IMPROVING MIN HASH FOR METAGENOMIC TAXONOMIC PROFILING

HOOMAN ZABETI[1], DAVID KOSLICKI[1*]

[1] *Mathematics Department, Oregon State University, Corvallis, OR.*

Abstract. Abstract here.

Keywords: *Min hash, k-mins sketch, metagenomics, taxonomic profiling, taxonomic classification, Jaccard index, containment.*

1. Introduction

(1) Min hash recently has been used to great success on biological data
(2) Mash, Sourmash, etc.
(3) originally designed for sets of relatively similar size and appreciable intersection size
(4) metagenomic taxonomic profiling the setup is different: many relatively small database entries, one very large metagenomic sample, very small intersection sizes in general
(5) we modify the min hash paradigm to this particular situation so it can handle a sample of much greater size than the reference database entries.

Min hash is great at comparing sets of similar size. When one set is much larger than the other, the Jaccard index is going to be smaller, which is precisely the case when you need a lot of hashes to get a good estimate (by the Chernoff bounds). In metagenomics, the typical paradigm is one/few very large set(s) (the metagenomic sample(s)) call it B and a bunch of small reference/database sets (call one A). Taking the classical min hash approach means sampling from $A \cup B$. Part a) of Figure 1 demonstrates such a situation while sampling 100 random points of $A \cup B$ and leads to 2 points lying in $A \cap B$. On the other hand, if we sample from just A (instead of $A \cup B$) and have some way to test if a point x is in $A \cap B$, we would get a much better estimate of $|A \cap B|$. Part b) of Figure 1 demonstrates this approach while sampling only 50 points from A and finds 26 points lying in $A \cap B$. They key to our approach is that the membership test $x \in A \cap B$ can be efficiently performed with a bloom filter of B. We analyze the time and space complexity, as well as the accuracy of this approach and find that for parameters typically used in metagenomics, our proposed approach is faster, uses less space, and is more accurate than the classical min hash approach.

2. Methods

Definitions, derivation of mathematical results here. *Need to carefully choose notation so we don't repeat the usage of X*

2.1. Definitions.

(1) Definitions of database entries, query sample, k-mer size, note size disparity
(2) k-mer size, genome = set of all k-mers from some genomes, give typical size. Sample is all k-mers from some large set of short reads, give typical size

Date: May 11, 2017.

* Corresponding Author: david.koslicki@math.oregonstate.edu.

