

# SeqyClean

## ver. 1.2.3

## User Manual

Ilya Y. Zhbannikov

January 27, 2013

## Introduction

The main purpose of SeqyClean is to clean next-generation (NG) sequences (“reads”): Roche 454 and Illumina. It provides adapter searching and trimming, quality and poly A/T trimming base on LUCY strategy[1]. SeqyClean accepts both FASTQ and SFF files and also zipped .gz files (zipped FASTQ files only).

## How to download

Follow the link: <https://bitbucket.org/izhbannikov/seqyclean>

## How to compile

```
cd < path_to_SeqyClean_directory >
make clean
make
```

## Usage

### Roche 454 sequences

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

### Main arguments

< -454 input\_filename >    The filename of file to be cleaned. Can be in SFF or FASTQ formats. "454" tells the program to process Roche 454 reads

< -o output\_prefix >        The files produced will be start with the output\_prefix followed by "\_"

### 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: <pre>./seqyclean -v vectors.fa -454 Small454Test.sff -o Test</pre>
-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: <pre>./seqyclean -c contaminants.fa -454 Small454Test.sff -o Test</pre>

<code>-p pcr_file_name</code>	<p>If DNA library was prepared with PCR method, use this option to remove PCR primers. You have to provide additional file <code>pcr_file_name</code> with set of used primers.</p> <p>Example:</p> <pre>./seqyclean -p PCR_primers.csv -454 Small454Test.sff -o Test</pre>
<code>-m file_of_RL_MIDS</code>	<p>This option works in 454 mode only. Use this option to provide your own RLMIDS. SeqyClean will use them and will not use its own provided by default.</p> <p>Example:</p> <pre>./seqyclean -m file_of_custom_RL_MIDS -454 Small454Test.sff -o Test</pre>
<code>-k k_mer_size</code>	<p>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 k-mer that will be used as a dictionary word size.</p> <p>Example:</p> <pre>./seqyclean -k 18 -454 Small454Test.sff -o Test</pre>
<code>-kc k_mer_size</code>	<p>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.</p> <p>Example:</p> <pre>./seqyclean -kc 25 -454 Small454Test.sff -o Test</pre>
<code>-f overlap</code>	<p>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</p> <p>Example:</p> <pre>./seqyclean -f 10 -454 Small454Test.sff -o Test</pre>
<code>-t number_of_threads</code>	<p>Specifies a number of threads in order to take advantage from using a multicore system.</p> <p>Example:</p> <pre>./seqyclean -t 16 -454 Small454Test.sff -o Test</pre>
<code>-qual max_avg_error max_error_at_ends</code>	<p>LUCY parameters for quality trimming. if "-qual" is set that means you have to provide <code>max_avg_error</code> and <code>max_error_at_ends</code>. Otherwise default values [0.01 0.01] will be used.</p> <p>Examples:</p> <pre>./seqyclean -qual -454 Small454Test.sff -o Test ./seqyclean -qual 0.01 0.05 -454 Small454Test.sff -o Test</pre>
<code>--qual_only</code>	<p>Use <code>--qual_only</code> parameter if you want to do only quality trimming.</p> <p>Example:</p> <pre>./seqyclean --qual_only -qual -454 Small454Test.sff -o Test</pre>
<code>--fastq</code>	<p>If input is given in SFF format, by default the output will be also in SFF format. Use this option only if you want to have FASTQ format on the output instead SFF.</p> <p>Example:</p> <pre>./seqyclean --fastq -454 Small454Test.sff -o Test</pre>
<code>--keep_fastq</code>	<p>Use this option only if you want to keep generated FASTQ file from your input SFF</p> <p>Example:</p> <pre>./seqyclean --keep_fastq -454 Small454Test.sff -o Test</pre>
<code>-minimum_read_length &lt;value&gt;</code>	<p>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 <code>minimum_read_length</code> parameter, such the read will be discarded. By default, the <code>minimum_read_length</code> is set to 50 base pairs.</p> <p>Example:</p> <pre>./seqyclean -minimum_read_length 100 -454 Small454Test.sff -o Test</pre>
<code>-polyat [cdna] [cerr] [crng]</code>	<p>This option provides trimming of poly A/T tails from nucleotide sequences.</p>

	Parameters: <b>cdna</b> – tail length (10 by default) <b>cerr</b> – maximum number of errors per tail (3 by default) <b>crng</b> – range to search poly A/T tails (50 by default) Examples: ./seqyclean -polyat -454 Small454Test.sff -o Test ./seqyclean -polyat 12 5 67 SmallTestPolyAT.fastq.gz -o Test_polyAT
--	---

