AlienTrimmer User Guide

[Version 0.3.0] *May 2013*
by Alexis Criscuolo

Criscuolo A, Brisse S (2013) AlienTrimmer: a tool to quickly and accurately trim off multiple short contaminant sequences from high-throughput sequencing reads. Genomics.

`ftp://ftp.pasteur.fr/pub/GenSoft/projects/AlienTrimmer/`

AlienTrimmer is a program that allows detecting and removing contaminant sequences (e.g. adaptors, primers) in both ends of high-throughput sequencing (HTS) read sequences. Based on the decomposition of the specified alien sequences into nucleotide *k*-mers of fixed length *k*, AlienTrimmer is able to determine whether such alien *k*-mers are occurring in both read ends by using a simple polynomial algorithm. AlienTrimmer can process typical HTS single- or paired-ends read files containing millions entries in few minutes with very low computer resources. When setting a Phred score value cut-off, AlienTrimmer is also able to perform quality-based filtering

Installation and Execution

AlienTrimmer runs on all operating systems that run Java 1.6 or later. Sun Java is freely available and presents very good performances. If Sun Java is not installed on your computer, select the last update of JDK/JRE at the following URL:

http://java.sun.com/products/archive/

and follow the 'Installation Instructions' to install the JRE (Java Runtime Environment) on your computer. This allows AlienTrimmer to be run. You can also install the JDK (Java Development Kit) that allows the AlienTrimmer source code to be compiled and run (see below).

On computers with Sun Java (1.6 or higher) installed, you just have to open a terminal, move to the directory containing the executable jar file AlienTrimmer.jar, and launch it with the following command-line:

```
java -jar AlienTrimmer.jar [options]
```

For example, you can get help by using the previous command-line without option:

```
java -jar AlienTrimmer.jar
```

It is not impossible (albeit uncommon) that the executable jar file AlienTrimmer.jar does not work. In this case, you must compile the source code AlienTrimmer/src/AlienTrimmer.java. To do so on Linux System, you must have the Sun JDK (1.6 or higher) installed (see above); then move to the AlienTrimmer/src/ directory, and launch the jar builder:

```
chmod a+x JarMaker.sh
./JarMaker.sh
```

to create a new executable jar file AlienTrimmer.jar.

If needed, you can also compile the source code to native machine code with the GNU compiler GCJ. To do so, you must have the GCJ compiler installed; then move to the AlienTrimmer/src/ directory, and launch the following command line:

```
gcj -fsource=1.6 -O3 --main=AlienTrimmer AlienTrimmer.java -o at
```

to create the binary named at. This binary could be executed with the following command line:

```
./at [options]
```

with options described hereafter. Depending on your computer system, compiling with Sun Java or GNU GCJ will lead to different running times. If fast speed is expected, the two alternative compilations must be assessed with several test files.

Quick Start

AlienTrimmer performs alien trimming on HTS reads inside FASTQ-formatted files (named here `read.fq`). Alien sequences to be trimmed (e.g. adaptors, barcodes, indexes, primers) are contained in a second text file (named here `alien.txt`) where each alien sequence is specified in one line. However, when reading this second file, AlienTrimmer ignores lines beginning by characters '#', '%' or '>'; therefore, comments could be added, or alien sequences could be saved inside a FASTA-formatted file, provided that each alien sequence is written on one line.

Given the FASTQ read file `read.fq` and the alien sequence file `alien.txt`, use the following command line to perform alien trimming:

```
java -jar AlienTrimmer.jar  -i read.fq  -c alien.txt
```

This will create the file `read.fq.at.fq` containing the trimmed reads in FASTQ format. However, to specify an output file name (here `trim.fq`), use the following command line:

```
java -jar AlienTrimmer.jar  -i read.fq  -c alien.txt  -o trim.fq
```

When using paired-ends reads contained in two files (here named `fwd.fq` and `rev.fq`), use the following command line:

```
java -jar AlienTrimmer.jar  -if fwd.fq  -ir rev.fq  -c alien.txt
```

This will create the three files `fwd.fq.at.fq`, `rev.fq.at.fq` and `fwd.fq.at.sgl.fq` containing trimmed paired-ends and singleton reads, respectively (i.e. singleton reads are those remaining when one of the two forward or reverse reads was discarded during the trimming process). However, output file names could be specified with options `-of`, `-or`, and `–os`. Specific alien sequences for forward and reverse reads can be used with options `-cf` and `-cr`, respectively.

