CHAPTER 2

# QTrim – A Novel Algorithm for Quality Trimming Tool

## Introduction

Different next generation DNA sequencing (NGS) technologies, (see Chapter 1 for details on NGS), have been developed and are widely in use. The NGS technologies have brought revolutions at genomic research and given hope to personal genome sequencing for $1000 (Drmanac, 2011). However, scientists have a notion that output sequence data from sequencing platforms have lower accuracy comparative to conventional Sanger sequencing method (Huse et al., 2007). The quality of sequence data is attributed to sequencing chemistry, hardware devices, reagents washing and base calling algorithms implemented in the sequencer (Ledergerber and Dessimoz, 2011). The huge amount of data generation attributing to higher coverage per base does compensate lower accuracy at some level. However, genome sequencing from sample with huge diversity or viral quasispecies would mean that every sequence read might represent a variant. In this case, a single base change might add value in final data analysis result.

Roche 454 FLX titanium technology generates approximately up to a billion base data (a read length of 1000 base and 1 million reads per run) ([www.454.com](http://www.454.com)). The phred software reads every base in DNA sequence trace file and assigns a log scaled quality score to it (Cock et al., 2010). Every base call accuracy is represented by a phred score ranging from 0 to 40, which from now on we call quality score. A quality score is just a probability of base call error and it is obtained from a formula:

phredd_score_formula.png

The higher quality score (close to 40) would mean higher probability of the base call being correct or higher accuracy and lower quality score (close to 0) would mean lower probability of the base being correct or lower accuracy. Given a threshold quality score, we consider a base is good quality, if its quality score is greater than the threshold value and poor quality, if lower than the threshold value. A limitation in Roche 454 sequencing is overall decreasing trend of quality score with increasing read length (Suzuki et al., 2011). Low accurate bases can compromise the downstream analysis. It is, therefore, highly recommended to quality check ultra deep sequence data. Manual removal of poor quality bases from a million sequences and a thousand read length is not feasible.

There are different approaches for removal of the poor quality bases from sequence reads. A widely accepted rule is to define a quality score as poor quality score using a threshold value. Quality score is inconsistent across sequence reads; it can decrease and increase across sequences making trough and crests as shown in figure 2.1.

Locating and removing every poor base from all sequences is not a good approach as it deletes the sequence information. This can misguide researches with huge number of deletion errors. One approach of quality trimming is the strict zero tolerance approach, in which only a sequence region is selected before a single base quality drops below threshold value. A quality-trimming tool applying this approach is HIV454 ([www.datamonkey.org](http://www.datamonkey.org)). This approach discards huge number of sequence reads; the output quality cleaned reads might not be sufficient for downstream analysis.

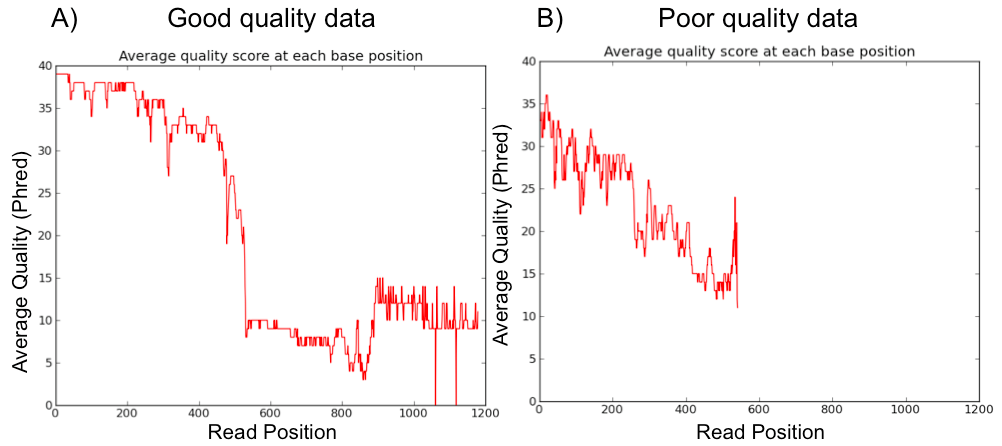


Fig 2.1: Figure: Graphical plot showing the average quality across the sequence reads in good quality data (data with mostly good quality bases) and poor quality data (data with most poor quality bases). Quality score of called base is inconsistent. A) In good sequence data, quality score slowly drops and then a sharp decrease at certain read length. B) In poor quality sequence data, quality score consistently drops.

