CHAPTER 2

# QTrim – A Novel Algorithm for Quality Trimming Tool

## 2.1 Introduction

The invention of high-throughput sequencing (HTS) technologies, such as Roche/454 pyrosequencing, has revolutionized the field of virology (Antonelli, 2013). The current Roche/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 (<http://www.454.com>). 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. One way to ensure high quality data is repetitive HTS of a genomic region generating large amount of data, resulting in higher coverage per base that 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. Therefore, each sequence read has to be quality controlled, independent of other sequence reads from same genomic region.

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 (Gilles et al., 2011; 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 methods range from strict approaches that have zero tolerance of low quality base calls in the output reads (Delport et al.; Gianella et al., 2011) through to averaging approaches allows the inclusion of a proportion of low quality base calls within an output read (Chou and Holmes, 2001; Schmieder and Edwards, 2011). Algorithms like PRINSEQ (Schmieder and Edwards, 2011), Geneious (Kearse et al., 2012) and LUCY (Li and Chou, 2004) that use averaging approach are available but the output reads have large number of poor quality bases or large number of reads are discarded. We have developed a quality trimming algorithm (QTrim) that uses a novel averaging approach to minimize poor quality bases and 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.

### 2.2 Methods and Materials

QTrim is a python quality trimming bioinformatics tool and takes as input a fastq file or a fasta file with its associated quality (.qual) file. If a combination of fasta and qual files are submitted, they are converted to fastq format. QTrim reads sequences from the input file one at a time using biopython package ([www.biopython.org](http://www.biopython.org)) (Cock et al., 2009). QTrim trims the input sequences based on the nucleotide quality score. The first step in QTrim is removal of any ambiguous base (Ns) that have quality score of zero at the 3’ end of sequences. It is then, followed by three sequential trimming steps detailed as below before final output of the clean reads (Figure 2.1):

1. QTrim checks if a sequence read is greater or equal to the minimum required sequence read length. If it satisfies then, the mean quality of the bases across a read is checked. If the mean quality is less than threshold quality, a single base is trimmed out from 3’ end. This process is looped until the mean quality score across the read satisfies the quality threshold (Figure 2.2 A). The read is discarded if it does not satisfy the required minimum sequence read length.
2. For a sequence read that satisfies both minimum read length and mean quality in step 1 above, further trimming is done using 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 less than the required mean quality, a single base is trimmed out from the 3’ end and the window is reset (Figure 2.2 B). The process is repeated until the mean quality in the window is satisfied.

The sequence is discarded if the read length is below the minimum read length although there can be further trimming.

1. In the third and the last trimming step, QTrim checks the quality score of the last nucleotides from the 3’ end. QTrim iteratively trims out the last nucleotide until the quality score of the last nucleotide in the sequence is greater or equal to quality threshold value (Figure 2.2 C). Finally, the read is saved in an output file.

For desired optimal quality trimming, user is required to set three parameters – mean quality across a sequence read, minimum output sequence read length and sliding window size. The quality threshold is the mean quality that each trimmed read must satisfy, the second defines the minimum allowed read length (base pairs) a read can reach during trimming before being discarded, while the final parameter (optional) defines the window size to be used during trimming. If no sliding window size is defined at input the default value is set to the user-defined minimum allowed read length.

The default mode of QTrim execution trims the poor quality bases from 3’ end of sequence reads, ignoring any ambiguous bases (Ns) interspersed among the high quality bases in the reads. Depending upon simultaneously trimming from 5’ and 3’ ends, trimming only from 3’ end and ignoring or removal of interspersed ambiguous Ns in the reads, there are four modes of QTrim execution.

1. Mode 1: Trimming from 3’ end with removal of interspersed ambiguous bases (Ns) in the reads.
2. Mode 2: Trimming from 3’ end without removal of interspersed ambiguous bases (Ns) in the reads. This is the default mode.
3. Mode 3: Trimming from 5’ and 3’ ends with removal of interspersed ambiguous bases (Ns) in the reads.
4. Mode 4: Trimming from 5’ and 3’ ends without removal of interspersed ambiguous bases (Ns) in the reads.

### 2.3 Graphical plots in QTrim

QTrim uses matplotlib (Hunter, 2007) and numpy to generate following analytical plots; each plot is plotted for both raw and trimmed data

1. **Distribution of number of reads by mean quality across the sequence read**

QTrim calculates the mean quality of every sequence reads and groups them by the mean quality. The number of sequences in each group of quality score is plotted as shown in figure 2.3. The plot gives an overview of the data quality with the spread of mean quality. The good quality data will have large number of sequence reads with high mean quality whereas poor quality data will have large number of sequence reads with low mean quality.

