SeqyClean User Manual

Ilya Y. Zhbannikov

June 21, 2017

1 Introduction

We developed SeqyClean – a bioinformatics software pipeline for next-generation sequence cleaning. The first purpose of SeqyClean is to incorporate all aspects of NGS cleaning: adapter, contaminant, poly A/T and quality trimming into a single bioinformatics pipeline. SeqyClean successfully recognizes and removes technological components (adapters, primers, barcodes), contaminants and vector. SeqyClean provides a comprehensive flexible quality trimming by incorporation the LUCY© quality trimming algorithm to remove bad-quality and poly-A/T erroneous data. In addition, SeqyClean has more features: extension paired-end reads by overlap and duplicates removal, which we consider important for genome assembly because it reduces data space by discarding duplicated reads.

2 Citing SeqyClean

2.1 BibText

```
@misc{seqyclean,
author =   {Ilya Zhbannikov and Samuel Hunter and James Foster and Matthew  Settles},
title =    {SeqyClean User Manual},
howpublished = {\url{https://github.com/ibest/seqyclean}},
year = {2012--2017}
}
```

2.2 Plain text

[ZHFS17] Ilya Zhbannikov, Samuel Hunter, James Foster, and Matthew Settles. SeqyClean User Manual. https://github.com/ibest/seqyclean, 2012–2017.

3 Installation

3.1 How to download

SeqyClean is an open-source software application available from the Bitbucket for free under this link: https://github.com/ibest/seqyclean. Save the file under some name you wish, unzip and compile:

```
$cd  path_to_SeqyClean_directory
$make
```

3.2 Usage

SeqyClean works on SFF files (454, Ion Torrent) and FASTQ Illumina (paired- and single-end reads).

Roche 454 libraries:

```
./seqyclean [options] -454 reads.sff -o output_prefix
```

Paired-end Illumina libraries:

```
./seqyclean [options] -1 R1.fastq -2 R2.fastq -o output_prefix
```

Single-end libraries:

```
./seqyclean [options] -U reads.fastq -o output_prefix
```

3.3 Options across different technology types

The options that can be used for all library types are shown in Table 1. See library-specific options in the following tables Table 2 (paired-end reads), Table 3 (Roche 454 pyrosequence reads) and Table 4 (single-end FASTQ libraries).

Table 1: Options for all libraries

`-v <filenme>`	This option does vector trimming. If you choose this option, the program assumes the file of vector sequences provided in `<filename>`. This file must be given in FASTA format.
	Examples:
	`./seqyclean -v vectors.fa -1 R1.fastq -2 R2.fastq -o Test`
	`./seqyclean -v vectors.fa -U R1.fastq -o Test`
	`./seqyclean -v vectors.fa -454 in.sff -o Test`

`-c <filename>`	This option is used for contaminants screening. If you choose this option, the program assumes the reference genome provided in `<filename>`. This file must be given in FASTA format. When SeqyClean recognizes contaminants in the sequence, the whole sequence gets discarded. Note: contaminant reference sequences must be provided!
	Examples:
	`./seqyclean -v contaminants.fa -1 R1.fastq -2 R2.fastq -o Test`
	`./seqyclean -v contaminants.fa -U R1.fastq -o Test`
	`./seqyclean -c contaminants.fa -454 in.sff -o Test`

`-k <value>`	Use this option in order to specify a size of k-mer. Default k-mer size is 15 bases.
`-kc <value>`	Special k-mer size for contaminant screening. Use this option only if you want to have different k-mer sizes for contaminant dictionary.