Another similar approach with slight modification is to select a region of sequence each time the quality score drops below the threshold value. This generates multiple fragments of the sequence with only high quality bases. HIV454 ([www.datamonkey.org](http://www.datamonkey.org)) and galaxy NGS quality filtering tool (Goecks et al., 2010) implement this approach. This approach deletes the sequence information between the sequence fragments from same sequence reads.

Another approach in a widely used Fastx-toolkit (<http://hannonlab.cshl.edu/fastx_toolkit>), is an iterative trimming of base from right to left until it reaches a high quality base. This approach considers a smooth decrease in quality score, not crest and trough. Therefore it fails to remove huge number of poor quality bases spread across the sequences.

A widely accepted approach is an averaging approach, in which a base is trimmed usually from right side (3’) or both from right and left side, until the average quality score across the read is greater or equal to user’s threshold value. This approach outputs huge number of sequence reads unlike HIV454 ([www.datamonkey.org](http://www.datamonkey.org)) and trims poor quality bases in between high quality bases unlike FASTX-toolkit (<http://hannonlab.cshl.edu/fastx_toolkit>). The averaging method sacrifices some good quality bases to remove many poor quality bases. The averaging approach is applied in various modes. One mode of averaging approach includes the mean of quality score across a sequence read. A base is trimmed from one end, usually from right end, until the mean score is greater or equal to users mean input value. An example of a tool that implements this mode is **Geneious** ([www.geneious.com](http://www.geneious.com)). The application of this mode outputs sequence reads with much low average read length. Another mode of averaging approach includes the mean quality score in certain number of bases from the ends i.e. window method of averaging. This method considers certain number of bases, say 10 bases, and averages the quality score of these bases. A base is trimmed from the end if the average is less than users input value and the window slides inwards. This mode of averaging is better than the previous one. It outputs higher mean read length and total reads output. A limitation in this approach is that it is unable to trim out islands of poor quality bases in the sequences. We have developed a quality trimming tool with a novel averaging approach called QTrim (Figure 2.2), that maximizes number of sequence reads output and mean read length both for good quality data and poor quality data. QTrim is available both in a standalone executable version for those with some level of computational knowledge and a web version (give website) for those with minimum computational expertise. In web version, quality trimming can be done simply uploading a sequence file, with default settings.

### Basic requirements

QTrim requires python v2.6+, but not v3.0. Other requirements are biopython ([www.biopython.org](http://www.biopython.org)), matplotlib v1.0+ ([www.matplotlib.org](http://www.matplotlib.org), Hunter 2007) and numpy ([www.numpy.org](http://www.numpy.org)).

### Implementation

QTrim has four different modes of trimming poor bases:

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.