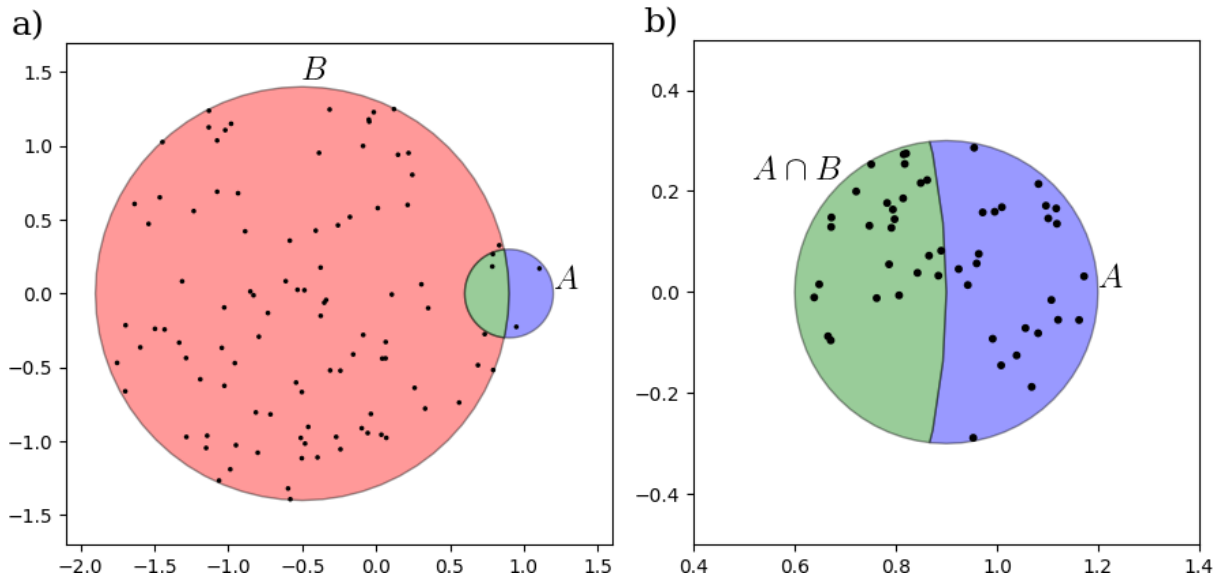

FIGURE 1. Conceptual comparison of classical min hash to the proposed containment approach when estimating the Jaccard index of very different sized sets. a) Sampling 100 points from $A \cup B$ (as is done in the classical min hash approach) leads to finding only 2 elements in $A \cap B$. b) Sampling just 50 points of A and testing if a point $x \in A \cap B$, finds 26 elements in $A \cap B$. This latter approach will be seen to lead to a better estimate of the Jaccard index.

2.2. **Classic min hash.** *describe classic approach.* Pick at random some number of hashes, compute minimum of said hash function on pair of sets, look for collision, prob. of collision is Jaccard. In the MinHash technique with different k hash functions, define

$$X_i = \begin{cases} 1 & h^{(i)}_{min}(A) = h^{(i)}_{min}(B) \\ 0 & o.w \end{cases}$$

Which give $E(X_i) = \dfrac{|A \cap B|}{|A \cup B|} = J(A, B)$. Hence for $X^k = \sum_{i=1}^{k} X_i$, the expectation is given by $E(X^k) = kJ(A, B)$.

2.3. **Min hash via containment.** *In this subsection, sketch the general idea of what we are doing.* Compute Bloom filter of larger set, randomly select some number of hashes, compute minimums of said hash function on the smaller set, check if minimum is in bloom filter. Prob(yes) is containment/coverage index (intersection over smaller set).

2.4. **Analytic comparison of classical min hash to containment min hash.**

2.4.1. *Chernoff bound estimates.*

Definition 2.1 (Give reference)**.** *Let X be a random variable such that $E(X) = \mu, \quad 0 < \delta < 1$. Then probability of relative error/deviation δ from mean μ is given by*

$$P\left(\left|\frac{X - \mu}{\mu}\right| \geq \delta\right) \leq 2e^{-\delta^2 \mu/3}$$

Applying this to the classic min hash approach

$$P\left(\left|\frac{\frac{X^k}{k} - J(A, B)}{J(A, B)}\right| \geq \delta\right) \leq 2e^{-\delta^2 k J(A,B)/3}$$

Now let k_J be the number of hash functions and t be the confident of the Chernoff bound($k_J := k, \quad t := 2e^{-\delta^2 k_J J(A,B)/3}$), therefore

$$k_J = \frac{-3\ln(t/2)|A \cup B|}{\delta^2 |A \cap B|}$$

In our new approach, for a bloom filter $(B)_b$ of a set B with false positive rate of p, let

$$X_i = \begin{cases} 1 & \text{if } a \in (B)_b \text{ for } a = argmin\{h^{(i)}(e) : e \in A\} \\ 0 & o.w \end{cases}$$

i.e. X_i will determine membership of the element of A with the smallest image under hash function $h^{(i)}$ in B.(So take elem of A that hashes to smallest value under $h^{(i)}$ and check for memberships in B). Therefore $E(X_i) = \dfrac{|A \cap B|}{|A|} + p$. Let $X^k = \sum_{i=1}^{k} X_i$, hence $E(X^k) = k\dfrac{|A \cap B|}{|A|} + kp$. Define $c := \dfrac{|A \cap B|}{|A|}$, $J := \dfrac{|A \cap B|}{|A \cup B|}$ and $\mu := kc + kp$ (i.c $\mu = E(X^k)$). Then

