SeqyClean User Manual

Ilya Y. Zhbannikov[1,3], Samuel S. Hunter[1,3], Matthew L. Settles[1,3]

[1]Institute for Bioinformatics and Evolutionary Studies, University of Idaho, Moscow, ID 83844-3051,
[2]Department of Bioinformatics and Computational Biology, University of Idaho, Moscow, ID 83844-3051,

Keywords:
Bioinfomatics, Genomics

Correspondence:
Ilya Y. Zhbannikov
Department of Bioinformatics and Computational Biology
University of Idaho
Moscow, ID 83844-30521
Email: zhba3458@vandals.uidaho.edu

Running Title:
SeqyClean User Manual

1 Abstract

Raw data from sequencing machines is not usually completely prepared for analysis since sequences often contain remnants of nucleotides that have been used during DNA library preparation. Such nucleotides can potentially case hurdles during data analysis and hereby need to be removed. In additional, the noise, which is presented as low-quality bases, has serious impact on genome assembly and mapping. We propose SeqyClean, a specialized cleaning pipeline that alleviates these issues.

2 Introduction

With newly developed next-generation sequencing technologies and increasing interest in gene discovery, DNA mapping, functional genomics and genome annotations, the amount of sequencing has exponential growth curve and doubles every month. Sequence data from automatic sequencing machines should not be considered as "ready-to-use" data for analysis due to contaminant remnants and adapters used during DNA sequencing. Poly A/T tails and low quality of sequenced data due to base-call errors make the library even worse. One can say that cleaning next-generations sequence data is not worthwhile because of large number of reads within a DNA library (1 million for Roche 454 and 10 millions for Illumina on average). However, we would argue with this and in order to support our claim we present several examples that show the importance of sequence cleaning. As an example, adapters left within a read along with base-call errors can potentially lead to unexpected results after genome assembly having large amount of small contigs, which indicates that genome was not assembled properly. Another example shows how important to trim pieces of adapters within a library of interest. Illumina technology employs specific adapters named TruSeq in order to immobilize DNA pieces on the surface. This situation is described in Figure XX. In this figure, two large contigs could not join together because of abundance of adapters at the end

of reads (marked yellow). The assembler stopped building the left contig and begun another one. Another example shows impact of contaminant screening on genome assembly when results lead to the wrong conclusion as the evidence of contaminants within a library of interest. However, after performing a BLAST search in order to find similar genomic sequences, it turned out that in addition to anolis data, the published genome contains genomic sequences of PhiX virus, which cannot be presented in anolis genome. Apparently, there was no contaminant screening provided before genome assembly. That is why it is very important to filter contaminants that may potentially be in a library. An additional cleaning step is highly desirable. This step is intended to filter adapters, contaminants, vector particles along with trimming of low-quality regions in order to prepare sequences for further analysis.

2.1 Sequence cleaning software

2.1.1 Individual cleaning aspects

We divide the cleaning tasks into the following categories:

- Adapter trimming

  Adapter trimming should be one of the major cleaning task. Adapters are short pieces of nucleotides used during sequencing. Different sequencing technologies employ different types of adapters. Cleaning software must be flexible enough to detect such artifacts within reads.

- Quality trimming

  Quality trimming is important part of cleaning pipeline intended to trim low-quality bases caused by base-call errors.

- Poly A/T tails trimming

  Trimming of poly A and poly T tails is used mostly for Roche 454 technology.

- Contaminants screening

  Removes those reads that can be contaminants.

- Vector trimming

  Removes vector pieces within reads. Necessary step if bacteria vector was used during the dna library preparation.

2.1.2 Existing cleaning software

Today's sequence cleaning software are mostly tools that perform individual cleaning tasks (e.g. TrimEST (removes poly A/T tails from sequence), VectorStrip and VecScreen (both trim bacteria vector particles), ESTPrep(Scheetz et al., 2003), Figaro(White, Roberts, Yorke, & Pop, 2008), Lucy(Chou & Holmes, 2001)), and AlienTrimmer (Criscuolo & Brisse, 2013) there are a few programs that offer cleaning pipelines. Also most of these tools perform cleaning of Sanger sequences, which is not NGS. A few tools perform several cleaning aspects such as adapters removal, contaminants/vector screening, poly-A/T and quality trimming, in single cleaning pipeline for next-generation sequencing. For these reasons there is a need for a next-generation sequence cleaning tool that combines all cleaning tasks into one single pipeline. We present SeqyClean – a bioinformatics software tool for next-generation sequence cleaning. The first purpose of SeqyClean is to incorporate various cleaning methods, such as adapter, contaminant, poly A/T and quality trimming into a single software pipeline; the second is an ability to clean second-generation sequencing data such as Roche 454 and Illumina sequences. SeqyClean can recognize and remove both technological components (adapters, primers, mid tags) along with contaminants and vector. The LUCY© quality trimming algorithm was incorporated to SeqyClean in order to remove low-quality and poly-A/T erroneous data. Below we conducted a survey which describes several sequence cleaning applications (Table 1). None of these application, except SeqyClean, offers

the full cleaning pipeline and the all aspects for complete next-generation sequence cleaning. In addition, SeqyClean has also advanced features, such as overlap and duplicates removal, which is helpful for genome assembly and mapping.

