Microbial eukaryotic 18S tag-sequence processing/QC - V4 region (automating) Version 2

Sarah Hu

Abstract

This protocol provides step-by-step instructions for implementing a perl script to easily automate quality checking of raw tag sequences. This was specifically constructed for the numerous 18S diversity projects (which target the V4 hypervariable specific for evaluting protistan diversity).

It is highly recommended that you follow along this Github repo: https://github.com/shu251/QC_steps_V4_tagsequencing

Last updated Sept 30th, 2017

Citation: Sarah Hu Microbial eukaryotic 18S tag-sequence processing/QC - V4 region (automating). **protocols.io** dx.doi.org/10.17504/protocols.io.j5acq2e
Published: 30 Sep 2017

Before start

See https://github.com/shu251/QC_steps_V4_tagsequencing

Programs/files required:

* Database (preferrably with associated taxa list) to align sequences to. A version of the PR2 database formatted for QIIME can be found [here](https://drive.google.com/drive/folders/0Bxzw_UrYS4IAaEQ0b0lMQ0ZiUkU?usp=sharing)
* [QIIME](http://qiime.org/install/install.html) - v.1.9.1 or higher. Note as of Jan 2018 QIIME1 will not be supported. Future iterations of this protocol will use QIIME2.
* fastqjoin - this should install with QIIME. Alternatively you can use [PEAR](https://sco.h-its.org/exelixis/web/software/pear/doc.html#installing) to merge PE sequences.
* [Trimmomatic](http://www.usadellab.org/cms/?page=trimmomatic) - v.0.32
* [vsearch](https://github.com/torognes/vsearch) - v1.11.1
* abyss (optional) - see step 9
* Make sure you know your primer sequences and generate a fasta file: 'trimPE_V4primer.fasta'. In the below example I'm using [V4 Stoeck et al. 2010

primers](http://onlinelibrary.wiley.com.libproxy1.usc.edu/doi/10.1111/j.1365-294X.2009.04480.x/full/)
* Create a "prefix.txt" file to index sample fastq files.
* Run 'create.map.pl'

Protocol

Pre-processing

Step 1.

You can download test files and scripts described below from git.

cmd COMMAND

```
git clone https://github.com/shu251/QC_steps_V4_tagsequencing
```

Pre-processing

Step 2.

Input raw fastq files that are already demultiplexed.

This tutorial uses V4 primers (Stoeck et al. 2010) for microbial eukaryotes

First create a text file with a list of file prefixes (prefix.txt). Example:

File lists:

Test01_L001_R1_001.fastq

Test01_L001_R2_001.fastq

Test02_L001_R1_001.fastq

Test02_L001_R2_001.fastq

Associated prefixes:

Test01

Test02

LINK:
https://github.com/shu251/QC_steps_V4_tagsequencing

Create map file (for use in QIIME)

Step 3.

Map files in QIIME serve to de-multiplex reads (based on barcodes and indices) and are used in 'split_library' QIIME step. Here, I've split up the V4 forward primer to fulfill the necessary barcode / index space required in the QIIME mapping file. However, we aren't using this feature of the mapping file.

Use this QIIME command to check mapping file:

http://qiime.org/1.6.0/scripts/check_id_map.html

Run create.map.pl script to generate map files based on your prefix file list.

cmd COMMAND

```
#!/usr/bin/perl -w
open INFILE, "prefix.txt";
@prefix = ();
while (<INFILE>)
{ chomp;
  push(@prefix, $_);
}
close INFILE;

#create map file
#specific only for V4 primers, Stoeck et al.

for $i(@prefix)
{
   $filename = $i.'_map.txt';
   open OUTFILE, ">$filename";
   print OUTFILE '#',"SampleID\tBarcodeSequence\tLinkerPrimerSequence\tDescription\n";
   $i =~ s/_//g;
   print OUTFILE $i, "\tCCAG\tCASCYGCGGTAATTCC\t", $i, "\n";
   close OUTFILE;
}
```

Contents of create.map.pl

Obtain other required scripts and files

Step 4.