## Illumina paired-end sequences

```
./seqyclean [options]* < -1 input_filename.1 > < -2 input_filename.2 > < -o output_prefix >
```

### Main arguments

< -1 input_filename.1 >	The filenames of the file to be cleaned. Can be in FASTQ formats only (the program also accepts .gz files).
< -2 input_filename.2 >	
< -o output_prefix >	The files produced will be start with the output_prefix followed by ”_”

### 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 SmallTestIllumina.R1.fastq.gz -2 SmallTestIllumina.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 SmallTestIllumina.R1.fastq.gz -2 SmallTestIllumina.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 SmallTestIllumina.R1.fastq.gz -2 SmallTestIllumina.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 SmallTestIllumina.R1.fastq.gz -2 SmallTestIllumina.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 [0.01 0.01] will be used. Examples: ./seqyclean -qual -1 SmallTestIllumina.R1.fastq.gz -2 -1 SmallTestIllumina.R2.fastq.gz -o Test ./seqyclean -qual 0.01 0.05 -1 SmallTestIllumina.R1.fastq.gz -2 SmallTestIllumina.R2.fastq.gz -o Test

<code>--qual_only</code>	Use <code>--qual_only</code> parameter if you want to do only quality trimming. Example: <code>./seqyclean --qual_only -qual -1 SmallTestIllumina_R2.fastq.gz -2 SmallTestIllumina_R2.fastq.gz -o Test</code>
<code>-minimum_read_length &lt;value&gt;</code>	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 <code>minimum_read_length</code> parameter such read will be discarded. By default, the <code>minimum_read_length</code> is set to 50 base pairs. Example: <code>./seqyclean -minimum_read_length 100 -1 SmallTestIllumina_R1.fastq.gz -2 SmallTestIllumina_R2.fastq.gz -o Test</code>
<code>-polyat [cdna] [cerr] [crng]</code>	This option provides trimming of poly A/T tails from nucleotide sequences. Parameters: <code>cdna</code> – tail length (10 by default) <code>cerr</code> – maximum number of errors per tail (3 by default) <code>crng</code> – range to search poly A/T tails (50 by default) Examples: <code>./seqyclean -polyat -454 Small454Test.sff -o Test</code> <code>./seqyclean -polyat 15 4 55 SmallTestPolyAT.fastq.gz -o Test.polyAT</code>

## Help

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

## Quick examples of usage

Example for 454 reads:

```
./seqyclean -v vectors.fasta -qual 0.05 0.05 -454 Small454Test.sff -o cleaned_data/Small454Test_cleaned
```

See Figure 2.

Example for Illumina reads:

```
./seqyclean -v vectors.fasta -v -c contaminants.fasta -qual -1 SmallTestIllumina_R1.fastq.gz -2 SmallTestIllumina_R2.fastq.gz -o cleaned_data/SmallTestIllumina_cleaned
```

See Figure 3.

## 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.

### Roche 454

After processing Roche 454 reads, SeqyClean outputs a cleaned file by default in Standard Flowgam Format (SFF) or (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>_<filename>_Report.csv` file which holds the detailed statistics for every read.

Filename	Adapters	Vector	Contaminants	Quality
<Output_prefix>_adp.sff — .fastq*	X			
<Output_prefix>_adp_Report.csv				
<Output_prefix>_adp_vec.sff — .fastq	X	X		
<Output_prefix>_adp_vec_Report.csv				
<Output_prefix>_adp_vec_cont.sff — .fastq	X	X	X	
<Output_prefix>_adp_vec_cont_Report.csv				
<Output_prefix>_adp_vec_cont_qual.sff — .fastq	X	X	X	X
<Output_prefix>_adp_vec_cont_Report.csv				
<Output_prefix>_adp_qual.sff — .fastq	X			X
<Output_prefix>_adp_qual_Report.csv				
<Output_prefix>_adp_vec_qual.sff — .fastq	X	X		X
<Output_prefix>_adp_vec_qual_Report.csv				
<Output_prefix>_adp_cont_qual.sff — .fastq	X			X
<Output_prefix>_adp_cont_qual_Report.csv				
<Output_prefix>_adp_cont.sff — .fastq	X		X	
<Output_prefix>_adp_cont_Report.csv				
<*Output_prefix>_qual.sff — .fastq				X*
<*Output_prefix>_qual_Report.csv				

## Illumina

After processing Illumina reads, SeqyClean generates two (shuffle 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.

