Sequences Dimensionality-Reduction by K-mer Substring Space Sampling Enables Effective Resemblance- and Containment-Analysis for Large-Scale omics-data

Huiguang Yi*, Yanling Lin & Wenfei Jin*

Department of Biology, South University of Science and Technology of China, Shenzhen, China
*Correspondence: yhg926@gmail.com & jinwf@sustech.edu.cn

Abstract

We proposed a new sequence sketching technique named k-mer substring space decomposition (kssd), which sketches sequences via k-mer substring space sampling instead of local-sensitive hashing. Kssd is more accurate and faster for resemblance estimation than other sketching methods developed so far. Notably, kssd is robust even when two sequences are of very different sizes. For containment analysis, kssd slightly outperformed mash screen—its closest competitor—in accuracy, while took testing datasets of 110,535 times less space occupation and consumed 2,523 times less CPU time than mash screen—suggested kssd is suite for quick containment analysis for almost the entire omics datasets deposited in NCBI. We detailed the kssd algorithm, provided proofs of its statistical properties and discussed the roots of its superiority, limitations and future directions. Kssd is freely available under an Apache License, Version 2.0 (https://github.com/yhg926/public_kssd)

Key words: sketch; distance estimation; sequence
1 Introduction

Up to this manuscript is composing, the size of NCBI SRA database alone has reached to 30 peta base-pairs, the number is seemingly will grow exponentially in the near future [1]. Additionally, the continuously emerging independent projects such as The Cancer Genome Atlas [2], 1000 Genomes [3], Human Microbiome Project [4], The Genotype-Tissue Expression (GTEx) project [5], Human Cell Atlas [6], etc. produced so-called omics-datasets (including genomics, metagenomics, transcriptomics, etc.) of terabytes to petabytes each [7]. Such a phenomena of omics-data explosion bring us to the ‘omics big data’ era [8]. On the one hand, the ‘omics big data’ is revolutionizing biomedical science profoundly [9]; on the other, such volume of data put a heavy burden for data storage and operations, and put a pressing need for effective sequence dimensionality-reduction techniques.

Recently, the omics ‘big data’ problem was largely alleviated by a class of sequence dimensionality-reduction techniques—minhash sketching, including methods such as mash [10], bindash [11], etc. Minhash sketching selects fix number (typically around 1000) of k-mers, termed sketch, which have minimum hashes among all the k-mers from the sequence, for the reduced representation of this sequence. No matter how large the original sequences are, taking their sketches of merely several Kbytes in size, minhash sketching could effectively estimate the pairwise distances, hence enabled first time rapid clustering of entire NCBI RefSeq. Other applications of minhash sketching include identifying mis-labeled samples, real-time outbreak pathogen surveillance, searching genomics database, etc. [10]. Notably, minhash sketching also play a crucial part in single-molecular long-reads assembly by overcome the computational bottleneck of all vs. all long-reads overlapping [12].

However, minhash sketching is not suite for comparing two k-mer sets of very different sizes [13], which is the scenario of many potential applications, e.g. prioritizing the reference genomes for a metagenomics dataset, composition analysis of a metagenomics dataset, searching viruses/bacteria contaminations in human omics samples, etc. Mash screen (mash version 2) then was developed to address this problem via estimating the proportion (termed containment-index or containment-coefficient) of a reference (the smaller k-mer set) contained in a sequences-mixture (the larger k-mer set). The containment-coefficient then is converted to containment-score representing identity between the two k-mer sets (see [14] for the definition); if there are many mixtures/references to be compared, the references are sketched as the reference-sketches database, then the mixtures are queried one by one against the reference-sketches database to calculate the containment-coefficient of each reference in each mixture [14]. But the mixtures, which consume the majority of the space, are not sketched at all, hence the merits of sketching technique—extremely high space- and time- efficiency—are largely lost, which make mash screen more suitable for answering the question “what reference genomes contained in my(one) omics run?” [14], but not “what reference genomes contained in which omics run?”.
To address the latter question, we here propose a new sequence sketching technique named \( k \)-mer substring space decomposition (kssd) which use random \( k \)-mer substring space sampling (see Methods) instead of minhash or more generally local-sensitive hashing [15] for sequences sketching. Kssd sketches both sequences-mixtures and references and can effectively estimate the distances using sketches even if they are of very different sizes. Conventionally, sketching technique is used to measure the extents of the two kinds of relationships—resemblance and containment, which capture the notions of “roughly the same” and “roughly contained” for two sequences (or documents), respectively [13]; more specifically, the resemblance captures the distance/similarity of two sequence of similar sizes, whereas the containment captures the distance/similarity of two sequence of very different sizes. Here we examine the performances of kssd and minhash-based sketching methods w.r.t. both resemblance- and containment-estimation.

