


SOFTWARE

Open Access



# SequelTools: a suite of tools for working with PacBio Sequel raw sequence data

David E. Hufnagel<sup>1,2\*</sup> , Matthew B. Hufford<sup>1</sup> and Arun S. Seetharam<sup>3</sup>

\*Correspondence:

david.hufnagel@usda.gov

<sup>2</sup>Virus and Prion Research Unit, National Animal Disease Center, USDA-ARS, Ames, IA 50010, USA

Full list of author information is available at the end of the article

## Abstract

**Background:** PacBio sequencing is an incredibly valuable third-generation DNA sequencing method due to very long read lengths, ability to detect methylated bases, and its real-time sequencing methodology. Yet, hitherto no tool was available for analyzing the quality of, subsampling, and filtering PacBio data.

**Results:** Here we present *SequelTools*, a command-line program containing three tools: Quality Control, Read Subsampling, and Read Filtering. The Quality Control tool quickly processes PacBio Sequel raw sequence data from multiple SMRTcells producing multiple statistics and publication-quality plots describing the quality of the data including N50, read length and count statistics, PSR, and ZOR. The Read Subsampling tool allows the user to subsample reads by one or more of the following criteria: longest subreads per CLR or random CLR selection. The Read Filtering tool provides options for normalizing data by filtering out certain low-quality scraps reads and/or by minimum CLR length. *SequelTools* is implemented in bash, R, and Python using only standard libraries and packages and is platform independent.

**Conclusions:** *SequelTools* is a program that provides the only free, fast, and easy-to-use quality control tool, and the only program providing this kind of read subsampling and read filtering for PacBio Sequel raw sequence data, and is available at <https://github.com/ISUgenomics/SequelTools>.

**Keywords:** Genomics, Next-generation sequencing, Third-generation sequencing, PacBio, Sequel

## Background

The third-generation of sequencing is here and making tremendous impact in the field of genomics. The primary contenders in third-generation sequencing are Pacific Biosciences (PacBio) (Sequel, Sequel2) and Oxford Nanopore (MinION, GridION, and PromethION). These new sequencing platforms are undergoing active development and pushing boundaries in terms of total output, read length, sequencing time, cost reduction and read accuracy [1, 2]. Recently introduced PacBio Sequel/Sequel2 platforms, which rely on Single-Molecule Real Time (SMRT) sequencing technology, are one of the most widely used long-read sequencing approaches [2, 3]. In contrast to second-generation methodologies, PacBio provides longer length reads, in much less time, with greatly



reduced-content bias, and an ability to distinguish between methylated and unmethylated bases [1–4]. Accuracy has also substantially improved from previous long-read platforms.

Similar to the previous RSII platform, PacBio Sequel uses the SMRTBell, a double stranded DNA molecule that loops around the ends, as the template for sequencing. The polymerase runs through the template continuously, sequencing the DNA by adding nucleotides in both the forward and reverse orientation. The contiguous sequence generated by the polymerase during sequencing is referred to as a “polymerase read” or a Continuous Long Read (CLR). This CLR read may include sequence from adapters and multiple copies of inserts, because it traverses the circular template many times. The CLRs are processed to remove adapter sequences and to retain only the insert sequence, called “subreads”. All other sequences sequenced from the CLR are called “scraps”. Multiple copies of subreads generated from the single SMRTBell can then be collapsed to a single, high-quality sequence, called the “read of insert” or Circular Consensus Sequence (CCS) [3, 5].

Sequencing is performed within a SMRTcell which contains tens of thousands of zero-mode waveguides (ZMWs). These ZMWs contain a light-detection module and an immobilized polymerase enzyme. The template (SMRTBell) is introduced into the ZMW and nucleotide bases labeled with different fluorophores are sequentially added. Each base incorporation will result in the release of a fluorophore, producing distinct light wavelengths per base. All ZMWs within a SMRTcell are processed in parallel, sequencing thousands of templates at the same time. Thus the number of productive ZMWs (ZMWs that received exactly one template) will directly indicate the productivity of the SMRTcell. The length distribution for the polymerase reads/subreads also provides a useful metric of run quality [5].

Here we present *SequelTools*, an efficient and user-friendly program with multiple tools including a QC tool that calculates multiple standardized statistics and creates publication-quality plots describing the quality of raw PacBio Sequel data, a Read Subsampling tool that allows the user to subsample their data by either longest subreads per CLR and/or random CLRs, and a Read Filtering tool that filters the user’s data by one or more chosen criteria.