The specificity and sensitivity of AlienTrimmer is mainly driven by the *k*-mer length option `-k`. In most cases, default option $k = 10$ leads to satisfactory results. However, alternative integer values could be set (from 5 to 15). AlienTrimmer is as conservative as *k* value is large, e.g. when $k = 15$, very few false positive alien residues are trimmed (i.e. high specificity), but reads with short remnants of alien residues (i.e. < 15 nucleotide long) could remain (i.e. low sensitivity). When dealing with reads containing many sequencing errors, it is recommended to set $k < 10$, e.g.

```
java -jar AlienTrimmer.jar  -i read.fq  -c alien.txt  -k 9
```

However, it should be stressed that lowering *k* improves the sensitivity, but could dramatically decrease the specificity, therefore leading to an unexpected overtrimming (i.e. a large number of non-alien residues will be trimmed with $k = 6$).

Other options are available, such as Phred quality-based trimming, and are described in the next section.

AlienTrimmer Command Line Options

Input/output files (single-ends data)

`-i <infile>`
This option allows the FASTQ read file to be indicated.

`-c <infile>`
This option allows the alien sequence file to be indicated. Each alien oligonucleotide sequence must be written in one line, and may not exceed 32,500 nucleotides. Standard degenerate bases are admitted, i.e. character states M, R, W, S, Y, K, B, D, H, V, N, and X. Input file name may not be a number (see last page).

`-o <outfile>`
This option allows indicating the name of the output FASTQ file that will contain the trimmed reads.

Input/output files (paired-ends data)

`-if <infile>`
This option allows forward read file to be indicated.

`-ir <infile>`
This option allows reverse read file to be indicated.

`-cf <infile>`
This option allows the forward alien sequence file to be indicated. When performing alien trimming on forward reads, AlienTrimmer will only consider alien sequences inside this file. Same requirements as option `-c` for single-ends data.

`-cr <infile>`
Same as option `-cf` but for reverse alien sequences.

`-c <infile>`
This option allows alien sequence file to be indicated. This option allows using the same alien sequence(s) for both forward and reverse reads. Same requirements as option `-c` for single-ends data.

`-of <outfile>`
This option allows forward output file to be indicated. By default, AlienTrimmer uses the forward input file name with the file extension `.at.fq`.

`-or <outfile>`
Same as option `-of` but for reverse reads. By default, AlienTrimmer uses the reverse input file name with the file extension `.at.fq`.

`-os <outfile>`
This option allows singleton output file to be indicated. Singleton reads are those remaining when one of the two forward or reverse reads was discarded during the trimming process; these are saved inside this file. By default, AlienTrimmer uses the forward input file name with the file extension `.at.sgl.fq`.

Trimming and filtering out options

`-k [5-15]`
This option allows specifying the integer value *k* used to perform *k*-mer decomposition. This value must lie between 5 and 15. Recall that $k = 5$ generally leads to over-trimming, whereas $k = 15$ leads to very conservative trimming. By default, AlienTrimmer uses $k = 10$.

`-m [0-15]`
This option allows specifying the maximum number of allowed mismatches *m* between alien and read sequences. By default, AlienTrimmer uses $m = \lceil k / 2 \rceil$.

`-l <integer>`
This option allows specifying the minimum read length *l* to output. All trimmed reads of length lower than *l* are discarded. By default, AlienTrimmer uses $l = 15$.

`-q <char>`
This option allows specifying the Phred quality score character cut-off. Every nucleotide associated with a Phred quality score character with value lower than this cut-off will be considered as an alien residue during the trimming process. See the table (next page) to select a Phred quality score character. In practice, it is recommended to set the cut-off character in quotation marks. By default, AlienTrimmer uses character `"!"`.

`-p [0-100]`
This option allows specifying the minimum allowed percentage *p* of correctly called nucleotides, a correctly called nucleotide being associated with a Phred quality score character with value higher than the cut-off specified with option `-q`. All reads (trimmed or not) with a percentage of correctly called nucleotides lower than this specified value will be discarded. By default, AlienTrimmer uses $p = 0$.

Displaying details

`-v`
When this option is set, AlienTrimmer displays trimming details each time a read is modified.