2 Results

2.1 Accuracy of resemblance estimation
To compare the accuracy of resemblance estimation of kssd with other sketching methods, we evolved a reference genome in-silico with 600 pre-defined mutation rates range from 0.001 to 0.60 with stepwise increase of 0.001 (use an in-house script, see supplementary file 1 for the dataset) which were served as ground truths hereafter. The mutants were divided into closely-related group and distantly-related group (with mutation rate 0.001 ~ 0.3 and 0.301 ~ 0.60, respectively). The mutation distances between the reference and its mutants were estimated then by kssd and two other sketching methods mash and bindash, with variated \( k \)-mer lengths and sketch-sizes. The Pearson correlation-coefficient of the ground truths and the estimates was adopted as the accuracy-measurement (Figure 1A, B).

The results show clear stratifications w.r.t. \( k \)-settings, where close \( k \) (20 vs. 21) yield generally similar curves whereas distant \( k \) (16 vs. 20, 21) yield dissimilar curves. When the sketch-size is relative smaller, using shorter \( k \) (16 here) is more accurate than longer \( k \) (20, 21 here); but when the sketch-size is larger (larger than 10,240 and 42,112 in closely-related group and distantly-related group, respectively), using longer \( k \) is more accurate (Figure 1A, B). However, for the sake of efficiency, sketching technique rarely use sketch-size greater than 10,000, so the shorter \( k \) is preferred. Given the same \( k \)-settings (\( k =16 \) or 20), kssd outperforms both mash and bindash on both groups, especially when the sketch-size is small (84, 1,268 and 21,077), where the accuracy of kssd is on par with that of mash/bindash with doubled sketch-size (Figure 1).

2.2 Accuracy of containment estimation
To compare kssd with mash screen for the accuracy of containment estimation, we benchmarked kssd by the same approach used by Ondov et al. [14], where a constitutes-known synthetic microbiome community (referred
as the shakya dataset, see supplemental file 2 for the 64 known constitutes), was adopted as the testing dataset [14]. Notably, due to sample contaminations, the real shakya dataset (accession: SRR606249) contains also exogenous genomes that are not included in the 64 known constitutes [16], and the contaminations could be revealed by comparing to the simulated shakya dataset that consisted of only the short-reads from the 64 known constitutes (served as the contamination-free control, see simulation method in [14]). However, Ondov et al. did not provide the exact list of the references they used, so we took as references the latest NCBI prokaryotic assembly which consisted of 138,743 genomes (see supplement file 2 for the summary) instead of the 82,086 genomes of Ondov et al.