When working with sequence data it is important to be aware of sequence quality before using the data for downstream analysis, otherwise poor quality reads could lead to spurious results. Programs or web applications that accomplish this task are often referred to as quality control (QC) tools. One popular QC tool for short-read sequence is *fastQC* [6]. *fastQC* works well for short reads, but is not appropriate for long reads found in third-generation sequences and does not provide all the metrics for quality control. While a quality control tool for Oxford Nanopore Technologies’ MinION sequence is available [7], there is no such tool available for the newest sequencing technology from PacBio, PacBio Sequel. Due to improvements in data formats and the technology itself, previous base quality programs for PacBio RSII [8, 9] data are no longer valid for assessing the quality of PacBio Sequel data. Currently, the only program that provides quality assessment for PacBio Sequel raw sequence data is the instrumentation software itself, SMRT Link, a linux-only, computationally intensive webtool where the user must upload their data files one at a time. Furthermore, SMRT Link can only be installed

by root users, requiring the installation of 23 external programs to run, and generates non-downloadable plots after setting up a web server [10]. In fact, the difficulty of using SMRT Link for high-throughput data was the motivation for writing *SequelTools*. Even for users who have the specialized skills required to install and run SMRT Link, running the QC tool via *SequelTools* would be much faster and simpler, free, platform independent, and would produce publication-quality plots. Hitherto, there are no freely available tools for assessing the quality of raw PacBio sequel sequence data. The development of a fast, free, and easy-to-install and use program to assess raw sequence quality is therefore crucial for any person making use of PacBio Sequel sequence data.

In addition to the QC tool, a Read Filtering tool has been implemented in *SequelTools*. When PacBio sequence is generated, the DNA template is often sequenced many times resulting in multiple subreads per CLR. One effective approach for handling these multiple subreads is to generate a consensus sequence [11]. PacBio has developed a tool for this purpose [12], which combines multiple subreads from the same ZMW using a statistical model to produce one highly accurate consensus sequence. However, this is a computationally intensive process and requires multiple passes for error correction to work reliably [12]. Considering the high accuracy of the latest PacBio Sequel chemistry [3], the improvement in accuracy due to generating a consensus sequence is small compared to using the Read Subsampling tool. As the accuracy of raw PacBio sequence improves over time and the error rate between raw reads and consensus sequence decreases the runtime advantage of using the Read Subsampling tool relative to generating a consensus sequence becomes more significant. When appropriate, subsampling longest subreads per CLR using *SequelTools* will reduce redundancy and data size. Random CLR subsampling, another function of *SequelTools*' Read Subsampling tool, will further reduce data size to any size desired. This could be useful for bootstrapping or generating a test data set for the purpose of developing or testing tools or computational pipelines. This could also be useful for situations where using large data sets is not feasible, like in phylogenetics. Random CLR subsampling using the read subsampling tool from *SequelTools* allows the user to generate a subsample of their raw sequence data in an unbiased way that retains all information associated with one CLR across subreads and, potentially, scraps files.

*SequelTools* also provides a read filtering tool, which would be very useful for those wishing to normalize their data. Read filtering can be done by minimum CLR length, having at least one complete pass of the DNA molecule through the polymerase, and/or having normal scraps adapters. We believe that the large majority of users that are using scraps reads will find value from filtering by the number of complete passes and normal scraps adapters. We believe that the Read Filtering tool will be valuable to the majority of people working with PacBio Sequel data who are working with scraps files due to the need to normalize data before performing downstream analysis.

*SequelTools* provides essential functions for analyzing and processing Pacbio Sequel raw sequence data, and is the only program to date of its kind. There is no publicly available tool for assessing the quality of this kind of data. There is also no known tool that allows users to subsample reads to reduce data size for testing or filter reads so as to normalize the data the way that *SequelTools* does. Despite the dearth of helpful tools, PacBio is a popular sequencing platform. We believe *SequelTools* will therefore further

facilitate the use of PacBio sequence data for researchers in a variety of disciplines who may not otherwise have had the technical expertise to utilize these valuable data in the absence of a tool for processing it.

