Run USEARCH for quality control of a 16s RNA dataset

Alise Ponsero,Bonnie Hurwitz

Abstract

This protocol explores how to run USEARCH (v11.0.667) (Edgar, 2010) on a 16s RNA dataset.

USEARCH is a sequence analysis tool that offers search and clustering algorithms. Here USEARCH will be used to run a quality control on a 16S RNA dataset. This protocol uses a staggered mock community as an example (see attached files).

Citation: Alise Ponsero,Bonnie Hurwitz Run USEARCH for quality control of a 16s RNA dataset. **protocols.io** dx.doi.org/10.17504/protocols.io.s8iehue
Published: 05 Sep 2018

Guidelines

For more information on USEARCH, please read the dedicated documentation.

Before start

You need to download USEARCH on your computer. To do so, request a dowload link here. Once the binary is downloaded, no need to install anything, just copy the binary to a directory that is accessible from the computer where you want to run the code.

For easier run, you can rename the binary to 'usearch' by running in your shell :

mv usearch11.0.667_i86osx32 usearch

On linux and Mac-OS systems, please make sure to have the execution rights on the binary. To add execution rights, run :

chmod u+rwx usearch

To run usearch, you can either add the path to the binary to your PATH variable, ou must type the full

path to the file :

./YOUR/PATH/TO/usearch -cluster_fast reads.fasta -id 0.9 -uc results.uc

Protocol

Get basic information about your read quality

Step 1.

Note : This protocol requires you to have USEARCH installed on your computer. For more information on the installation, please read the 'Guidelines and warnings' section of this protocol. Moreover, you should be able to use a command line interface. To learn the basic of a command line interface, please read this protocol.

Note 2: This protocol uses the 16s RNA Staggered mock community, expected results are provided for this dataset.

The fastx_info provides you with a quick summary of the length distribution and quality of the reads in a FASTQ file. *I*f you have paired reads, review the R1 and R2 files separately because they are usually different, e.g. the R2s tend to have lower quality.

*T*o run this command, open your shell, and run :

```
DIR=$PWD
mkdir fastq_info
#creates a folder named 'fastq_info' where we will store the results

FILE1='$DIR/16S-mockS.1_V4_R1-hdr.fastq'
OUT1='$DIR/fastq_info/16S-mockS.1_V4_R1-hdr.log'
./usearch -fastx_info $FILE1 -output $OUT1

FILE2='$DIR/16S-mockS.1_V4_R2-hdr.fastq'
OUT2='$DIR/fastq_info/16S-mockS.1_V4_R2-hdr.log'
./usearch -fastx_info $FILE2 -output $OUT2
```

This command should create two files in the folder fastq_info, containing the scores for each of those files.

EXPECTED RESULTS

Result for 16S-mockS.1_V4_R1-hdr.log

File size 7.0M, 11.2k seqs, 3.1M letters and quals
Lengths min 274, lo_quartile 276, median 278, hi_quartile 279, max 282
Letter freqs G 33.8%, A 26.2%, T 21.2%, C 18.8%
0% masked (lower-case)
ASCII_BASE=33
EE mean 7.7; min 0.1, lo_quartile 0.9, median 2.9, hi_quartile 8.9, max 161.6

Result for 16S-mockS.1_V4_R2-hdr.log

saguaro:16s_QC aponsero$ ./usearch -fastx_info $FILE1 -output $OUT1
File size 7.0M, 11.2k seqs, 3.1M letters and quals
Lengths min 271, lo_quartile 275, median 276, hi_quartile 278, max 283
Letter freqs C 36.5%, T 25.5%, A 19.4%, G 18.7%
0% masked (lower-case)
ASCII_BASE=33
EE mean 34.1; min 0.7, lo_quartile 21.4, median 30.8, hi_quartile 42.7, max 176.0

Choose your QC parameters

Step 2.

The **fastq_eestats2** command creates a summary report showing how many reads will pass an expected error filter at different length thresholds. This will be useful for choosing parameters for fastq_filter. If you have paired reads, review the R1 and R2 files separately because they are usually different, e.g. the R2s tend to have lower quality.

To run this command, open your shell and run :

```
DIR=$PWD

FILE1='$DIR/16S-mockS.1_V4_R1-hdr.fastq'
OUT1='$DIR/fastq_info/16S-mockS.1_V4_R1-hdr-eestats2.txt'
./usearch usearch -fastq_eestats2 $FILE1 -output $OUT1

FILE2='$DIR/16S-mockS.1_V4_R2-hdr.fastq'
OUT2='$DIR/fastq_info/16S-mockS.1_V4_R2-hdr-eestats2.txt'
./usearch usearch -fastq_eestats2 $FILE2 -output $OUT2
```

This command has two optional options :

- **-length_cutoffs** : specifies three integers separated by commas giving the shortest length, longest length and length increment. An asterisk (*) indicates no upper length limit. Default is 50,*,50 which means that length cutoffs of 50, 100, 150 ... maximum read length will be used.
- **-ee_cutoffs** : which is given as one or more floating-point values separated by commas giving a list of the cutoffs to use. Default is 0.5,1.0,2.0. An asterisk (*) indicates that no e.e. cutoff should be applied, so all reads of at least the given length are included.