$$\begin{aligned} P\left(\left|\frac{\left(\frac{X^k}{k} - p\right) - c}{c}\right| \geq \delta\right) &= P\left(X^k \geq \left(1 + \frac{kc\delta}{\mu}\right)\mu\right) + P\left(X^k \leq \left(1 - \frac{kc\delta}{\mu}\right)\mu\right) \\ &= P\left(X^k \geq \left(1 + \left(\frac{c}{c+p}\right)\delta\right)\mu\right) + P\left(X^k \leq \left(1 - \left(\frac{c}{c+p}\right)\delta\right)\mu\right) \\ &\leq 2e^{-\left(\frac{c}{c+p}\right)^2 \delta^2 k(c+p)/3} \end{aligned}$$

So for $t := 2e^{-\left(\frac{c}{c+p}\right)^2 \delta^2 k(c+p)/3}$, the desired Chernoff bound, let $k_c = k$ be the number of hash functions which is required to achieve this bound. Then

$$k_c = \frac{-3(c+p)\ln(t/2)}{c^2\delta^2}$$

Define $c_{est} := \frac{X^k}{k} - p$. Similarly

$$\begin{aligned} P\left(\left|\frac{\frac{|A|c_{est}}{|B|} - \frac{|A\cap B|}{|B|}}{\frac{|A\cap B|}{|B|}}\right| \geq \delta\right) &= P\left(\left|\frac{\frac{|A|c_{est}}{|B|} - \frac{|A|c}{|B|}}{\frac{|A|c}{|B|}}\right| \geq \delta\right) \\ &= P\left(\left|\frac{\left(\frac{X^k}{k} - p\right) - c}{c}\right| \geq \delta\right) \\ &\leq 2e^{-\left(\frac{c}{c+p}\right)^2 \delta^2 k(c+p)/3} \end{aligned}$$

For c_{est}, our estimation of containment, define $J_{est} := \frac{|A|c_{est}}{|A|+|B|-|A|c_{est}}$. Therefore,

$$\begin{aligned}
P\left(\left|\frac{J_{est}-J}{J}\right| \geq \delta\right) &= P(J_{est} \geq (1+\delta)J) + P(J_{est} \leq (1-\delta)J) \\
&= P\left(\frac{|A|c_{est}}{|A|+|B|-|A|c_{est}} \geq (1+\delta)J\right) + P\left(\frac{|A|c_{est}}{|A|+|B|-|A|c_{est}} \leq (1-\delta)J\right) \\
&= P\left(c_{est} \geq \frac{(1+\delta)J(|A|+|B|)}{|A|(1+(1+\delta)J)}\right) + P\left(c_{est} \leq \frac{(1-\delta)J(|A|+|B|)}{|A|(1+(1-\delta)J)}\right) \\
&= P\left(c_{est} \geq \frac{(1+\delta)(\frac{|A|c}{|A\cup B|})(|A|+|B|)}{|A|(\frac{|A\cup B|+(1+\delta)|A\cap B|}{|A\cup B|})}\right) + P\left(c_{est} \leq \frac{(1-\delta)(\frac{|A|c}{|A\cup B|})(|A|+|B|)}{|A|(\frac{|A\cup B|+(1-\delta)|A\cap B|}{|A\cup B|})}\right) \\
&= P\left(c_{est} \geq \frac{(1+\delta)(|A|+|B|)}{|A\cup B|+(1+\delta)|A\cap B|}c\right) + P\left(c_{est} \leq \frac{(1-\delta)(|A|+|B|)}{|A\cup B|+(1-\delta)|A\cap B|}c\right) \\
&= P\left(c_{est} \geq \frac{(1+\delta)(|A\cup B|+|A\cap B|)}{|A\cup B|+(1+\delta)|A\cap B|}c\right) + P\left(c_{est} \leq \frac{(1-\delta)(|A\cup B|+|A\cap B|)}{|A\cup B|+(1-\delta)|A\cap B|}c\right) \\
&= P\left(c_{est} \geq \left(1+\frac{\delta|A\cup B|}{|A\cup B|+(1+\delta)|A\cap B|}\right)c\right) \\
&+ P\left(c_{est} \leq \left(1-\frac{\delta|A\cup B|}{|A\cup B|+(1-\delta)|A\cap B|}\right)c\right)
\end{aligned}$$

