Trimming and Filtering

A Beginner's Guide

Dr. Princess Rodriguez

2025-02-12

Contents

1 Learning Objectives

- Understand what scenarios warrant trimming
- Be able to clean FASTQ reads using Trimmomatic if required

2 Introduction

During the last class, we took a high-level look at the quality of each of our samples using FastQC. We visualized per-base quality graphs showing the distribution of read quality at each base across all reads in a sample and extracted information about which samples fail which quality checks. Some of our samples failed quite a few quality metrics used by FastQC. This does not mean, though, that our samples should be thrown out! It is very common to have some quality metrics fail, and this may or may not be a problem for your downstream application.

Now, we want to make sure that as many reads as possible map or align accurately to the genome. To ensure accuracy, only a small number of mismatches between the read sequence and the genome sequence are allowed, and any read with more than a few mismatches will be marked as being unaligned.

Therefore, to make sure that all the reads in the dataset have a chance to map/align to the genome, unwanted information can be trimmed off from every read, one read at a time. The types of unwanted information can include one or more of the following:

- leftover adapter sequences
- known contaminants (strings of As/Ts, other sequences)
- poor quality bases

3 Where does this "unwanted information" come from?

- During the sequencing process, adapter sequences are added to the ends of the reads. These sequences are used to attach the RNA molecules to the sequencing platform but they do not map to regions of the genome. These adapter sequences can interfere with downstream analysis and lead to inaccurate results.

- RNA-seq data may contain low-quality reads that are caused by sequencing errors, PCR bias, or other factors such as poor sample quality. By removing these low-quality reads researcher can ensure that the data is of high quality.

4 Steps of trimming data

Step 1: Quality Control

The first step in trimming RNA-seq data is to assess the quality of the raw reads. This can be done using software such as FastQC, which generates a report that includes information on read length, GC content, base quality, and sequencing adapter contamination. If the data is of poor quality, it may need to be re-sequenced or excluded from further analysis.

Step 2: Adapter Trimming The next step is to remove adapter sequences from the reads. These sequences are added during library preparation and can interfere with downstream analysis. Many tools (some listed below) can be used to trim adapters from the reads by searching for the adapter sequence and removing it.

Step 2b: Quality Trimming During adapter trimming, we could also remove low-quality reads by setting a minimum quality threshold and removing any bases that fall below this threshold. Trimming tools often have built-in quality trimming options that can be used for this purpose.

Step 2c: Short read filtering

Reads that are too short *after trimming* can also be removed. This step is important to ensure that only high-quality reads are used for downstream analysis. Trimming tools also have this built-in option as well.

5 Tools for Trimming

There are a number of tools that can be used for read trimming, some include:

- Cutadapt
- Trimmomatic
- fastp
- Trim Galore

They have a varying range of clipping and trimming features, but for the most part they all work similarly.

6 Trimming is not always required

- There are some aligners that are available which will "soft-clip" low-quality bases or adapter sequences *during* alignment. If you are working with short reads (<50 bp), trimming can actually prevent the aligner from discarding poor-quality reads.

We will compare some aligners next week

In the below class exercise we will be using Trimmomatic.

7 Trimmomatic options

```
module load gcc/13.3.0-xp3epyt
```

Search for the module with:

```
module avail
```

Use the following to check that the program was loaded

```
module list
```

Trimmomatic has a variety of options to trim your reads. If we run the following command, we can see some of our options.

```
trimmomatic
```

```
Usage:
       PE [-version] [-threads <threads>] [-phred33|-phred64] [-trimlog <trimLogFi
   or:
       SE [-version] [-threads <threads>] [-phred33|-phred64] [-trimlog <trimLogFi
```

This output shows us that we must first specify whether we have paired end (PE) or single end (SE) reads. Next, we specify what flag we would like to run. For example, you can specify threads to indicate the number of processors on your computer that you want Trimmomatic to use. In most cases using multiple threads (processors) can help to run the trimming faster. These flags are not necessary, but they can give you more control over the command. The flags are followed by positional arguments, meaning the order in which you specify them is important. In paired end mode, Trimmomatic expects the two input files, and then the names of the output files. These files are described below. While, in single end mode, Trimmomatic will expect 1 file as input, after which you can enter the optional settings and lastly the name of the output file.

option	meaning
inputFile1	Input reads to be trimmed. Typically the file name will contain an _1 or _R1 in the name.
inputFile2	Input reads to be trimmed. Typically the file name will contain an _2 or _R2 in the name.
outputFile1P	Output file that contains surviving pairs from the _1 file.
outputFile1U	Output file that contains orphaned reads from the _1 file.
outputFile2P	Output file that contains surviving pairs from the _2 file.
outputFile2U	Output file that contains orphaned reads from the _2 file.

In addition, trimmomatic expects to see is the trimming parameters:

step	meaning
ILLUMINACLIP	Perform adapter removal.
SLIDINGWINDOW	Perform sliding window trimming, cutting once the average quality within the window falls below a threshold.
LEADING	Cut bases off the start of a read, if below a threshold quality.
TRAILING	Cut bases off the end of a read, if below a threshold quality.
CROP	Cut the read to a specified length.
HEADCROP	Cut the specified number of bases from the start of the read.
MINLEN	Drop an entire read if it is below a specified length.
TOPHRED33	Convert quality scores to Phred-33.
TOPHRED64	Convert quality scores to Phred-64.

We will use only a few of these options and trimming steps in our analysis. It is important to understand the steps you are using to clean your data. For more information about the Trimmomatic arguments and options, see the Trimmomatic manual.

