Development of a simple and streamlined bioinformatics pipeline to extract custom database sequences from next-generation sequencing data

Ebony Stretch[1], Erin O'Brien[1], Leanna Apodaca[1], Nicole DeNamur[2], and Elinne Becket[1†]

[1]College of Science and Mathematics, California State University San Marcos, CA
[2]University of California, Berkeley, CA
[†]Corresponding Author

ABSTRACT

The transfer of antibiotic resistance genes in microbial communities is of great importance to study because they affect many aspects of clinical and ecological health. Using next-generation sequencing (NGS) of microbial communities, one can investigate which antibiotic resistance genes occur in a given population. Currently, several bioinformatic pipelines exist that align DNA sequences to antibiotic gene databases and then utilize visualization tools for the analyses of that DNA sequence. However, these pipelines are often not user-friendly, are computationally intensive, and can be prone to sequence alignment limitations. Therefore, we sought to develop a user-friendly streamlined code that could rapidly extract antibiotic resistance genes from any reference database using basic terminal commands. To accomplish this, we initially developed a set of Linux-based commands by combining Trimmomatic, Seqtk, Bowtie2, SAMtools, BEDTools and GROOT, while utilizing the Comprehensive Antibiotic Resistance Database (CARD) as reference for NGS datasets. This resulted in rapid and effective analyses of antibiotic resistance genes from Southern California coastal water samples. The pipeline was further improved by piping multiple steps into one single command, strongly reducing large datafile outputs and hard disk space. The resulting pipeline can analyze the antibiotic resistance profiles from full metagenome samples on a standard laptop computer. We also included the option to run multiple samples in a single line of code, which can easily be utilized on personal machines as well. Additionally, this pipeline has been used for RNA-seq analyses in a classroom setting, demonstrating flexibility in its application. Future goals include compiling these codes into a single executable file for any novice researcher to use in analyzing next-generation sequencing data. The completed bioinformatics pipeline may be used to produce comprehensive antibiotic resistome gene profiles from a personal computer, and it can be customized to use with any desired database.

BACKGROUND

Next-generation sequencing bioinformatic pipelines are often:

- Not novice-friendly,
- Computationally intensive, and
- Prone to sequence alignment limitations

GOAL

Generate a streamlined pipeline, accessible to non-experts, that rapidly extracts genes/genetic regions of interest from any reference database using basic terminal commands on a personal computer.

PIPELINE REQUIREMENTS

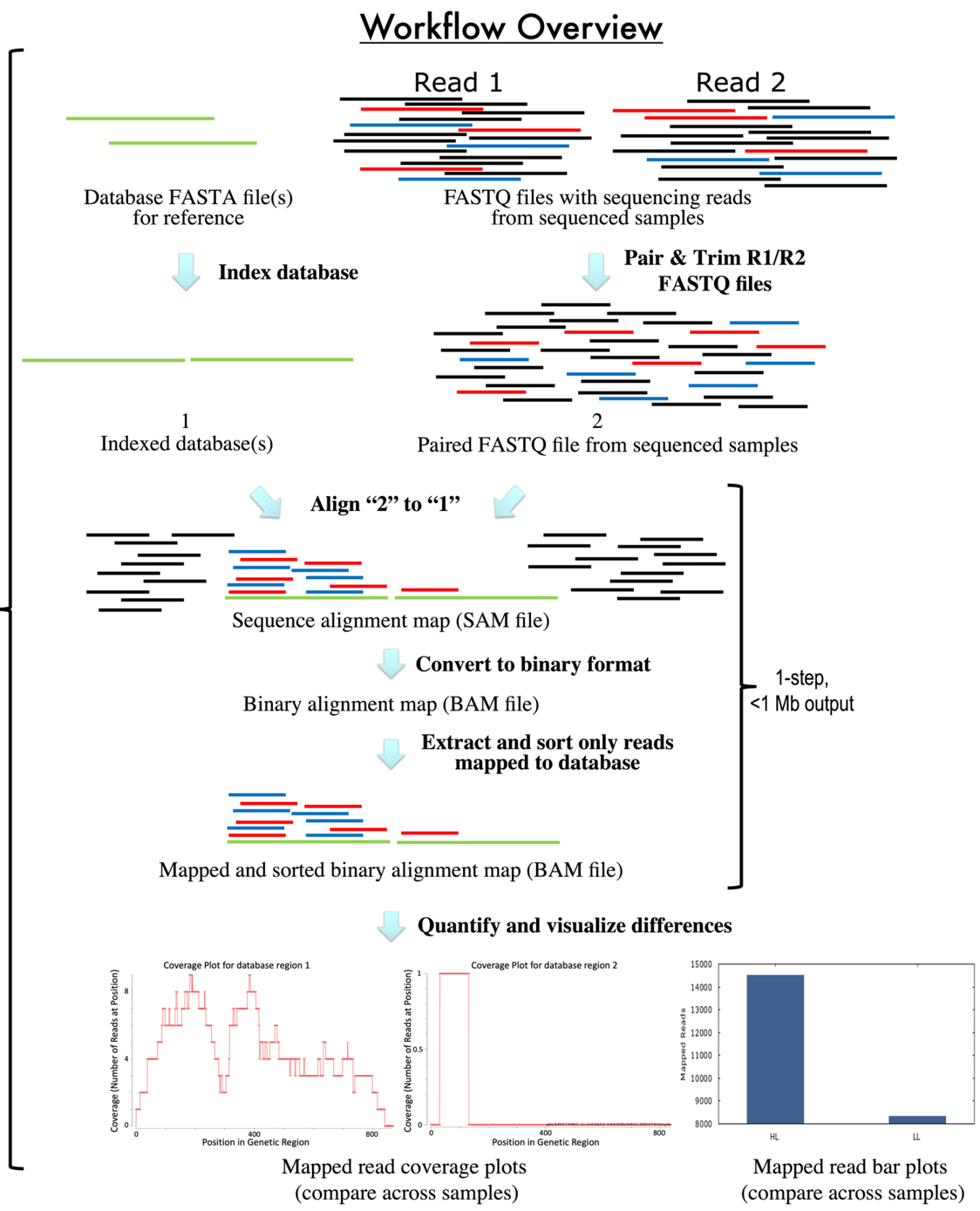
Dependencies: Trimmomatic, Seqtk, FastQC, Bowtie2, SAMtools, BEDtools, GROOT