Table 1: Comparison of existing cleaning tools

Application	SFF	FastQ	Contaminants	Vector	Adapter	Quality	Poly A/T	Overlap	Duplicates
SeqyClean	X	X	X	X	X	X	X	X	X
AlienTrimmer		X	X	X	X*				
Lucy				X		X	X		
SeqClean			X			X			
TrimEST							X		
VectorStrip				X					
VectorScreen				X					
Figaro				X					

*It considers adapter as a contaminant

SeqyClean has been intensively used in our lab with constant improving and we believe that research community will benefit from using it.

3 Getting SeqyClean

Follow the link to download: https://bitbucket.org/izhbannikov/seqyclean/get/stable.zip. Save the file under some name you wish.

Then compile:

```
cd path_to_SeqyClean_directory
make clean
make
```

4 Usage

4.1 Roche 454

```
./seqyclean [options] -454 input_filename -o output_prefix
```

4.1.1 Main arguments

`-454 input_filename`	The filename of library to be cleaned. Can be in SFF or FASTQ formats. "-454" tells the program to clean Roche 454 reads
`-o output_prefix`	The files produced will start with the output_prefix followed by some formatted ending (see section "Output files: naming convention")

4.1.2 Roche 454 options

Options are shown in Table 2

4.2 Illumina paired- and single-end sequences

4.2.1 Paired-end sequences

```
./seqyclean [options] -1 input_filename_R1 -2 input_filename_R2 -o output_prefix
```

The descriptions of main arguments for paired-end libraries are shown in Table 3.

4.2.2 Single-end sequences

```
./seqyclean [options] -U input_filename -o output_prefix
```

The descriptions of main arguments for single-end libraries are shown in Table 4.

The descriptions of `[options]` for Illumina libraries are shown in Table 5.

Table 2: Roche 454 options

-v vector_file	This option is used for vector trimming. If you choose this option, the program assumes the reference genome provided in vector_file. This file must be given in FASTA format. Note: vector reference genome(s) must be provided! Example: ./seqyclean -v vectors.fa -454 in.sff -o Test
-c file_of_contaminants	This option is used for contaminants screening. If you choose this option, the program assumes the reference genome provided in file_of_contaminants. This file must be given in FASTA format. When SeqyClean recognizes contaminants in the sequence, the whole sequence gets discarded. Note: contaminant reference genome(s) must be provided! Example: ./seqyclean -c contaminants.fa -454 in.sff -o Test
-m file_of_RL_MIDS	This option works in 454 mode only. Use this option to provide your own RLMIDs. SeqyClean will use them and will not use those provided by default. Example: ./seqyclean -m file_of_custom_RL_MIDS -454 in.sff -o Test
-k k_mer_size	Use this option in order to specify a size of k-mer. Default size is 15 bases. Example: ./seqyclean -k 18 -454 in.sff -o Test
-kc k_mer_size	Special k-mer size for contaminant screening. Use this option only if you want to have different k-mer sizes for contaminant dictionary. Sometimes this option is useful because it prevents false detection of contaminants when program discards too many reads. Example: ./seqyclean -kc 25 -454 in.sff -o Test
-f overlap	For Roche 454 only. This option is intended to impose an overlap between two consecutive kmers. By default it is set to 1 bp. Refer to Fig. 1 Example: ./seqyclean -f 10 -454 in.sff -o Test
-t number_of_threads	Specifies a number of threads in order to take advantage from using a multicore system. Example: ./seqyclean -t 16 -454 in.sff -o Test
-qual max_avg_error max_error_at_ends	LUCY parameters for quality trimming. if "-qual" is set that means you have to provide max_avg_error and max_error_at_ends. Otherwise default values [20 20] will be used. Examples: ./seqyclean -qual -454 in.sff -o Test ./seqyclean -qual 25 30 -454 in.sff -o Test
--qual_only	Use --qual_only parameter if you want to do only quality trimming. Example: ./seqyclean --qual_only -qual -454 in.sff -o Test
--fastq	If input is given in SFF format, by default the output will be also in SFF format. Use this option if you want to have FASTQ format on the output in addition to SFF. Example: ./seqyclean --fastq -454 in.sff -o Test
--keep_fastq	Use this option only if you want to keep original FASTQ file from your input SFF Example: ./seqyclean --keep_fastq -454 in.sff -o Test
-minimum_read_length value	Use this option in order to define the minimum number of base pairs when read is still considered as acceptable. If after the cleaning process the read has a length which is less than minimum_read_length parameter, such the read will be discarded. By default, the minimum_read_length is set to 50 base pairs. Example: ./seqyclean -minimum_read_length 100 -454 in.sff -o Test
-polyat [cdna] [cerr] [crng]	This option provides trimming of poly A/T tails from nucleotide sequences. cdna – tail length (10 by default); cerr – maximum number of errors per tail (3 by default); crng – range to search poly A/T tails (50 by default) Examples: ./seqyclean -polyat -454 in.sff -o Test ./seqyclean -polyat 12 5 67 poly_test.fastq.gz -o Test_polyAT

