CHAPTER 2

# QTrim – A Novel Algorithm for Quality Trimming Tool

## Introduction

The invention of ultra deep high throughput next generation sequencing (NGS) technology (see Chapter 1 for details on NGS), like Roche/454 pyrosequencing, has revolutionized the field of virology (Antonelli, 2013). The current 454 pyrosequencing technology allows for the generation of as many as one million high quality sequence reads with read lengths of up to 1000 base pairs. This technology therefore provides unprecedented sampling depth to study highly heterogeneous HIV quasispecies (Beerenwinkel and Zagordi, 2011). Since large volumes of sequence data are produced, the data quality has to be high because the manual curation of quality and sequencing errors, as could be done with traditional Sanger sequencing, is no longer feasible. The large amount of data, resulting in higher coverage per base, compensates for the lower quality bases to a certain level. However, sequencing from a sample of HIV quasispecies would mean that every sequence read could represent a unique variant. One of the major limitations of pyrosequencing is that sequence quality is not consistent, either within a read or between reads generated in the same sequencing run (Huse et al., 2007) and, thus, downstream analysis of such data may be compromised as a result of low quality data (Mardis, 2008). The quality scores for the current generation 454 sequencing platforms are similar to PHRED scores (Ewing and Green, 1998) and represent the probability of a base call error at each individually base in a read (Brockman et al., 2008). These quality scores range from 0 to 40 and are log-scaled (Cock et al., 2010), meaning that scores of 30 and 40 represent a probability of an incorrect base call of 1 in 1000 and 1 in 10000 respectively. As with most sequencing approaches, the quality of sequence data generated using 454 pyrosequencing decreases linearly across a sequence read (Figure 2.1) (Huse et al., 2007; Suzuki et al., 2011). The identification of a true base with a high quality score is pertinent, particularly in HIV drug resistance studies where low quality sequence data might represent artificial viral mutations (Kunin et al., 2009) affecting resistance test on the whole. Thus, in many instances it is imperative to undertake quality filtering of 454 sequence data to remove those low scored bases prior to subsequent analysis.

Quality trimming generally entails some form of iterative removal from one or both ends of a sequence read with the primary goal to ensure that the resultant read is of high quality. Quality trimming tools range from strict approaches that have zero tolerance of low quality base calls in the output reads through to averaging approaches that maximize read length by allowing the inclusion of a proportion of low quality base calls within an output read (Chou and Holmes, 2001b). Here, we describe a quality trimming algorithm (QTrim) that uses a novel averaging approach to maximize the output of high quality reads from 454 sequence data. To enable its use by a broad range of researchers, QTrim is available as a standalone python executable script for individuals with computational expertise and as a web-interface for individuals with little, or no, bioinformatics experience.

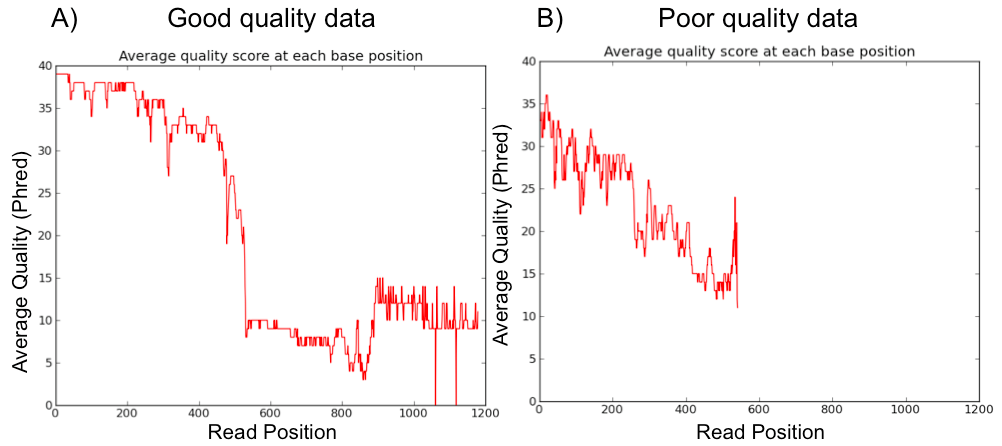


Fig 2.1: Figure: Graphical plot showing the average quality across the sequence reads in good quality data (data with mostly good quality bases) and poor quality data (data with most poor quality bases). Quality score of called base is inconsistent. A) Quality score slowly drops in good quality data. B) Quality score drops down quick in poor quality data