Given the references, the containment-measurements (namely, the aaf-distance of kssd and the containment-score of mash screen, see Equation 5 and [14] for the respective definitions) of each reference to the real- and the simulated-shakya datasets were plotted against the minimum of the distances of this reference to the 64 known constitutes (referred as the minimum reference-to-constitutes distance hereafter, Figure 2). The Fig. 3 in the paper of Ondov et al. [14] were faithfully reproduced (Figure 2, 2nd row), though we used more references here. The kssd containment-analysis (Figure 2, 1st row) conveys almost the same information with that of mash screen, where the data points are biased due to the same low-abundance constitutes (Figure 2, circled in purple) and the same contaminations (Figure 2, circled in orange and red). However, the data points for mash screen get increasingly discrete with the increase of the minimum reference-to-constitutes distance (x-axis) or the decrease of containment-score (y-axis) due to its sketch-size is fixed. In contrast, the data points for kssd are continuous with the change of x and y, which is more realistic. Since kssd estimates the distance (Equation 5) whereas mash screen estimates the identity between the reference and the shakya dataset, the containment-measurement of kssd is positively correlated with the minimum reference-to-constitutes distance whereas that of mash screen is negatively correlated. Therefore, we used the absolute correlation-coefficients for accuracy comparison. Both the kssd and the mash screen containment-measurements are correlated well with the minimum reference-to-constitutes distances less than 0.15, but kssd slightly outperforms mash screen, where the absolute correlation-coefficients of kssd for the simulated- and the real-shakya dataset are 0.9869 and 0.9611, respectively, and of mash screen are 0.9808 and 0.9592, respectively.

Notably, kssd took as queries the sketches of the shakya datasets combined of only 580Kbytes—110,535 times less space occupation than that of mash screen which took the raw datasets combined of 52Gbytes. Moreover, kssd took only 23.5 CPU minutes for the raw dataset sketching and 3 CPU seconds for the containment estimation, whereas mash screen took totally 126 CPU minutes for screening and the containment estimation. There is no need for kssd to sketch the raw datasets again for the containment-analysis for new references whereas mash screen would take another 126 CPU minutes for another 138,743 new references, which is likely happen in the near future since the NCBI Refseq database is growing rapidly [17], meaning the asymptotic time-
consumption of kssd is only $\frac{1}{2523}$ of that of mash screen (3 CPU seconds vs. 126 CPU minutes). Such a high space- and time-efficiency of kssd should enable building a sketch database for almost all the short-read datasets deposited in NCBI for fast containment-analysis.

### 2.3 Computational efficiency

We also tested the time-complexities of kssd, mash and bindash using a dataset consisted of 100K bacteria reference genomes under a 12-cores machine. It took 10531, 19094 and 328835 CPU seconds for kssd, bindash and Mash, respectively and took 933, 11,335 and 27,600 elapsed seconds for kssd, bindash and mash, respectively (Figure 3A and B) to complete the all pairwise distance computation. Therefore, kssd is 12 times and 30 times faster than bindash and mash respectively while using same number of threads (12 threads here). The comparison of time-complexity was conducted using similar sketch-size (1,780 and 2,048 for kssd/mash and bindash, respectively), as illustrated in Figure 1, kssd can achieved similar accuracies by using only half of the sketch-sizes of mash/bindash. Therefore, kssd could outperform mash/bindash even more under the accuracy criterion.

### 3 Discussion

We showed in this manuscript that kssd has greater accuracy and computational efficiency versus minhash sketching methods, and notably, kssd distance estimation is robust even when two sequences are of very different sizes. Mash screen estimates containment-coefficients from the raw data of the sequences-mixtures, whereas kssd estimates from the sketches which are hundreds of thousands of times smaller. Such a high-efficient summarizing ability of kssd facilitates large-scale containment analysis. We illuminated that the Jaccard- and the containment-coefficients estimated by kssd are essentially the sample proportions which are asymptotically gaussian distributed. Therefore, the standard deviations and 95% CIs could be resolved analytically (see Methods), which is not yet framed by other sketching methods. Kssd is also the only method able to decompose a k-mer set into a collection of mutual-exclusive sketches, which enables calculating the ground truth Jaccard- and containment-coefficient even when fully loading of the whole k-mer set is restrained by the memory limitation.