Table 3: Paired-end main arguments

`-1 input_filename_R1`
 	
`-2 input_filename_R2`
	The filenames of the library to be cleaned. Must be in FASTQ formats only (the program also accepts zipped (.gz) FASTQ files).

`-o output_prefix`	The files produced will start with the output_prefix followed by some formatted ending (see section "Output files: naming convention")

Table 4: Single-end main arguments

`-U input_filename`	The filenames of the library to be cleaned. Can be in FASTQ formats only (the program also accepts .gz files).
`-o output_prefix`	The files produced will start with the output_prefix followed by some formatted ending (see section "Output files: naming convention")

4.3 Help

For help please use: `seqyclean -?  or --help`

4.4 Quick examples of usage

Example for 454 reads: `./seqyclean -v test_data/vectors.fasta -qual 30 25 -454 in.sff -o cleaned_data/Small454Test_cleaned`

Example for Illumina reads: `./seqyclean -v test_data/vectors.fasta -c test_data/contaminants.fasta -qual -1 test_data/R1.fastq.gz -2 test_data/R2.fastq.gz -o cleaned_data/cleaned`

5 Output files: naming conventions

Depending on the given parameters and the cleaning strategy, the name of output file can be different and has the formats described below.

Table 5: Illumina options

-v vector_file	This option is used for vector trimming. If you choose this option, the program assumes the reference genome provided in vector_file . This file must be given in FASTA format. Example: ./seqyclean -v vectors.fa -1 R1.fastq.gz -2 R2.fastq.gz -o Test
-c file_of_contaminants	This option is used for contaminants screening. If you choose this option, the program assumes the reference genome provided in file_of_contaminants . This file must be given in FASTA format. When SeqyClean recognizes contaminants in the sequence, the whole sequence gets discarded. Example: ./seqyclean -c contaminants.fa -1 R1.fastq.gz -2 R2.fastq.gz -o Test
-k k_mer_size	Use this option in order to specify a size of k-mer. Default size is 15 bases. In Illumina mode this option defines a size of kmer that will be used as a dictionary word size. Example: ./seqyclean -k 14 -1 R1.fastq.gz -2 R2.fastq.gz -o Test
-kc k_mer_size	Special k-mer size for contaminant screening. Use this option only if you want to have different k-mer sizes for contaminant dictionary. Sometimes this option is useful because it prevents false detection of contaminants when program discards too many reads. Example: ./seqyclean -kc 31 -1 R1.fastq.gz -2 R2.fastq.gz -o Test
-qual max_avg_error max_error_at_ends	LUCY parameters for quality trimming. if "-qual" is set that means you have to provide max_avg_error and max_error_at_ends. Otherwise default values [20 20] will be used. Examples: ./seqyclean -qual -1 R1.fastq.gz -2 -1 R2.fastq.gz -o Test ./seqyclean -qual 30 25 -1 R1.fastq.gz -2 R2.fastq.gz -o Test
--qual_only	Use --qual_only parameter if you want to do only quality trimming. Example: ./seqyclean --qual_only -qual -1 R2.fastq.gz -2 R2.fastq.gz -o Test
-minimum_read_length value	Use this option in order to define the minimum number of base pairs when read is still considered as acceptable. If after cleaning process the read has length which is less than minimum_read_length parameter such read will be discarded. By default, the minimum_read_length is set to 50 base pairs. Note: in this case no adapter/vector/contaminants cleaning is performed. Example: ./seqyclean -minimum_read_length 100 -1 R1.fastq.gz -2 R2.fastq.gz -o Test
-polyat [cdna] [cerr] [crng]	This option provides trimming of poly A/T tails from nucleotide sequences. Parameters: cdna – tail length (10 by default) cerr – maximum number of errors per tail (3 by default) crng – range to search poly A/T tails (50 by default) Examples: ./seqyclean -polyat -454 in.sff -o Test ./seqyclean -polyat 15 4 55 poly_test.fastq.gz -o Test_polyAT
-overlap <value>	This option provides overlap of read 1 and read 2. The consensus single-end sequence is stored under the name ¡output_prefix¿_SEOLP.fastq. Parameter: <value¿ – the minumum length of overlap in bases. By default it is set to 10 bases.
-ot value	This option sets the similarity threshold for adaprer trimming by overlap. By default its value is set to 0.9.
-adapter_length value	This option sets the maximum adapter length for adaprer trimming by overlap. By default its value is set to 40 bases.
--ow	This option allows SeqyClean overwrite output files.