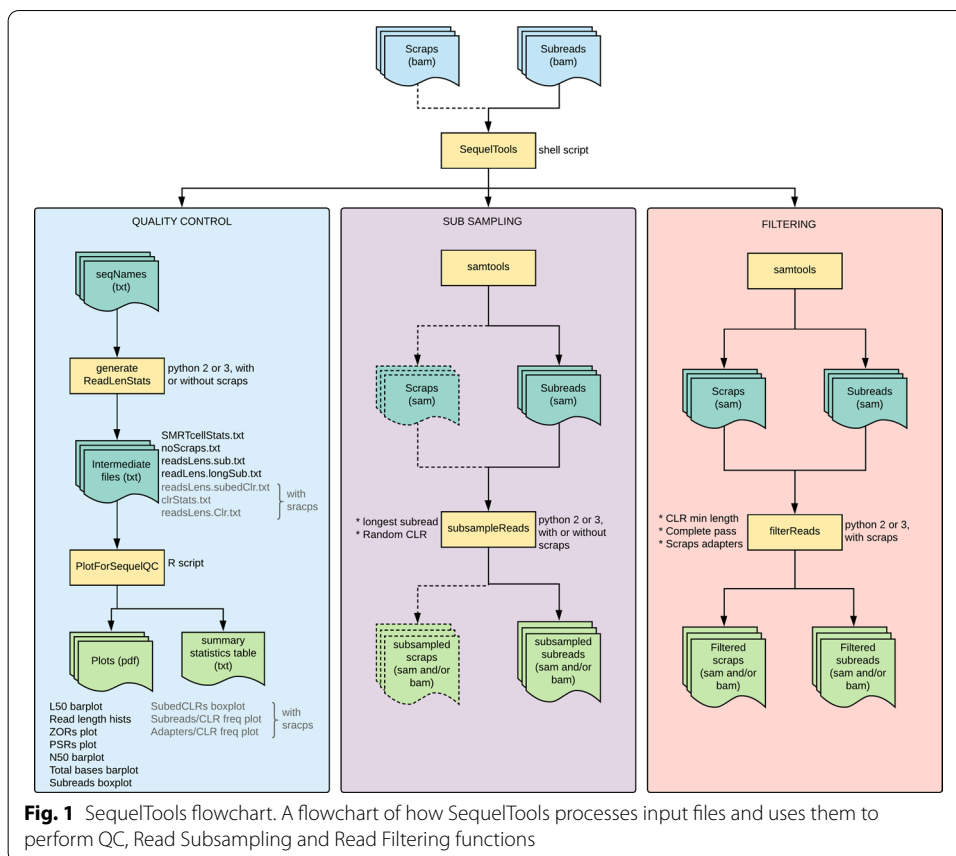
### Implementation

*SequelTools* uses only standard libraries and packages within bash, R and Python in order to facilitate quality assessment, data filtering and normalization of raw PacBio Sequel data. While it can be run on a single SMRTcell, *SequelTools* is designed to run across multiple SMRTcells simultaneously. The main script is written in bash which calls Samtools for converting between BAM (Binary Alignment/Map) and SAM (Sequence Alignment/Map) format, Python for calculations, and R for plotting. Python 2 or 3 can be used, and the version is determined automatically by the program. *SequelTools* is fast (with the exception of subsampling for longest subreads), easy to use, and works on any operating system from the command-line. *SequelTools*, in its current form, is composed of three tools, which can be used one at a time using regular command-line arguments. The three tools implemented in this program are 1) Quality Control, 2) Read Subsampling, and 3) Read Filtering. *SequelTools* uses BAM format files as input because raw PacBio Sequel sequence files come in BAM format. PacBio sequence files include both subreads files containing reads of interest and scraps files with additional reads generated during the sequencing process like barcodes and adapters. For all tools subreads files are required, and for some scraps files are also required.