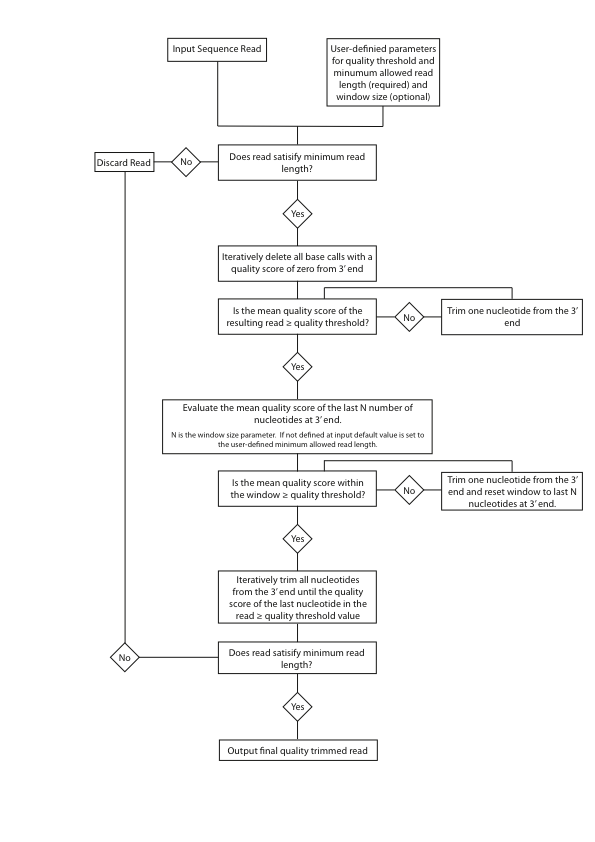


Fig 2.2: Systematic work flow of QTrim

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

QTrim accepts input file in two different file formats: FASTQ and FASTA with associated QUAL file. Submission of the input file run QTrim with all parameters in optimized default values. However, users can set three parameters for better sensitive quality trim:

1. Mean quality score
2. Minimum sequence read length
3. Window size, which is optional and the default is same as minimum sequence read length

QTrim execution begins with removal of all the ambiguous ‘N’ bases at the ends irrespective of the mode chosen. The trimming process, then, involves three sequential steps as shown in figure 2.2. Each step loops in itself until all the users conditions are met or ultimately the sequence gets discarded for having its length shorter than minimum read length. The three steps are:

1. Iterative trimming of a base from 3’ end until the average quality score across the sequence read is below the user required mean score (Figure 2.3 A).
2. Iterative trimming of a base from 3’ end using averaging approach with window sliding method. QTrim selects nucleotides of window size from 3’ end of the sequence read and evaluates for average score. A base is trimmed from 3’ end if the average score is below user required mean score. The window is slided one base inward and the process is repeated until mean quality score in the window is ≥ users mean score or the sequence is discarded (Figure 2.3 B).
3. Iterative trimming of a base from 3’ end until the last nucleotide quality score is ≥ users mean score. In this step, QTrim selects only the last nucleotide to evaluate the quality score for iterative trimming. QTrim stops trimming once the sequence read length is equal to users minimum read length (Figure 2.4 C).

The lower steps are not invoked once the sequence read is discarded in a step directly above. The sequence read is finally output in to a file after successfully going through the above sequential steps.

Selection of QTrim mode involving trimming from both 5’ and 3’ ends, invokes the above sequential steps at both the ends of sequence reads.

### Graphical plots in QTrim

Observation of data in graphical plots gives its overall view about quality. QTrim is able to generate plots invoking an the specific option in the command line. The plots are generated for the dataset before and after quality trimming, which allows direct observation and comparison of the dataset in both states (Figure 2.4, 2.5 and 2.6).