Kssd first selects k-mers and then maps the selected k-mers to integers one-to-one using a pre-determined recoding scheme. This process is reversible and could be seen as a special hash function that has no hash-collision (namely, two different k-mers are mapped to the same integer). In contrast, other sketching methods typically first hash from the whole k-mers space to a much smaller integers space, which caused hash-collision, before the hashes are selected for sketching. Since the final sketch would be smaller than the integers space, sketching methods like kssd putting k-mers selection before hashing could avoid hash-collision. Hash-collision, usually infrequent though, would increase the variation of the Jaccard- or containment-coefficient estimates. Thanks to the resolution of the hash-collision, kssd acquires its superior accuracy over other methods.
Though kssd possess the above-mentioned merits, there are still limitations: Firstly, Kssd loses its efficiency when searching individual-gene- or small-virus-references contained in sequences-mixtures, since these sequences are too short to be dimensionality-reduced further to an informative sketch [14]. And kssd requires the sequences-mixtures to be dimensionality-reduced with the same rate with the references, so the sequences-mixtures cannot be dimensionality-reduced either. We suggest the references should be at least 10K base-pairs, so that the lowest-level sketches (with 16 folds of dimensionality-reduction) would still be informative enough for effective distance estimation. For single-gene searching, the tool BIGSI [18] would be a more suitable choice, and for numerous small-virus genomes searching, mash screen would be better. Secondly, in current version of kssd implementation, the choices for the dimensionality-reduction rate are restricted to $16^n$ ($n$ ranges from 0 to $\frac{k}{2}$, where $k$ is k-mer length and kssd only work with even $k$), which prevents fine-tuning of the dimensionality-reduction rate. In addition, kssd currently cannot handling amino-acid sequences and 10X genomic single-cell sequences, and cannot overlapping single-molecular-long-reads (the latter two require sketching multiple sequences in a single file individually). Extensions including these functionalities are our future directions.

Another important direction is to estimate directly from the sketch the abundance of each reference contained. Though it is easy to tracking in sketch the k-mer occurrences from the sequences-mixture, the relationship between the k-mer occurrences and the abundance of each reference contained is complicated, since other genomes contained may also contribute to the k-mer occurrences thus confounding abundance estimation. However, the relationship is not totally chaotic, it is evidenced that there is clear average nucleotide identity (ANI) boundary (ANI = 95%) between two difference prokaryotic species [19], which means roughly 5% or larger genome regions are species-specific, thus the k-mer occurrences of these regions are not or much less confounded by other species and probably reflect the truth abundance. Therefore, a robust inferring of species-specific k-mer occurrences would be the key for sketch-based abundance estimation.

Though minhash sketching was introduced for omics-data handling recently, it has long history of development that initially used for detecting similar web pages and clustering documents [13]. In contrary, kssd was introduced first time in this study, but we speculate that a modified version of kssd can perform tasks like documents clustering or similar web pages detection as well. After all, the documents, webpages content and DNA are essentially all sequences but with a different alphabet set.

4 Methods

4.1 The main idea

The kssd idea originated from the naive sketching method of sampling k-mers directly from the sequence as its sketch. However, such a sketching method is ill-suited for distance estimation since the two sketches $S(A)$ and
$S(B)$ are probably drawn from two unrelated regions of two genome $A$ and $B$. Therefore, they have very few shared $k$-mers (with a Jaccard-coefficient approximate to 0), even when $A$ and $B$ are nearly identical (Figure 4A). Notwithstanding its naivety, this thought inspired us the idea of $k$-mer space sampling—Firstly, a subset of $k$-mers $s$, termed $k$-mer subspace, is drawn randomly from $k$-mer space $S$ (namely the collection of all possible string of length $k$ defined in a given alphabet set $\Sigma$); then the sketch of any given sequence is built by overlapping $s$ with the $k$-mers set of this sequence. Since $s$ is an unbiased sampling of the $k$-mer space $S$, it is independent of any instance $k$-mer sets. After sketching, two sequences $A$ and $B$, their intersection $A \cap B$ and union $A \cup B$ should go through dimensionality-reductions of the same expectation rate of $\frac{|S|}{|s|}$. Therefore, it enables measuring both the resemblance and the containment of the two sequences directly using their sketches, even if they are of very difference sizes (Figure 4B). The $k$-mer space sampling could be extended to $k$-mer space shuffling—where $k$-mer space $S$ is shuffled and split into $N$ subspaces of equal size, so that $N$ sketches could be obtained for a $k$-mer set by sketching using the $N$ subspaces individually. All the $N$ sketches combined is a lossless representative of the original $k$-mer set, such a lossless property is particularly useful when the ground truth Jaccard- or containment-coefficient is needed when the memory is limited.

