

Metagenomics Processing

Generate and review quality control report on reads:

```
forward='/srv/data/mg/projects/LCY/short_reads/LC_H08_080105_Bio5slurp_B1.forward.fastq.gz'  
reverse='/srv/data/mg/projects/LCY/short_reads/LC_H08_080105_Bio5slurp_B1.reverse.fastq.gz'
```

```
srun fastqc --noextract --outdir readstats_preqc ${forward} ${reverse}
```

Trim adapters from reads:

```
srun --cpus-per-task ${cpus} bbdduk.sh -Xmx10g zipllevel=9 threads=${cpus} qin=33 interleaved=t  
ref=/srv/databases/contaminants/truseq_adapters.fa  
in1=data/LC_H08_080105_Bio5slurp_B1.forward.fastq.gz  
in2=data/LC_H08_080105_Bio5slurp_B1.reverse.fastq.gz out=${sample}.interleaved.atrim.fq.gz  
stats=${sample}.adapt_stats.txt ftm=5 ktrim=r k=23 mink=9 rcomp=t hdist=2 tbo tpe minlength=0  
2>${sample}.adapters.log &
```

Discard contaminant reads:

```
mkdir ${qc}/discarded
```

```
srun --cpus-per-task ${cpus} bbdduk.sh -Xmx10g threads=${cpus} qin=33 interleaved=t  
ref=/srv/databases/contaminants/phix174.fa.gz in=${sample}.interleaved.atrim.fq.gz  
out=${sample}.interleaved.atrim.decontam.fq.gz outm=${sample}.phix.fq.gz k=31 hdist=1 mcf=0.9  
stats=${sample}.phix_stats.txt 2>${sample}.phix.log &
```

* DNA from the genome of bacteriophage PhiX174 is often used as a spike-in control during Illumina sequencing runs and should be removed when present.

Estimate the metagenome's average coverage and compute an accumulation curve:

```
srun readstats.py --csv --output ${sample}.atrim.decontam.readstats.tsv  
${sample}.interleaved.atrim.decontam.fq.gz 2>/dev/null &
```

```
Seqs in sample = 359483250
```

```
nreads=$(expr $(tail -n 1 ${sample}.atrim.decontam.readstats.tsv | awk -F "\""'"' '{ print $2 }') / 20)
```

```
minlength=75
```

```
echo $nreads  
17974162
```

```
srun sample-reads-randomly.py --num_reads ${nreads} --output /dev/stdout  
${sample}.interleaved.atrim.decontam.fq.gz 2>/dev/null | srun qtrim --interleaved --qual-type 33 -o  
${sample}.forward.subset.fq -v ${sample}.reverse.subset.fq --trunc-n --min-len ${minlen} --leading 20 --  
trailing 20 --sliding-window 4:20 - 2>${sample}.subset.qtrim.log &
```

```
srn --cpus-per-task ${cpus} nonpareil -t ${cpus} -f fastq -T alignment -s ${sample}.forward.subset.fq -b  
${sample}.forward.cov &
```

```
srn --cpus-per-task ${cpus} nonpareil -t ${cpus} -f fastq -T alignment -s ${sample}.reverse.subset.fq -b  
${sample}.reverse.cov &
```

```
srn --pty R  
library(Nonpareil)
```

```
> library(Nonpareil)  
> svg('H08.np_curve.svg', height=7, width=7)  
> ncurve <- Nonpareil.curve.batch(c('H08.forward.cov.npo', 'H08.reverse.cov.npo'),  
libnames=c('H08_forward', 'H08_reverse'))  
> Nonpareil.legend('bottomright')_reverse'))> dev.off()  
> dev.off()  
> ncurve[, 'LRstar']
```

```
H08_forward H08_reverse  
5100253858 5416681662
```

If LR* is less than the dataset size, a larger threshold (Phred ≥ 10) can be used during quality trimming.
(it's not)

```
Bp (from read stats interleaved)  
37545924676
```

Generate a base composition histogram for the adapter-trimmed reads:

```
srn --cpus-per-task ${cpus} bbdut.sh -Xmx10g threads=${cpus} qin=33 interleaved=t  
in=${sample}.interleaved.atrim.decontam.fq.gz bhist=${sample}.base_freq_dist.hist &
```

```
> library(ggplot2)  
> rlength <- 125  
> bhist <- data.frame(read.table("H08.base_freq_dist.hist", sep="\t", row.names=1), strand=rep(c("Forward",  
"Reverse"), times=c(rlength, rlength)), base=rep(0:(rlength-1), times=2))  
> colnames(bhist) <- c("A", "C", "G", "T", "N", "strand", "base")  
> bhist <- data.frame(bhist[,c("base", "strand")], stack(bhist, select=c("A", "C", "G", "T", "N")))  
> svg("H08.base_freq_dist.svg", height=6, width=9)  
> ggplot(bhist, aes(x=base, y=values, color=ind)) + geom_line() + facet_grid(~strand) + xlab("Base Position") +  
ylab("Frequency") + theme(legend.title=element_blank())  
> dev.off()
```

```
quit()
```