The command will generate two output files containing the reports.

EXPECTED RESULTS

Result for 16S-mockS.1_V4_R1-hdr.fastq, with the default parameters

11241 reads, max len 282, avg 277.5

Length	MaxEE 0.50	MaxEE 1.00	MaxEE 2.00
50	11030(98.1%)	11202(99.7%)	11237(100.0%)
100	10578(94.1%)	10989(97.8%)	11192(99.6%)
150	10082(89.7%)	10615(94.4%)	11020(98.0%)
200	9090(80.9%)	10048(89.4%)	10671(94.9%)
250	6152(54.7%)	8066(71.8%)	9472(84.3%)

Result for 16S-mockS.1_V4_R2-hdr.fastq, with the default parameters

11241 reads, max len 283, avg 276.4

Length	MaxEE 0.50	MaxEE 1.00	MaxEE 2.00
50	10816(96.2%)	11060(98.4%)	11192(99.6%)
100	10232(91.0%)	10712(95.3%)	11032(98.1%)
150	8680(77.2%)	9946(88.5%)	10676(95.0%)
200	4330(38.5%)	7227(64.3%)	9178(81.6%)
250	74(0.7%)	469(4.2%)	1634(14.5%)

Filter reads

Step 3.

The **-fastq_filter** command performs quality filtering and conversion of a FASTQ to a FASTA file.

This command has several output parameters :

-fastqout filename	FASTQ output file.
-fastaout filename	FASTA output file.
-fastqout_discarded filename	FASTQ output file for discarded reads.
-fastaout_discarded filename	FASTA output file for discarded reads.
-relabel prefix	Generate new labels for the output sequences. They will be labeled prefix1, prefix2 and so on. For example, if you use -relabel SampleA. then the labels will be SampleA.1, SampleA.2 etc.
-fastq_eeout	Append the expected numner of errors according to the Q scores to the label in the format 'ee=xx;'. Expected errors are calculated after truncation, if applicable.
-sample string	Append sample=string; to the read label

The command has also several filtering options :

-fastq_truncqual N	Truncate the read at the first position having quality score <= N, so that all remaining quality scores are >N.
-fastq_maxee E	Discard reads with > E total expected errors for all bases in the read after any truncation options have been applied.
-fastq_trunclen L	Truncate sequences at the L'th base. If the sequence is shorter than L, the read is discarded.
-fastq_minlen L	Discard sequences with < L letters.
-fastq_stripleft N	Delete the first N bases in the read.
-fastq_maxee_rate E	Discard reads with > E expected errors per base.Calculated after any truncation options have been applied. For example, with the fastq_maxee_rate option set to 0.01, then a read of length 100 will be discarded if the expected errors is >1, and a read of length 1,000 will be discarded if the expected errors is >10.
-fastq_maxns k	Discard if there are >k Ns in the read.

examples of use :

1. Simple conversion FASTQ to FASTA format :

```
./usearch -fastq_filter 16S-mockS.1_V4_R1-hdr.fastq -fastaout 16S-
mockS.1_V4_R1-hdr.fasta
```

2. Truncate to length 150, discard if expected errors > 0.5, and convert to FASTA:

```
./usearch -fastq_filter 16S-mockS.1_V4_R1-hdr.fastq -fastq_trunclen 150 -
fastq_maxee 0.5 -fastaout 16S-mockS.1_V4_R1-hdr.fasta
```

EXPECTED RESULTS

Results for the command

./usearch -fastq_filter 16S-mockS.1_V4_R1-hdr.fastq -fastq_trunclen 150 -fastq_maxee 0.5

00:00 1.9Mb FASTQ base 33 for file 16S-mockS.1_V4_R1-hdr.fastq
00:00 2.4Mb 100.0% Filtering, 89.7% passed
11241 Reads (11.2k)
0 Discarded reads length < 150
1159 Discarded reads with expected errs > 0.50
10082 Filtered reads (10.1k, 89.7%)

Result for the command

./usearch -fastq_filter 16S-mockS.1_V4_R2-hdr.fastq -fastq_trunclen 150 -fastq_maxee 0.5

00:00 1.9Mb FASTQ base 33 for file 16S-mockS.1_V4_R2-hdr.fastq
00:01 2.4Mb 100.0% Filtering, 77.2% passed
11241 Reads (11.2k)
0 Discarded reads length < 150
2561 Discarded reads with expected errs > 0.50
8680 Filtered reads (8680, 77.2%)