# Supplemental

## 1 Proof of equivalence to harmonic number

Equation 2 stated this relationship between our recursive formula for $\mathbf{E}[S_n]$ and the harmonic number.

$$\text{Simple Random Model: } \mathbf{E}[S_n] = 1 + \frac{1}{n}\sum_{j=0}^{n-1}\mathbf{E}[S_j]$$

$$\text{Harmonic Series Sum: } H_n = \sum_{i=1}^{n}\frac{1}{i}$$

We now show that these are equivalent using induction.

*Proof.* Base case: Letting $n = 1$ for both equations:

$$\mathbf{E}[S_1] = 1 + \frac{1}{1}(0) = 1$$
$$H_1 = \frac{1}{1} = 1\checkmark$$

Induction step: We now assume $\mathbf{E}[S_n] = H_n$ and show that $\mathbf{E}[S_{n+1}] = H_{n+1}$.

$$\mathbf{E}[S_n] = 1 + \frac{1}{n}\sum_{j=0}^{n-1}\mathbf{E}[S_j]$$
$$\mathbf{E}[S_{n+1}] = 1 + \frac{1}{n+1}\sum_{j=0}^{n}\mathbf{E}[S_j]$$

Now we rewrite $\mathbf{E}[S_{n+1}]$ in terms of $\mathbf{E}[S_n]$:

$$\begin{aligned}
\mathbf{E}[S_{n+1}] &= 1 + \frac{1}{n+1}\sum_{j=0}^{n}\mathbf{E}[S_j] \\
&= \left(1 + \frac{1}{n}\sum_{j=0}^{n-1}\mathbf{E}[S_j]\right)\frac{n}{n+1} - \frac{n}{n+1} + 1 + \frac{1}{n+1}\mathbf{E}[S_n] \\
&= \frac{n}{n+1}\mathbf{E}[S_n] + \frac{1}{n+1}\mathbf{E}[S_n] - \frac{n}{n+1} + 1 \\
&= \mathbf{E}[S_n] + \frac{-n+n+1}{n+1} \\
&= \mathbf{E}[S_n] + \frac{1}{n+1}
\end{aligned}$$

Now apply the assumption of $S_n = H_n$.

$$\begin{aligned}
\mathbf{E}[S_{n+1}] &= \mathbf{E}[S_n] + \frac{1}{n+1} \\
&= \sum_{i=1}^{n} \frac{1}{i} + \frac{1}{n+1} \\
&= \sum_{i=1}^{n+1} \frac{1}{i} \\
\mathbf{E}[S_{n+1}] &= H_{n+1} \ \checkmark
\end{aligned}$$

□

## 2 Proof of membership in $\Theta(\log d)$

In the main manuscript, we proved the expectation of our recurrence sum for a simple random model is equivalent to the harmonic series sum ($\mathbf{E}[S_n] = H_n$). Next, we want to show that $H_n \in \Theta(\log n)$ in order to prove the average space complexity for the cliff-compressed document array profiles.

*Proof.* Using Stolz–Cesàro's rule (L'Hôpital's rule for sequences), we can show that $H_n \in \Theta(\log n)$ by proving that as $n \to \infty$ the limit of the ratio of the functions is equivalent to a constant. If so, that means we can identify constants $c_1, c_2$ such that $c_1 \log n \leq H_n \leq c_2 \log n$ thereby proving that $H_n \in \Theta(\log n)$.

$$\begin{aligned}
&\lim_{n\to\infty} \frac{H_n}{\log n} = \lim_{n\to\infty} \frac{H_{n+1} - H_n}{\log(n+1) - \log(n)} = \lim_{n\to\infty} \frac{1/(n+1)}{\log(1+1/n)} \\
&= \lim_{n\to\infty} \frac{1}{\log(1+1/n)^n + \log(1+1/n)^1} = \lim_{n\to\infty} \frac{1}{\log e + \log(1+1/n)^1}
\end{aligned}$$