### Methods and Materials

QTrim takes as input a fastq file or a fasta file with its associated .qual file. Sequence data, from NGS technology, in the input file might have undetermined ambiguous bases or “N”s spread across. These ambiguous bases, with quality score of zero, do not contribute data analysis, and thus, QTrim users have an option to remove these “N”s before initiation of quality trimming. Although the quality of bases decreases towards the end of sequence reads, that might not always be the case, where the sequencing might begin with low quality bases. QTrim provides an option to quality trim from both 5’ and 3’ ends of a sequence read. These user options are provides as four different modes of quality trimming in QTrim.

1. Trimming from 3’ end only with removal of ambiguous ‘N’ bases in the middle of reads. This is mode 1.
2. Trimming from 3’ end only without removing ambiguous ‘N’ bases in the middle of reads. This is mode 2 and is the default mode.
3. Trimming from 5’ and 3’ ends with removal of ambiguous ‘N’ bases in the middle of reads. This is mode 3.
4. Trimming from 5’ and 3’ ends without removing ambiguous ‘N’ bases in the middle of reads. This is mode 4.

QTrim execution requires three parameters to be set by the user:

1. Mean quality score: It is the threshold value that each trimmed read must satisfy
2. Minimum read length: It is the minimum allowed read length (base pairs) a read can reach during trimming before being discarded
3. Sliding window size: It is the size of a subsequence that slides inward from the ends of a sequence during trimming. If no window size is defined at input the default value is set to the user-defined minimum allowed read length meaning that this parameter is optional.

::QTrim:BMCBioinformatics 2:FiguresAndTables:Figure1.pdf

Fig 2.2: Systematic work flow of QTrim

QTrim execution begins with removal of the trailing ambiguous ‘N’ bases at the 3’ end of a sequence read and, depending upon mode chosen, the ‘N’ bases between high quality bases across a sequence reads are either removed or left untouched.

The QTrim algorithm comprises three sequential steps (Figure 2.2) with the first step iteratively trimming single nucleotides from the 3’ end of a read until it’s mean quality score satisfies the quality threshold (Figure 2.3 A). The second step is a sliding window approach that evaluates the mean quality score of the last N number of nucleotides at 3’ end (N is equal to the window size). If the mean quality score of the bases within the window is not satisfied, a single base is deleted from the 3’ end and the window is reset (Figure 2.3 B). Once the quality threshold within the window is satisfied the final step iteratively trims all nucleotides from the 3’ end until the quality score of the last nucleotide in the read ≥ quality threshold value (Figure 2.3 C). If the length of the resulting trimmed read is less than the minimum allowed read length the read is discarded. Depending upon the mode chosen, QTrim enables trimming to occur simultaneously at both the 5’ and 3’ ends.

### Graphical plots in QTrim

QTrim generates graphical plots invoking the option for plotting. The plots are generated for the dataset before and after quality trimming states, which allows direct observation and comparison of the dataset at both states for quality score at every 10th base position across sequence reads (Figure 2.4), read length distribution (Figure 2.5) and average read length (Figure 2.6).

QTrim_sequential_steps.pdf

Fig 2.3: Sequential steps of trimming in QTrim. A) Iterative trimming with mean across sequence B) Iterative trimming with mean in window C) Iterative trimming with mean in last nucleotide. The last nucleotide has score 25, which satisfies user mean. QTrim does no further trimming. The final read length is greater than users read length. Therefore, the sequence is output to a file.

:Figures_for_thesis:qtrim_quality_score_trend.pdf

Fig 2.4: QTrim generated plot showing the average quality score across the sequence reads at every 10th base (Bin size: 10) in (A) good quality dataset and (B) poor quality dataset. The green line represents the number of sequences (represent by secondary axis at the right) for evaluating the average quality score.

qtrim_readlength_distribution.pdf

Fig 2.5: QTrim generated plot showing distribution of sequence reads with read length before trimming (A) and after trimming (B)

:Figures_for_thesis:qtrim_mean_score.pdf

Fig 2.6: QTrim generated plot showing distribution of sequence reads with mean score before trimming (A) and after trimming (B).