`-qual [mae mee -w0 <value> -w1 <value>]`	Quality trimming. Default values for `mae` (maximum average error) and `mee` (maximum error at ends) are [20 20]. "w0" and "w1" are window parameters.
	Examples:
	`./seqyclean -1 R1.fastq -2 R2.fastq -o Test -qual`
	`./seqyclean -qual 21 -w0 40 -w1 5 -U R1.fastq -o Test`
	`./seqyclean -qual 25 33 -454 in.sff -o Test`

`-bracket [bracket length] [max avg error]`	Bracket parameters: minimum length (default=10) and maximum average error (default=0.794 or 1 phred) - these maximum average error values means that checking for bracket error is OFF)
`-window window_size max_avg_error [window_size max_avg_error ...]`	Parameters for window trimming. By default two windows are used: large window, 50 bp long, with maximum average error of 0.794 and small window, 10 bp long, with maximum average error of 0.794. By default checking for error at this stage of quality trimming algorithm is OFF.
`-minlen value`	Use this option `-minlen` 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 `-minlen` parameter, the read will be discarded. By default, the `-minlen` is set to 100 base pairs. Example: `./seqyclean -minlen 10 -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 -1 R1.fastq -2 R2.fastq -o Test`
	`./seqyclean -polyat 12 5 120 -U R1.fastq -o Test`
	`./seqyclean -polyat -454 in.sff -o Test`

`-dup [startdw] [sizedw] [maxdup]`	This option provides duplicates screening.
	
	`startdw` – search starting position (10 by default); `sizedw` – size of window (35 by default); `maxdup` – maximum number of duplicates (3 by default)
	Examples:
	`./seqyclean -dup -1 R1.fastq -2 R2.fastq -o Test`
	`./seqyclean -dup -sizedw 50 -U R1.fastq -o Test`
	`./seqyclean -dup -startdw 5 -sizedw 30 -maxdup 12 -454 in.sff -o Test`

`-verbose`	Verbose output, default=off.
`-detrep`	Generate detailed report for each read, default=off.
`-no_adapter_trim`	This option turns off adapter trimming. Default=off.

Table 2: Illumina paired-end libraries

`-shuffle`	With this option SeqyClean will combine output paired-end libraries into one single file named `<output_prefix>_shuffled.fastq`. However, SeqyClean still does keep single-end reads (reads without corresponding pairs) in `<output_prefix>_SE.fastq` file. Example: `./seqyclean -shuffle -1 R1.fastq -2 R2.fastq -o Test`
`-at <value>`	This option sets the similarity threshold for adaprer trimming by overlap. By default its value is set to 0.75.
`-alen <value>`	This option sets the maximum adapter length for adaprer trimming by overlap. By default its value is set to 60 bases.
`-overlap <value>`	This option turns on merging overlapping paired-end reads and `<value>` is the minimum overlap length. By default the minimum overlap length is 16 base pairs.
`-i64`	Turns on 64-quality base, default = off.
`-new2old`	A switch to fix read IDs, default=off (As is detailed in: http://contig.wordpress.com/2011/09/01/newbler-input-iii-a-quick-fix-for-the-new-illumina-fastq-header/)

Table 3: Roche 454 pyrosequence libraries

`-t <value>`	Number of threads (not yet applicable to Illumina mode), default=4.
`-fastq`	Output in FASTQ format, default=off.
`-fasta`	Output in FASTA format, default=off.
`-m <filename>`	Using custom barcodes, default=off. <filename> - a path to a FASTA-file with custom barcodes.
`-d <value>`	This option `-d` is intended to tweak an overlap between two consecutive k-mers. By default the length of overlap it is set to 1 bp. Example: `./seqyclean -d 10 -454 in.sff -o Test`

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

3.4 Description of seqyclean output

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

3.4.1 SFF (454, Ion Torrent)

- Output_prefix.sff , .fastq (optionally)
- Output_prefix_Report.tsv - if `-detrep` flag is on.
- Prefix_SummaryStatistics.txt
- Prefix_SummaryStatistics.tsv

3.5 FASTQ

After processing FASTQ reads, SeqyClean generates PE1 and PE2 files that contain paired-end reads, SE file with single-end reads OR 'shuffled' file and file with single-end reads (SE) if `-shuffle` flag was set. output files in FASTQ format.

