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: ./seqyclean -v vectors.fa -454 Small454Test.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. Example: ./seqyclean -c contaminants.fa -454 Small454Test.sff -o Test

`-p pcr_file_name`	If DNA library was prepared with PCR method, use this option to remove PCR primers. You have to provide additional file pcr_file_name with set of used primers. Example: `./seqyclean -p PCR_primers.csv -454 Small454Test.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 its own provided by default. Example: `./seqyclean -m file_of_custom_RL_MIDS -454 Small454Test.sff -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 k-mer that will be used as a dictionary word size. Example: `./seqyclean -k 18 -454 Small454Test.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 Small454Test.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 Small454Test.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 Small454Test.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 [0.01 0.01] will be used.
	Examples:
	`./seqyclean -qual -454 Small454Test.sff -o Test`
	`./seqyclean -qual 0.01 0.05 -454 Small454Test.sff -o Test`

`--qual_only`	Use `--qual_only parameter if you want to do only quality trimming.` Example: `./seqyclean --qual_only -qual -454 Small454Test.sff -o Test`
`--fastq`	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. Example: `./seqyclean --fastq -454 Small454Test.sff -o Test`
`--keep_fastq`	Use this option only if you want to keep generated FASTQ file from your input SFF Example: `./seqyclean --keep_fastq -454 Small454Test.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 Small454Test.sff -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 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 >`
 	
`< -2 input_filename_2 >`
	The filenames of the file to be cleaned. Can be in FASTQ formats only (the program also accepts .gz files).

`< -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`

`--qual_only`	Use `--qual_only` parameter if you want to do only quality trimming. Example: `./seqyclean --qual_only -qual -1 SmallTestIllumina_R2.fastq.gz -2 SmallTestIllumina_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. Example: `./seqyclean -minimum_read_length 100 -1 SmallTestIllumina_R1.fastq.gz -2 SmallTestIllumina_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 Small454Test.sff -o Test` `./seqyclean -polyat 15 4 55 SmallTestPolyAT.fastq.gz -o Test_polyAT`

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

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	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

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.

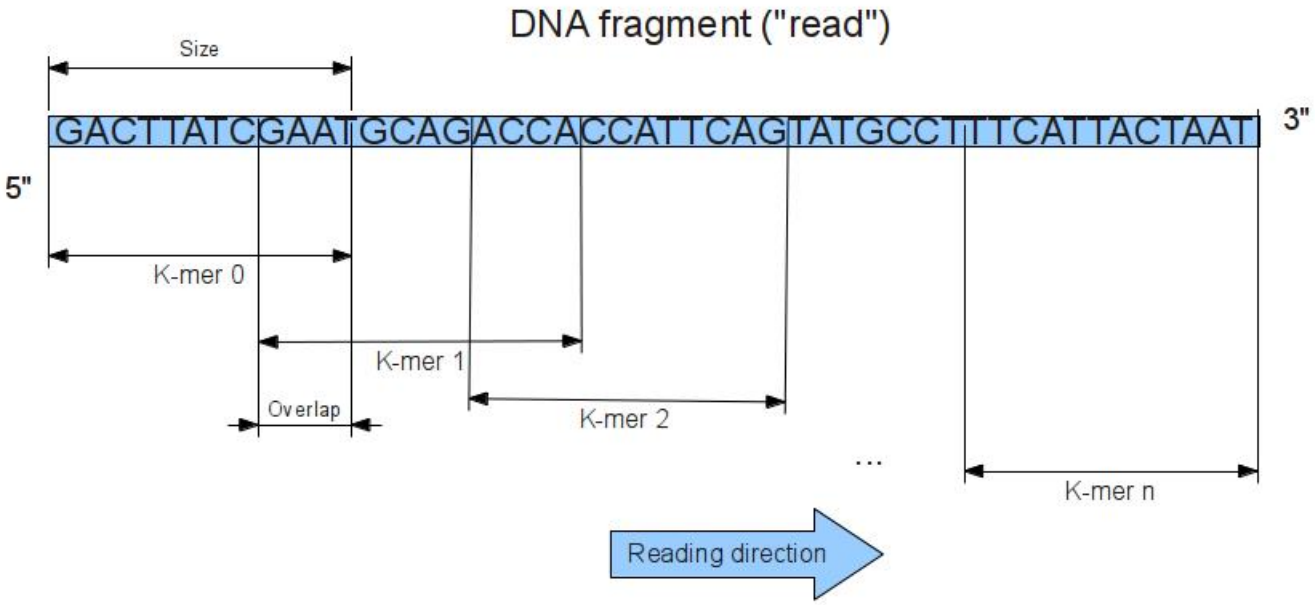

Figure 1: Making a set of consecutive kmers.