define $\delta' = \frac{\delta|A\cup B|}{|A\cup B|+(1+\delta)|A\cap B|}$ and $\delta'' = \frac{\delta|A\cup B|}{|A\cup B|+(1-\delta)|A\cap B|}$. Therefore,

$$\begin{aligned}
P\left(\left|\frac{J_{est}-J}{J}\right| \geq \delta\right) &\leq e^{-(\frac{c}{c+p})^2\delta' k(c+p)/3} + e^{-(\frac{c}{c+p})^2\delta'' k(c+p)/2} \\
&\leq 2e^{-(\frac{c}{c+p})^2\delta' k(c+p)/3}
\end{aligned}$$

Let $k_{est} := k$ be the number of hash functions which is required to achieve a desire confident $t := 2e^{-(\frac{c}{c+p})^2\delta' k(c+p)/3}$. Therefore

$$k_{est} = \frac{-3(c+p)ln(t/2)\left(|A\cup B|+(1+\delta)|A\cap B|\right)^2}{c^2\delta^2|A\cup B|^2}$$

2.4.2. *Number of hashes required.* By comparing the ratio of the number of hash functions required for the Jaccard approach and the containment approach (our new approach),

$$\frac{k_J}{k_c} = \frac{-2\ln(t/2)}{\delta^2 J}\frac{c^2\delta^2}{-2(c+p)\ln(t/2)} = \frac{c}{J}\left(\frac{c}{c+p}\right)$$

When $p = 0.01$ and ≥ 0.01, $\frac{c}{c+p} \geq .90$

$$\frac{k_J}{k_c} \geq .9\frac{c}{J} = .9\frac{|A\cup B|}{|A|} = .9\frac{(|A|+|B|-|A\cap B|)}{|A|} \geq .9\frac{|B|}{|A|}$$

Therefore to obtain a specific accuracy and particular Chernoff boundary, containment approach needs $\frac{|B|}{|A|}$ times fewer hash functions than the Jaccard approach.($k_J \approx \frac{|B|}{|A|}k_c$)

2.4.3. *Time complexity.* Both approaches linear in the number of hashes. (So containment approach is m times faster for typical set sizes).

