FASTQ Manipulation Tools

Trimming low quality bases

There are a number of open source tools that can trim off 3' bases and produce a FASTQ file of the trimmed reads to use as input to the alignment program.

FASTX Toolkit

The **FASTX-Toolkit** provides a set of command line tools for manipulating **fasta** and **fastq** files. The available modules are described on their website. They include a fast **fastx_trimmer** utility for trimming fastq sequences (and quality score strings) before alignment.

FASTX-Toolkit is available via the TACC module system.

FASTX_toolkit module description

```
#remember to load  load biocontainers module if you haven't already- fastx_toolkit is part of that

module spider fastx
module load fastx_toolkit
```

Let's run **fastx_trimmer** to trim all input sequences down to 90 bases:

Run fastx_Trimmer

```
fastx_trimmer -i data/Sample1_R1.fastq -l 90 -Q 33 -o Sample1_R1.trimmed.fastq
```

- The **-l 90** option says that base 90 should be the last base (i.e., trim down to 90 bases)
- the **-Q 33** option specifies how base qualities on the 4th line of each fastq entry are encoded. The FASTX toolkit is an older program, written in the time when Illumina base qualities were encoded differently. These days Illumina base qualities follow the Sanger FASTQ standard (Phred score + 33 to make an ASCII character).

Exercise: What if you just want to get rid of reads that are too low in quality?

fastx_quality_filter syntax

```
fastq_quality_filter -q <N> -p <N> -i <inputfile> -o <outputfile>
-q N: Minimum Base quality score
-p N: Minimum percent of bases that must have [-q] quality
```

Let's try it on our data- trim it to only include reads with atleast 80% of the read having a quality score of 30 or above.

Run fastx_quality_filter

```
fastq_quality_filter -q 20 -p 80 -i data/Sample1_R1.fastq -Q 33 -o Sample1_R1.filtered.fastq
```

- The **-q 20** option says that minimum base quality is 20
- The **-p 80** option says at least 80% of the read must meet the minimum quality requirement in order to be retained
- the **-Q 33** option specifies how base qualities on the 4th line of each fastq entry are encoded. The FASTX toolkit is an older program, written in the time when Illumina base qualities were encoded differently. These days Illumina base qualities follow the Sanger FASTQ standard (Phred score + 33 to make an ASCII character).

Exercise: Compare the results of fastq_trimmer vs fastq_quality_filter

Compare results

```
grep '^@HWI' Sample1_R1.trimmed.fastq |wc -l
grep '^@HWI' Sample1_R1.filtered.fastq |wc -l
```

Adapter Trimming

Data from RNA-seq or other library prep methods that resulted in very short fragments can cause problems with moderately long (50-100bp) reads since the 3' end of sequence can be read through to the 3' adapter at a variable position. This 3' adapter contamination can cause the "reql" insert sequence not to align because the adapter sequence does not correspond to the bases at the 3' end of the reference genome sequence.

Unlike general fixed-length trimming (e.g. trimming 100 bp sequences to 40 or 50 bp), adapter trimming removes differing numbers of 3' bases depending on where the adapter sequence is found.

The GSAF website describes the flavaors of Illumina adapter and barcode sequence in more detail https://wikis.utexas.edu/display/GSAF/Illumina+-+all+flavors

WARNING: ADAPTORS FOR RNA LIBRARIES

```
For old ligation-based RNA-libraries, the R1 adaptor is different from the R2 adaptor.

For current stranded RNA libraries (such as dUTP, used by GSAF), the R1 adaptor would be the same as the R2
adaptor.

Fastqc reports will give you an idea about which adaptor is present in your data. Further, it's always a good
idea to grep -c <partofdaptorseq> <fastqfile> to make sure you have the right adaptor sequence before trimming.
```

FASTX Toolkit

One of the programs available as part of the fastx toolkit does a crude job of clipping adaptors out of sequences.

fastx_clipper will clip a certain nucl. sequence (eq: adapter) from your reads.

fastx_Clipper general syntax

```
fastx_clipper -a <adapter> -i <inputfile> -o <outputfile> -l <discardSeqsShorterThanN>
```

fastx_Clipper example

```
fastx_clipper -a GATCGGAAGAGCACACGTCTGAACTCCA -i data/Sample1_R1.fastq -o Sample1_R1.cutadapt.fastq -l 20
```

More sophisticated tools like Trimgalore and cutadapt may be suitable, particularly with dealing with paired end data.

Cutadapt example

```
module load cutadapt
cutadapt -a GATCGGAAGAGCACACGTCTGAACTCCA -o <R1_outputfile> -p <R2_outputfile> <inputR1file> <innputR2file>
```

Please refer to https://wikis.utexas.edu/display/GSAF/Illumina+-+all+flavors for Illumina library adapter layout.

Below information holds true from old ligation-based RNA-libraries. In this case, the R1 adaptor is different from the R2 adaptor.

For current stranded RNA libraries (such as dUTP, used by GSAF), the R1 adaptor would be the same as the R2 adaptor.

Fastqc reports will give you an idea about which adaptor is present in your data. Further, it's always a good idea to grep -c <partofdaptorseq> <fastqfile> to make sure you have the right adaptor sequence before trimming.

Appendix: Illumina Adapter Information

https://wikis.utexas.edu/display/GSAF/Illumina+-+all+flavors

<P5 primer/capture site> <IndexRead2> <Read1 primer site>
<template - gDNA, RNA, amplicon, whatever>
<Read2 primer site> <IndexRead1> <P7 primer/capture site>

Standard DNA Library

- Read 1- Look for <Read 2 primer site>

GATCGGAAGAGCACACGTCTGAACTCCAGTCAC

- Read 2 - Look for <RevComp of TruSeq Read 1 primer>

GATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTAGA

Small-rna or RNA library

- Read 1- Look for <Read 2 primer site>

GATCGGAAGAGCACACGTCTGAACTCCAGTCAC

- Read 2 - Look for <RevComp of Read 1 primer site(NEB)>

TGATCGTCGGACTGTAGAACTCTGAACGTGTAGA

Let's look at how we can aggregate multiple FastQC reports into one report with MultiQC

BACK TO THE COURSE OUTLINE