE.fa:2:30:10 \
> LEADING:3 \
> TRAILING:3 \
> SLIDINGWINDOW:4:15 \
> MINLEN:36
```

Notes:
1. "$" and ">" are cursors. You don't type them. Be sure to type "\" as appropriate.
2. You get four files for output: paired and unpaired for each strand type (_1 or _2). You'll only need the paired files for STAR (aligning the reads to a reference genome).
3. Your sample's name for this example is: "ctrl_rep1_1" and "ctrl_rep1_2" (control_replicate1_forward and control_replicate1_reverse).
4. You need to specify an adapters file to run Trimmomatic. In the above example, we are using the TruSeq3-PE.fa adapter file (-PE = paired end).
5. You need to specify certain options (see below for descriptions).
6. "$ADAPTERS" = path/to/adapters.fa
7. The above command is only for one file at a time. Use a SLURM file to run Trimmomatic on more than one file. See the example SLURM files for Trimmomatic on the workshop website: https://github.com/elinorv21/RNA-Seq_workshop/

Options for Trimmomatic (for both Savio and Galaxy):

1. ILLUMINACLIP:$ADAPTERS:2:30:10:

a. Maximum mismatch count which will still allow a full match to be performed (seed mismatches): (2, default) Allows minor mismatches, such as sequencing errors or degraded adapter sequences (if you want no errors allowed, use 0)

b. How accurate the match between the two 'adapter ligated' reads must be for PE palindrome read alignment (palindrome clip threshold): 10 - 30 (30, default).

c. How accurate the match between any adapter etc. sequence must be against a read (simple clip threshold): 7 - 15 (10, default)

2. Sliding Window Trimming: SLIDINGWINDOW: <window size>:<required quality>

a. Number of bases to average across <window size>: 4 - 5(4, default)
b. Average quality required <required quality>: 15 - 20 (20, default)

3. Drop reads below a specified length: MINLEN:<length>
Minimum length of reads to be kept: <length> 30 - 50 (20, default)

4. Cut bases off the start of a read, if below a threshold quality: LEADING:<quality>
Minimum quality required to keep a base: <quality> 3 - 10 (3, default)

5.Cut bases off the end of a read, if below a threshold quality: TRAILING:<quality>
Minimum quality required to keep a base: <quality> 3 - 10 (3, default)

6. Cut the read to a specified length: CROP:<length>
Number of bases to keep from the start of the read: <length>(Should not use)

7. Cut the specified number of bases from the start of the read: HEADCROP:<length>
Number of bases to remove from the start of the read: <length> (Should not use)

8.Drop reads with average quality lower than a specified level: AVGQUAL:<required quality>
Minimum average quality required to keep a read: 20 or higher

9. Trim reads adaptively, balancing read length and error rate to maximize the value of each read: MAXINFO:<targetLength>:<fraction>

a. Target read length: <targetLength> Example: 150 (original read length)
b. Strictness: <fraction> 0.8 - 0.9 (0.8)

10. (Galaxy only) Quality score encoding: phred33 (for recent sequencing)

11. (Galaxy only) Output trimlog file? Yes since you're planning to use multiQC

12. (Galaxy) Output trimmomatic log messages? Yes since you're planning to use multiQC

@@@ 5. (Alternative) Run Trimmomatic with a SLURM: @@@

Notes:
1. Make sure that mouse_env is activated; run this operation in the Scratch base directory:
$ cd /global/scratch/users/your-username
2. Be sure to download and edit the file's code to match the name and location of your specific data and also the name and location of your adapter file.

$ sbatch bioconda_trim_example.sh
(give this command in your scratch base directory)

GitHub link to SLURM files:
https://github.com/elinorv21/RNA-Seq_workshop/blob/main/bioconda_trim.sh (generic file)
https://github.com/elinorv21/RNA-Seq_workshop/blob/main/bioconda_trim_example.sh (example)

Note:

This SLURM file applies Trimmomatic to only one sample (with forward and reverse reads). It may be helpful to run Trimmomatic on many samples with just one command (use a loop):

$ sbatch trim_loop_example.sh
(give this command in your scratch base directory unless you use absolute paths for everything)

Savio replies with your job number (jobID).