5.1 Roche 454

After processing Roche 454 reads, SeqyClean outputs a cleaned file by default in Standard Flowgam Format (SFF) and (if option `--fastq` was chosen) in FASTQ format. Also two report files: Prefix_SummaryStatistics.txt (which contains information about how many reads were processed, trimmed, discarded and some other information) and Prefix_Report.csv file which holds the detailed statistics for every read.

- Output_prefix.sff , .fastq (optionally)
- Output_prefix_Report.tsv
- Prefix_SummaryStatistics.txt
- Prefix_SummaryStatistics.tsv

5.2 Illumina

After processing Illumina reads, SeqyClean generates two (shuffled file and file with single-end reads) or three (PE1 and PE2 files that contain paired-end reads and one file with single-end reads) output files in FASTQ format.

- Output_prefix_PE1.fastq
- Output_prefix_PE2.fastq
- Output_prefix_shuffled.fastq
- Output_prefix_SE.fastq
- Output_prefix_PE1_Report.tsv

- Output_prefix_PE2_Report.tsv
- Prefix_SummaryStatistics.txt
- Prefix_SummaryStatistics.tsv

6 Supported RLMIDs

The set of supported Roche 454 RL MIDs is shown in Table 6.

Table 6: Supported RLMIDs by default

#	Left MID	Right MID	#	Left MID	Right MID
RL1	ACACGACGACT	AGTCGTGGTGT	RL19	ATAGTATACGT	ACGTATAGTAT
RL2	ACACGTAGTAT	ATACTAGGTGT	RL20	CAGTACGTACT	AGTACGTGCTG
RL3	ACACTACTCGT	ACGAGTGGTGT	RL21	CGACGACGCGT	ACGCGTGGTCG
RL4	ACGACACGTAT	ATACGTGGCGT	RL22	CGACGAGTACT	AGTACTGGTCG
RL5	ACGAGTAGACT	AGTCTACGCGT	RL23	CGATACTACGT	ACGTAGTGTCG
RL6	ACGCGTCTAGT	ACTAGAGGCGT	RL24	CGTACGTCGAT	ATCGACGGACG
RL7	ACGTACACACT	AGTGTGTGCGT	RL25	CTACTCGTAGT	ACTACGGGTAG
RL8	ACGTACTGTGT	ACACAGTGCGT	RL26	GTACAGTACGT	ACGTACGGTAC
RL9	ACGTAGATCGT	ACGATCTGCGT	RL27	GTCGTACGTAT	ATACGTAGGAC
RL10	ACTACGTCTCT	AGAGACGGAGT	RL28	GTGTACGACGT	ACGTCGTGCAC
RL11	ACTATACGAGT	ACTCGTAGAGT	RL29	ACACAGTGAGT	ACTCACGGTGT
RL12	ACTCGCGTCGT	ACGACGGGAGT	RL30	ACACTCATACT	AGTATGGGTGT
RL13	AGACTCGACGT	ACGTCGGGTCT	RL31	ACAGACAGCGT	ACGCTGTGTGT
RL14	AGTACGAGAGT	ACTCTCGGACT	RL32	ACAGACTATAT	ATATAGTGTGT
RL15	AGTACTACTAT	ATAGTAGGACT	RL33	ACAGAGACTCT	AGAGTCTGTGT
RL16	AGTAGACGTCT	AGACGTCGACT	RL34	ACAGCTCGTGT	ACACGAGGTGT
RL17	AGTCGTACACT	AGTGTAGGACT	RL35	ACAGTGTCGAT	ATCGACAGTGT
RL18	AGTGTAGTAGT	ACTACTAGACT	RL36	ACGAGCGCGCT	AGCGCGCGCGT

7 Method

The general workflow diagram is shown in 1 and described below. The workflow consists of several atomic steps: (1) Input data pre-processing; (2) Trimming poly A/T tails; (3) Vector and contam-

inants trimming; (4) Adapter trimming; (5) Quality trimming; (6) Establishing clip points; (7) Generating output files and summary statistics. Stages 2, 3, 4, 5 are optional depending on chosen cleaning strategy.