Filename	Adapters	Vector	Contaminants	Quality
<Output_prefix>_shuffle_adp.fastq <Output_prefix>_SE_adp.fastq <Output_prefix>_PE1_adp.fastq <Output_prefix>_PE2_adp.fastq <Output_prefix>_PE1_adp_Report.csv <Output_prefix>_PE2_adp_Report.csv	X			
<Output_prefix>_shuffle_adp_vec.fastq <Output_prefix>_SE_adp_vec.fastq <Output_prefix>_PE1_adp_vec.fastq <Output_prefix>_PE2_adp_vec.fastq <Output_prefix>_PE1_adp_vec_Report.csv <Output_prefix>_PE2_adp_vec_Report.csv	X	X		
<Output_prefix>_shuffle_adp_vec_cont.fastq <Output_prefix>_SE_adp_vec_cont.fastq <Output_prefix>_PE1_adp_vec_cont.fastq <Output_prefix>_PE2_adp_vec_cont.fastq <Output_prefix>_PE1_adp_vec_cont_Report.csv <Output_prefix>_PE2_adp_vec_cont_Report.csv	X	X	X	
<Output_prefix>_shuffle_adp_vec_cont_qual.fastq <Output_prefix>_SE_adp_vec_cont_qual.fastq <Output_prefix>_PE1_adp_vec_cont_qual.fastq <Output_prefix>_PE2_adp_vec_cont_qual.fastq <Output_prefix>_PE1_adp_vec_cont_qual_Report.csv <Output_prefix>_PE2_adp_vec_cont_qual_Report.csv	X	X	X	X
<Output_prefix>_shuffle_adp_qual.fastq <Output_prefix>_SE_adp_qual.fastq <Output_prefix>_PE1_adp_qual.fastq <Output_prefix>_PE2_adp_qual.fastq <Output_prefix>_PE1_adp_qual_Report.csv <Output_prefix>_PE2_adp_qual_Report.csv	X			X
<Output_prefix>_shuffle_adp_vec_qual.fastq <Output_prefix>_SE_adp_vec_qual.fastq <Output_prefix>_PE1_adp_vec_qual.fastq <Output_prefix>_PE2_adp_vec_qual.fastq <Output_prefix>_PE1_adp_vec_qual_Report.csv <Output_prefix>_PE2_adp_vec_qual_Report.csv	X	X		X
<Output_prefix>_shuffle_adp_cont_qual.fastq <Output_prefix>_SE_adp_cont_qual.fastq <Output_prefix>_PE1_adp_cont_qual.fastq <Output_prefix>_PE2_adp_cont_qual.fastq <Output_prefix>_PE1_adp_cont_qual_Report.csv <Output_prefix>_PE2_adp_cont_qual_Report.csv	X		X	X
<Output_prefix>_shuffle_qual.fastq <Output_prefix>_SE_qual.fastq <Output_prefix>_PE1_qual.fastq <Output_prefix>_PE2_qual.fastq <Output_prefix>_PE1_qual_Report.csv <Output_prefix>_PE2_qual_Report.csv				X

## Supported RL MIDs 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	ACGTAGTGTCTG
RL6	ACGCGTCTAGT	ACTAGAGGCGT	RL24	CGTACGTCTGAT	ATCGACGGACG
RL7	ACGTACACACT	AGTGTGTGCGT	RL25	CTACTCGTAGT	ACTACGGGTAG
RL8	ACGTACTGTGT	ACACAGTGCGT	RL26	GTACAGTACGT	ACGTACGGTAC
RL9	ACGTAGATCGT	ACGATCTGCGT	RL27	GTCGTACGTAT	ATACGTAGGAC
RL10	ACTACGTCTCT	AGAGACGGAGT	RL28	GTGTACGACGT	ACGTCTGTGCAC
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	AGACGTCTGACT	RL34	ACAGCTCGTGT	ACACGAGGTGT
RL17	AGTCGTACACT	AGTGTAGGACT	RL35	ACAGTGTCTGAT	ATCGACAGTGT
RL18	AGTGTTAGTAGT	ACTACTAGACT	RL36	ACGAGCGCGCT	AGCGCGCGCGT

## Acknowledgements

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

## References

- [1] Chou, H. and Holmes, M *DNA sequence cleaning and vector removal* 2001, BMC Bioinformatics, 12, 1093 — 1104.
- [2] <http://www.idtdna.com/pages/products/nextgen/454-adapters>

## Contacts

For any questions regarding SeqyClean (i.e. usage, bugs found, performance and so on) please contact Ilya by email: zhba3458@vandals.uidaho.edu. I appreciate every feedback provided by users!  
Thank you.