1. **Distribution of number of reads by sequence read length**

QTrim calculates the length of every sequence reads and groups them by the length. The number of sequences in each group of sequence length is then plotted as shown in figure 2.4. The plot gives an overview of the data quality with the range of sequence lengths. The good quality data will have large number of sequence reads clustered together whereas poor quality data will have sequence reads with wide range of read lengths.

1. **The trend of the mean quality score at the interval of 10 bases in the reads**

The mean quality score is calculated from every 10th base positions of every reads. The base position, the average quality score and the number of sequences contributing to the average quality score at the base position are plotted as shown in figure 2.5. The plot shows variation in the quality score of nucleotides with increase in the sequence read length. The good quality data will have consistent high mean quality before the quality reduces while poor quality data will have steep drop in mean quality.

## 2.3.1 Test Data

Two previously sequenced Roche/454 pyrosequenced fastq datasets (A and B) were chosen to test QTrim quality trimming and compare with other widely used methods. The dataset A has only 9% of nucleotides with quality score below 20, has large number of sequences with higher mean quality score of 20 (Figure 2.6 A) and has low range of read length distribution (Figure 2.7 A). The dataset B has significantly higher (59%) number of nucleotides with quality score below 20, has large number of sequences with lower mean quality score (Figure 2.6 B) and has wide range of read length distribution (Figure 2.7 B). This shows that there is large difference in quality between the two test data. We define test data A and B as good quality and poor quality data respectively.

The poor quality data was obtained from amplicon based ultra deep sequencing of the HIV reverse transcriptase gene. The original study aimed to characterize the emergence and persistence of drug resistant mutations in HIV-1 subtype C infected individuals from the Karonga district in Malawi (Bansode et al., 2013). (ADD THE DETAILS OF GOOD QUALITY DATA.)

### 2.4 Results

Both the good and poor quality test datasets were trimmed using QTrim with the parameters set as follows: mean quality in sequence reads of 20 (Q20) and 30 (Q30) with a constant minimum read length of 50. The data sets were also trimmed with other widely used methods, including PRINSEQ (Schmieder and Edwards, 2011), the Modified-Mott algorithm implemented in Geneious (Kearse et al., 2012), Roche/454 Newbler v2.6, FASTX (Blankenberg et al., 2010) and the Lucy algorithm (Chou and Holmes, 2001; Li and Chou, 2004) implemented in clean\_reads (Blanca et al., 2011). Apart from Newbler v2.6, which is preset to trim at Q20, all other methods were executed at Q20 and Q30 with a constant minimum read length of 50. The performance of QTrim was compared to the above mentioned tools on the basis of the total number of reads in the output, longest average read length in the output, number of poor quality bases in the output, and time of execution. The best tool in the comparison should generate the highest number of trimmed sequencing reads that satisfy the quality threshold with the longest average read length.

When applied to the good quality dataset, QTrim and PRINSEQ performed at an equivalent level (Figure 2.7 A and C; Table 2.2 Q20 and Table 2.3) and outperformed all the other methods (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 terms of the percentage of total bases in the output, again QTrim and PRINSEQ outperformed the other methods while they both allowed some poor quality bases (Table 2.1).

In the more stringent Q30 analysis, the number of produced reads remained similar to that of the Q20 analysis (Table 2.1) however the mean read length reduced to 422 and 426 nucleotides for QTrim and PRINSEQ respectively. The percentage of bases and poor quality bases in QTrim and PRINSEQ output are reduced as well; PRINSEQ has one percent bases greater than QTrim while QTrim has half percent poor quality bases less than PRINSEQ (Table 2.1). For both the Q20 and Q30 analysis all of the other approaches produced a comparable number of trimmed reads to QTrim and PRINSEQ, however the average read lengths were significantly shorter (Figure 2.8 A and C).

When applied to the poor quality data, PRINSEQ and QTrim were, by far, the two best performing approaches (Figure 2.8 B and D). A total of 32818 trimmed reads with a mean length of 273 nucleotides was produced by QTrim and 32381 trimmed reads with a mean length of 282 nucleotides was produced by PRINSEQ in the Q20 threshold analyses. The lower quality of this data is reflected in the much shorter trimmed reads produced and over 50% of bases trimmed out by all methods (Table 2.2) from this analysis when compared to the trimmed read lengths produced during the analysis of the good quality data. The performance was further evident when the stringent Q30 analysis of the poor quality data was undertaken. The average trimmed read length reduced from 273 nucleotides (Q20) to 162 nucleotides (Q30) for QTrim and from 282 nucleotides (Q20) to 176 nucleotides (Q30) for PRINSEQ. PRINSEQ has the highest percentage of poor quality bases (Table 2.2). Further, the dramatic reduction in the number of bases produced for all methods in the Q30 analysis (ranging from a 80% reduction in the number of bases Q30 analysis in QTrim, to 99.8% reduction in FASTX (Table 2.2)), indicates that, for many reads, the sequences were of too low quality to pass the minimum read length threshold.