However, $k$-mer space sampling/shuffling is computational expensive when $k$ is large, since it consumes $O(|\Sigma|^k)$ memory and time. For example, when $k = 22$, a common $k$ choice for mammalian genome analysis, it occupies 16Tbytes memory and takes $4^{22}$ CPU operations ($|\Sigma| = 4$ here) for the $k$-mer space shuffling alone, which is an unrealistic resource demand. To address this, we generalized $k$-mer space sampling/shuffling to $k$-mer substring space sampling/shuffling (kssd)—where a substring of the $k$-mer is selected according to a pre-defined pattern, so that we can use $k$-mer substring space sampling/shuffling instead of whole $k$-mer space sampling/shuffling (Figure 5). In this way, the computational complexity is dramatically reduced.

4.2 Mathematical characterization of kssd method

4.2.1 $k$-mer substring selection pattern, $k$-mer substring space sampling/shuffling and sketching

A $k$-mer substring selection pattern (kssp) $p$ is a ’01’ string of length $k$, by which the letters at the 1s of the given $k$-mer (or $k$-mer set) $K$ are concatenated, termed as the $p$-selected-$K$-substring(s) and denoted by $p^1(K)$, and the letters at 0s are also concatenated, termed as $p$-unselected-$K$-substring(s) and denoted by $p^0(K)$. The weight of $p$, denotated by $w$, is the number of 1s in $p$ (also the length of $p^1(K)$). Given alphabet set $\Sigma$ and weight $w$, the $k$-mer substring space $S$ is the collection of all length $w$ strings defined in $\Sigma$, thus has dimensionality $|S| = |\Sigma|^w$ (we view each element in $S$ as a dimension). $S$ is first Fisher-Yates shuffled [20] and then partitioned into $N$ subspaces of equal size, denoted by $s_1, s_2, ..., s_N$; for computational convenience, we choose $N = |\Sigma|^m$, where $m$ could be any positive integer $\leq w$, so that $|s| = |\Sigma|^w - m (s \in \{s_1, s_2, ..., s_N\})$, and the dimensions of $s$ could be represented by lexically-ordered strings of length $w-m$ (Figure 5A).
Let $A$ be a $k$-mer set, the subset $R(A) = \{K| K \in A, p^1(K) \in s\}$ is mapped to sketch $S(A)$ as follow: for each $k$-mer $K \in R(A)$, the $p$-selected-$K$-substring $p^1(K)$ is recoded by the lexical order of its dimension in the subspace $s$, denoted by $s(p^1(K))$, then the $p$-unselected-$K$-substring $p^0(K)$ and the string $s(p^1(K))$ are concatenated into the new string $p^0(K)s(p^1(K))$, which has length of $k$-$m$ (Figure 5B). This $k$-mer recoding scheme could be summarized by the function $r(K) = p^0(K)s(p^1(K))$, and the overall sketching process could be summarized by:

$$S(A) = r(R(A)) = \{ p^0(K)s(p^1(K)) | K \in A, p^1(K) \in s \} \quad (1)$$

### 4.2.2 kssd distance

The Jaccard- and the containment-coefficients for a $k$-mer sets pair $(A, B)$ are estimated by:

$$\hat{f}(A, B) = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|} \quad (2)$$

and

$$\hat{C}(A, B) = \frac{|S(A) \cap S(B)|}{\min(|S(A)|, |S(B)|)} \quad (3)$$

respectively. $\hat{f}$ and $\hat{C}$ could be further converted to mash- and aaf-distances by

$$\hat{D}_m = -\frac{1}{k} \ln \left( \frac{2\hat{f}}{1+\hat{f}} \right) \quad (4)$$

and

$$\hat{D}_a = -\frac{1}{k} \ln \left( \hat{C} \right) \quad (5)$$

respectively; both the mash- and aaf-distances estimate the mutation distance between $A$ and $B$ [10, 21].