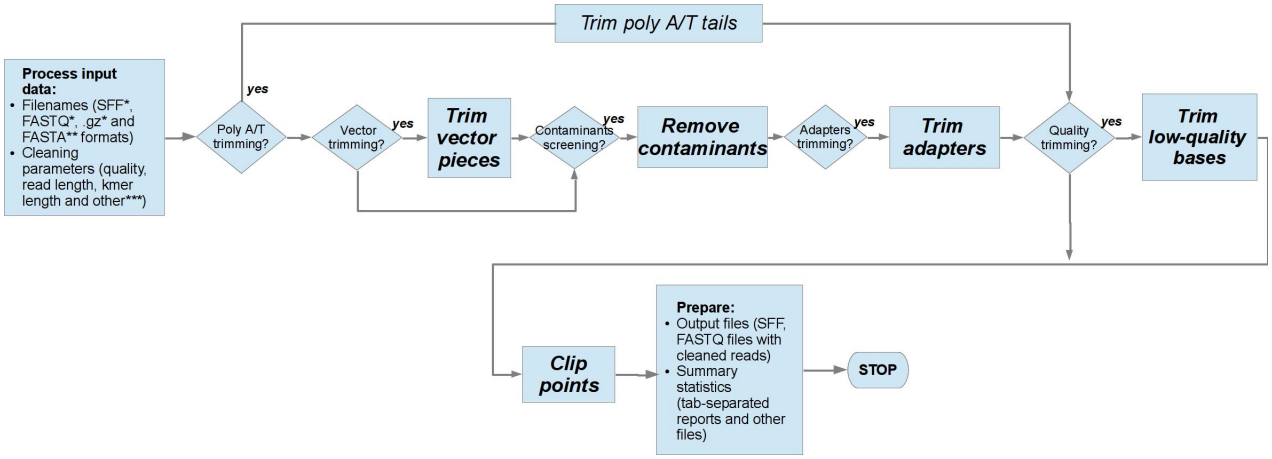

Figure 1: The workflow diagram for SeqyClean.

7.1 Input data pre-processing step

The first processing of the input data is performed on the preprocessing stage. SeqyClean accepts FastQ, SFF and FastA-formatted files the following types:

- DNA library files (format can be SFF or FastQ) that contain sequence data
- References that are contaminant/vector genomes in FastaA - formatted file.

Preprocessing stage analyzes user's inputs and depending on chosen strategy, call appropriate cleaning methods.

7.2 Contaminants and vector detection and trimming

In majority of cases DNA library contains foreign DNA at the same time. Contaminants in DNA library are one of the main obstacles for sequence assembly and mapping. If vector is used (Figure

7.2), it is highly possible that reads contain remnants of vector genome, which is also a contaminant. Contaminant genomes are reference genomes and read as a query sequence. It is up to user to decide which types of contaminants are likely to be in a library and therefore provide reference genomes. Algorithm uses exact kmer matching and works as follows. (1) Reference genomes along with reverse complements are sampled into consecutive kmers of n bases (15 bases by default) long and then each kmer is stored in a hash table named "dictionary". Each dictionary item stores a key, which is a kmer string and a value that represents an array of 2-tuples: pos, id where pos represents the position within a reference of each occurrence of each kmer and id represents a record name in provided FastA file. Searching for a contaminant within a query sequence is performed by dictionary lookup of each kmer from the query sequence. Refer to Figure 7.2. There are two cases: (1) Vector trimming when vector sequence occupies only a part of a read. When vector coordinates are approximately found, then exact coordinates are obtained by applying a pairwise alignment of a vector site in the query sequence and corresponding area in the reference vector genome. The read is discarded if a vector is more than 80% of the read. (2) In case of contaminant screening the whole read considered as contaminant and discarded as soon as it meets several consecutive successful lookups.

7.3 Adapter trimming

Roche 454 and Illumina Next-Generation Sequencing technologies employ short synthetic nucleotides called adapters or primers to immobilize DNA pieces on a surface. Depending on technology, adapters may be different types (Roche 454 and Illumina adapters) and attached to the ends of the DNA particle. Adapters can potentially have a significant impact on genome assembly as it is shown in 7.2. In this case due to adapter on the end, assembler does not extend the growing contig but stops and initializes another one. Two different algorithms are used for adapter trimming: (1) Banded Pairwise alignment (Fickett, 1984) is used for Roche 454 adapters. This is a variation of