Table 2.1 Total reads, mean read length and number of poor bases in the test data.

|  |  |  |  |
| --- | --- | --- | --- |
| Test Data | Total Reads | Mean read length | Number of bases with q<20 |
| Good Quality Data | 15868 | 484 | 687945 |
| Poor Quality Data | 33022 | 583 | 11256293 |

## Test Data

Two sequence data with extreme difference in untrimmed quality scores, which we define as good quality and poor quality, were used for testing the efficiency and the sensitivity of QTrim quality trimming. The good quality 454 sequence data used here was defined as sequence reads with consistently high quality scores (q≥20) at called bases until close to 400bp (2.6 A). The poor quality data, on the other hand, exhibited low quality calls (q<20) spread throughout the entire read for all generated sequence reads (2.6 B). The presumed poor quality data, which is amplicon based ultra deep sequencing of HIV reverse transcriptase gene, was obtained from a study to characterize the emergence and persistence of drug resistant mutations in HIV-1 subtype C infections in Karonga district in Malawi (Bansode et al., 2013). (ADD THE DETAILS OF GOOD QUALITY DATA.)

### Software used

QTrim algorithm for quality trimming is fully coded with python high level programming language and biopython package ([www.biopython.org](http://www.biopython.org)). The plots are generated using python plotting modules – Matplotlib (Hunter, 2007), along with numpy ([www.numpy.org](http://www.numpy.org)).

### Results

The details on total reads, average read length and number of poor quality bases of two selected test data is shown in Table 2.1.

Both the data sets were trimmed with QTrim at parameters set of mean quality in sequence reads as 20 (Q20) and 30 (Q30) with constant minimum read length as 50. The data sets were also trimmed with other widely used tools - PRINSEQ (Schmieder and Edwards, 2011), the Modified-Mott algorithm implemented in Geneious (Kearse et al., 2012), 454/Roche’s newbler v2.6, FASTX (Blankenberg et al., 2010) and the lucy algorithm (Chou and Holmes, 2001a; Li and Chou, 2004) implemented in clean\_reads (Blanca et al., 2011). Except Roche’s Newbler, which only trims at Q20, all other tools were executed at Q20 and Q30 with constant minimum read length of 50. The result of trimming from all the tools is shown in Table 2.2 for poor quality data and in Table 2.3 for good quality data. The performance of QTrim was compared at two quality threshold levels (Q20 and Q30) against the above mentioned tools on the basis of total output reads, longest average read length in the output, lower number of poor quality bases and time of execution. We assume the best tool should be the one that output highest number of cleaned sequence reads with longest average read length.

When applied to the good quality dataset, our observation showed that QTrim and PRINSEQ performed at equivalent level (Figure 2.7 A and C) (Table 2.2 Q20 and Table 2.3 Q20) and outperformed all the other tools (Figure 2.7 A and C), with 15829 trimmed reads with a mean length of 448 nucleotides output by QTrim and 15825 trimmed reads with a mean length of 450 nucleotides output by PRINSEQ in the Q20 threshold analysis. In the more stringent Q30 analysis, the number of output reads remained similar to that of the Q20 analysis (Table 2.2 Q30 and Table 2.3 Q30) however the mean read length reduced to 422 and 426 nucleotides for QTrim and PRINSEQ respectively. For both the Q20 and Q30 analysis all of the other approaches output a comparable number of trimmed reads to QTrim and PRINSEQ, however their average read lengths were significantly shorter (Figure 2.7 A and C).

Table 2.2: Quality trimming of poor quality data with QTrim and other tested tools at mean quality of 20 (Q20) and 30 (Q30) and minimum read length 50

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Total output reads | | Mean read length | | Total bases with q<20 | |
| Q20 | Q30 | Q20 | Q30 | Q20 | Q30 |
| QTrim | 32818 | 23321 | 273 | 162 | 2107979 | 251894 |
| Clean\_reads | 31379 | 16940 | 157 | 89 | 536342 | 535 |
| PRINSEQ | 32381 | 20717 | 282 | 176 | 2281685 | 300029 |
| FASTX | 1242 | 279 | 154 | 139 | 11460 | 681 |
| Geneious | 29142 | 13218 | 178 | 90 | 436947 | 231 |
| Roche’s Newbler | 32047 | NA | 252 | NA | 1743863 | NA |