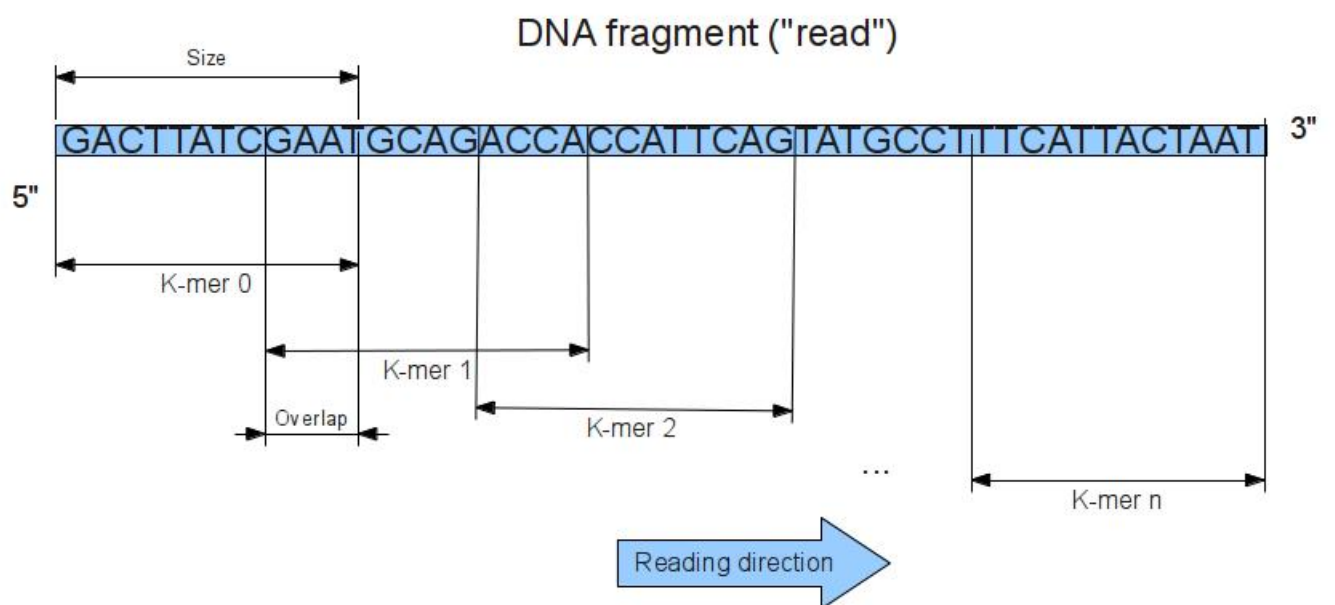


Figure 1: Making a set of consecutive kmers.



```
~/bio/app/SeqyClean: bash
File Edit View Bookmarks Settings Help
o sff.o poly.o abi.o gzstream.o -lpthread -Xlinker -zmuldefs -lz
ilya@kwt:~/bio/app/SeqyClean$ ./seqyclean -v vectors.fasta -qual 0.05 0.05 -454 Small454Test.sff -o cleaned_data/Small454Test_cleaned
=====Parameters=====
Version: 1.2.3
-----Basic parameters-----
Provided data files :
Small454Test.sff
Adapters trimming: YES. Custom RLMIDS: NO. Default RL MIDs will be used.
Vector screening: YES. Vector file provided: vectors.fasta
K-mer_size for for vector trimming: 15 bp
Distance between the first bases of two consecutive kmers: 1 bp
Contaminants screening: NO
Quality trimming: YES
Maximum error: 0.05
Maximum error at ends: 0.05
-----Output files-----
Output prefix: cleaned_data/Small454Test_cleaned
Report file: cleaned_data/Small454Test_cleaned_adp_vec_qual_Report.csv
Roche output file: cleaned_data/Small454Test_cleaned_adp_vec_qual.sff
-----Other parameters-----
k-mer_size: 15 bp
Maximum number of mismatches allowed in alignment: 5
Minimum read length to accept: 50 bp
Distance between the first bases of two consecutive kmers: 1 bp
number_of_threads: 4
=====Starting the process=====
Parsing screening file vectors.fasta
Parsing file: Small454Test.sff...
File is in SFF format, starting conversion...
Conversion finished. Total number of reads read from given file(s): 1000
Running the main pipeline...
Trimming small pieces of vector...
Right ends...
Left ends...
Making a report...
Making output files...
=====Summary Statistics=====
Reads analyzed: 1000, Bases:988199
Found ->
Left mid tag: 1000, 100%
Right mid tag: 1000, 100%
# of reads with vector: 212, 21.2%
Reads left trimmed ->
By adapter: 968
By quality: 4
By vector: 128
Average left trimmed length: 68.746 bp
Reads right trimmed ->
By adapter: 172
By quality: 767
By vector: 61
Average right trimmed length: 392.747 bp
Reads discarded: 0 ->
By read length: 0
-----
Reads accepted: 1000, %100
Average trimmed length: 19.0958 bp
Average read length: 977.238 bp
=====
Program finished.
Elapsed time = 7.578092e+00 seconds
ilya@kwt:~/bio/app/SeqyClean$
```