- Output_prefix_PE1.fastq
- Output_prefix_PE2.fastq
- Output_prefix_shuffled.fastq (if `-shuffle` flag was set)
- Output_prefix_SE.fastq
- Output_prefix_PE1_Report.tsv (if `-detrep` flag was set)
- Output_prefix_PE2_Report.tsv (if `-detrep` flag is on)
- Prefix_SummaryStatistics.txt
- Prefix_SummaryStatistics.tsv

3.6 Workflow

The general workflow diagram of SeqyClean is shown in Figure 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 contaminants trimming; (4) Adapter trimming; (5) Quality trimming; (6) Extension by overlap; (7) PCR duplicates removal; (8) Establishing clip points; (9) Generating output files and summary statistics. Stages 2, 3, 4, 5, 6, 7 are optional depending on chosen cleaning strategy.

3.6.1 Supported RLMIDs

The set of supported Roche 454 RL MIDs is shown in Table 5.

4 Acknowledgements

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

References

[ZHFS17] Ilya Zhbannikov, Samuel Hunter, James Foster, and Matthew Settles. Seqyclean user manual. https://github.com/ibest/seqyclean, 2012–2017.

Table 4: Single-end FASTQ libraries

`-U <filename>`	Turns on single-end mode.
`-i64`	Turns on 64-quality base, default = off.
`-new2old`	A switch to fix read IDs, default=off (As is detailed in: http://contig.wordpress.com/2011/09/01/newbler-input-iii-a-quick-fix-for-the-new-illumina-fastq-header/)

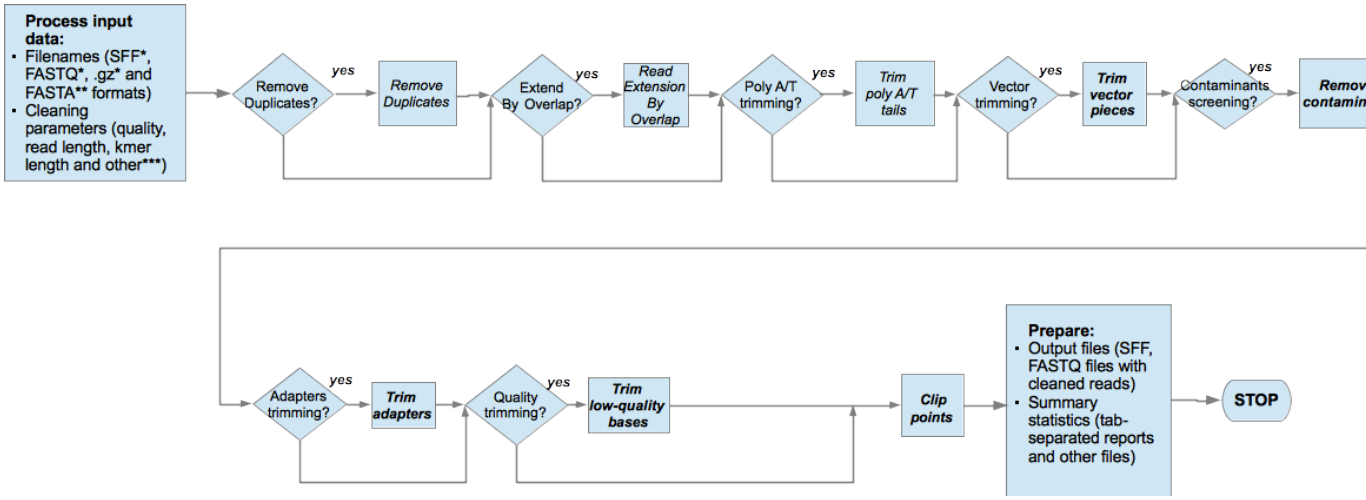

Figure 1: The workflow diagram for SeqyClean.

Table 5: 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