Table 2.3: Quality trimming of good quality data with QTrim and other tested tools at mean quality of 20 (Q20) and 30 (Q30) and minimum read length 50

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Total output reads | | Mean read length | | Total bases with q<20 | |
| Q20 | Q30 | Q20 | Q30 | Q20 | Q30 |
| QTrim | 15829 | 15450 | 448 | 422 | 300135 | 115302 |
| Clean\_reads | 15781 | 15267 | 417 | 315 | 140302 | 1464 |
| PRINSEQ | 15825 | 15471 | 450 | 426 | 324949 | 143736 |
| FASTX | 13176 | 9787 | 466 | 450 | 129765 | 38340 |
| Geneious | 15709 | 15113 | 428 | 338 | 134715 | 5410 |
| Roche’s Newbler | 15788 | NA | 437 | NA | 195610 | NA |

:Figures_for_thesis:qtrim_results.pdf

Fig 2.7: Comparison of QTrim with other tools on account of total output reads, mean and mean read length of output reads using two datasets (good quality – A and C, poor quality – B and D) at mean threshold quality score 20 and 30.

When applied to the poor quality data, PRINSEQ and QTrim were, by far, the two best performing approaches (Figure 2.7 B and D) with 32818 trimmed reads with a mean length of 273 nucleotides output by QTrim and 32381 trimmed reads with a mean length of 282 nucleotides output by PRINSEQ in the Q20 threshold analysis. The lower quality of this data is reflected in the much shorter trimmed reads output from the analysis when compared to the trimmed read lengths output by the analysis of the good quality data.

This is further evident when the stringent Q30 analysis of the poor quality data was undertaken with the average trimmed read length reducing from 273 nucleotides (Q20) to 162 nucleotides (Q30) for QTrim and from 282 nucleotides (Q20) to 176 nucleotides (Q30) for PRINSEQ. Further, the dramatic reduction in the number of reads output for all methods in the Q30 analysis (ranging from a 29% reduction in the number of high quality reads output between the Q20 and Q30 analysis in QTrim to a 77% reduction in FASTX), indicates that, for many sequence reads, they were of too low quality to pass the minimum read length threshold.

Upon comparison with all other approaches, QTrim performs equally as well as the best of these methods (PRINSEQ). The trimmed reads output by PRINSEQ are, on average, slightly longer than those output by QTrim (Figure 3). Upon further examination, however, this is as a result of PRINSEQ outputting a higher number of low quality bases (quality score < 20) at the 3’ end of its trimmed reads. For example, PRINSEQ output 8% as many low quality bases than QTrim in the Q20 trimming of both datasets tested here, and outputs 17% and 25% as many low quality bases in the Q30 trimming of the poor quality and good quality datasets respectively. We find that this is the case in all of the methods that use an averaging approach for quality trimming. As soon as the minimum quality score in a read satisfies the quality threshold, the read is output as trimmed without any further analysis. In QTrim, however, we employ a further two steps that ensures that low quality bases at the 3’ end of quality trimmed reads are removed. Thus, while the reads may be slightly shorter than those output by PRINSEQ, users can have a high level of confidence that quality is consistent across the length of the quality trimmed reads output by QTrim.

Finally, the operation of QTrim is twice as fast as PRINSEQ (379372 versus 189966 bases trimmed per second) on a standard desktop computer with a 2 GHz Intel**®** Core**™** Duo CPU and 2GB of RAM.

### Discussion

QTrim is a novel algorithm implementing averaging approach for sensitive quality trimming of 454 sequence data. QTrim does not filter singleton reads or duplicated reads and does not check or trim out adaptor sequence.

Some best performing tools like PRINSEQ has lots of options that confuse users. QTrim options are simple to use with less options. In fact, only the input file will suffice QTrim to execute with default optimized settings, including the output filename. Some tools use secondary software for quality trimming. This software has to be preinstalled before executing the trimming tools. Users refrain away from these tools to avoid hassles with secondary software. QTrim is standalone executable file. Though QTrim requires biopython, matplotlib and numpy, these are packed within the executable file.

