

Pipeline for metagenomic sequence data processing:

Sequence QC:

FASTQ sequences were pre-processed with KneadData (v0.12.0) prior to taxonomic classification. The steps in KneadData are:

- 1) Use Trimmomatic¹ to trim Illumina adapters, low-quality bases (using a sliding window of 4, and clipping the read if the average window Phred score drops below 20), and filter resulting reads if their length is <50% of the input read length
- 2) Remove repetitive sequences using Tandem Repeats Finder (TRF)²
- 3) Remove reads from human DNA using bowtie2³, searching against a database of human DNA (T2T-CHM13v2.0)

KneadData was run on each of the paired end samples separately (option: --unpaired), using a bowtie2 database built from the T2T-CHM13v2.0 human reference genome, and otherwise was run with default settings:

```
kneaddata --unpaired $DATA_DIR/*S${SGE_TASK_ID}_R1*.fastq.gz --reference-db $DB_DIR --output $INT_OUTPUT_DIR --threads $NSLOTS
```

```
kneaddata --unpaired $DATA_DIR/*S${SGE_TASK_ID}_R2*.fastq.gz --reference-db $DB_DIR --output $INT_OUTPUT_DIR --threads $NSLOTS
```

FASTQC reports are generated before and after pre-processing with KneadData to assess the sequence quality of samples. If the FASTQC reports before pre-processing show the presence of overrepresented sequences in samples, KneadData is provided an additional option to trim the overrepresented sequences (--run-trim-repetitive).

K-mer-based taxonomic classification: Kraken2^{4,5} is a tool for taxonomic classification of shotgun metagenomic sequencing data using a k-mer-based approach. The classifier assigns taxonomic labels to each sequencing read by querying k-mer minimizer sequences within a read against a database. A custom database is built from complete genomes in RefSeq for the bacterial, archaeal, viral, and plasmid domains, along with the human T2T-CHM13v2.0 genome, and collections of eukaryotic pathogens (EuPathDB) and known vectors (UniVec_Core) (downloaded and built September 29, 2023).

1. Download taxonomy

```
kraken2-build --download-taxonomy --db $DBNAME
```

2. Download RefSeq complete genome libraries

```
kraken2-build --threads 28 --download-library archaea --db $DBNAME
```

```
kraken2-build --threads 28 --download-library bacteria --db $DBNAME
```

```
kraken2-build --threads 28 --download-library plasmid --db $DBNAME
```

```
kraken2-build --threads 28 --download-library viral --db $DBNAME
```

```
kraken2-build --threads 28 --download-library human --db $DBNAME
```

```
kraken2-build --threads 28 --download-library UniVec_Core --db $DBNAME
```

3. Download T2T-CHM13v2.0 human reference sequence to be added to human reference

```
kraken2-build --add-to-library /restricted/projectnb/uh2-  
sebas/data/metagenomics/Kraken2-DB/Kraken2Uniq-DB-09222023/chm13v2.0.fa  
--db $DBNAME
```

4. Download EuPathDB sequences

```
curl -O -s http://ccb.jhu.edu/data/eupathDB/dl/eupathDB.tar.gz
```

5. Build database based on libraries

```
kraken2-build --threads 28 --build --db $DBNAME
```

Kraken2 v. 2.1.2 is run on each sample with paired-end fastq input files from KneadData output (\$SAMPLE_R1_001_kneaddata.fastq.gz) and with the Kraken2Uniq option to report the k-mer minimizer counts (--report_minimizer_data). Additional options are used to set a minimum number of overlapping k-mers sharing the same minimizer (--minimum-hit-groups) and report all taxa including those that have zero read counts (--report-zero-counts):

```
kraken2 --threads 16 --db $DBNAME \  
--output $SAMPLE.kraken.txt \  
--report $SAMPLE.aggregated.report.txt \  
--minimum-hit-groups 4 \  
--report-minimizer-data \  
--report-zero-counts \  
--use-names \  
--paired \  
--gzip-compressed $SAMPLE_R1_001_kneaddata.fastq.gz  
$SAMPLE_R2_001_kneaddata.fastq.gz
```

Kraken2 with the Kraken2Uniq option provides two outputs:

- 1) Standard output file (\$SAMPLE.kraken.txt) which indicates how each input sequence pair was classified including the taxa that each k-mer mapped back to, to determine the final taxon label.
- 2) Report file (\$SAMPLE.aggregated.report.txt) which identifies the taxa classified for the read sequences and the number of reads and k-mer minimizer sequences mapped to each taxon.

The Kraken2 report files are merged across all samples and stored in a single BIOM table using python package kraken-biom v. 1.0.1 (<https://pypi.org/project/kraken-biom/>). The BIOM table is imported into R using the phyloseq package⁶ to generate an OTU table of read counts with samples as columns and taxa as rows, and a taxonomic table of taxa names with each level of the taxonomy (e.g., kingdom to species) as columns and taxa as rows. A custom R script (02_kraken2uniq-generate-table.R) is used to merge the k-mer minimizer counts across samples into a table of k-mer minimizer counts with samples as columns and taxa as rows.