Database Requirements: FASTA database of choice, one or more files

Hard disk Requirements: 2x the size of each sample fastq file + database

Hardware Requirements: Personal laptop/desktop

Illustration of the DATAlign Workflow



Mapped read coverage plots (compare across samples)

Mapped read bar plots (compare across samples)

Command Line Code

```
######### DATAlign Pipeline

######### Take all filenames ending in .fq.gz to make filenames reference list for all subsequent codes
ls *.fq.gz | cut -f 1 -d '.' > filenames
ls *.fq.gz | sed 's/ _R1.*//' | sed 's/_R2.*//' | uniq > filesname_trim

######### Build and index desired database using bowtie2
bowtie2-build CARD_mobile_merged.fasta ARG_db

######### Prepare fastq files

#Trim/filter fastq files based on quality (Phred score 25) and length (149 nt) – For paired-end reads
while read p; do trimmomatic PE -phred33 "$p"_R1.fq.gz "$p"_R2.fq.gz "$p"_R1_paired.fq.gz
"$p"_R1_unpaired.fq.gz "$p"_R2_paired.fq.gz "$p"_R2_unpaired.fq.gz ILLUMINACLIP:TruSeq3-PE.fa:2:30:10
LEADING:25 TRAILING:25 SLIDINGWINDOW:4:15 MINLEN:149; done < filenames_trim

#QC trimmed fastq files
while read p; do fastqc "$p"_paired.fq.gz; done < filenames

######## Normalize read numbers from multiple fastq files

#Compute the number of reads in each fastq
while read p; do gunzip -c "$p"_paired.fq.gz | wc -l | awk '{print $1 / 4}' >> numreads; done < filenames

#Compute the minimum number of reads across all fastqs in the folder
minreads=$(cat numreads | sort -n | head -1)

#subsample fastq files
while read p; do seqtk sample -s100 "$p"_paired.fq.gz $minreads | gzip > "$p".sub.fastq.gz; done < filenames

######## Alignment pipeline

# first pipe: runs alignment of fastq to database, creates .sam file, extracts and outputs only mapped/aligned reads
# second pipe: converts .sam to .bam
# third pipe: sorts the bam file in order of database sequence appearance
while read p; do bowtie2 --threads 2 -x ARG_db -1 "$p"_R1.sub.fastq.gz -2 "$p"_R2.sub.fastq.gz --no-unal |
samtools view -bS - | samtools sort -o - > "$p"_ARG.bam; done < filenames_trim

######## create comparison plots of mapped reads

## Generate groot report and coverage plots
groot report -i []sorted.bam -c 0.1 --plotCov > [].report

# create table of total mapped reads in each sample
while read p; do mapped=$(samtools view -c -F 4 ${p}_ARG.bam); printf "$p\t$mapped\n"; done < filenames_trim
> ARG-mapped.tsv

# create barplot to compare TOTAL mapped reads across samples
gnuplot -e "set terminal png size 640,300; set output 'ARG_mapped_read_barplot.png'; set boxwidth 0.4; set style
fill solid 1.00; set terminal png nocrop enhanced font 'verdana,10' size 640,300; set nokey; set title 'ARG Mapped
Reads'; set ylabel 'Mapped Reads'; plot 'ARG-mapped.tsv' using 2:xticlabels(1) with boxes lt rgb '#406090'"
```

CONCLUSIONS

- Generated streamlined alignment code which is flexible to database of choice and can be run locally
- Successfully run on shotgun metagenomics and RNA-seq data against regions of interest
- Accessible to students/non-coders
- Can be run on a personal laptop computer (OSX/Linux)
- <10 minutes per sample for most files, can be run in a laboratory classroom

FUTURE DIRECTIONS

- Remove GROOT report dependency – custom data visualization
- Use HMM Meta Mark instead of Bowtie2 for better sensitivity
- Add rRNA removal step for RNA-seq normalization
- "Black box" the code to allow custom input -> output in single step

References:

Rowe WPM, Winn MD. Indexed variation graphs for efficient and accurate resistomes profiling. Bioinformatics. 2018. doi: bty387
Biller SJ, et al. Marine microbial metagenomes sampled across space and time. Sci Data. 2018. doi: 10.1038/sdata.2018.176

Acknowledgements:

We would like the thank Dr. Sethuraman, Mellissa Lynch, Dr. Miller, and Dr. Pfeiffer for their guidance and help