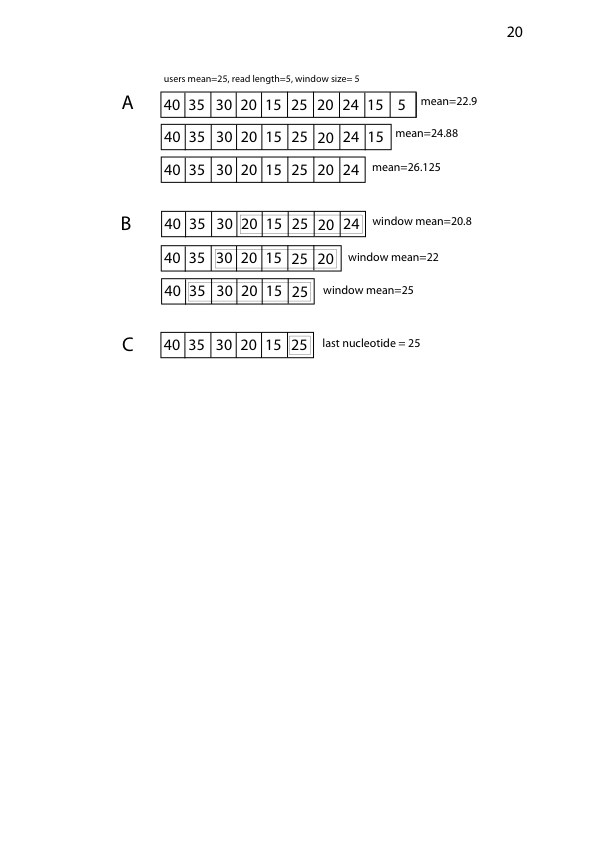


Fig 2.3: Sequential steps of trimming in QTrim. A) Iterative trimming with mean across sequence B) Iterative trimming with mean in window C) Iterative trimming with mean in last nucleotide. The last nucleotide has score 25, which satisfies user mean. QTrim does no further trimming. The final read length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

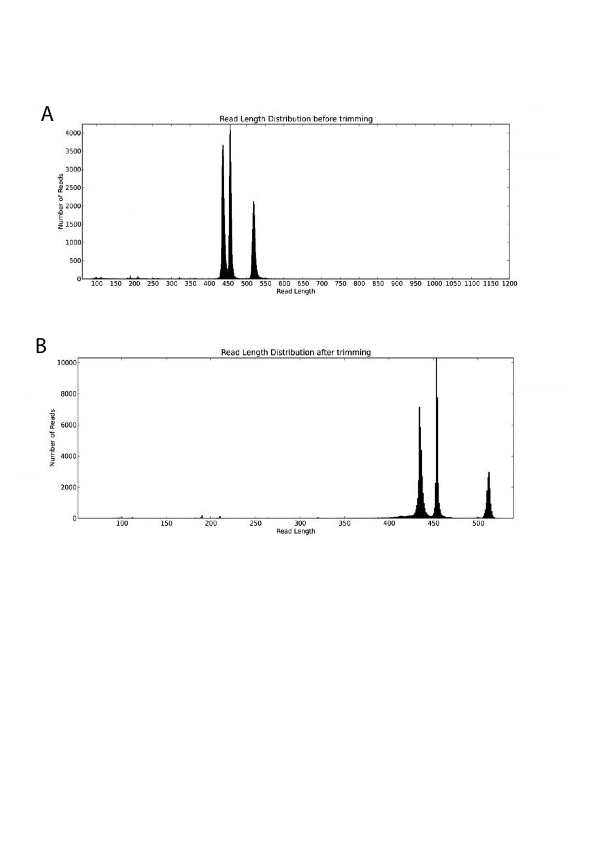


Fig 2.4: QTrim generated plot showing distribution of sequence reads with read length before trimming (A) and after trimming (B)