### The quality control tool

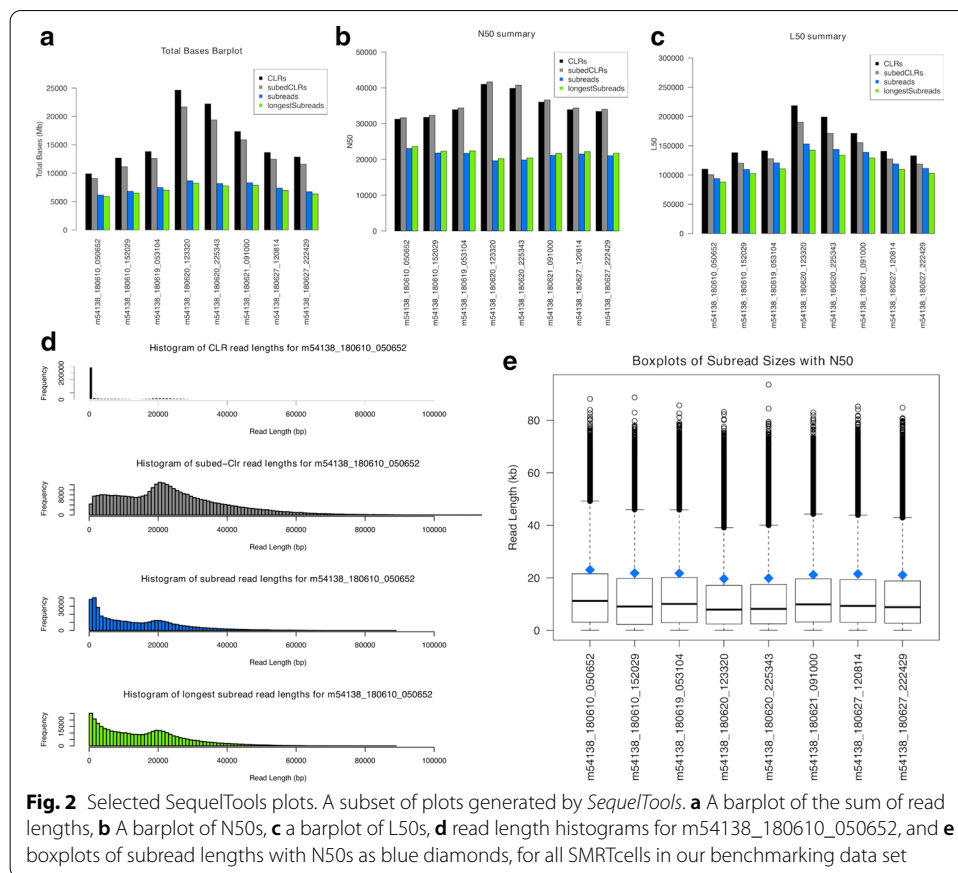
One of *SequelTools*' tools is the QC tool. This tool creates tables and plots summarizing the quality of PacBio Sequel data. The QC tool does not require scraps files. With scraps files, the QC tool takes longer to run, but also produces additional plots and provides more information in standard plots using additional information concerning CLR. The QC tool calls Samtools [13] and awk to convert BAM files to SAM format and extracts only needed information (Fig. 1). Then Python is used to make all necessary calculations, producing intermediate data files that are passed to R. Note that when including scraps files only normal scraps reads are used for downstream QC analyses. By default these intermediate data files will be deleted at the end of the program's operation, but they will be retained if the user selects the appropriate arguments. At this point, reads are organized into up to four read groups: (1) subreads, (2) longest subreads, (3) CLR, and (4) subCLR (CLR containing subreads). When scraps are included, the default is to use all four read groups, but the user can request only two groups if preferred: (1) subreads and (2) subCLR. Alternatively, if scraps are excluded the two read groups are (1) subreads and (2) longest subreads.

Final plots and tables are produced in R, including a tab-delimited table of summary statistics (Additional file 1: Tables S1–S7), which can be viewed easily in Microsoft Excel, as well as several publication-quality PDF (Portable Document Format) plots. A subset of these plots can be seen in Fig. 2; all plots can be seen in Additional file 1: Figure S1. The summary statistics table includes information for all chosen read groups for each SMRTcell. Statistics include number of reads, total bases, mean and median read length, N50, L50, PSR, and ZOR. PSR is the polymerase-to-subread ratio and is calculated as



follows: total bases from the longest subreads per CLR divided by the total bases from subreads. This is a measure of the effectiveness of library preparation. When PSR is close to one the DNA template is mostly the reads of interest, whereas a total failure of library preparation would result in no reads of interest and a PSR of zero. ZOR is the ZMW-occupancy-ratio and is calculated as the number of CLRs with subreads divided by the number of subreads. This is a measure of the effectiveness of matching DNA templates with ZMW's. When ZOR is zero there are no DNA templates in ZMWs. When ZOR is above one then there are more than one DNA template per ZMW on average; ideally, this value is exactly one.

While the summary statistics table (Additional file 1: Tables S1–S7) is always produced, the user can request more or fewer plots based on their needs. The full suite of plots with scraps files includes barplots of (A) N50s, (B) L50s, (C) total bases, and (D) read length; frequency plots of (E) subreads per subedCLR, and (F) adapters per CLR; boxplots of (G) subread lengths and (H) subedCLR lengths; and (I) ZOR and (J) PSR plots (Additional file 1: Figure S1). The user can also request an intermediate (A, C, G, H, I, and J) or basic (A and C) suite of plots. Without scraps files, the full suite of plots is A, B, C, D, G, I, and J, the intermediate collection is A, C, G, I, and J, and the basic set is A and C (Fig. 1). With or without scraps the intermediate selection of plots is default. The summary statistics table as well as all plots, except the read length histograms and frequency plots, present the data from all SMRTcells together. For read



length histograms and frequency plots, separate files are generated for each SMRTcell with either one plot per group for histograms or one plot total for frequency plots.