- Fasta file with listed forward and reverse primer. (

Perl script to QC each fastq file

Step 5.

Steps of script noted here:

(1) Import prefix file, named 'prefix.txt' and starting loop

(2) Run Trimmomatic to clip primers (see

(2) Merge paired end (R1 and R2) reads with a 20 bp overlap

(3) Perform split library command in QIIME, filter sequences with Q score 30 (-q 29)

(4) Clip primers. Allows sequences to have either forward or reverse primers

Note I am using Stoeck et al. 2010 V4 primers specifically for microbial eukaryotes

(5) Combine clipped sequences

(6) Move excess files to 'split_*' directories which were created during the split library step

(7) Length filter: seqlength_cutoff.pl [input.fasta] [min] [max] [output.fasta]

(8) Chimera check with vsearch (uchime) using a reference database

- You will need to acquire reference database that is best for your sample type. I am using the PR2 database. Alternatively, you can change the command to run vsearch chimera checking de novo. This will take longer.

(9) Move excess files (chimeras) to split directories

(10) Combine all reads together

Required programs:

- QIIME

- cutadapt

- vsearch

Required components:

- Forward and reverse primer sequences, I'm using V4 Stoeck et al. 2010 primers

- reference database

- seqlength_cutoff.pl (see script abo)

cmd COMMAND

```
#! /usr/bin/perl -w
open INFILE, "prefix.txt";
@prefix = ();
while (<INFILE>)
{ chomp;
  push(@prefix, $_);
}
```

```
close INFILE;
#(1) Import prefix file, named "prefix.txt" and starting loop

for $i(@prefix)
{
#(2) Merge paired end (R1 and R2) reads with a 20 bp overlap
print "join_paired_ends.py -f ",$i,"_L001_R1_001.fastq -r ",$i,"_L001_R2_001.fastq -
o joined_",$i," -j 20\n";
print "mv joined_",$i,"/fastqjoin.join.fastq ",$i,"_merged.fastq\n";

#(3) Perform split library command in QIIME, filter sequences with Q score 30 (-q 29)
print "split_libraries_fastq.py -i ",$i,"_merged.fastq -m ",$i,"_map.txt --
barcode_type 'not-barcoded' --sample_ids ",$i," -q 29 -n 0 -o split_",$i,"\n";
print "mv split_",$i,"/seqs.fna ",$i,".merged.Q30.fasta\n";

#(4) Clip primers. Allows sequences to have either forward or reverse primers
#(4.1) Searches for forward primer, discard seqs without those primers
print "cutadapt -g CCAGCASCYGCGGTAATTCC -O 3 --discard-untrimmed -m 10 -
o ",$i,".assembled.clipped.regF.fasta ",$i,".merged.Q30.fasta >> ",$i,".filter.log\n";

#Name discarded reads from 4.1 "discard_regF.fasta"
print "cutadapt -g CCAGCASCYGCGGTAATTCC -O 3 --untrimmed-output discard_regF.fasta -m 10 -
o tmp.fasta ",$i,".merged.Q30.fasta >> ",$i,".filter.log\n";
print "rm tmp.fasta\n"; #remove tmp (which is a duplicate)
#4.2 Check discarded for reverse primer
print "cutadapt -a TYRATCAAGAACGAAAGT -O 3 --discard-untrimmed -m 10 -
o ",$i,".assembled.clipped.regR.fasta discard_regF.fasta>> ",$i,".filter.log\n";

#(5) Combine clipped sequences
print "cat ",$i,".assembled.clipped.regF.fasta ",$i,".assembled.clipped.regR.fasta >> ",$i,
".assembled.clipped.fasta\n";

#(6) Move excess files to "split_*" directories which were created during the split library
 step
print "mv ",$i,"*reg* split_",$i,"/\n"; #move excess files from trimming to split file
print "mv ",$i,".filter.log split_",$i,"/\n";

# (7) Length filter: seqlength_cutoff.pl [input.fasta] [min] [max] [output.fasta]
print "seqlength_cutoff.pl ",$i,".assembled.clipped.fasta 150 500 ",$i,".assembled.clipped.
len.fasta\n";

#(8) Chimera check with vsearch (uchime) using a reference database
print "vsearch --uchime_ref ",$i,".assembled.clipped.len.fasta --
db /galadriel/sarah/PR2/pr2.qiime.fasta --uchimeout ",$i,".uchimeinfo_ref --
chimeras ",$i,".chimeras_ref.fasta --strand plus --
nonchimeras ",$i,".assembled.clipped.len.nc.final.fasta \n";

#(9) Move excess files (chimeras) to split directories
print "mv ",$i,".chimeras_ref.fasta split_",$i,"/\n"; #move excess chimera files to split d
ir
print "mv ",$i,".uchimeinfo_ref split_",$i,"/\n";

#(10) Combine all reads together
print "cat ",$i,".assembled.clipped.len.nc.final.fasta >> allseqs_test.fasta\n";

}
```