### 4.2.3 Statistics properties of kssd

Due to the nature of Fisher-Yates shuffle [20], subspace $s$ and sketch $S(A)$ are an unbiased sampling (without replacement) of space $S$ and the recoded $k$-mer set $r(A)$, respectively; and we have $|r(A)| = |A|$, since the $k$-mer recoding function $r$ is injective. Thus, we have:

$$E[|S(A)|] = \frac{1}{N} |r(A)| \quad (6)$$

namely, the expected rate of dimensionality-reduction is $\frac{1}{N}$ for sketch $S(A)$.

For two $k$-mer sets $A$ and $B$, we have:

$$S(A \cap B) = \{ p^0(K)s(p^1(K)) | K \in A \cap B, \ p^1(K) \in s \}$$

$$= \{ p^0(K)s(p^1(K)) | K \in A, \ p^1(K) \in s \} \cap \{ p^0(K)s(p^1(K)) | K \in B, \ p^1(K) \in s \} \quad (7)$$

$$= S(A) \cap S(B)$$

and similarly, we have:

$$S(A \cup B) = S(A) \cup S(B), \quad (8)$$

$$S(A \cup B - A) = S(A) \cup S(B) - S(A) \quad (9)$$

and
\[ S(\mathcal{A} \cup B - B) = S(\mathcal{A}) \cup S(B) - S(B) \]  

(10)

namely, the orders of kssd sketching operation and set operations are interchangeable. Since \((A \cap B) \cap (A \cup B - A \cap B) = \emptyset\), we have \(|S(A \cap B)| + |S(A \cup B - A \cap B)| = |S(A \cup B)|\), where \(|S(A \cap B)|\) and \(|S(A \cup B - A \cap B)|\) are independent. Therefore, we have:

\[
E\left\{ \frac{1}{\hat{J}} \right\} = E\left\{ \frac{|S(\mathcal{A}) \cup S(B) - S(\mathcal{A}) \cap S(B) + S(\mathcal{A}) \cap S(B)|}{|S(\mathcal{A}) \cap S(B)|} \right\} + 1 \\
= E\left\{ \frac{|S(\mathcal{A}) \cup S(B) - S(\mathcal{A}) \cap S(B)|}{|S(\mathcal{A}) \cap S(B)|} \right\} + 1 \\
= E\left\{ \frac{|S(\mathcal{A} \cup B - A \cap B)|}{E[|S(\mathcal{A} \cup B)|]} \right\} + 1 \\
= \frac{1}{n} \frac{|S(\mathcal{A} \cup B - A \cap B)|}{n} + 1 \\
= \frac{1}{J},
\]

(11)

namely,

\[ E\{\hat{J}\} = J, \]

similarly, we have:

\[ E\{\hat{C}\} = C, \]

(12)

which means \(\hat{J}\) and \(\hat{C}\) are unbiased estimates of \(J\) and \(C\), respectively. Both \(\hat{J}\) and \(\hat{C}\) are essentially sample proportions, thus have asymptotical gaussian distribution \(N(\hat{p}, \sqrt{\frac{\hat{p}(1-\hat{p})}{n}})\), \(np > 10\), where \(\hat{p} = \hat{J}, n = |S(\mathcal{A} \cup B)|\) or \(\hat{p} = \hat{C}, n = \min(|S(\mathcal{A})|, |S(\mathcal{B})|)\) when \(p = J\) or \(C\), respectively. Therefore, the population standard deviation is given by

\[ sd(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \]

(13)

and the 95% confidence interval (CI) is \([\hat{p} - 1.96 \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + 1.96 \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}]\). The P-value is defined by:

\[ P-value = P (z \geq \frac{\hat{p}}{\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}}) \]

(14)

Where the random variable \(z\) follows the standardized normal distribution. To account for multiple testing problem, \(Q\)-value \(Q\) is obtained by multiplying the \(P\)-value \(P\) by the total number of comparisons, for example, for \(x\) queries search against \(y\) references, \(Q = Px_y\).