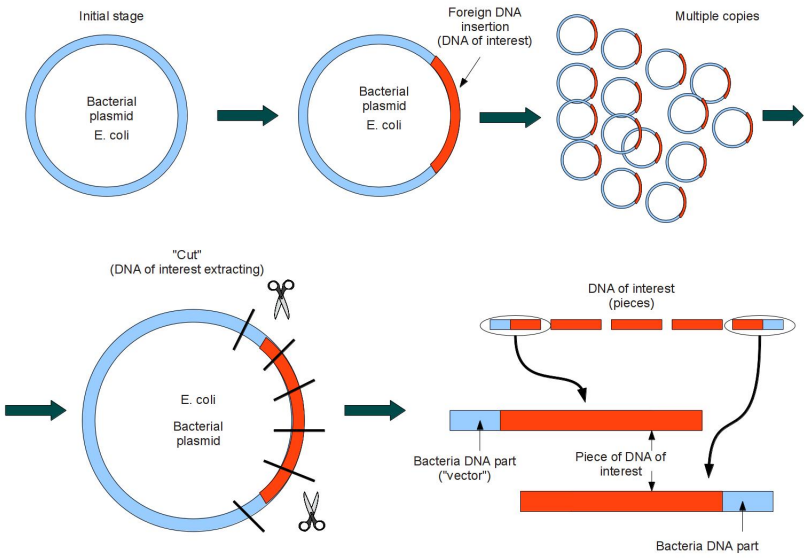

Figure 2: Bacteria vector cloning technology.

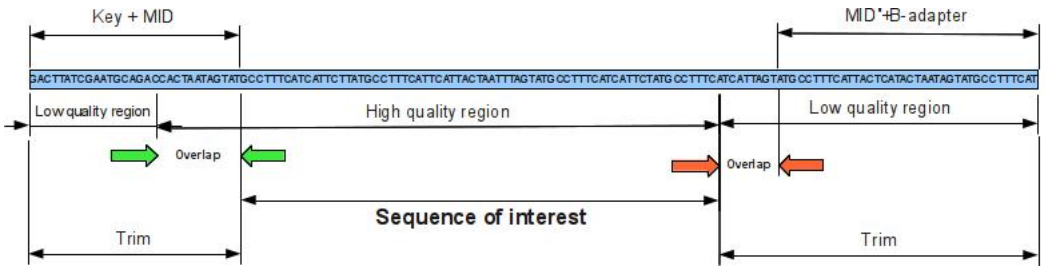

Figure 3: Artifacts within a read (Roche 454).

Smith-Waterman algorithm with reduced omputational complexity in order to increase the alignment performance. (2) SSAHA kmer matching algorithm (Ning, Cox, & Mullikin, 2001), which is used to detect Illumina TruSeq adapters.

7.4 Quality trimming

Errors in reads are unavoidable and assumption that reads are error-free is often misleading. In fact, majority of reads in DNA library have base-call errors and it is important to obtain as less

errors as possible. The quality of nucleotide bases decreases toward the end of the read 7.4. To obtain nucleotide sequence and corresponding quality scores, a program known as a base caller converts the raw output from DNA sequencer into nucleotide bases, each of which is typically assigned a quality score that estimates the probability that the base has been sequenced correctly. Quality scores are needed to trim the low quality ends of a read (Roche 454) and for determination a consensus sequence. SeqyClean utilizes quality trimming approach described in (Chou & Holmes, 2001) and employs the scoring model with an integer scale derived from the error probability p (where p is the probability that a particular base has been read correctly) via the formula $Q = -10log_{10}p$, a scoring scheme named Phred scheme. Nucleotides and corresponding Phred quality scores with other metadata are stored in ascii – delimited text FastQ format which Illumina employs (files are usually compressed with gzip) and binary Sff format which is used by Roche 454. Having the raw data, algorithm discards low-quality regions that may exist within a read by computing average quality of a window. Windows with quality lower than pre-defined threshold are trimmed. The minimum read length to keep is by default 50 nucleotide bases. In order to be able to process short Illumina reads with length about 30 bases, the size of the window can be adjusted.

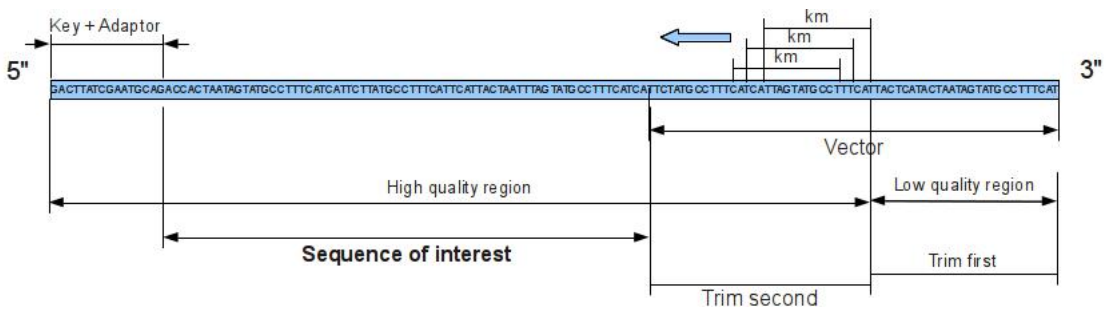

Figure 4: Quality trimming.