Finally, when compared by execution time, QTrim is twice as fast as PRINSEQ (379372 versus 189966 bases trimmed per second) while other methods take less time (Figure 2.9) on a standard desktop computer with a 2 GHz Intel**®** Core**™** Duo CPU and 2GB of RAM.

### 2.4.1 QTrim Web Service

QTrim web service has been developed to facilitate quick and easy quality trimming of HTS sequence data for bioinformaticians with little knowledge on command line tool execution. The web service is available at <http://hiv.sanbi.ac.za/tools#/qtrim> (Figure 2.10). First time users are required to register to get access for trimming job submission. Registration is free for academic users whereas commercial users need to pay. Users can login using their userid/email and password after registration. On the QTrim job submission page (Figure 2.11), users can upload the HTS sequence data files, provide a job name and submit the job straight away with default parameter settings or change the parameter settings like mean quality, minimum read length, mode of trimming in the advance settings and then submit the job for quality trimming.

Users can view their job details using the job name. The plots generated for the uploaded raw data and trimmed result data will be displayed for the selected job. Fore more details on plots, see the section “**Graphical plots in QTrim**” under **Methods and Material**. Users can also download the trimmed results files from a link on the result web page of the selected job.

### 2.5 Discussion and Conclusion

QTrim is a novel algorithm implementing an averaging approach for sensitive quality trimming of 454 sequence data. Upon comparison with all other approaches, QTrim

performs equally as well as the best of these methods (PRINSEQ (Schmieder and Edwards, 2011)) on the basis of total output reads and the mean output read length. The trimmed reads produced by PRINSEQ are, on average, slightly longer than those from QTrim. Upon further examination, however, this is as a result of PRINSEQ allowing a higher number of low quality bases (quality score < 20) at the 3’ end of its trimmed reads. For example, PRINSEQ generates 8% more low quality bases than QTrim in the Q20 trimming of both datasets tested here, and 17% and 25% more low quality bases in the Q30 trimming analyses 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 defined as trimmed without any further analysis. In QTrim, however, we employ two further steps, which ensure that low quality bases at the 3’ end of quality trimmed reads are removed. Thus, while the reads may be slightly shorter than those produced by PRINSEQ, users can be confident that the quality of the generated reads is consistent across the length of the quality trimmed data produced by QTrim.

QTrim performance comparison with other approaches is distinctive when the trimming is done on poor quality data. While most other tools trim out almost all reads, QTrim is able to “rescue” high quality nucleotides more than PRINSEQ, which has comparable total reads output and mean read length. This means QTrim 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.

Fastx (Blankenberg et al., 2010) has distinctly the highest speed of execution and this is because it trims out all nucleotides once the required percentage of high quality nucleotides is obtained. Other methods, including QTrim that check the mean quality of nucleotides trim single base at each time. Although QTrim is slower than Geneious (Kearse et al., 2012), it is comparable with Clean\_reads (Blanca et al., 2011) and Roche’s Newbler v2.6 and as much as twice the speed of PRINSEQ.

From users prospective, QTrim is simple to use with less parameter to define while other methods require complex combinations of parameters to be defined. QTrim is available as standalone executable file with required library files of Biopython (Cock et al., 2009), matplotlib (Hunter, 2007) and numpy, which allows users to extract and execute straight away without installation of any secondary software. The executables are available for Linux and MacOSX and are downloadable from <https://hiv.sanbi.ac.za/software/qtrim>. This makes QTrim to be easily integrated into next generation sequence analysis pipeline.

**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 Res* **18**: 763-770.

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

Cock, PJ, Antao, T, Chang, JT, Chapman, BA, Cox, CJ, Dalke, A, Friedberg, I, Hamelryck, T, Kauff, F, Wilczynski, B (2009) Biopython:freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**: 1422-1423.

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.

Delport, W, Young, JA, Poon, AF, Pond, SLK A bioinformatics pipeline for the analysis and interpretation of HIV-1 ultradeep sequence data.

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

Gianella, S, Delport, W, Pacold, ME, Young, JA, Choi, JY, Little, SJ, Richman, DD, Kosakovsky Pond, SL, Smith, DM (2011) Detection of minority resistance during early HIV-1 infection: natural variation and spurious detection rather than transmission and evolution of multiple viral variants. *J Virol* **85**: 8359-8367.

Gilles, A, Meglecz, E, Pech, N, Ferreira, S, Malausa, T, Martin, JF (2011) Accuracy and quality assessment of 454 GS-FLX Titanium pyrosequencing. *BMC Genomics* **12**: 245.

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 biol* **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.