Figure 2: Program output: Roche 454 reads

```
~/bio/app/SeqClean : seqclean
File Edit View Bookmarks Settings Help
ilya@ilort:~/bio/app/SeqClean$ ./seqclean -v vectors.fasta -c contaminants.fasta -qual -1 SmallTestIllumina_R1.fastq.gz -2 SmallTestI
llumina_R2.fastq.gz -o cleaned_data/SmallTestIllumina_cleaned
-----Parameters-----
Version: 1.2.3
-----Basic parameters-----
Provided data files :
PE1: SmallTestIllumina_R1.fastq.gz, PE2: SmallTestIllumina_R2.fastq.gz
Adapters trimming: YES
Vector screening: YES. Vector_file provided: vectors.fasta
K-mer_size for for vector trimming: 15
Distance between the first bases of two consecutive kmers: 1
Contaminants screening: YES. File_of_contaminants: contaminants.fasta
K-mer size for contaminants: 15
Quality trimming: YES
Maximum error: 0.01
Maximum error at ends: 0.01
-----Output files-----
Output prefix: cleaned_data/SmallTestIllumina_cleaned
Report files: cleaned_data/SmallTestIllumina_cleaned_PE1_adp_vec_cont_qual_Report.csv, cleaned_data/SmallTestIllumina_cleaned_PE2_adp_
vec_cont_qual_Report.csv
PE1 file: cleaned_data/SmallTestIllumina_cleaned_PE1_adp_vec_cont_qual.fastq
PE2 file: cleaned_data/SmallTestIllumina_cleaned_PE2_adp_vec_cont_qual.fastq
Single-end reads: cleaned_data/SmallTestIllumina_cleaned_SE_adp_vec_cont_qual.fastq
-----Other parameters-----
Maximum number of mismatches allowed in alignment: 5
Minimum read length to accept: 50
New to old-style Illumina headers: NO
-----Starting the process-----
Parsing screening file vectors.fasta
Parsing screening file contaminants.fasta
Running the Illumina cleaning process...
17 adapter in forward first found in the read #0
15 adapter in forward first found in the read #100
-----Summary Statistics-----
PE1 reads analyzed: 126788, Bases: 19144988
Found ->
Adapters: 2738, 2.15951%
# of reads with vector: 41493, 32.6553%
# of reads with contaminants: 4530, 3.57289%
Reads left trimmed ->
By quality: 1695
By vector: 30695
Average left trimmed length: 1.19972 bp
Reads right trimmed ->
By adapter: 2609
By quality: 117320
By vector: 2329
Average right trimmed length: 7.72159 bp
PE1 reads discarded: 46284
By contaminants: 4530
By read length: 46284
-----
PE2 reads analyzed: 126788, Bases: 19144988
Found ->
Adapters: 804, 0.634129%
# of reads with vector: 39336, 31.025%
# of reads with contaminants: 4479, 3.53267%
Reads left trimmed ->
By quality: 2919
By vector: 36430
Average left trimmed length: 1.22776 bp
Reads right trimmed ->
By quality: 119420
By vector: 2261
By adapter: 628
Average right trimmed length: 8.43108 bp
PE2 reads discarded: 49284
By contaminants: 4479
By read length: 49284
-----
-----Summary for PE & SE-----
Pairs kept: 75467, 59.5222%, Bases: 22405291, 58.5148%
Pairs discarded: 44167, 34.8363%, Bases: 13237434, 34.8327%
Single Reads PE1 kept: 5937, Bases: 720424
Single Reads PE2 kept: 2117, Bases: 299452
Average trimmed length PE1: 2.03248 bp
Average trimmed length PE2: 3.07894 bp
Average read length PE1: 148.968 bp
Average read length PE2: 147.921 bp
-----
Done cleaning
Program finished.
```

Figure 3: Program output: paired-end Illumina reads.