7.5 Extending Illimina reads by overlaps between paired-end sequences

Illumina paired-end sequencing technology produces a very large amount of short reads. In spite of high quality and significant amount (up to 100 million) paired-end reads, their length (100-250 nucleotides) still makes genome assembly a difficult and challenging task. There is a potential benefit to take advantage of paired-end paradigm to extend the reads that may potentially overlap into one single-end sequence with length up to almost twice as the length of a pair. This is can be considered as a library pre-processing before conducting the downstream analysis (Magoč & Salzberg, 2011). We included such length extension algorithm to SeqyClean. The algorithm works as follows:

1. Align the pair of reads so that they overlap completely by the full length of the shorter read.
2. Repeat while the overlap is longer than minoverlap:
    a Calculate the overlap length.
    b Calculate the score for the overlap as the ratio between the number of mismatches and the overlap length.
    c If the score of the overlap is equal or bigger that the pre-defined mismatch threshold:
        - Calculate the new quality values of all bases within the overlap by taking maximum quality scores if bases are equal. Otherwise assign the new quality value as a subtraction of quality scores.
        - Save the overlap and progress toward the next pair.
    d Otherwise slide the reads apart by one base, reducing the overlap by one.
3. If the score is less than the mismatch threshold, report that no good overlap was found.

7.6 Removing duplicates

Removing duplicates can be considered to mitigate the effects of PCR amplification bias introduced during the library construction. We define duplicates as reads with identical sequence. Removal of duplicates is dependent on an error rate in the library. Removing duplicates also reduces the pool of reads before mapping and thereby reduces alignment time. The algorithm considers bases within the position from 10 to 25 in the read and does not look at the whole sequence (Figure 5).

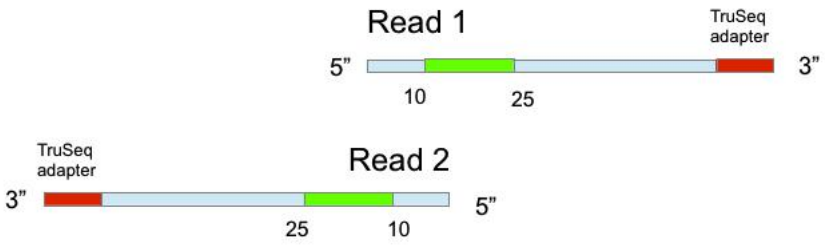

Figure 5: Paired-end Illumina sequences. Green regions are to be checked for duplicates.

7.7 Establishing clip points

Clip point is a position within a read where all bases are discarded after this position. SeqyClean establishes the most conservative clip points, in other words it chooses the right-most clip point from the left and left-most clip point from the right side of the read. This implies, for example, that if quality clip points are located inside of vector clip points, SeqyClean uses the first ones as the actual trim points of the read (Fig. is needed).

7.8 Output files and summary statistics

The output files are FastQ or SFF-formatted files containing either trimmed reads (FastQ-file) or reads with new established clip points (SFF-file). In addition, detailed statistics for each read and summary statistics are generated both in human- (TXT) and machine-readable (TSV) formats.

7.9 Quality and adapter trimming impacts on genome assembly and mapping

We evaluated SeqyClean from two sides: genome assembly and mapping. Our interest was (1) to explore impacts of quality and adapter trimming on genome assembly and (2) genome mapping. We also determined the optimal quality parameters for sequence cleaning. Finally we cleaned human genome data with found optimal quality parameters for SeqyClean.

We tested the impact of quality and adapter trimming on two NGS libraries: Escherichia coli MG-1655 which is Roche 454 pyrosequencing data, 621,578 reads, 327,471,374 bases, and (Yeast) W303 which is Illumina MiSeq data, 2x 3,875,453 reads, 2x 972,738,703 bases. As a reference we used E.coli MG-1655 strain obtained from GenBank [Accession number]. In case with Yeast we used Yeast W303-K60001 strain, which was provided by M. Ralser at al. [CITE]. We cleaned both E.coli and Yeast libraries with SeqyClean by applying quality trimming parameter pairs maximum_average_error, maximum_error_at_ends [1] presented in Phred scores. We varied both of them from 1 to 40 (inclusive), where 1 denotes 20.57% of chance that the base was read correctly and 40 implies 99.99% that a base was read correctly. Some of values of these parameters are presented in Table 1 and compete table is in supplementary materials. We call the pair maximum_average_error, maximum_error_at_ends as a quality parameters. Parameters for adapter trimming were default.