## Conclusion

QTrim is a fast, highly sensitive and accurate algorithm that outperforms all available approaches for quality trimming of 454 sequence data. A noteworthy feature is that it enables sensitive trimming of sub-optimal sequence data thereby enabling researchers to undertake downstream analysis on lesser quality sequence data that otherwise may have been discarded. The command line python version of QTrim can be easily incorporated into sequence analysis pipelines, while the web interface maintained at <https://hiv.sanbi.ac.za/software/qtrim> enables users with little or no bioinformatics experience to undertake quality trimming of their high-throughput sequencing data. It is free for all academic researchers and commercial users need to contact [simon@sanbi.ac.za](mailto:simon@sanbi.ac.za) for proper license.

**Bibliography**

Antonelli, G (2013) Emerging new technologies in clinical virology. *Clin Microbiol Infect* **19**: 8-9.

Bansode, V, McCormack, GP, Crampin, AC, Ngwira, B, Shrestha, RK, French, N, Glynn, JR, Travers, SA (2013) Characterizing the emergence and persistence of drug resistant mutations in HIV-1 subtype C infections using 454 ultra deep pyrosequencing. *BMC Infect Dis* **13**: 52.

Beerenwinkel, N, Zagordi, O (2011) Ultra-deep sequencing for the analysis of viral populations. *Curr Opin Virol* **1**: 413-418.

Blanca, JM, Pascual, L, Ziarsolo, P, Nuez, F, Cañizares, J (2011) ngs\\_backbone: a pipeline for read cleaning, mapping and SNP calling using Next Generation Sequence. *BMC Genomics* **12**: 285.

Blankenberg, D, Gordon, A, Kuster, GV, Coraor, N, Taylor, J, Nekrutenko, A (2010) Manipulation of FASTQ data with Galaxy. *Bioinformatics* **26**: 1783-1785.

Brockman, W, Alvarez, P, Young, S, Garber, M, Giannoukos, G, Lee, WL, Russ, C, Lander, ES, Nusbaum, C, Jaffe, DB (2008) Quality scores and SNP detection in sequencing-by-synthesis systems. *Genome research* **18**: 763-770.

Chou, H-H, Holmes, MH (2001a) DNA sequence quality trimming and vector removal. *Bioinformatics* **17**: 1093–1104.

Chou, HH, Holmes, MH (2001b) DNA sequence quality trimming and vector removal. *Bioinformatics* **17**: 1093-1104.

Cock, PJA, Fields, CJ, Goto, N, Heuer, ML, Rice, PM (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Research* **38**: 1767-1771.

Ewing, B, Green, P (1998) Base-Calling of Automated Sequencer Traces UsingPhred. II. Error Probabilities. *Genome Research* **8**: 186-194.

Hunter, JD (2007) Matplotlib: A 2D Graphics Environment. *Computing in Science and Engineering* **9**: 90-95.

Huse, SM, Huber, JA, Morrison, HG, Sogin, ML, Welch, DM (2007) Accuracy and quality of massively parallel DNA pyrosequencing. *Genome biology* **8**: R143.

Kearse, M, Moir, R, Wilson, A, Stones-Havas, S, Cheung, M, Sturrock, S, Buxton, S, Cooper, A, Markowitz, S, Duran, C, Thierer, T, Ashton, B, Meintjes, P, Drummond, A (2012) Geneious Basic: An integrated and extendable desktop software platform for the organization and analysis of sequence data. *Bioinformatics* **28**: 1647-1649.

Kunin, V, Engelbrektson, A, Ochman, H, Hugenholtz, P (2009) Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. *Environ Microbiol* **12**: 118-123.

Li, S, Chou, H-H (2004) LUCY2: an interactive DNA sequence quality trimming and vector removal tool. *Bioinformatics* **20**: 2865–2866.

Mardis, ER (2008) The impact of next-generation sequencing technology on genetics. *Trends in genetics* **24**: 133.

Schmieder, R, Edwards, R (2011) Quality control and preprocessing of metagenomic datasets. *Bioinformatics* **27**: 863-864.

Suzuki, S, Ono, N, Furusawa, C, Ying, B-W, Yomo, T (2011) Comparison of Sequence Reads Obtained from Three Next-Generation Sequencing Platforms. *PLoS ONE* **6**: e19534.