Here, we know that as $n$ approaches $\infty$ the expression $\log(1+1/n)$ will approach $\log(1)$ which is equivalent to 0.

$$= \lim_{n\to\infty} \frac{1}{\log e + 0} = \lim_{n\to\infty} \frac{1}{1} = 1$$

□

Therefore, we have shown that $H_n \in \Theta(\log n)$, and based off of the result from the previous section (1), we have also shown that $\mathbf{E}[S_n] \in \Theta(\log n)$

## 3 Running Kraken 2 and Braken

Here are command used to build both the Kraken 2 and Bracken indexes for the SILVA database.

```
kraken2-build --db silva --special silva --threads 1
bracken-build -d silva -t 1 -l 250 -k 35
```

Here are the commands used to run the Kraken 2 query followed by using Bracken to do genus abundance estimation.

```
kraken2 --db silva
        --threads 1
        --report output.kreport2
        --paired mate_1.fastq mate_1.fastq
```

```
        > output.kraken2
bracken -d silva
        -r 250
        -l G
        -i output.kreport2
        -o output.bracken
```

# 4 MicrobeMixer: Simulating realistic 16S rRNA reads

The code to use `MicrobeMixer` can be found at `https://github.com/oma219/MicrobeMixer`. There are two main steps; the first step is to summarize public data to identify the top 100 genera for a particular biome and the second step is to simulate reads for that biome using `art_illumina` [Huang et al., 2012].

Here is an example of how to run `MicrobeMixer` to generate an abundance file that contains the top 100 genera found in human gut.

```
python3 sim_16s_reads.py stats
                  --biome root:Host-associated:Human:Digestive system
                  --taxonomy ../data/tax_slv_ssu_138.1.txt
                  --output abundance.tsv
```

The previous code generates a file called `abundance.tsv`. Next we will use that file to generate simulate reads from those genera.

```
python3 sim_16s_reads.py simulate
                 --biome-abundance abundance.tsv
                 --silva-ref ../data/SILVA_138.1_SSURef_NR99_tax_silva.fasta
                 --silva-taxonomy ../data/tax_slv_ssu_138.1.txt
                 --primers ../data/V3_V4_primers.txt
                 --temp-dir <OUTPUT_DIR>
```

This command will generate two FASTQ files and file called `seqtax.txt` that contains the mapping for each genus id (1 to 100) to the taxonomic path for that genus. Here is the command that `MicrobeMixer` uses to simulate reads using `art_illumina`.

```
art_illumina -ss MSv1 -amp -p -na -i genus_{genus_num}_seqs.fna -l 250
             -f {coverage} -o genus_{genus_num}_reads_
```

Additionally, the primers used by `MicrobeMixer` to extract the hypervariable regions are list below which were same ones used by Almeida et al [Almeida et al., 2018]:

```
V1-V2 F: AGMGTTYGATYMTGGCTCAG        V4-V4 F: GTGCCAGCMGCCGCGGTAA
V1-V2 R: GCTGCCTCCCGTAGGAGT          V4-V4 R: GACTACHVGGGTATCTAATCC
V3-V4 F: CCTACGGGNGGCWGCAG           V4-V5 F: GTGCCAGCMGCCGCGGTAA
V3-V4 R: GACTACHVGGGTATCTAATCC       V4-V5 R: CCCGTCAATTCMTTTRAGT
```

# 5 Read-level classification results for additional datasets

Figures S1 and S2 show the read-level classification on the Human Gut and Soil datasets.