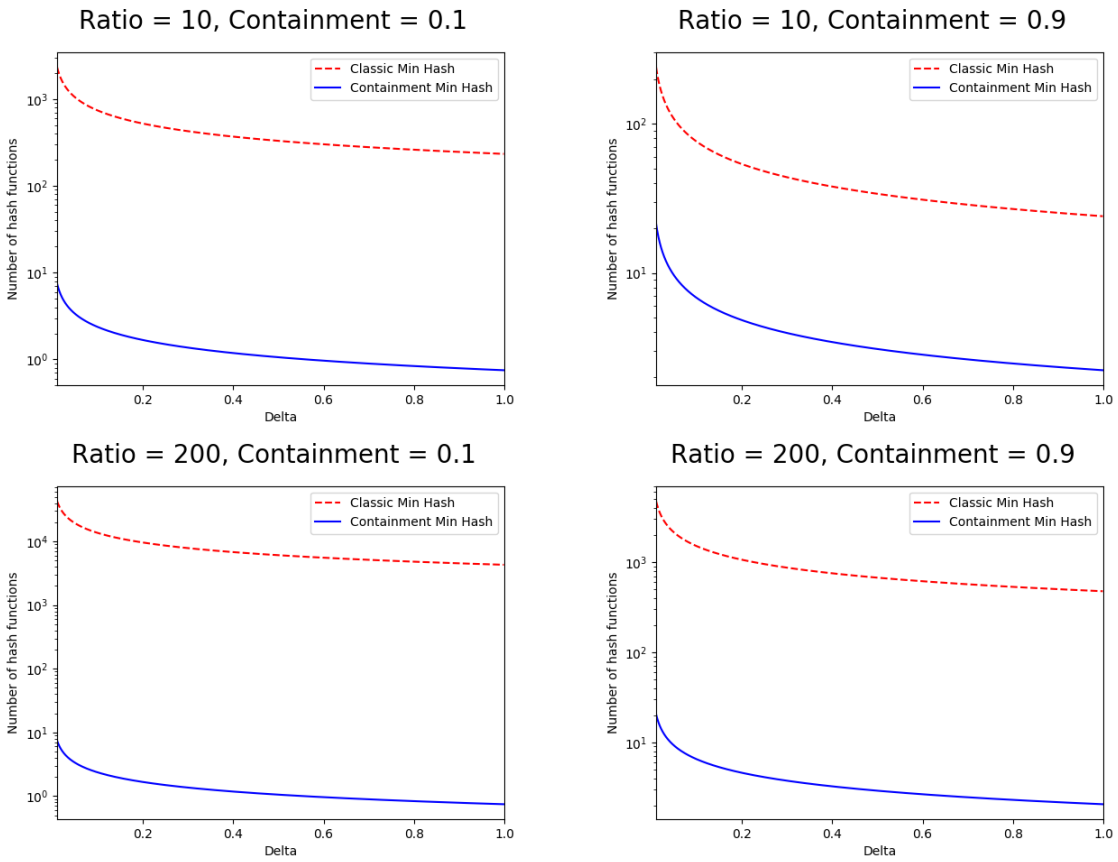

FIGURE 2. Explanation+ example

2.4.4. *Space complexity (all with examples of the numbers in practice).* Even though we must form a bloom filter for a metagenomic sample (which can potentially contain many billions of k-mers), due to the many fewer hashes we must store (orders of magnitude fewer) there actually turns out to be a cost savings due to the size of training databases.

Let G be the number of training genomes utilized, M the maximum number of k-mers in any of the training genomes. Let N be the number of k-mers in the sample. Typical values are XXXX. Recall that k_c and k_J are the number of hashes required in the containment approach and classic approach respectively.

In practice, we see 48 bits per hash seen in practice, so the total number of bits used by the classical approach is given by

$$S_J = 48k_J \cdot G$$

Typical bloom filter implementations use approximately $1.44 \log_2(1/p)$ bits per item where p is the false positive rate. Hence, when $p = 0.01$, the total number of bits used by the proposed containment approach is give by

$$S_c = 48k_c \cdot G + 9.6N$$

It is reasonable to assume that in practice, the number of k-mers in the sample S is some (large) multiple m of the largest number of k-mers found in any one of the training genomes M: $S \leq m \cdot M$.

Hence, comparing the ration of the size requirements of the approaches, we obtain:

$$\frac{S_c}{S_J} = \frac{1}{5}\frac{M}{G \cdot k_c} + \frac{1}{m}$$

Typically G is greater than 30 thousand, M is less than 4.3 million, $k_c = 500$ (change this to confidence and delta), and $m \geq 200$. Hence, $\frac{S_c}{S_J} \leq 0.06$, so the containment approach actually uses significantly less space.

Include a plot of tradeoff in sizes (metagenome k-mers vs number of training genomes)

3. Results

In this section, we compare classic min hash to the proposed method.