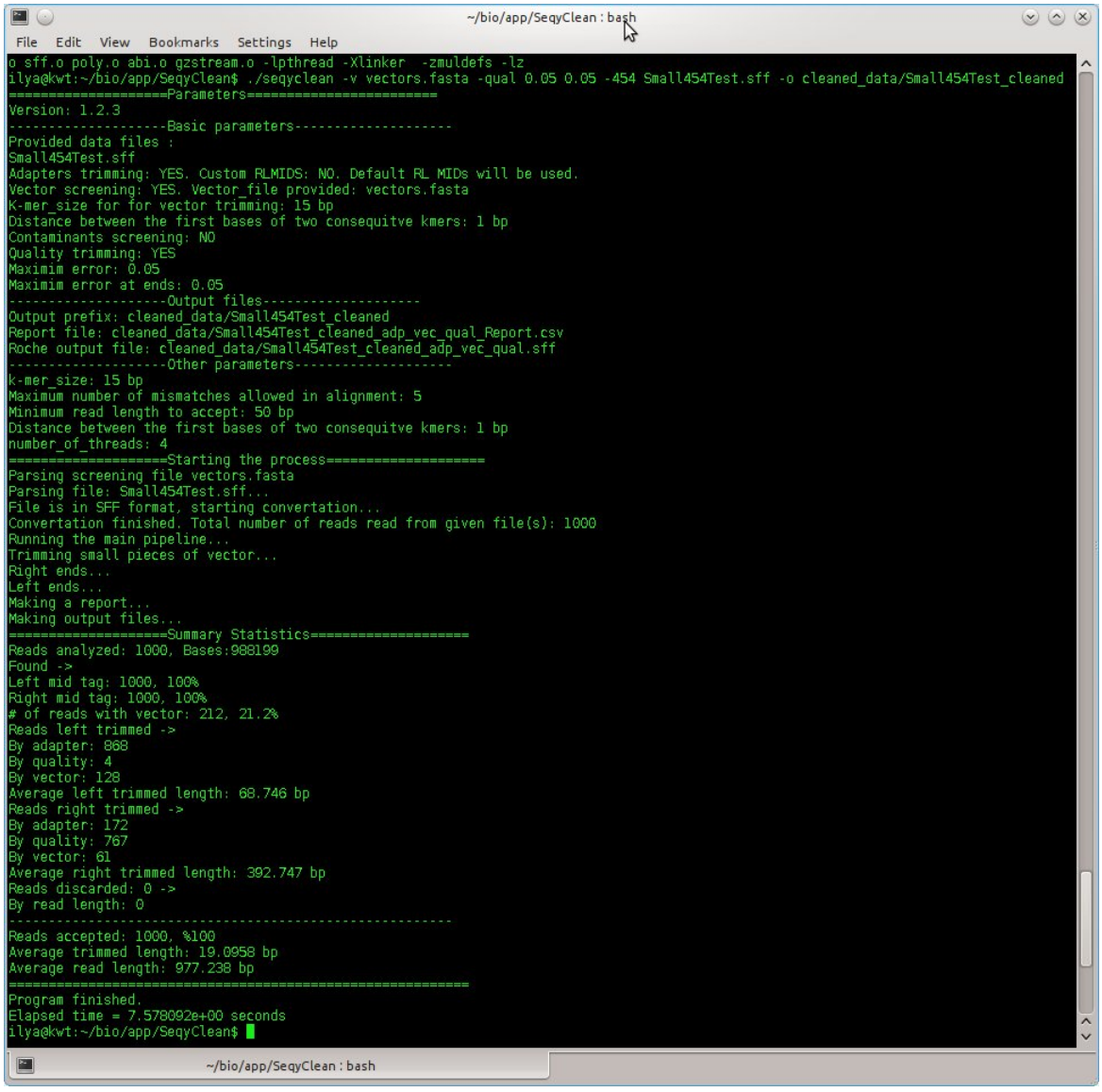

Figure 2: Program output: Roche 454 reads