To estimate the taxonomic relative abundances at the species level for each sample, Bracken⁷ is applied to the Kraken2 classification report files to compute the estimated species abundances using a Bayesian algorithm. To reestimate the species abundances with Bracken, a kmer distribution file is generated to provide the expected number of kmers classified for each genome in the Kraken2 database. This is

performed in three steps with the Kraken2 database files (--db), kmer length (-k), and read length of the data (-l) provided as input.

1. Search all library input sequences against the database

```
kraken2 --db $DBNAME --threads 16 <( find -L $DBNAME/library \( -name
"*.fna" -o -name "*.fa" -o -name "*.fasta" \) -exec cat {} + ) >
$DBNAME/database.kraken
#rm ${DBNAME}/input.fasta
```

2. Compute classifications for each perfect read from library input sequences

```
$Bracken/kmer2read_distr --seqid2taxid ${DBNAME}/seqid2taxid.map \
--taxonomy ${DBNAME}/taxonomy \
--kraken ${DBNAME}/database.kraken \
--output ${DBNAME}/database${READ_LEN}mers.kraken \
-k ${KMER_LEN} -l ${READ_LEN} -t 16
```

3. Generate the kmer distribution file

```
$Bracken/generate_kmer_distribution.py -i
${DBNAME}/database${READ_LEN}mers.kraken \
-o ${DBNAME}/database${READ_LEN}mers.kmer_distrib
```

Bracken v.2.9 is run on the Kraken2 report of each sample (\$SAMPLEARG.aggregated.report.txt) with input arguments kmer distribution file (-k), taxonomic classification level to estimate abundances (-l), threshold of reads to remove low abundance taxa (-t).

```
python ${Bracken}/est_abundance.py -i ${SAMPLEARG}.aggregated.report.txt \
-k ${DBNAME}/database${READ_LEN}mers.kmer_distrib \
-l ${CLASSIFICATION_LVL} \
-t ${THRESHOLD} \
-o ${SAMPLEARG}.bracken.species.txt
```

The output of this step is one Bracken output report for each sample (\$SAMPLEARG.bracken.species.txt) which provides the identified taxa classified for the read sequences with the number of additional reads mapped to each taxon and the total number of reads and relative abundances based on the additional reads.

Scripts from KrakenTools⁵ are implemented to filter the Bracken reports to include particular species of interest and remove species due to host contamination such as Homo sapiens (NCBI ID 9606).

```
python ${BRACKEN}/filter_bracken.out.py -i ${SAMPLEARG}.bracken.species.txt \
-o ${SAMPLEARG}.bracken.species.filtered.txt \
--include ${TAXID} \
--exclude 9606
```

In addition, the Bracken reports are merged across all samples into a txt file for downstream analyses (merged.bracken.species.filtered.txt).

```
python ${BRACKEN}/combine_bracken_outputs.py --files ${BRACKEN_FILES} \
--output merged.bracken.species.filtered.txt
```

Marker-gene-based taxonomic classification:

MetaPhlAn4⁸ is a tool for taxonomic characterization of shotgun metagenomic sequencing data using a curated marker gene database (vJun23). Paired-end fastq input files, the output from KneadData, are concatenated (cat read1.fastq read2.fastq > out.fastq) and run with MetaPhlAn 4 v.4.1. The following options are used:

```
--nproc 16
--bowtie2db $DB_DIR
--input_type fastq
-o $OUTPUT_FILE
```

The output of this step is one file per sample with relative abundance of taxa that sums to 1 at each level of taxonomy (e.g. species-genome-bin (SGB), genus, family). These outputs are merged into a single file that represents samples as columns and taxa as rows using the utility script within MetaPhlAn4 merge_metaphlan_tables.py

References

1. Bolger, A. M., Lohse, M. & Usadel, B. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* **30**, 2114–2120 (2014).
2. Benson, G. Tandem repeats finder: a program to analyze DNA sequences. *Nucleic acids research* **27**, 573–580 (1999).
3. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nature methods* **9**, 357–359 (2012).
4. Wood, D. E., Lu, J. & Langmead, B. Improved metagenomic analysis with Kraken 2. *Genome Biology* **20**, 1–13 (2019).
5. Lu, J. *et al.* Metagenome analysis using the Kraken software suite. *Nature protocols* **17**, 2815–2839 (2022).
6. McMurdie, P. J. & Holmes, S. phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. *PLoS ONE* **8**, e61217 (2013).
7. Lu, J., Breitwieser, F. P., Thielen, P. & Salzberg, S. L. Bracken: estimating species abundance in metagenomics data. *PeerJ Computer Science* **3**, e104 (2017).
8. Blanco-Míguez, A. *et al.* Extending and improving metagenomic taxonomic profiling with uncharacterized species using MetaPhlAn 4. *Nature Biotechnology* 1–12 (2023).