3.1. **Synthetic data.** Here we illustrate the improved accuracy of containment min hash over classical min hash in estimating the Jaccard index. To that end, we generated two random strings w_A and w_B on the alphabet $\{A, C, T, G\}$. We set $|w_A| = 10,000$ and $|w_B| = 15$ to simulate the situation of interest where one wishes to estimate the Jaccard index of two sets of very different size. We then appended a common string w_C of increasing length to each of w_A and w_B so that $\text{Jac}_k(w_A w_C, w_B w_C)$ ranges between 0 and 1. We picked the k-mer size of 11 and utilized a signature size of 100. Figure 3 depicts the comparison of containment min hash with the classical min hash Jaccard estimate on this data and effectively illustrates the results in section 2.4.1 which proved that the containment approach has a higher probably of being closer to the true Jaccard than the classic approach. The mean and variance of the classic min hash approach on this data was 0.000577 ± 0.001776 while using the containment approach was 0.000717 ± 0.000005 demonstrating a substantial decrease in variance. This improved variance was observed over a range of k-mer sizes, number of hashes, and lengths of input strings.

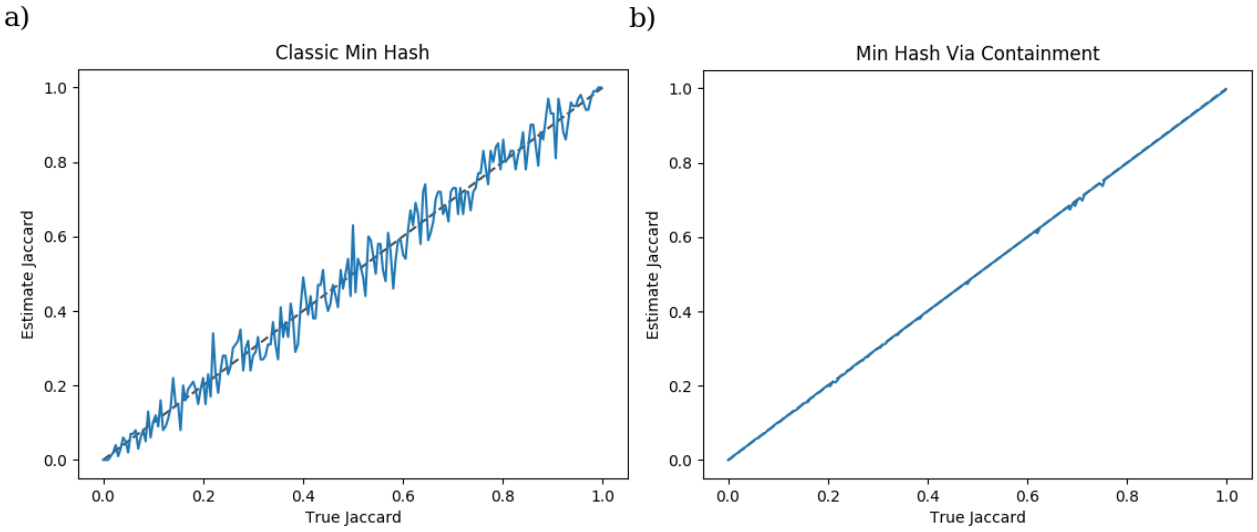

Figure 3. Comparison of containment min hash to the classical min hash estimate of the Jaccard index on synthetic data. Each method utilized the 100 smallest hashes of the murmer3 hash function on the 11-mers of two randomly generated strings with sizes 10,000 and 15 respectively after appending a common substring of increasing size. a) Classical min hash estimate of the Jaccard index. b) The proposed containment min hash method on the same data.

