Illumina Fastq Filtering

Bryce Foster

Abstract

Method to filter Illumina sequence data by removing low quality reads and reads mapping to artifact and contamination databases using BBTools.

DNA sequencing produces low quality reads, reads that contain sequencing artifacts and contamination from lab processes. Removing the bad data from the fastq produces more accurate results for mapping to references and assembly.

BBTools is a bioinformatics software package developed at the DOE Joint Genome Institute and used worldwide.

http://bbtools.jgi.doe.gov

Citation: Bryce Foster Illumina Fastq Filtering. **protocols.io**
dx.doi.org/10.17504/protocols.io.gydbxs6
Published: 07 Jul 2017

Protocol

Reorder reads in fastq file (Clumpify)

Step 1.

Use clumpify to reorder the reads in the fastq file so that similar reads are near each other.

Clumpifying the fastq can reduce the file size by 30% depending on the amount of duplicated reads.

Clumpify only has this affect on gzipped and bzipped fastq files.

This step is optional.

$ clumpify.sh zl=4 reorder in=(raw fastq file) out=(temp fastq file)

- zl=4: gzip level for out file

- reorder: reorder reads based on similar kmers (pairs kept together)
- in: input fastq file
- out: temporary fastq file

Remove Sequencing Adapters

Step 2.

This command removes reads that match the JGI sequencing adapters database.

Commands:

$ bbduk.sh ktrim=r ordered minlen=51 minlenfraction=0.33 mink=11 tbo tpe rcomp=f k=23 ow=t hdist1=1 hdist2=1 ftm=5 zl=4 in=(temp fastq 1) out=(temp fastq 2) ref=(JGI sequencing adapters db.fa)

- ktrim = r: trim read to remove bases matching reference on the right of the read
- ordered: keep reads in the same order as the input fastq (temp fastq 1)
- minlen=51: keep reads with at least 51 bases
- minlenfraction=0.33: keep reads that are at least 33% of the original size
- mink=11: length of shortest kmer to use to identify kmer as matching the reference
- tbo: trim by overlap on (trim adapters based on where paired reads overlap)
- tpe: when right-trimming, trim both read of the minimum length of each other
- rcomp=f: do not look for reverse compliments of kmers
- k=23: kmer length used to find kmers matching 'ref'
- ow=t: overwrite 'out' file
- hdist1=1: hamming distance of 1, used to identify kmers as matching 'ref'
- hdist2=1: hamming distance of 1 for short kmers
- ftm=5: right trim length to be equal to module 'ftm'
- zl=4: zip level (gzip)
- in: input fastq file
- out: intermediate fastq file after this first step
- ref: JGI sequencing artifacts database [1]

Quality Filtering

Step 3.

This step removes reads and regions with low quality or match the JGI sequencing artifacts database [2].

$ bbduk.sh maq=5 trimq=10 qtrim=f ordered ow=t maxns=1 minlen=51 minlenfraction=0.33 k=25 hdist=1 zl=6 cf=t ref=(JGI sequencing artifacts db #2.fa) in=(temp fastq STEP 2) out=(temp fastq

STEP 3)

- maq=5: reads with average quality below 5 will be discarded (after trimming)
- trimq=10: regions with quality below 10 will be trimmed
- qtrim=f: trim read ends to remove bases with quality below trimq (f = do not do this)
- ordered: keep reads in the same order as the input fastq (temp fastq 2)
- ow=t: overwrite 'out' file
- maxns=1: remove reads with more than 1 N in the read
- minlen=51: keep reads with at least 51 bases
- minlenfraction=0.33: keep reads that are at least 33% of the original size
- k=25: kmer length used to find kmers matching 'ref'
- hdist1=1: hamming distance of 1, used to identify kmers as matching 'ref'
- zl=6: zip level
- cf=t: discard reads if the chastity filter = Y in the read header
- in: input fastq file from STEP 2 'out='
- out: intermediate fastq file for this STEP
- ref: JGI sequencing artifacts database [2]

Artifact Filtering

Step 4.