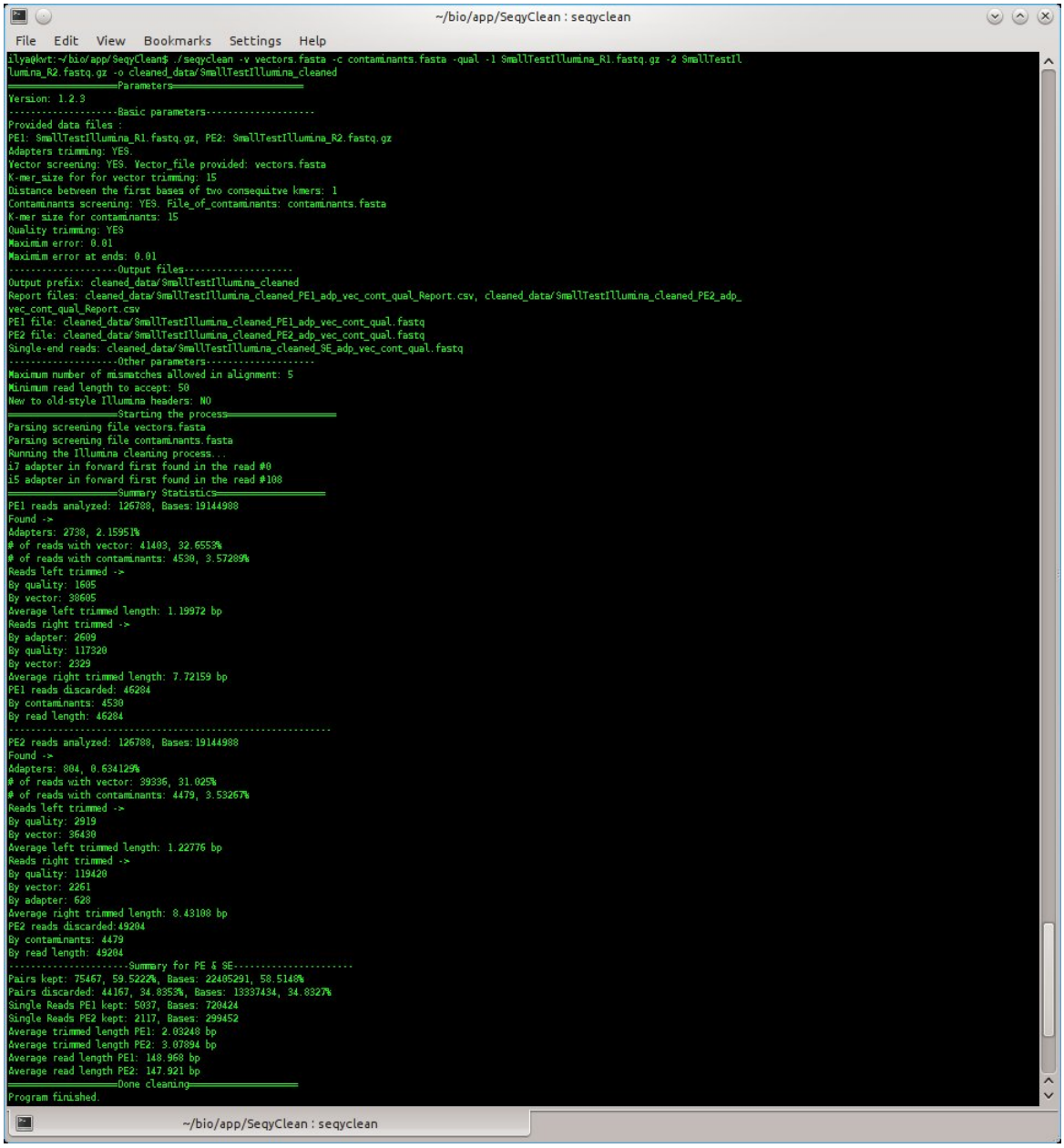

Figure 3: Program output: paired-end Illumina reads.