3.2. **Simulated biological data.** To demonstrate the exponential improvement of containment min hash over classical min hash for increasing sample sizes, we contrast here the mean relative performance of the classical min hash estimate to the containment approach on simulated biological data. We utilized GemSIM [2] to simulate two sets of metagenomic data. The first set had an average number of k-mers in the sample was only 58.254% of the size of the average number of k-mers in the genomes used to simulate the data. The second set had an average number of k-mers in the sample equal to 196.506% of the average size of the number of k-mers in the genomes used to simulate the data. As demonstrated in Section 2.4.1 we expect that once the number of k-mers in the sample is large in comparison to the number of k-mers used to simulate the data, the containment approach will give an exponentially better estimate of the Jaccard index in comparison to the classical min hash approach. Figure 4 depicts the relative error of the classic min hash approach and the containment approach on these two sets of simulated data. Observe that the containment approach has significantly less error when, as is commonly seen in practice, the number of k-mers in the sample is appreciable in comparison to the number of k-mers in a given reference organism. As demonstrated in section 2.4.1, this improvement of the containment approach over the classic approach continues to grow as the metagenome size grows in relation to the reference genome sizes.

For the first set of simulated data, we used GemSIM to simulate 10000 reads from 20 randomly selected bacterial genomes for the k-mer size $k = 11$. We then repeated this 20 times. A false positive rate of 0.001000 was used for the false positive rate of the bloom filter used for the containment approach.

For the second set of simulated data, we used GemSIM to simulate 1000000 reads from 20 randomly selected bacterial genomes for the k-mer size $k = 11$. We then repeated this 20 times. A false positive rate of 0.001000 was used for the false positive rate of the containment approach.

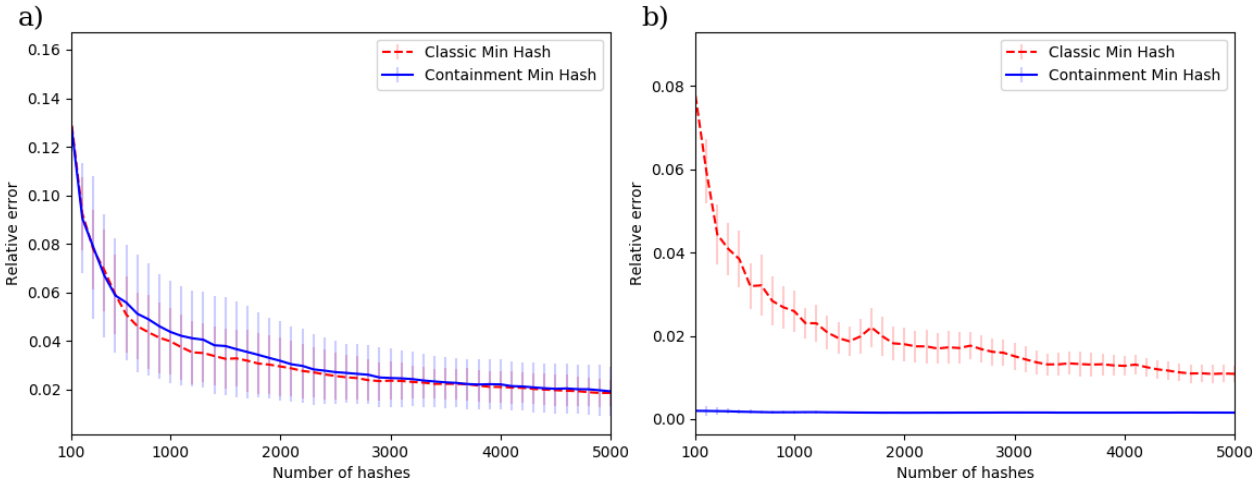

FIGURE 4. Comparison of relative error of containment min hash to the classical min hash estimate of the Jaccard index on simulated biological data. a) On 20 replicates of samples consisting of 20 genomes with only 10000 reads. b) On 20 replicates of samples consisting of 20 genomes with 1000000 reads.