Perl script to automate running QC steps on paired end sequences. See step 4 directions for required components and programs.

Check all steps

Step 6.

Product from previous script is a bunch of intermediate sequence files and the final product with all sequences combined "allseqs_test.fasta" in this case.

Here, run these commands to get simple stats on each intermediate step. It is important to make sure a large percentage of sequences was not lost at a certain step - this can be indicative of processing or sequencing error. For example, if most of my samples lost >50% of their sequences following the "split_library" command (step 3 above), I would consider reducing the Q score threshold to something less conservative (perhaps 25?). In another example, if all of the sequences were removed (or >90%) during the sequence length cut off (step 7), this sample should be re-sequenced or thrown out.

cmd COMMAND

```
abyss-fac -s 0 *_merged.fastq >>stats_merged.txt
abyss-fac -s 0 *.merged.Q30.fasta >> stats_Q30.txt
abyss-fac -s 0 *.assembled.clipped.fasta >> stats_primerclipped.txt
abyss-fac -s 0 *.assembled.clipped.len.fasta >> stats_length_cutoff.txt
abyss-fac -s 0 *.assembled.clipped.len.nc.final.fasta >> stats_chimeras.txt
```

Check all steps

Step 7.

R script to compile all of the .txt files made in previous step.

Again, do a final check of all samples to make sure the QC was appropriate. For our lab, I see between 20-50% sequences lost per sample (from raw reads to post-chimera check). As long as the final sequence count is > 10,000 I keep the sample. Our threshold of at least 10,000 sequences (typically more) is very conservative.

Once this is done, you can clean up directory by removing any of these intermediate files.

cmd COMMAND

```
#Run R:
stats<-
c("stats_merged.txt", "stats_Q30.txt", "stats_primerclipped.txt", "stats_length_cutoff.txt"
, "stats_chimeras.txt")

for (file in stats){
  if (!exists("dataset")){
    dataset<-read.delim(file, header=TRUE, sep="\t",row.names=NULL)
    dataset<-dataset[c(2,4,8,10)]
    colnames(dataset)[1:4]<-c(file,"Min (bp)", "Max (bp)", "File name")
  }
  if (exists("dataset")){
    tmpdata<-read.delim(file, header=TRUE, sep="\t",row.names=NULL)
    tmpdata<-tmpdata[c(2,4,8,10)]
    colnames(tmpdata)[1:4]<-c(file,"Min (bp)", "Max (bp)", "File name")
    dataset<-cbind(dataset, tmpdata)
    rm(tmpdata)
  }
}
```

```
head(dataset)
write.csv(dataset, file="Seq_stats_QC.csv")
```

Run R script to compile a QC datasheet

OTU-clustering

Step 8.

Depending on the questions you want to ask your data, decide about what type of OTU clustering you want to do (both algorithm and percent similarity).

QIIME is a great resource for tutorials on OTU clustering -
http://qiime.org/tutorials/otu_picking.html#running-the-otu-picking-workflows

Analysis

Step 9.

For examples of preliminary analyses and figure generation in R see:
https://github.com/shu251/PreliminaryFigures_V4_tagseq