

---

# **libqc++ Documentation**

***Release 0.4.0-dirty***

**Kevin Murray**

**Apr 04, 2017**



---

## Contents

---

<b>1</b>	<b>libqcpp Applications</b>	<b>3</b>
1.1	Trimit . . . . .	3
<b>2</b>	<b>libqcpp API</b>	<b>5</b>
2.1	Overview . . . . .	5
2.2	Streams . . . . .	5
2.3	Processors . . . . .	6
<b>3</b>	<b>Developer documentation</b>	<b>9</b>
3.1	Compiling static binaries . . . . .	9
<b>4</b>	<b>Indices and tables</b>	<b>11</b>



Contents:



Libqcpp ships with some applications built using the library.

### Contents

- *libqcpp Applications*
  - *Trimit*
    - \* *QC Steps*
    - \* *Usage*

## Trimit

Trimit ties together several common QC measures applied to short read sequencing data. It works with paired end Illumina and similar sequencing experiments.

### QC Steps

- Measure per-base quality scores
- Trim/Merge reads: does a global alignment between read pairs to detect read-through. Read pairs from fragments less than the read length are trimmed at the fragment length, discarding the second read. Read pairs from fragments that are longer than the read length but less than twice the read length are merged. Read pairs from fragments longer than twice the read length are not modified.
- Windowed quality control: a sliding-window based quality score trimmer, which uses a slightly improved version of the [sickle](#) trimming algorithm.
- Optional length filtering and/or truncation

## Usage

See *trimit -h*.

### Contents

- *libqcpp API*
  - *Overview*
  - *Streams*
  - *Processors*
    - \* *AdaptorTrimPE*
    - \* *WindowedQCTrim*
    - \* *PerBaseQuality*
    - \* *ReadLenFilter*
    - \* *ReadLenCounter*
    - \* *ReadTruncator*

## Overview

libqcpp's API is built around two concepts: Stream and Processors.

## Streams

`ReadStream` are streams of sequence reads. These streams parse reads from or write reads to a file or stream. `ReadInputStream` and `ReadOutputStream` do so without any manipulation. A `ProcessedReadStream`

processes reads using a pipeline of processors. `ThreadedQCProcessor` is a high-level, multi-threaded read processor that reads from and writes to files directly. Streams can report, as member variables or as a YAML report, statistics on reads that have been parsed or written.

## Processors

Processors mutate or calculate statistics on a read or read pair. They may also report statistics on all reads they have processed in their lifetime, as member variables or as YAML reports.

The following processors are implemented (shown with constructor arguments).

### AdaptorTrimPE

```
AdaptorTrimPE(const std::string &name, int min_overlap=10,
               const QualityEncoding &encoding=SangerEncoding);
```

Aligns a read pair to each other, and detect either adaptor read-through, or read overlap. Operates only on paired reads. Yields single ended reads if the read pair is either shorter than the read length (thus each read contains adaptor sequence) or the read ends overlap.

### WindowedQCTrim

```
WindowedQualTrim(const std::string &name, int8_t min_quality,
                  size_t min_length, size_t window_size=0,
                  const QualityEncoding &encoding=SangerEncoding);
```

Uses a sliding-window based approach to trim reads at the point where base quality decreases below a threshold. Reads are trimmed at the first position at which a window's mean base quality is below `min_quality`. The 5' end of a read is also trimmed. Reads less than `min_length` bases long are removed from the stream. The window size may be set using `window_size`; a `window_size` value of 0 causes the window length to be 10% of the read length.

### PerBaseQuality

```
PerBaseQuality(const std::string &name,
                const QualityEncoding &encoding=SangerEncoding);
```

Records statistics on per-cycle quality across all read sets, reporting the distribution of base quality scores for each cycle.

### ReadLenFilter

```
ReadLenFilter(const std::string &name,
               size_t threshold = 1,
               const QualityEncoding &encoding=SangerEncoding);
```

Filters reads less than `threshold` bases out of a stream.

## ReadLenCounter

```
ReadLenCounter(const std::string &name,  
               const QualityEncoding &encoding=SangerEncoding);
```

Counts the length distribution of all reads.

## ReadTruncator

```
ReadTruncator(const std::string &name,  
              size_t threshold=64,  
              const QualityEncoding &encoding=SangerEncoding);
```

Truncates reads to `threshold` bases long, and removes reads from the stream less than `threshold` bases long.



---

### Developer documentation

---

#### Contents

- *Developer documentation*
  - *Compiling static binaries*

### Compiling static binaries

```
make -f src/Makefile.static
```



## CHAPTER 4

---

### Indices and tables

---

- `genindex`
- `search`