However, a complete command for Trimmomatic will look something like the command below. This command is an example and will not work, as we do not have the files it refers to:

```
trimmomatic PE SRR_1056_1.fastq SRR_1056_2.fastq  \
        SRR_1056_1.trimmed.fastq SRR_1056_1un.trimmed.fastq \
        SRR_1056_2.trimmed.fastq SRR_1056_2un.trimmed.fastq \
        SLIDINGWINDOW:4:20 ILLUMINACLIP:SRR_adapters.fa
```

In this example, we have told Trimmomatic:

code	meaning
PE	that it will be taking a paired end file as input
SRR_1056_1.fastq	the first input file name
SRR_1056_2.fastq	the second input file name
SRR_1056_1.trimmed.fastq	the output file for surviving pairs from the _1 file
SRR_1056_1un.trimmed.fastq	the output file for orphaned reads from the _1 file
SRR_1056_2.trimmed.fastq	the output file for surviving pairs from the _2 file
SRR_1056_2un.trimmed.fastq	the output file for orphaned reads from the _2 file
SLIDINGWINDOW:4:20	to use a sliding window of size 4 that will remove bases if their phred score is below 20
ILLUMINACLIP:SRR_adapters.fa	to clip the Illumina adapters using the sequences listed in SRR_adapters.fa

Note Some of the commands we ran in this lesson are long! When typing a long command into your terminal, you can use the character to separate code chunks onto separate lines. This can make your code more readable.

8 Running Trimmomatic

Class Exercise

This Exercise will take ~20 mins. Please work with your neighbor if you have questions. I will begin answering questions at the 5 minute mark.

Part A: Make a copy of the exercise materials. The folder can be found in this location:

```
/gpfs1/cl/mmg3320/course_materials/trimmomatic_exercise
```

Part B: Modify the `trim.sh` script using either Jupyter Notebook or Nano to run `trimmomatic` on the provided FASTQ sample. Ensure that you include the arguments listed below, using the **exact syntax** required by `trimmomatic`.

argument to add	meaning
PE	that it will be taking a paired end file as input
inputFile1	SRR2589044_1.fastq.gz
inputFile2	SRR2589044_2.fastq.gz
outputFile1P	Output file that contains surviving pairs from the _1 file.
outputFile1U	Output file that contains orphaned reads from the _1 file.
outputFile2P	Output file that contains surviving pairs from the _2 file.
outputFile2U	Output file that contains orphaned reads from the _2 file.
SLIDINGWINDOW:4:20	to use a sliding window of size 4 that will remove bases if their phred score is below 20
filter short reads	drop an entire read if it is below 25
adapter to add	NexteraPE-PE.fa:2:40:15

We are adding additional parameters the NexteraPE-PE.Fa adapter

- NexteraPE-PE.fa : This is a FASTA file that contains adapter sequences specific to Nextera paired-end libraries.
- `2:40:15`
  - `2` : Seed mismatch threshold (maximum number of mismatches allowed in the adapter sequence match).
  - `40` : Palindrome mode threshold (minimum match length for identifying adapter sequences when the paired-end reads overlap).
  - `15` : Simple adapter trimming threshold (minimum length of a match required to trigger adapter removal).

Once you have a working script, run it with the following command:

```
sh trim.sh
```

You will know that your script is correct and working properly if you see the following output:

```
TrimmomaticPE: Started with arguments:
SRR2589044_1.fastq.gz SRR2589044_2.fastq.gz SRR2589044_1.trim.fastq.gz SRR2589044_
 Using PrefixPair: 'AGATGTGTATAAGAGACAG' and 'AGATGTGTATAAGAGACAG'
Using Long Clipping Sequence: 'GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAG'
Using Long Clipping Sequence: 'TCGTCGGCAGCGTCAGATGTGTATAAGAGACAG'
Using Long Clipping Sequence: 'CTGTCTCTTATACACATCTCCGAGCCCACGAGAC'
Using Long Clipping Sequence: 'CTGTCTCTTATACACATCTGACGCTGCCGACGA'
ILLUMINACLIP: Using 1 prefix pairs, 4 forward/reverse sequences, 0 forward only se
Quality encoding detected as phred33

Input Read Pairs: 1107090 Both Surviving: 885220 (79.96%) Forward Only Surviving: 
TrimmomaticPE: Completed successfully
```

Part C: Check that the file outputs are the correct size

```
total 453M
-rw-r--r-- 1 pdrodrig pi-jdragon 124M Feb 11 14:30 SRR2589044_1.fastq.gz
-rw-r--r-- 1 pdrodrig pi-jdragon  94M Feb 11 18:32 SRR2589044_1.trim.fastq.gz
-rw-r--r-- 1 pdrodrig pi-jdragon  18M Feb 11 18:32 SRR2589044_1un.trim.fastq.gz
-rw-r--r-- 1 pdrodrig pi-jdragon 128M Feb 11 14:30 SRR2589044_2.fastq.gz
-rw-r--r-- 1 pdrodrig pi-jdragon  91M Feb 11 18:32 SRR2589044_2.trim.fastq.gz
-rw-r--r-- 1 pdrodrig pi-jdragon 271K Feb 11 18:32 SRR2589044_2un.trim.fastq.gz
```

Part D: Run FASTQC on all the `.gz` files and visualize the HTML files to see whether your **per base sequence quality** is higher after trimming. In addition, please note if the **adapter content/contamination** has been removed.