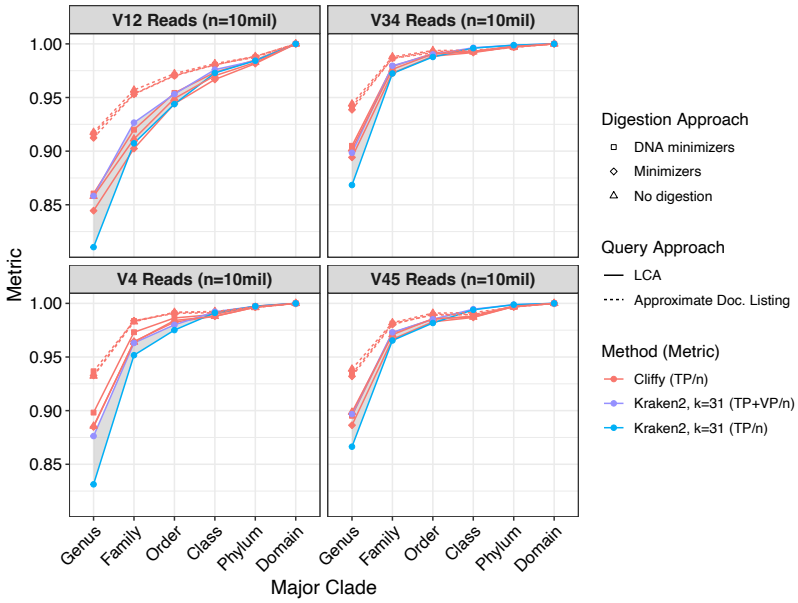


Figure S1: Read-level classification accuracy across taxonomic levels on Human Gut datasets. Read-level classification results across different levels of taxonomy on the Human Gut datasets corresponding to four different hypervariable regions. Each dataset consisted of 10 million 250-bp paired-end Illumina reads.

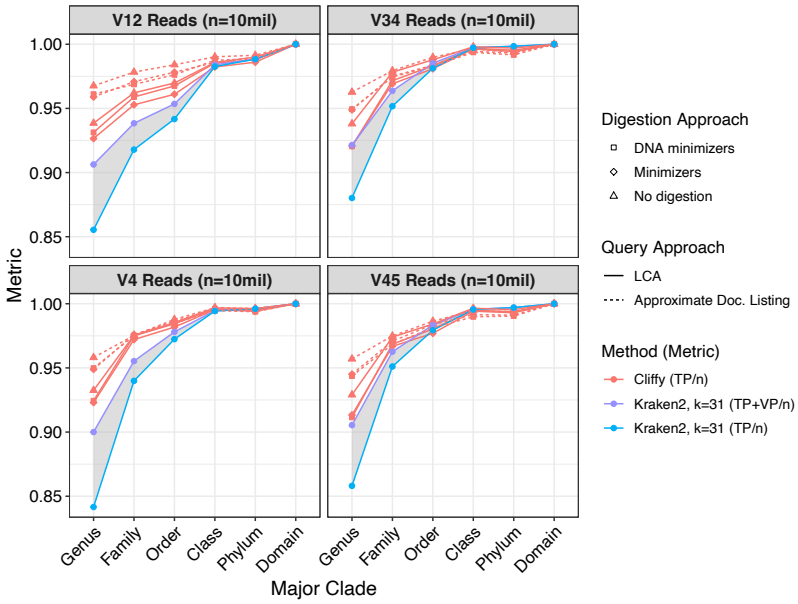


Figure S2: Read-level classification accuracy across taxonomic levels on Soil datasets. Read-level classification results across different levels of taxonomy on the Soil datasets corresponding to four different hypervariable regions. Each dataset consisted of 10 million 250-bp paired-end Illumina reads.

# 6 Genera abundance results for additional datasets

Figures S3 and S4 show the genera abundance estimation and Bray-Curtis distances on the Human Gut and Soil datasets.