This step removes reads in the JGI's short sequencing artifacts database from the fastq from STEP 3

$ bbduk.sh ordered ow=t k=20 hdist=1 zl=6 in=(temp fastq3) out=(temp fastq4) ref=(JGI short sequencing artifacts db.fa)

- ordered: keep reads in the same order as the input fastq (temp fastq 3)
- ow=t: overwrite 'out' file
- k=20: kmer length used to find kmers matching 'ref'
- hdist=1: hamming distance of 1, used to identify kmers as matching 'ref'
- zl=6: zip level
- in: input fastq file from STEP 3 'out='
- out: intermediate fastq file
- ref: JGI short sequencing artifacts database [3]

Contamination Removal

Step 5.

Identify reads matching references in the JGI contamination databases using bbmap and remove them from the fastq file from STEP 4.

This step is optional.

$ filterbytaxa.sh names=(ncbi tax id) include=f tree=tree.taxtree.gz level=order ow=t in=(JGI contamination database) out=(temp taxa.fa.gz)

- this step is optional. It removes references matching the (ncbi tax id). If the sequence is E. Coli and E. Coli is in the contamination database then this step excludes it from being removed
- names: list of ncbi taxonomy ids to exclude from removing (e.g. cat = 9685)
- level=order: use references at the order level to exclude from removal (e.g. cat = 9685, order = Carnivora, anything in the Carnivora order will not be removed)
- tree=tree.taxtree.gz: taxonomy tree file
- include=f: discard filtered sequences
- ow=t: overwrite the out parameter
- in: JGI contamination database [4]
- out: output contamination database with 'names' removed, used for mapping

$ bbmap.sh ordered quickmatch k=13 idtag=t printunmappedcount ow=t qtrim=rl trimq=10 untrim build=1 ref=(temp taxa fa.gz) in=(temp fastq 4) out=(temp fastq 5)

- (temp taxa fa.gz) can be replaced by the JGI contamination database if the JGI contamination database did not need to be modified to exclude NCBI taxonomy ids
- ordered: keep the reads in the output fastq the same order as the input fastq
- quickmatch: generate cigar strings quickly
- k=13: kmer length used to find kmers matching 'ref'
- idtag=t: write tag indicating percent identity
- printunmappedcount: print the total number of unmapped reads and bases
- ow=t: overwrite the out parameter
- qtrim=rl: quality trim left and right ends of reads before mapping
- trimq=10: trim regions with average quality below 10
- untrim: undo trimming after mapping
- build=1: use a unique id for references indexed in the same directory
- ref: JGI contamination database or the (temp taxa.fa.gz) from the previous command
- in: input fastq from STEP 4
- out: intermediate fastq file

$ bbmap.sh ordered k=14 idtag=t usemodulo printunmappedcount ow=t qtrim=rl trimq=10 untrim kfilter=25 maxsites=1 tipsearch=0 minratio=.9 maxindel=3 minhits=2 bw=12 bwr=0.16 fast=true maxsites2=10 zl=8 in=(temp fastq 5) out=(final filtered fastq)

- ordered: keep the reads in the output fastq the same order as the input fastq
- k=14: kmer length used to find kmers matching 'ref'
- itag=t: write tag indicating percent identity
- usemodulo: throw away 80% of kmers based on remainder moduloa number to reduce RAM usage
- printunmappedcount: print the total number of unmapped reads and bases
- ow=t: overwrite the out parameter
- qtrim=rl: quality trim left and right ends of reads before mapping
- trimq=10: trim regions with average quality below 10
- untrim: undo trimming after mapping
- kfilter=25: potential mapping sites must have at least 25 consecutive exact matches
- maxsites=1: maximum number of total alignments to print per read

- tipsearch=0: do not look for read end deletions
- minratio=.9: minratio is 0.9
- maxindel=3: do not look for indels longer than this
- minhits=2: minimum number of seed hits required for candidate sites
- bw=2: restrict bandwidth ratio to 2
- bwr=0.16: restrict alignment band to this fraction of the read length
- fast=true: macro to set other parameters to run faster at reduced sensitivity
- maxsites2=10: do not analze more than this many alignments per read
- zl=8: gzip compression level
- in: input fastq from previous command (temp fastq 5)