Some users may want to modify the provided R script to change plot format or to make entirely new plots. In the case that a user wishes to use a custom R script for plotting, we recommend the user run the QC tool once using the ‘-k’ argument to generate the intermediate data files and retain them at the end of the QC tool’s operation. Next, the user will need to create their custom R script. If the user wishes to modify *SequelTools*’s QC plots we recommend the user start by copying and renaming either ‘plotForSequelTools\_wScraps.R’ or ‘plotForSequelTools\_noScraps.R’ depending on whether the user is running the QC tool with or without scraps files, respectively. To aid in the process of testing the custom R plotting script we have added an argument which will skip the read length calculations with Samtools and the statistical calculations with Python which, together, generate the intermediate data files. Together these steps make up most of the runtime of the QC tool, therefore skipping these steps allows for rapid testing of an alternative plotting script. Whether the user is modifying a *SequelTools* R plotting script or using one created from scratch the user will need to provide the custom plotting script to *SequelTools*. When testing the custom R script the ‘-k’ argument will remain necessary, otherwise the intermediate files will all be deleted at the end of the QC tool’s operation.

### The read subsampling tool

Another of *SequelTools*' tools is the Read Subsampling tool. This tool allows the user to subsample their BAM format sequence files by longest subreads per CLR or random CLR selection. Subsampling by the longest subreads per CLR simply creates a new sequence file for each SMRTcell containing only the longest subread for each CLR. Similarly, subsampling by random CLR creates a new sequence file for each SMRTcell containing only reads associated with randomly selected CLRs. The Read Subsampling tool does not require scraps files, but can take advantage of them if provided by the user when subsampling for randomly selected CLRs. When subsampling by longest subreads alone if scraps files are provided, which is not recommended, a new subsampled scraps file will be generated identical to the original scraps file. As with the QC tool, using scraps files will increase *SequelTools*' runtime, mostly due to the need to convert additional BAM files to SAM format.

The Read Subsampling tool first converts BAM sequence files to SAM format using Samtools (Fig. 1). Next the SAM files are processed using Python. Python parses through the SAM files and saves necessary information. When subsampling by longest subreads coordinate information associated with CLR IDs are saved by Python, and when subsampling by random CLRs all CLR IDs are saved. Subsampling CLR IDs and/or read IDs based on chosen criteria comes next, followed by parsing input SAM files again and outputting all reads associated with subsampled longest subreads and/or randomly subsampled CLRs in the SAM format. Finally, upon request the Read Subsampling tool will convert the output files to BAM format using Samtools.

### The read filtering tool

An additional tool available via *SequelTools* is the Read Filtering tool. This tool provides the user a filtering functionality and requires both scraps files and subreads files. Filtering can be done using one or more of the following criteria: (1) minimum CLR length, (2) having at least one complete pass of the DNA molecule through the polymerase, or (3) normal adapters for scraps. The latter two criteria apply only to scraps files, and are recommended for most downstream analyses involving scraps files. Filtering by minimum CLR length takes input BAM format subreads and scraps files and yields files with only CLRs having a total length, including all provided scraps, greater or equal to the threshold value provided by the user. If the user wishes for CLR length to be calculated for the purpose of applying a minimum length threshold to include scraps reads, but to exclude these less desirable scraps reads, they must run the filters for number of complete passes of the DNA molecule and normal adapter scraps reads first and then run the filter for minimum CLR length using BAM files generated by the Read Filtering tool in the first run. Filtering by complete passes of the DNA molecule takes input BAM format subreads and scraps files and creates an output containing only scraps reads with at least one full pass of the DNA molecule through the polymerase. Filtering by normal scraps adapters takes BAM format subreads and scraps files as input and creates an output with only normal scraps adapters, defined as having a ZMW classification annotation of 'N' for 'normal', as opposed to "control", "malformed", or "sentinel", and a scrap region-type annotation of 'A' for 'adapter', as opposed to "barcode" or "LQRegion" [14].