Characters to specify with option -q depending on the FASTQ encoding format, with corresponding Phred quality score and associated probability values

Probability of incorrect base call	Phred quality score	Encoding format				
		Sanger	Solexa	Illumina 1.3	Illumina 1.5	Illumina 1.8
1.000	-5		;			
1.000	-4		<			
1.000	-3		=			
1.000	-2		>			
1.000	-1		?			
1.000	0	!	@	@		!
0.794	1	"	A	A		"
0.631	2	#	B	B	B	#
0.501	3	$	C	C	C	$
0.398	4	%	D	D	D	%
0.316	5	&	E	E	E	&
0.251	6	'	F	F	F	'
0.200	7	(	G	G	G	(
0.158	8	)	H	H	H	)
0.126	9	*	I	I	I	*
0.100	**10**	**+**	**J**	**J**	**J**	**+**
0.079	11	,	K	K	K	,
0.063	12	-	L	L	L	-
0.050	**13**	**.**	**M**	**M**	**M**	**.**
0.040	14	/	N	N	N	/
0.032	15	0	O	O	O	0
0.025	16	1	P	P	P	1
0.020	17	2	Q	Q	Q	2
0.016	18	3	R	R	R	3
0.013	19	4	S	S	S	4
0.010	**20**	**5**	**T**	**T**	**T**	**5**
0.008	21	6	U	U	U	6
0.006	22	7	V	V	V	7
0.005	**23**	**8**	**W**	**W**	**W**	**8**
< 0.005	24	9	X	X	X	9
< 0.005	25	:	Y	Y	Y	:
< 0.005	26	;	Z	Z	Z	;
< 0.005	27	<	[	[	[	<
< 0.005	28	=	\	\	\	=
< 0.005	29	>	]	]	]	>
< 0.005	30	?	^	^	^	?
< 0.005	31	@	_	_	_	@
< 0.005	32	A	`	`	`	A
< 0.005	33	B	a	a	a	B
< 0.005	34	C	b	b	b	C
< 0.005	35	D	c	c	c	D
< 0.005	36	E	d	d	d	E
< 0.005	37	F	e	e	e	F
< 0.005	38	G	f	f	f	G
< 0.005	39	H	g	g	g	H
< 0.005	40	I	h	h	h	I
< 0.005	41					J

Using AlienTrimmer with Pre-compiled Alien Sequences

When a user always deals with reads produced by the same HTS technique, it is expected that the same putative contaminant oligonucleotides (e.g. adaptors, primers) are used to perform alien trimming. In this case, AlienTrimmer allows directly storing these alien sequences, instead of using the same alien sequence file every time.

To do so, simply edit the source code file `AlienTrimmer.java`, and write the different alien sequences inside one of the 9 arrays named `ALIEN1`, `ALIEN2`, …, `ALIEN9` (approximately line 100). A name could also be given for each alien sequence array by filling the empty strings `ALIEN1NAME`, …, `ALIEN9NAME`. By default, AlienTrimmer stores homopolymer, dimer and trimer sequences inside array `ALIEN0`, `ALIEN1`, and `ALIEN2`, respectively. These oligonucleotide sequence set are named "Homopolymers", "Dimers", and "Trimers", respectively, and could be used to filter out low-complexity reads.

For example, to set four alien sequences inside `ALIEN4`, and name this alien sequence set "four putative alien sequences", the corresponding source code of AlienTrimmer should be updated like this (approximately line 100):

```
static final String ALIEN4NAME = “four putative alien sequences”;
static final String[] ALIEN4 = { “GATCGGAAGAGCGGTTCAGCAGGAATGCCGAG”,
                                 “ACACTCTTTCCCTACACGACGCTCTTCCGATCT”,
                                 “GATCGGAAGAGCACAACGTCT”,
                                 “GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT” };
```

After saving and compiling this new version of `AlienTrimmer.java` (see Installation and Execution), alien sequence set(s) could be directly used with the option `-c`. Indeed, for using the four alien sequences inside array `ALIEN4`, simply use option `-c 4`. Different alien sequence sets could also be used simultaneously: for example, to use alien sequence sets `ALIEN0`, `ALIEN1` and `ALIEN4`, simply use option `-c 014`. Of course, pre-compiled sequence sets could be also used with options `-cf` and `-cr` when using paired-ends data; for example, to trim off alien sequences from `ALIEN1` and `ALIEN2` in file `fwd.fq`, and alien sequences from `ALIEN0`, `ALIEN1` and `ALIEN4` in file `rev.fq`, use the following command line:

```
java -jar AlienTrimmer.jar  -if fwd.fq  -cf 12  -ir rev.fq  -cr 014
```

To display the content of one alien sequence set, simply use the option `-d`: for example, knowing that AlienTrimmer stores homopolymers inside `ALIEN0` by default, the four homopolymeric sequences could be displayed with the following command line:

```
java -jar AlienTrimmer.jar  -d 0
```