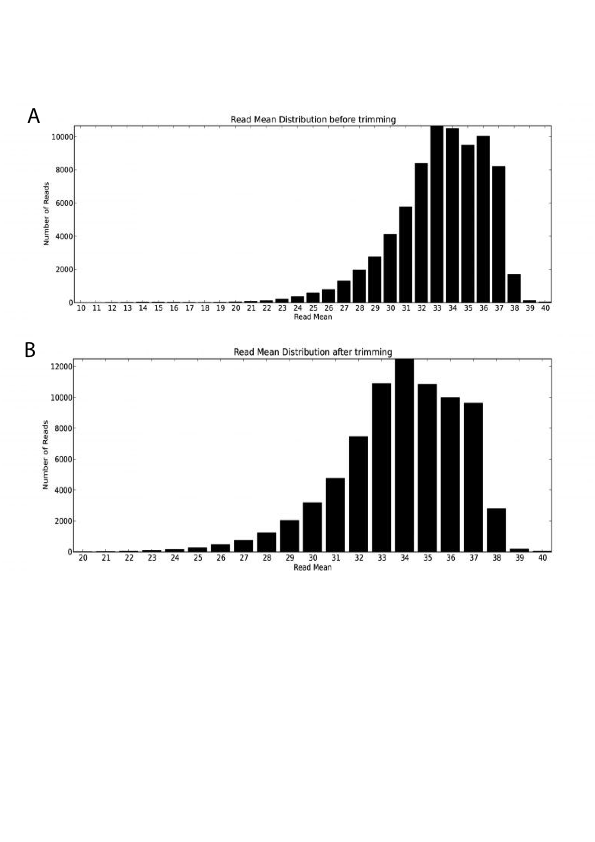


Fig 2.5: QTrim generated plot showing distribution of sequence reads with mean score before trimming (A) and after trimming (B).

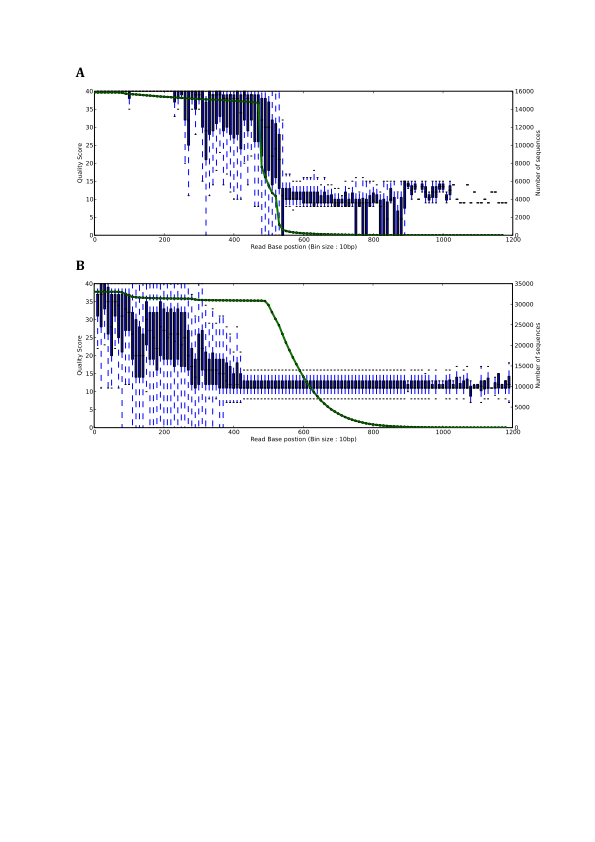


Fig 2.6: QTrim generated plot showing the average quality score across the sequence reads at every 10th base (Bin size: 10) in good quality dataset (A) and poor quality dataset (B). The green line represents the number of sequences (represent by secondary axis at the right) for evaluating the average quality score.

### Results

QTrim performance was compared 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, 2001; Li and Chou, 2004) implemented in clean\_reads (Blanca et al., 2011). Two raw ultra deep 454 sequence datasets were chosen with extreme difference in quality scores. One dataset is good quality data, which is defined as sequence data with consistent high quality score nucleotides for certain read length (Figure 2.1 A). The next dataset is poor quality data, which is defined as sequence data with huge inconsistent quality score. The quality score drops sharply very quick in poor quality dataset (Figure 2.1 B).

QTrim performance was compared with the above mentioned tools on the basis of total output reads, average read length in the output and time of execution. The two datasets were run through all the tools with similar input parameters. The best tool should be the one that output highest number of cleaned sequence reads with highest average read length.