Slurm files:

To see status of a job:
$ squeue -u your-username

To cancel a job:

$ scancel jobID

GitHub link to relevant files:
https://github.com/elinorv21/RNA-Seq_workshop/blob/main/trim_loop.sh (generic file)
https://github.com/elinorv21/RNA-Seq_workshop/blob/main/trim_loop_example.sh (example)

@@@ 6. Example of Trimmomatic Applied to Data: @@@

Check for adapter sequence, "AGATCGGAAGAGC," in your raw data, by the following bash command:

$ grep 'AGATCGGAAGAGC' raw_reads.fastq

Savio replies:

AGCGGGAAGAAAATGTAAAGGCAGATGTTTTTCATGCATACCTGTCTCTTCTGAAACAAACTCG
TCCAGTGCAAAGTTGGCTAGATCGGAAGAGCACACGT
GTATTGAAGGTTTCAAACATTATCTGCGTCATCTTCTCTCTGTTAGCTTTGGGGTTCAGGGAGAT
CGGAAGAGCACACGTCTGAACTCCAGTCACCGCATG
CCACGCTCTCCCTTGTGTCCAGGCTGACCATCACGACCTGGGGGACCATCGCTGCCAGGGTT
ACCATCACGACCAGCTTCACCAGGGAGATCGGAAGAGCA
CCAGCCCTCTTGGTGAGGTCGATGTCTGCTTTCCTCAACACCACATGAGCATATCAGATCGGA
AGAGCACACGTCTGAACTCCAGTCACCGCATGATATCT
AGACCAGTCAGACCACGGGCACCATCTTTACCAGGAGAACCATCAGCACCTTTGGGACCAGC
ATCAAGATCGGAAGAGCACACGTCTGAACTCCAGTCACC

Check for the same adapter sequence after Trimmomatic:

$ grep 'AGATCGGAAGAGC' trimmed_reads.fastq

Results: Nothing!

@@@ 7. (Alternative) Trimmomatic on Galaxy @@@

1. Type trimmomatic in the "Tool" bar on the left side.

2. Complete the “Tool Parameters”:

Single-end or paired-end reads? Click “paired-end (two separate paired-end files)” or “paired-end as a collection”. If your paired-end reads are from a single sample, click “paired-end (two separate paired-end files)”. If you are analyzing more samples (say, 10 samples), click the “paired-end as a collection”.

Select datasets:

a. Input FASTQ file (R1/first of pair): input forward strand
b. Input FASTQ file (R2/second of pair): input reverse strand

ILLUMINACLIP (Example in bash: ILLUMINACLIP:TruSeq3-PE.fa:2:30:10)

Simple clip mode: Suppose a complete technical sequence is contained in the read. Then a standard alignment method (such as the Smith-Waterman local alignment) between the contaminant and the read can be used to trim the contaminant from the read. If a partial technical sequence is attached to an end of the read (or possibly within the read), many shorter contaminant pieces may remain at the read end or within the read, if the alignment score < simple_clip_threshold.

Palindrome clip mode: Assuming paired-end reads, the forward read is aligned to the reverse complement of the reverse read. If the insert piece is shorter than the read, then the sequencer will ‘read-through’ to the whole or partial portion of the adapter ligated to the insert. While the insert will match perfectly with the reverse complement of the insert, the contaminant will have the wrong orientation. Since Trimmomatic uses local alignment (with a sliding window), only a few bases need to be examined in order to trim the adapter at the ends of the reads, if the alignment score > palindrome_clip_threshold.

1) Perform Initial ILLUMINACLIP Step? (Remove adapter and other illumina-specific sequences from the read) Choose yes (because Illumina reads almost always contain adapter contamination, especially with short insert sizes)

2) Select standard adapter sequences or provide custom?
<fastaWithAdaptersEtc> (bash: TruSeq3-PE.fa: your specific adapter sequence)

a) Standard:

Adapter sequences to use:

1. TruSeq2 (single-ended, for Illumina GAII) (Obsolete)
2. TruSeq3 (single-ended, for MiSeq or HiSeq)
3. TruSeq2(paired-ended, for Illumina GAII) (Obsolete)
4. TruSeq3 (paired ended, for MiSeq or HiSeq) (Most common for recent sequencing)
5. TruSeq3 (additional seqs, paired-ended, for MiSeq or HiSeq) (Use only if standard TruSeq3 misses adapters)
6. Nextera (paired-ended) use only if “grep 'CTGTCTCTTATACACATCT' raw_269.fastq” is not empty; Nextera XT or Nextera RNA-Seq kit is used

b) Custom:

Example: Custom adapter sequences in fasta format (paste into website window)**:**

Example: Illumina Universal adapter**:**
>Read1Adapter
AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGATCTCGGTGGTCGCCGTATCATT

>Read2Adapter
GATCGGAAGAGCACACGTCTGAACTCCAGTCACGGATGACTATCTCGTATGCCGTCTTCTGCT
TG

3) Maximum mismatch count which will still allow a full match to be performed: <seed mismatches> Choose 2, which allows minor mismatches, such as sequencing errors or degraded adapter sequences (if you want no errors allowed, use 0)

4) How accurate the match between the two 'adapter ligated' reads must be for PE palindrome read alignment: <palindrome clip threshold> 10 - 30 (30, default).

5) How accurate the match between any adapter etc. sequence must be against a read: <simple clip threshold> 7 - 15 (10, default)

6) Minimum length of adapter that needs to be detected (PE specific/palindrome mode): <minAdapterLength> Minimum length of adapter that needs to be detected (PE specific/ palindrome mode): 8 - 12 (8, default)

7) Always keep both reads (PE specific/palindrome mode)? <keepBoth> Yes

(Trimmomatic Operations)

8) Sliding Window Trimming (SLIDINGWINDOW) - MAXINFO: See options above.

9) Quality score encoding: phred33 (for recent sequencing)

10) Output trimlog file? Yes since you're planning to use multiQC

11) Output trimmomatic log messages? Yes since you're planning to use multiQC