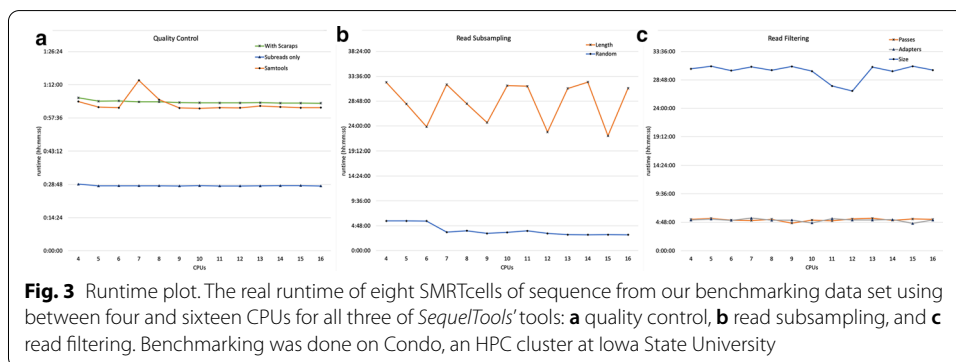
The Read Filtering tool starts by using Samtools to convert BAM sequence files to SAM format (Fig. 1). Then the SAM files are processed with a Python script. Python first extracts coordinates from scraps and subreads files. Next, if filtering by CLR length, CLRs are assembled from subreads and scraps coordinate data, CLR lengths are calculated, and CLR IDs are stored that do not pass the minimum length threshold. If filtering by CLR length the subreads input file is iterated through again and only reads containing CLRs that pass the minimum length threshold are written to a new output file. If filtering by either the number of passes of the DNA molecule or normal scraps adapters, the scraps file is then iterated through again and only reads containing information indicating they pass all chosen scraps thresholds are written to a new output file. Lastly, upon the user’s request, the Read Filtering tool will convert the output files to BAM format using Samtools.

### Results/discussion

#### Benchmarking

Each tool within *SequelTools*’ was tested and benchmarked on Condo, a High-Performance Computing Cluster, at Iowa State University, running the Red Hat Linux operating system. Varying number of CPU’s (Central Processing Units) (4 to 16, with increments of 1) and 8 SMRTCells from the PacBio reads of the NC358 maize genome (Bioproject ID PRJEB32404) [15] was used for benchmarking. For the QC tool, both with and without scraps modes were benchmarked with fixed memory. All default QC tool options were used. The UNIX time command was used to collect the ‘real’ usage time for each run. On average it took less than 30 min to run with only scraps and little over an hour with both scraps and subreads (Fig. 3). The runtime of the quality control tool is therefore tightly correlated with total input file size. However, using a greater number of cores did not affect the runtime with or without scraps files. This is probably due to using the same amount of memory for each run or the processing time is disk read/write bound. We also noticed similar behavior for Samtools, which is the most time intensive component of the QC tool (Fig. 3).

For the Read Subsampling tool, both random subsampling and subsampling for longest subreads was benchmarked with 4 to 16 processors with the same sized input data (8 SMRTcells). Subsampling by random CLRs took less than 6 h with few processors but the runtime decreased steadily with more processors (~ 3 h with 16 processors) (Fig. 3).





However, for subsampling for longest subreads, we did not find any improvements in runtime with additional CPUs. The bulk of the processing for subsampling for longest subreads is done serially via a Python script and additional CPUs will not help speed up this process. The 8 SMRTcell data, amounting to more than 130Gb and containing 4 million reads, takes about 32 h for subsampling for longest subreads regardless of number of CPUs used (Fig. 3).

For the Read Filtering tool, all three sub-tools were tested with similar benchmarking runs. Filtering by both normal scraps adapters and by number of passes takes about 5 h to complete (Fig. 3). Again, since the bulk of the time is used for read filtering serially within a Python script, the number of CPUs does not improve runtime for this tool. For length-based filtering, the minimum CLR length threshold of 1Kb was used and read filtering was performed on all 8 SMRTcells. This process took about 30 h to complete (Fig. 3). All three of *SequelTools'* tools were run with both subreads and scraps files. The total size of this benchmarking data set is 277Gb.

In order to test the efficacy of the read filtering tool for improving downstream analysis we assembled a genome de novo before and after read filtering. A sequel II *Arabidopsis thaliana* data set was downloaded from the official PacBio demo data set website [16]. The data set contained two SMRTcells of scraps and subreads data which were filtered using a minimum CLR length threshold of 5000 bases for the test data set. The unfiltered data set was used as a control data set. The assembler Canu [17] was run on both of these data sets after converting data files to fasta format using samtools [13]. Assembly statistics were then computed using Assemblathon 2 [18]. Detailed methods along with the commands used can be found on the *SequelTools* GitHub repository. Even using an assembler with an automatic 1000 base minimum read-length cutoff and a genome with very high read quality and depth the Read Filtering Tool improved the amount of bases that were incorporated during de novo genome assembly (Additional file 1: Table S8). With a poorer data set or a different assembler, filtering out short CLRs would likely have an even greater impact.