Our observation with good quality dataset (total sequence reads = 15868) showed that QTrim and PRINSEQ performed at equivalent level (Table 2.1 Q20 and Table 2.2 Q20) and outperformed all the rest tools (Figure A and C). However, the result showed that the number of nucleotides with quality scores lower than mean is much higher in PRINSEQ. This was seen at both the parameters Q20 and Q30. Though the rest tools output comparable trimmed reads, the mean read length was significantly lower than QTrim and PRINSEQ (Figure 2.7 A and C).

|  |  |  |
| --- | --- | --- |
| Q20 | QTrim | PRINSEQ |
| Reads output | 15829 | 15825 |
| Mean read length | 448 | 450 |
| Total nucleotides with quality score < 20 | 300135 | 324949 |

|  |  |  |
| --- | --- | --- |
| Q30 | QTrim | PRINSEQ |
| Reads output | 15450 | 15471 |
| Mean read length | 422 | 426 |
| Total nucleotides with quality score < 30 | 112571 | 140691 |

Table 2.1: Total reads output and mean read length of output sequence reads from two top most performing tools QTrim and PRINSEQ while trimming **good quality data** at the parameter of mean quality score 20 (Q20) and 30 (Q30).

|  |  |  |
| --- | --- | --- |
| Q20 | QTrim | PRINSEQ |
| Reads output | 32818 | 32381 |
| Mean read length | 273 | 282 |
| Total nucleotides with quality score < 20 | 2107979 | 2281658 |

|  |  |  |
| --- | --- | --- |
| Q30 | QTrim | PRINSEQ |
| Reads output | 23321 | 20717 |
| Mean read length | 162 | 176 |
| Total nucleotides with quality score < 30 | 221391 | 259526 |

Table 2.2: Total reads output and mean read length of output sequence reads from two forerunner tools QTrim and PRINSEQ while trimming **poor quality data** at the parameter of mean quality score 20 (Q20) and 20 (Q30).

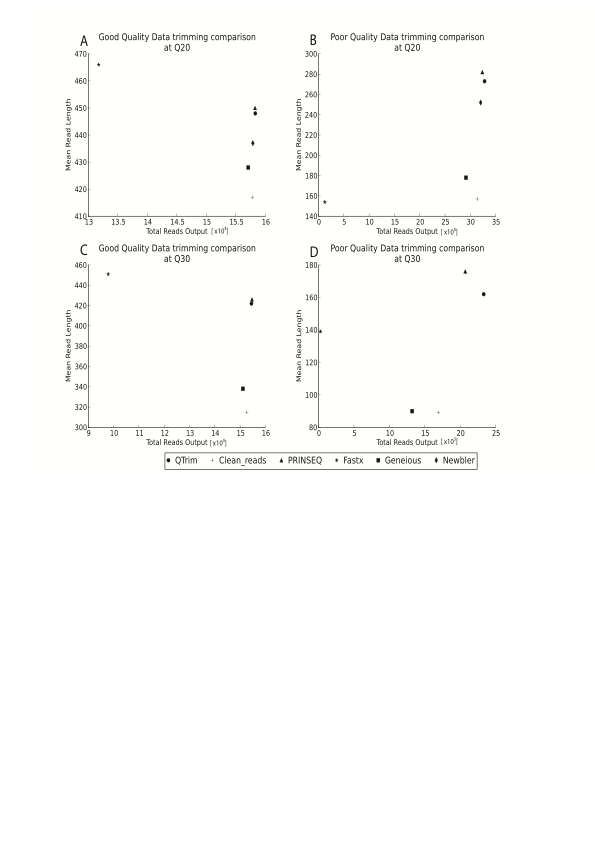


Fig 2.7: Comparison of QTrim with other tools on account of total output reads, mean and mean read length of output reads using two datasets (good quality – A and C, poor quality – B and D) at mean threshold quality score 20 and 30.