3.3. **Real biological data.** Real metagenomes contain many magnitudes more k-mers than those found in any reference organisms [] which indicates the advantage of utilizing the proposed containment approach to the classical min hash estimate of the Jaccard index. To evaluate the utility of the containment min hash approach on real biological data, we analyzed a subset of DNA generated by the study in [1] consisting of those reads contained in the sample 4539585.3.fastq. This sample

consisted of 25.4M reads with average length of 65bp. We formed a bloom filter consisting of all 21-mers of this sample and formed sketches of size 500 from 4,798 viral genomes obtained from NCBI. Utilizing the proposed containment min hash approach, we found the largest containment index between the reference viral metagenomes and the sample to be 0.0257 for the virus *Sauropus leaf curl disease associated DNA beta* which corresponds to a Jaccard index of 2.398e-08. As demonstrated in section 2.4.1 we can be XX% sure that the true Jaccard index between this genome and the sample is within a relative error of XX% of the true Jaccard index value. If we were to use the classical min hash approach, the Chernoff bounds dictate that we would min hash sketches of size XXX to achieve this same confidence bound on the relative error.

To evaluate if this extremely low-abundance organism is actually present in the sample, we utilized the SNAP alignment tool [3] to align the sample to the *Sauropus leaf curl disease associated DNA beta* genome. The script *MakeCoveragePlot.sh* provides the exact commands and parameters used to perform the alignment. We found that 288 reads aligned with a MAPQ score above 20 (XXgrab valueXX). The coverage of the viral genome is depicted in Figure 5 using a square-root scale and a window size of 10. These high-quality mapped reads to such a small genome lends evidence to support the claim that this particular virus is actually present in the sample metagenome.

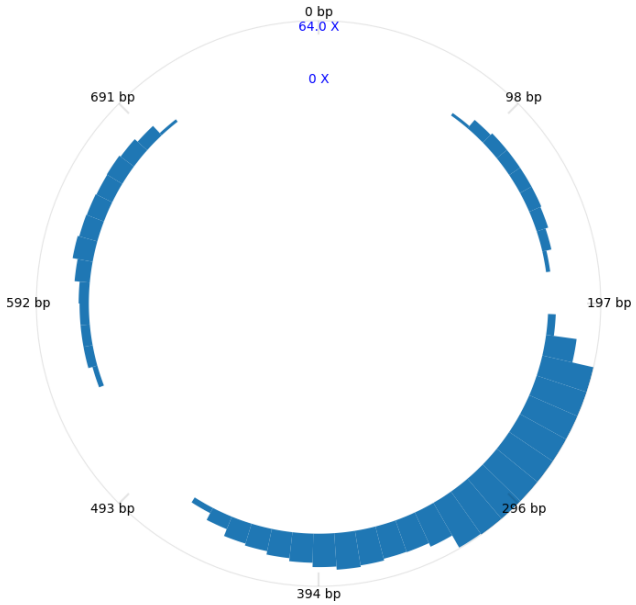

FIGURE 5. Plot of the real metagenomic sample alignment coverage to the virus *Sauropus leaf curl disease associated DNA beta* detected by the proposed containment min hash approach. A total of 288 reads aligned with a MAPQ score above 20 (XXgrab valueXX) using the SNAP aligner [3]. A square root scale and a window size of 10 was used for the plot, resulting in an average per-window coverage of 24.217X.

4. DISCUSSION

References

[1] A. C. Howe, J. K. Jansson, S. A. Malfatti, S. G. Tringe, J. M. Tiedje, and C. T. Brown. Tackling soil diversity with the assembly of large, complex metagenomes. *Proceedings of the National Academy of Sciences*, 111(13):4904–4909, 2014.

[2] K. E. McElroy, F. Luciani, and T. Thomas. Gemsim: general, error-model based simulator of next-generation sequencing data. *BMC genomics*, 13(1):74, 2012.

[3] M. Zaharia, W. J. Bolosky, K. Curtis, A. Fox, D. Patterson, S. Shenker, I. Stoica, R. M. Karp, and T. Sittler. Faster and more accurate sequence alignment with snap. *arXiv preprint arXiv:1111.5572*, 2011.