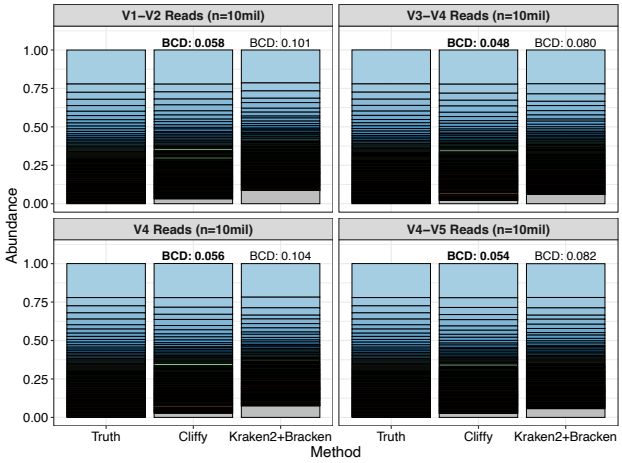


Figure S3: Genera abundance estimation for the Human Gut microbiome datasets. Genera abundance estimation for the 100 genera present in each of the simulated datasets for the Human Gut microbiome. The grey bar present in the Cliffy and Kraken 2 bars present the percentage of reads classified to genera not present in the true set. Each bar is annotated with the Bray-Curtis distance to the true distribution (lower means closer).

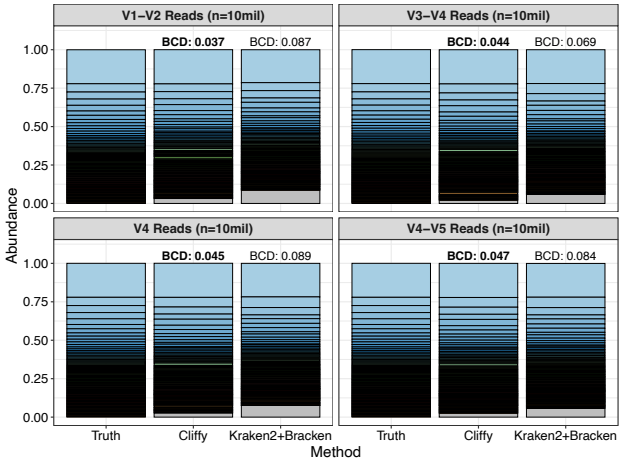


Figure S4: Genera abundance estimation for the Soil microbiome datasets. Genera abundance estimation for the 100 genera present in each of the simulated datasets for the Soil microbiome. The grey bar present in the Cliffy and Kraken 2 bars present the percentage of reads classified to genera not present in the true set. Each bar is annotated with the Bray-Curtis distance to the true distribution (lower means closer).

# 7 Read-level classification results when varying $k$ for Kraken 2

Figures S5, S6, S7 show Kraken 2's accuracy as we vary the length of the minimizer parameter used for classification.