4.2.4 k-mer set decomposition.
By applying Equation 1 with subspace $s_1$, $s_2$, ..., $s_N$ individually, $k$-mer set $A$ can be decomposed into a collection of sketches $d(A) = \{S_1(A), S_2(A), ..., S_N(A)\}$, respectively. Since $\{s_1, s_2, ..., s_N\}$ is a partition of $k$-mer substring space $S$, we have $\bigcup_{i=1}^{N} s_i = S$ and $s_i \cap s_j = \emptyset$, for $1 \leq i \leq N$ and $1 \leq j \leq N$, therefore,

$$\bigcup_{i=1}^{N} S_i(A) = \{p^i(K)_{s_i}(p^i(K)) \mid K \in A, p^i(K) \in S\} = r(A),$$

and

$$S_i(A) \cap S_j(A) = \{p^i(K)_{s_i}(p^i(K)) \mid K \in A, p^i(K) \in s_i \} \cap \{p^j(K)_{s_j}(p^j(K)) \mid K \in A, p^j(K) \in s_j \} = \emptyset, \ \text{for} \ \ 1 \leq i \leq N, \ 1 \leq j \leq N.$$

Thus, we have:

$$|A| = |\bigcup_{i=1}^{N} S_i(A)| = \sum_{i=1}^{N} |S_i(A)|. \quad (17)$$

The above conversion from $A$ to $d(A)$ is termed as $A$ decomposition and sketches $S_i(A)$ are termed as the components of $A$. Based on Equations 7-8,17 the ground truth Jaccard-coefficient $J$ and containment-coefficient $C$ of two $k$-mer sets $A$ and $B$ could be recovered by:

$$J = \frac{|A \cap B|}{|A \cup B|} = \frac{\sum_{i=1}^{N} |S_i(A \cap B)|}{\sum_{i=1}^{N} |S_i(A \cup B)|} = \frac{\sum_{i=1}^{N} |S_i(A) \cap S_i(B)|}{\sum_{i=1}^{N} |S_i(A) \cup S_i(B)|}.$$ \quad (18)

and

$$C = \frac{\sum_{i=1}^{N} |S_i(A) \cap S_i(B)|}{\min(\sum_{i=1}^{N} |S_i(A)|, \sum_{i=1}^{N} |S_i(B)|)}.$$ \quad (19)

respectively. Equations 18-19 allow estimating the intersection and union of $A$ and $B$ component by component, where each component take only $\frac{1}{N}$ memory of the $k$-mer set, which is particularly useful when ground truth $J$ and $C$ is needed in a memory-limited computer.

### 4.2.5 $k$-mer length

The optimal $k$-mer length for alignment-free sequences comparisons is roughly (but not strictly) framed by the Equation S2 in our previous work [22], namely:

$$k = \log_4 \frac{3g}{2m},$$ \quad (20)

where $g$ is the genome-size, and $m$ is the allowed upper bound of the probability of random $k$-mer hitting (two $k$-mers are identical due to coincidence of random combination) which typically ranges from 0.001 to 0.01. Equation 20 is proposed under the considerations that $k$ should be long enough to ensure $k$-mer uniqueness, while should be as short as possible to hold sensitivity [22]. It yields $k = 16$ for bacterial or smaller genomes, $k = 20$ or 22 for metagenomics/mammals’ or larger genomes and $k = 18$ for other genomes in-between. There is no necessary to use $k > 22$ and kssd avoids to do so to prevent $k$-mer substring space overflow (since the size of $k$-mer substring space is proportional to that of $k$-mer space by default).
4.2.6 Implementation

k-mer substring space sampling/shuffling is implemented by the subcommand shuffle, for example:

```
kssd shuffle -k 9 -s 6 -l 3 -o out
```

this command will generate a file named ‘out.shuf’ which keeps the shuffled k-mer substring space, this file would then took as input for sequences sketching or decomposition. In current version kssd 1.0, users have no need to set kssp by himself, since kssd using an internal symmetric kssp ‘(0)n(1)2m(0)n’, where n and 2m are the lengths of 0 or 1, respectively. Users need only to choose half-length of k (here set by -k 9) and half-length of k-mer substring (m, here set by -s 6). Such a design simplifies the k-mer substring selection step. The rate of dimensionality-reduction is controlled by -l, for this example which uses -l 3 will have an expected rate of dimensionality-reduction rate 163. For decomposition purpose set -l 0.