Trimming performance at Q20 (Fig 2.7 B) and Q30 (Fig 2.7 D) on poor quality data (Total reads =32814) showed that QTrim and PRINSEQ are, once again, the best performing tools showing high trimmed sequence reads output and high mean read length (Table 2.2). The huge volume of high quality data discarded in between Q20 and Q30 (minimum 29% by QTrim to maximum 77% by FASTX) explains that the input poor quality dataset has too many sequences with higher number of low quality bases. QTrim output highest number trimmed sequence reads for poor quality dataset. This explains the number of bases with quality score less than users mean value is much lower in QTrim. This point puts QTrim at stronger position than PRINSEQ. This extensive trimming result to few bases shorter reads, but long enough for data analysis. The comparative analysis of results from the best performing tools QTrim and PRINSEQ showed that the total number of output reads was higher for QTrim but the mean read length was longer for PRINSEQ. The close examination of output sequence reads showed that PRINSEQ higher number of low quality bases (Table) resulting to longer read. The low quality bases are ~8% more in PRINSEQ at Q20 trimming on both datasets and 17% and 25% more low quality bases in PRINSEQ than QTrim result at Q30 trimming of good and poor quality datasets respectively.

Most of the trimming tools using averaging approach, output reads soon the mean quality score is above users threshold value. This leaves low quality bases at 3’ end and islands of low quality bases in the middle of reads. However, QTrim is designed to trim those low quality bases as well. A shorter mean read length in QTrim output explains that the tool trims more poor quality bases at expense of few high quality bases. Although, QTrim outputs shorter reads, removal of more low quality bases gives consistent high quality across trimmed sequence and high confidence for downstream analysis.

The best performing tools above – QTrim and PRINSEQ was finally compared on the basis of execution speed in time. QTrim (189966 bases per sec) execution speed was twice as fast as PRINSEQ (379372 bases per sec). The tools were executed on standard laptop with 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.

Besides, averaging method, percentage based method is also in use. This method iteratively trims a nucleotide from 3’ end or selects the longest region of sequence reads with the percentage of low quality bases is lower than expected value; for e.g. Seqtrim (Falgueras et al., 2010). The same method is applied in reverse gear that trims a nucleotide until the percentage of high quality nucleotides in sequence reads is greater than expected value; for e.g. FastQ quality filter implemented in Galaxy (Blankenberg et al., 2010). Prior knowledge on raw input data is essential for users to determine the percentage of high or low quality bases in their final output. This only puts users in confusion or repeat the quality trimming process multiple times before users are happy with final trimmed data. Pyrocleaner (Jerome et al., 2011) cleans 454 sequence reads depending on factors like read length, standard deviation of whole reads, complexity across sequence reads or within sequence reads and number of ambiguous bases. These tools with complex and confusing options make users refrain away from applying. QTrim options are simple to use. In fact, only the input file will suffice QTrim to execute with default options, 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. However, user should have python v2.6 or 2.7 for QTrim execution,

### Conclusion

We have developed a highly sensitive and easy to use algorithm – QTrim for quality trimming Roche 454 sequence data. The visualizing point about QTrim is that the tool is able to trim sub-optimal quality data and rescue the high quality data, which otherwise are discarded. QTrim can be easily incorporated in to next generation sequence analysis pipeline for quality trimming. A web interface has been maintained at <https://hiv.sanbi.ac.za/software/qtrim> for those who have little knowledge on use of bioinformatics tools in command line interface. 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**

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.

Chou, H-H, Holmes, MH (2001) 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.

Drmanac, R (2011) The advent of personal genome sequencing. *Genetics in Medicine* **13**: 188-190.

Falgueras, J, Lara, AJ, Fernández-Pozo, N, Cantón, FR, Pérez-Trabado, G, Claros, MG (2010) SeqTrim: a high-throughput pipeline for pre-processing any type of sequence read. *BMC Bioinformatics* **11**: 38.

Goecks, J, Nekrutenko, A, Taylor, J, Team, TG (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol* **11**: R86.

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

Jerome, M, Noirot, C, Klopp, C (2011) Assessment of replicate bias in 454 pyrosequencing and a multi-purpose read-filtering tool. *BMC Res Notes* **4**: 149.

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.

Ledergerber, C, Dessimoz, C (2011) Base-calling for next-generation sequencing platforms. *Briefings in Bioinformatics* **12**: 489-497.

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

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.