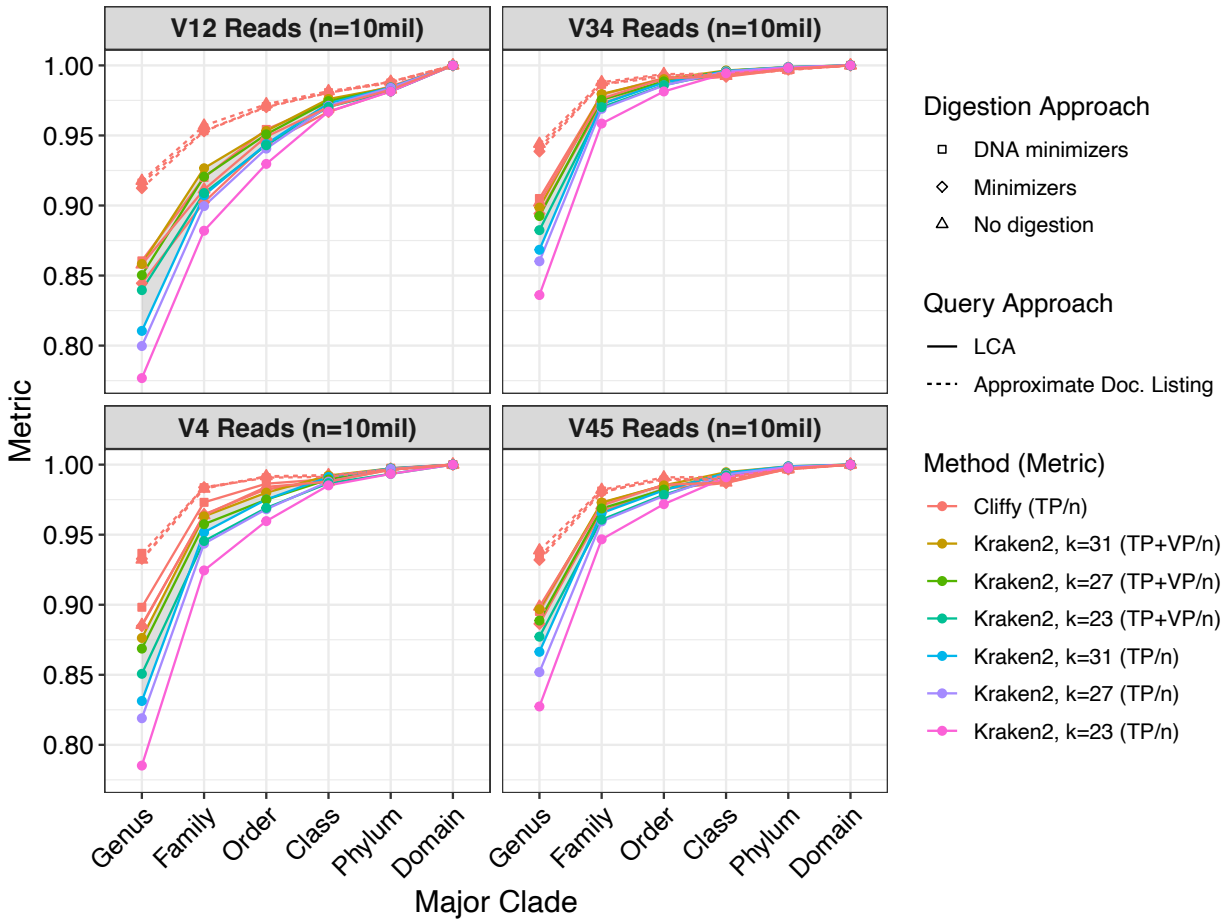


Figure S5: Read-level classification accuracy on the Human Gut dataset with varying $k$-mer sizes in Kraken 2. Read-level classification results across different levels of taxonomy on the Human-Gut dataset corresponding to four different hypervariable regions. Each dataset consisted of 10 million 250-bp paired-end Illumina reads. We tested out Kraken 2 using $k$-mer values of $k = 31$ (default), 27, 23 to see it would improve the accuracy, but instead we saw a downtrend as $k$ got smaller.