## Conclusion

*SequelTools* is an easy-to-install-and-use program that provides a variety of utilities for working with Pacbio Sequel raw sequence data including quality control, read subsampling, and read filtering. The QC tool calculates key statistics and generates publication-quality plots, providing all standard metrics for overall sequence quality including N50, read length and count statistics, PSR, and ZOR. *SequelTools'* QC tool can evaluate eight SMRTcells from NC358 maize genome (Bioproject ID PRJEB32404) [15] in about 30 minutes with subreads alone and in about an hour with the addition of scraps reads on our High-Performance Computing Cluster. Other than the proprietary PacBio *SMRTlink* program, which is time intensive, does not produce downloadable plots, and requires the user to set up a web server to install, there is currently no program available to compute these statistics. In addition to the QC tool, *SequelTools* has read subsampling and read filtering functions which allow the user to reduce their data size and to normalize their data, respectively. The author is not aware of any other program that provides these additional functionalities for PacBio Sequel data, except the *bamsieve* tool from SMRT Link [10] which does random CLR

subsampling but does not include scraps reads or longest subread subsampling. We therefore conclude that *SequelTools* is the only reasonable choice for quality control, the best choice for read subsampling, and the only choice for read filtering for users of PacBio Sequel sequencing data. We believe *SequelTools* will therefore contribute to the expansion of the use of PacBio's highly valuable, and already popular, Sequel sequencing system.

## Availability and requirements

*Project name* SequelTools.

*Project home page* <https://github.com/ISUgenomics/SequelTools>.

*Operating systems* Platform independent.

*Programming languages* Bash, Python, and R.

*Other requirements* samtools, awk.

*License* GNU GPL v3.0.

*Any restrictions to use by non-academics* None.

## Supplementary information

**Supplementary information** accompanies this paper at <https://doi.org/10.1186/s12859-020-03751-8>.

Additional file 1.

### Abbreviations

BAM: Binary alignment/map format; CLR: Continuous long read; CPU: Central processing unit; PDF: Portable document format; PSR: Polymerase-to-subread ratio; QC: Quality control; RAM: Random-access memory; SAM: Sequence alignment/map format; SMRT: Single molecule real time sequencing technology; ZMW: Zero-mode waveguide; ZOR: Zero-mode waveguide occupancy ratio.

### Acknowledgements

We would like to thank Maggie Woodhouse for proving comments on this manuscript, as well as Jim Coyle, director of HPC at Iowa State University and Andrew Severin at the Genome Informatics Facility for supporting the development of this tool.

### Authors' contributions

D.E.H. designed, implemented and maintained the code, and coordinated the research activity. D.E.H., M.B.H., and A.S. formed the project goals including suggesting new features, and contributed to data acquisition and visualization. M.B.H. and A.S. provided oversight and mentorship. A.S. performed benchmarking. M.B.H. acquired financial support for the project. D.E.H. and A.S. tested code components, wrote the manuscript, and designed the methodology. M.B.H. edited the manuscript. All authors read and approved the final manuscript.

### Funding

This work was supported by the National Science Foundation Plant Genome Research Program Grant number IOS-1744001 and Principle Investigator Kelly Dawe. This grant was used to support the stipends of D.E.H and A.S as well as to generate the data used for initial development and testing. D.E.H. was funded, during the editing and revision process, by the USDA Agricultural Research Service Research Participation Program of the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the U.S. Department of Energy (DOE) and USDA Agricultural Research Service (contract number DE-AC05-06OR23100). Mention of trade names or commercial products in this article is solely for the purpose of providing specific information and does not imply recommendation or endorsement by the USDA, DOE, or ORISE. USDA is an equal opportunity provider and employer. No funding bodies had any role in study design, data collection, analysis and interpretation, writing, or the decision to submit the work for publication.

### Availability of data and materials

The test data set used for benchmarking is available in the NCBI repository, BioProject PRJEB32404.

### Ethics approval and consent to participate

Not applicable.

### Consent for publication

Not applicable.

### Competing interests

The authors declare that they have no competing interests.

**Author details**

<sup>1</sup> Department of Ecology, Evolution and Organismal Biology, Iowa State University, Ames, IA 50011, USA. <sup>2</sup> Virus and Prion Research Unit, National Animal Disease Center, USDA-ARS, Ames, IA 50010, USA. <sup>3</sup> Genome Informatics Facility, Iowa State University, Ames, IA 50011, USA.