Tadpole with a subset of reads:

```
srn interleave-reads.py -o H08.interleaved.subset.fq H08.forward.subset.fq H08.reverse.subset.fq &
```

```
srn --cpus-per-task 3 tadpole.sh -Xmx20g threads=3 mode=contig interleaved=t minprob=0.8 k=31  
in=H08.interleaved.subset.fq out=H08.quick_assem.subset.fa &
```

Use for qtrim:

Lower qscore, lower window
qscore = between 2-5 (use 5)
window = 4

```
srunch --cpus-per-task ${cpus} bbmap.sh threads=${cpus} nodisk=t interleaved=t reads=100000  
in=${sample}.interleaved.subset.fq ref=${sample}.quick_assem.subset.fa  
mhist=${sample}.map_error_rates.hist & ick_assem.subset.fa mhist=${sample}.map_error_rates.hist &
```

```
srunch --cpus-per-task ${cpus} bbmap.sh threads=${cpus} nodisk=t interleaved=t reads=100000  
in=${sample}.interleaved.subset.fq ref=${sample}.quick_assem.subset.fa  
mhist=${sample}.map_error_rates.hist & ick_assem.subset.fa mhist=${sample}.map_error_rates.hist &
```

In R:

```
> mhist <- read.table("H08.map_error_rates.hist", sep="\t")  
> colnames(mhist) <- c("Base", "MatchForward", "SubForward", "DelForward", "InsForward",  
"NForward", "OtherForward", "MatchReverse", "SubReverse", "DelReverse", "InsReverse", "NReverse",  
"OtherReverse")  
> mhist <- data.frame(Base=mhist[, "Base"], stack(mhist, select=c("SubForward", "DelForward",  
"InsForward", "SubReverse", "DelReverse", "InsReverse")))  
> svg("H08.mapping_error_rates.svg", height=6, width=8)  
> ggplot(mhist, aes(x=Base, y=values, color=ind)) + geom_line() + xlab("Base Position") +  
ylab("Mapping Error Rate") + theme(legend.title=element_blank())  
> dev.off()
```

Trim reads based on quality score and filter by length:

qscore=5
window=4

```
srunch qtrim --qual-type 33 --interleaved -o ${sample}.interleaved.atrim.decontam.qtrim.fq.gz -s  
${sample}.singles.atrim.decontam.qtrim.fq.gz --min-len ${minlen} --leading ${qscore} --trailing  
${qscore} --sliding-window ${window}:${qscore} ${sample}.interleaved.atrim.decontam.fq.gz  
2> ${sample}.qtrim.log &
```

```
srunch filter_replicates --interleaved -o ${sample}.interleaved.atrim.decontam.qtrim.derep.fq.gz --log  
${sample}.interleaved.replicates.log.gz --prefix --rev-comp ${sample}.interleaved.atrim.decontam.qtrim.fq.gz 2>  
${sample}.interleaved.derep.log &
```

Check the quality of the remaining reads to see if additional quality control is needed:

```
srunch fastqc --noextract --outdir readstats_postqc ${sample}.interleaved.atrim.decontam.qtrim.derep.fq.gz
```

Assemble paired-end and single-end reads together:

```
srunch --cpus-per-task 4 megahit -t 4 --12 H08.interleaved.atrim.decontam.qtrim.derep.fq.gz -r  
H08.singles.atrim.decontam.qtrim.fq.gz --out-prefix H08 --out-dir H08_new.assembly --min-contig-len  
200 --k-min 27 &
```

Rename contigs for downstream analysis:

```
srunch anvi-script-reformat-fasta H08_assembly/H08.contigs.fa -o H08.renamed.fasta -l 0 --simplify-names  
--report-file contig_names_map.tsv
```

```
srun metaquast.py -o H08.assembly_stats --fast --max-ref-number 0 H08.renamed.fasta
```

```
srun bowtie2-build H08.renamed.fasta ${sample}.contigs &
```

```
srun bmap.sh -Xmx24g interleaved=t in=${sample}.interleaved.atrim.decontam.qtrim.derep.fq.gz  
out=/dev/null ihist=${sample}.insert_size.hist reads=100000 ref=H08.renamed.fasta
```

Mapping Reads:

```
ismean=151
```

```
issd=78
```

```
ismin=$(value=$(expr ${ismean} - ${issd} \* 3); if [[ ${value} -lt 0 ]]; then echo 0; else echo $value; fi)
```

```
ismax=$(expr ${ismean} + ${issd} \* 3)
```

```
srun --cpus-per-task ${cpus} bowtie2 --very-sensitive -I ${ismin} -X ${ismax} -x H08.contigs -p ${cpus} --  
interleaved ${sample}.interleaved.atrim.decontam.qtrim.derep.fq.gz 2>${sample}.mapping.log | srun samtools sort -  
| srun samtools view -b -h -F 4 -o ${sample}.mapped.sorted.filtered.bam - &
```

```
srun samtools index H08.mapped.sorted.filtered.bam
```