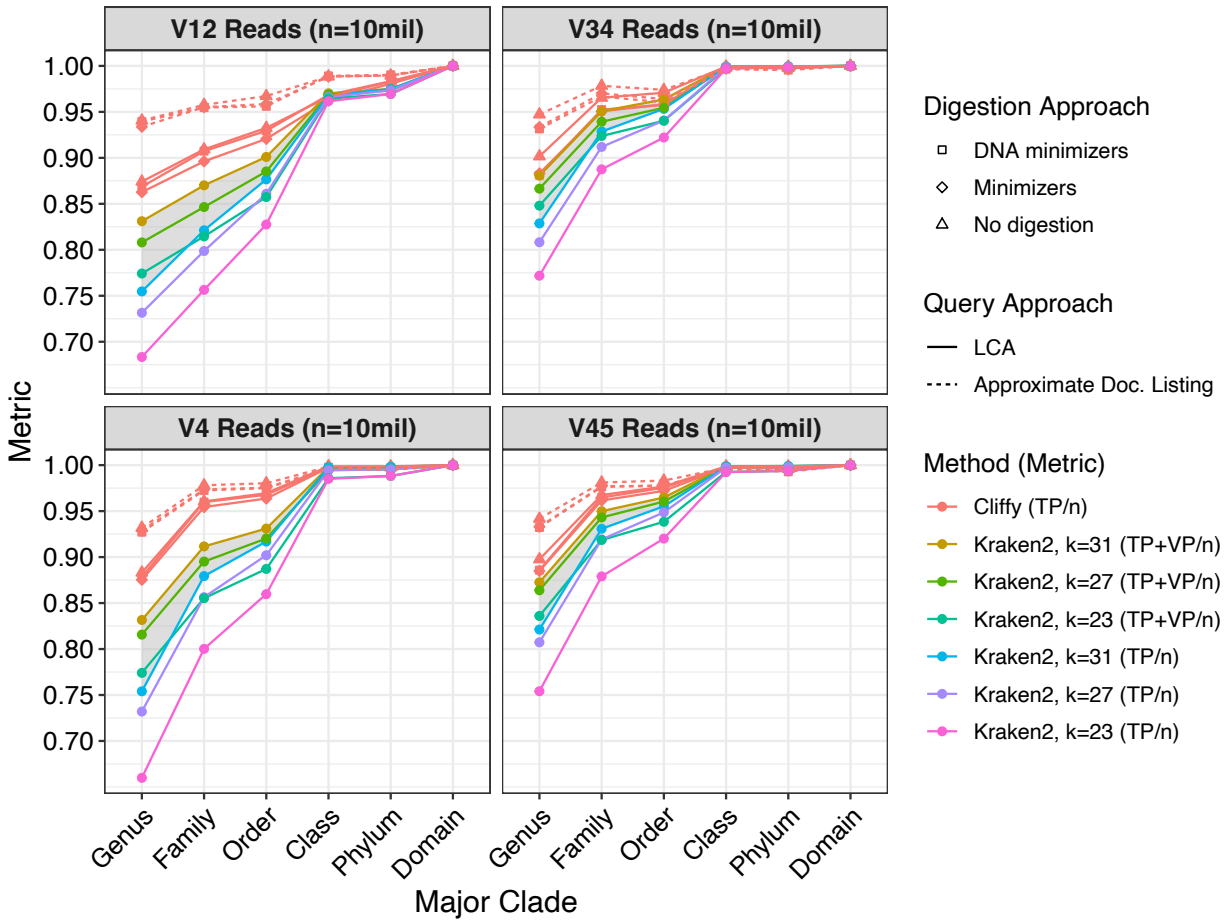


Figure S6: Read-level classification accuracy on the Aquatic dataset with varying $k$-mer sizes in Kraken 2. Read-level classification results across different levels of taxonomy on the Aquatic dataset corresponding to four different hypervariable regions. Each dataset consisted of 10 million 250-bp paired-end Illumina reads. We tested out Kraken 2 using $k$-mer values of $k = 31$ (default), 27, 23 to see it would improve the accuracy, but instead we saw a downtrend as $k$ got smaller.