Received: 19 December 2019 Accepted: 11 September 2020

Published online: 01 October 2020

**References**

- Schadt EE, Turner S, Kasarskis A. A window into third-generation sequencing. *Hum Mol Genet.* 2010;19(R2):227–40.
- Goodwin S, McPherson JD, McCombie WR. Coming of age: ten years of next-generation sequencing technologies. *Nat Rev Genet.* 2016;17(6):333–51.
- Rhoads A, Au KF. PacBio sequencing and its applications. *Genom Proteom Bioinform.* 2015;13(5):278–89.
- Ross MG, Russ C, Costello M, Hollinger A, Lennon NJ, Hegarty R, Nusbaum C, Jaffe DB. Characterizing and measuring bias in sequence data. *Genome Biol.* 2013;14(5):51.
- Ardui S, Ameur A, Vermeesch JR, Hestand MS. Single molecule real-time (SMRT) sequencing comes of age: applications and utilities for medical diagnostics. *Nucleic Acids Res.* 2018;46(5):2159–68. <https://doi.org/10.1093/nar/gky066>.
- Andrews S, Krueger F, Segonds-Pichon A, Biggins L, Krueger C, Wingett S. FastQC. Babraham: Babraham Institute; 2012.
- Lanfear R, Schalamun M, Kainer D, Wang W, Schwesinger B. MinIONQC: fast and simple quality control for MinION sequencing data. *Bioinformatics.* 2019;35(3):523–5.
- Desvillechabrol D, Legendre R, Rioualen C, Bouchier C, Van Helden J, Kennedy S, Cokelaer T. Sequanix: a dynamic graphical interface for snakemake workflows. *Bioinformatics.* 2018;34(11):1934–6.
- PacificBiosciences/stsPlots. Plot primary analysis quality control metrics. <https://github.com/PacificBiosciences/stsPlots>. Accessed 8 June 2019
- Software Downloads. Accessed 5 July 2019. <https://www.pacb.com/support/software-downloads/>
- Wenger AM, Peluso P, Rowell WJ, Chang PC, Hall RJ, Concepcion GT, Ebler J, Fungtammasan A, Kolesnikov A, Olson ND, Topfer A, Alonge M, Mahmoud M, Qian Y, Chin CS, Phillippy AM, Schatz MC, Myers G, DePristo MA, Ruan J, Marschall T, Sedlazeck FJ, Zook JM, Li H, Koren S, Carroll A, Rank DR, Hunkapiller MW. Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nat Biotechnol.* 2019;37(10):1155–62.
- Pacific Biosciences of California I. ccs. GitHub. <https://github.com/PacificBiosciences/ccs>. Accessed 7 Oct 2019
- Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. The sequence alignment/map format and SAMtools. *Bioinformatics.* 2009;25(16):2078–9.
- BAM format specification for PacBio. Accessed 12 Oct 2019. <https://pacbiofileformats.readthedocs.io/en/5.1/BAM.html>.
- Ou S, Liu J, Manchanda N, Gilbert AM, Wei X, Chin C-S, Hufnagel DE, Pedersen S, Snodgrass S, Fengler K, et al. Effect of sequence depth and length in long-read assembly of the maize inbred nc358. *Nat Biotechnol.* 2019.
- The PacBio *Arabidopsis Thaliana* Genome. [https://downloads.pacbcloud.com/public/SequelData/ArabidopsisDemoData/SequenceData/1\\_A01\\_customer/](https://downloads.pacbcloud.com/public/SequelData/ArabidopsisDemoData/SequenceData/1_A01_customer/). Accessed 12 July 2020.
- Sergey K, Walenz BP, Berlin K, Miller JR, Bergman NH, Phillippy AM. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res.* 2017;27:722–36.
- Bradnam KR, Fass JN, Alexandrov A, Baranay P, Bechner M, Birol I, Boisvert S, Chapman JA, Chapuis G, Chikhi R, Chitsaz H, Chou W-C, Corbeil J, Del Fabbro C, Docking TR, Durbin R, Earl D, Emrich S, Fedotov P, Fonseca NA, Ganapathy G, Gibbs RA, Gnerre S, Godzaridis Goldstein S, Haimel M, Hall G, Haussler D, Hiatt JB, Ho IY, Howard J, Hunt M, Jackman SD, Jaffe DB, Jarvis ED, Jiang H, Kazakov S, Kersey PJ, Kitzman JO, Knight JR, Koren S, Lam T-W, Lavenier D, Laviolette F, Li Y, Li Z, Liu B, Liu Y, Luo R, MacCallum I, MacManes MD, Maillet N, Melnikov S, Naquin D, Ning Z, Otto TD, Paten B, Paulo OS, Phillippy AM, Pina-Martins F, Place M, Przybylski D, Qin X, Qu C, Ribeiro FJ, Richards S, Rokhsar DS, Ruby JG, Scalabrin S, Schatz MC, Schwartz DC, Sergushichev A, Sharpe T, Shaw TI, Shendure J, Shi Y, Simpson JT, Song H, Tsarev F, Vezzi F, Vicedomini R, Vieira BM, Wang J, Worley KC, Yin S, Yiu S-M, Yuan J, Zhang G, Zhang H, Zhou S, Korf IF. Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *GigaScience.* 2013. <https://doi.org/10.1186/2047-217X-2-10>.

**Publisher's Note**

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.