To explore impacts of quality and adapter trimming on assembly we performed assemblies both cleaned E.coli and Yeast libraries with Newbler 2.6, a Roche 454 assembler. We conducted five assemblies for each of the quality parameter pair in order to exclude impacts of assembler's variations. Roche 454 sequencing offers its own clip points stored alongside with the nucleotide data and other information in SFF-formatted file [CITE]. Such "native" clip points can be potentially used as an alternative to SeqyClean's clip ponts. In order to evaluate the impact of Roche's native clip points we performed assembly of E.coli data with Roche clip points only. We also conducted assembly of the raw, non-cleaned sequences for both E.coli and Yeast libraries. In order to ex-

plore effects of quality and adapter trimming on the assembly we estimated the following assembly parameters: (1) N50; (2) total number of contigs; (3) average read length; (4) read coverage; (5) insertions/deletions (indels); (6) total length of the assembly.

We also evaluated impacts of quality and adapter cleaning on mapping. With SeqyClean we cleaned *E.coli* and *Yeast* libraries with the same strategy as before. For those libraries we estimated number of single-nucleotide polymorphisms and insertions/deletions and then compared with raw data. For mapping we used bowtie2 [CITE] aligner along with samtools [CITE] in order to extract SNPs and indels.

8 Acknowledgements

This work was supported by IBEST COBRE, grant NIH/NCRR P20RR16448 and the University Research Office at the University of Idaho.

References

Chou, H., & Holmes, M. H. (2001). DNA sequence quality trimming and vector removal. *Bioinformatics/computer Applications in The Biosciences*, *17*, 1093–1104. doi: 10.1093/bioinformatics/17.12.1093

Criscuolo, A., & Brisse, S. (2013). Alientrimmer: A tool to quickly and accurately trim off multiple short contaminant sequences from high-throughput sequencing reads. *Genomics*(0), -. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0888754313001481` doi: http://dx.doi.org/10.1016/j.ygeno.2013.07.011

Fickett, J. W. (1984). Fast optimal alignment. *Nucleic Acids Research*, *12*, 175–179. doi: 10.1093/nar/12.1Part1.175

Magoč, T., & Salzberg, S. L. (2011). Flash: fast length adjustment of short reads to improve genome assemblies. *Bioinformatics*, *27*(21), 2957-2963. Retrieved from `http://bioinformatics.oxfordjournals.org/content/27/21/2957.abstract` doi: 10.1093/bioinformatics/btr507

Ning, Z., Cox, A. J., & Mullikin, J. C. (2001). SSAHA: A Fast Search Method for Large DNA Databases. *Genome Research*, *11*, 1725–1729. doi: 10.1101/gr.194201

Scheetz, T. E., Trivedi, N., Roberts, C. A., Kucaba, T., Berger, B., Robinson, N. L., ... Casavant, T. L. (2003). ESTprep: Preprocessing CDNA Sequence Reads. *Bioinformatics/computer Applications in The Biosciences*, *19*, 1318–1324. doi: 10.1093/bioinformatics/btg159

White, J. R., Roberts, M., Yorke, J. A., & Pop, M. (2008). Figaro: a novel statistical method for vector sequence removal. *Bioinformatics/computer Applications in The Biosciences*, *24*, 462–467. doi: 10.1093/bioinformatics/btm632

Table 7: Quality parameter values along with corresponding real error

Phred value for the pair {maximum_average_error, maxim_error_at_ends}	Corresponding error probability	Phred value for the pair {maximum_average_error, maxim_error_at_ends}	Corresponding error probability
1,1	0.2057	21,21	0.9921
2,2	0.3690	22,22	0.9937
3,3	0.4988	23,23	0.9950
4,4	0.6019	24,24	0.9960
5,5	0.6838	25,25	0.9968
6,6	0.7488	26,26	0.9975
7,7	0.8005	27,27	0.9980
8,8	0.8415	28,28	0.9984
9,9	0.8741	29,29	0.9987
10,10	0.9000	30,30	0.9990
11,11	0.9206	31,31	0.9992
12,12	0.9369	32,32	0.9994
13,13	0.9499	33,33	0.9995
14,14	0.9602	34,34	0.9996
15,15	0.9684	35,35	0.9997
16,16	0.9749	36,36	0.9997
17,17	0.9800	37,37	0.9998
18,18	0.9842	38,38	0.9998
19,19	0.9874	39,39	0.9999
20,20	0.9900	40,40	0.9999