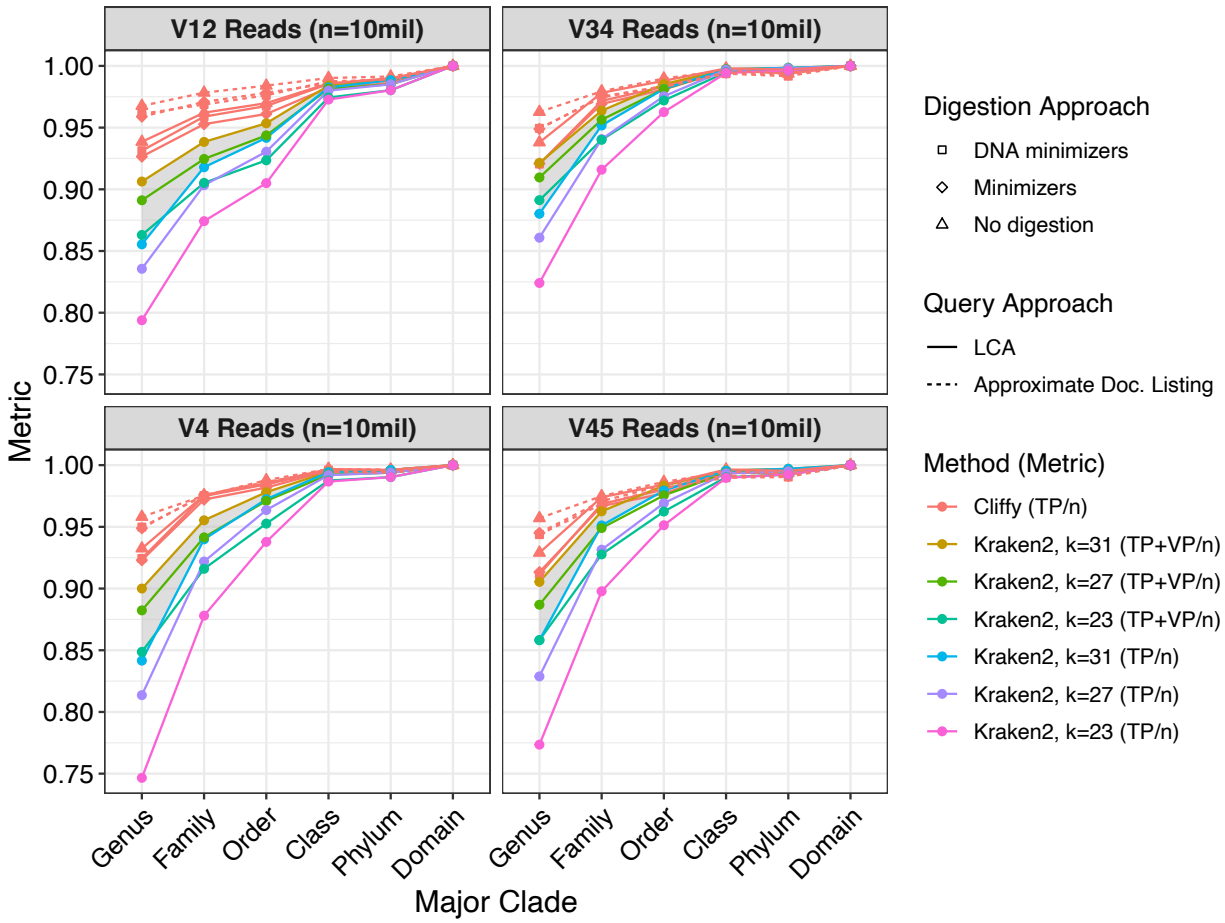


Figure S7: Read-level classification accuracy on the Soil dataset with varying $k$-mer sizes in Kraken 2. Read-level classification results across different levels of taxonomy on the Soil dataset corresponding to four different hypervariable regions. Each dataset consisted of 10 million 250-bp paired-end Illumina reads. We tested out Kraken 2 using $k$-mer values of $k = 31$ (default), 27, 23 to see it would improve the accuracy, but instead we saw a downtrend as $k$ got smaller.

# 8 Index scaling on whole-genome databases from GTDB

In order to test the scaling of Cliffy to larger input datasets, we decided to build indexes over subtrees of increasing size from the GTDB [Parks et al., 2020] metagenomic database. Table S1 shows the nine different input datasets that we constructed Cliffy indexes for where each index is represented by a root node.

| Index #: | Taxonomic Rank: | Name: | # of Genomes/Documents: | Reference Size (GB): |
|---|---|---|---|---|
| 1 | Species | Salmonella | 5 | 0.044 |
| 2 | Species | Escherichia | 11 | 0.099 |
| 3 | Species | Citrobacter | 18 | 0.170 |
| 4 | Species | Klebsiella | 24 | 0.248 |
| 5 | Species | Enterobacter | 49 | 0.441 |
| 6 | Family | Aeromonadaceae | 72 | 0.526 |
| 7 | Family | Pasteurellaceae | 198 | 0.798 |
| 8 | Family | Vibrionaceae | 348 | 3.090 |
| 9 | Family | Alteromonadaceae | 530 | 4.178 |

Table S1: Collections of whole genome sequences of increasing size selected from the GTDB (release 220) [Parks et al., 2020]. Each index represents a subtree of the GTDB taxonomy beginning at the species. In this experiment, we utilized the representative genomes curated by GTDB [Parks et al., 2020].

Figure S8 shows the temporary disk usage used by Cliffy during the index building step, as well as the final size of the index. We observed that Cliffy's temporary disk usage grows quickly as size of the index and number of document grows which makes it difficult currently to scale to whole-genome databases like GTDB. Furthermore, even with the cliff-compressed document array profiles, the index size can grow quite large for large reference sequences such as whole-genome databases. Future work will be needed to reduce the disk usage during the build phase as well as additional ideas to further compress the data-structure for large-scale inputs. Nevertheless, for smaller inputs such as 16S rRNA classification using the SILVA database, Cliffy provides a reasonably sized index with improved classification accuracy.

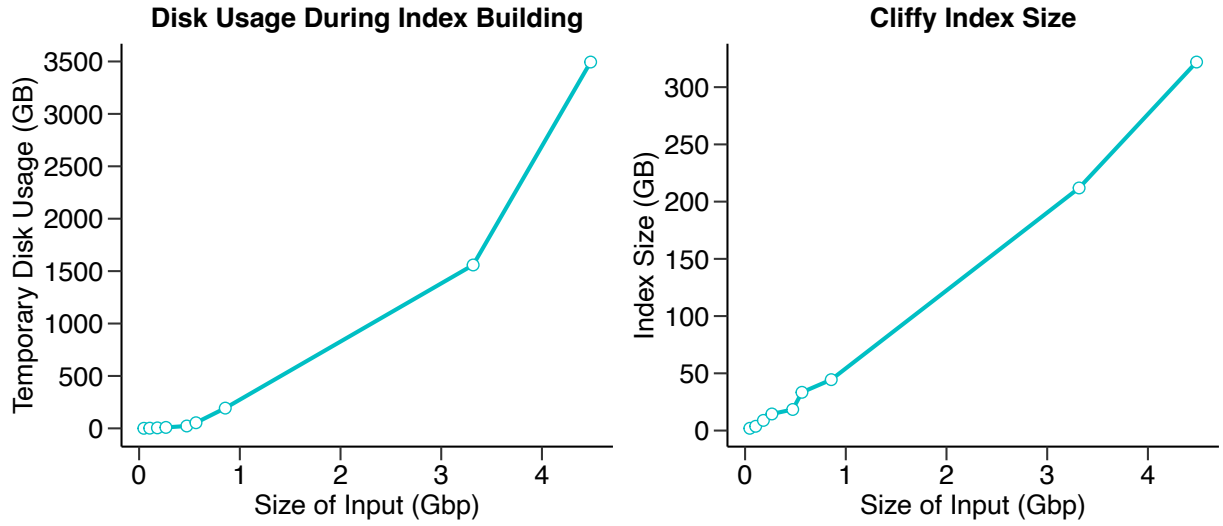


Figure S8: Cliffy index building statistics for GTDB whole-genome collections. Cliffy index building statistics for the collections of whole genome sequences from GTDB [Parks et al., 2020] described in Table S1. (a) Shows the temporary amount of disk space used during the index building phase, primarily used to store the uncompressed document array profiles prior to applying cliff compression. (b) Shows the final Cliffy index size for each of the whole-genome collections which contains the run-length encoded BWT and the cliff-compressed document array profiles.