To sketch/decompose reference sequences, just run:

```
kssd dist -r <reference files dir> -L out.shuf -o <ref_outdir>
```
or

```
kssd dist -r <reference files dir> -L 3 -k 9 -o <ref_outdir>
```
either command will generate a database of reference-sketches in folder ref_outdir/ref; the latter one actually combines the two commands below internally:

```
kssd shuffle -k 9 -s 6 -l 3 -o <ref_outdir>/default.shuf &&
kssd dist -r <reference files dir> -L ref_outdir/default.shuf -o <ref_outdir>
```
it simplifies the process at the cost of losing control for option ‘-s’.

To sketch/decompose query sequences, make sure using the same ‘.shuf’ file with the references, and just run:

```
kssd dist -o <qry_outdir> -L out.shuf|<ref_outdir>/default.shuf <query files dir>
```
it will generate a database of query-sketches in folder qry_outdir/qry. Then we can perform all queries versus all references comparisons by:

```
kssd dist -r <ref_outdir>/ref -o <outdir> <qry_outdir>/qry
```
To compare references to themselves, just run:

```
kssd dist -r <ref_outdir>/ref -o <outdir> <ref_outdir>/qry
```
then the distance will output to the ‘distance’ file in the folder <outdir>.

5 Supplementary Files

1. ResemblanceAccuracyExperiment

The testing data sets, script, the shuffled k-mer substring space file (.shuf file), the detailed workflow (README.md) to generate Figure 1 and the source data of Figure 1.

2. ContainmentAccuracyExperiment
The testing data sets, the shuffled $k$-mer substring space file, sketches, the detailed workflow to generate Figure 2 and the source data of Figure 2.

6 Acknowledgments

This study was supported by National Key R&D Program of China (2018YFC1004500), the National Science Foundation of China (81872330, 31741077), and Basic Research Grant from Science and Technology Innovation Commission of Shenzhen Municipal Government (JCYJ20170817111841427).

7 References

1. NCBI. SRA database growth. 2019 [cited 2019 August 8]; Available from: https://trace.ncbi.nlm.nih.gov/Traces/sra/.
2. Cancer Genome Atlas Research, N., et al., The Cancer Genome Atlas Pan-Cancer analysis project. Nat Genet, 2013. 45(10): p. 1113-20.
3. Zhang, W., et al., Comparing genetic variants detected in the 1000 genomes project with SNPs determined by the International HapMap Consortium. J Genet, 2015. 94(4): p. 731-40.
4. Integrative, H.M.P.R.N.C., The Integrative Human Microbiome Project. Nature, 2019. 569(7758): p. 641-648.
5. Consortium, G.T., The Genotype-Tissue Expression (GTEx) project. Nat Genet, 2013. 45(6): p. 580-5.
6. Regev, A., et al., The Human Cell Atlas. Elife, 2017. 6.
7. Muir, P., et al., The real cost of sequencing: scaling computation to keep pace with data generation. Genome Biol, 2016. 17: p. 53.
8. Field, D., et al., Megascience. 'Omics data sharing. Science, 2009. 326(5950): p. 234-6.
9. Costa, F.F., Big data in biomedicine. Drug Discov Today, 2014. 19(4): p. 433-40.
10. Ondov, B.D., et al., Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol, 2016. 17(1): p. 132.
11. Zhao, X., BinDash, software for fast genome distance estimation on a typical personal laptop. Bioinformatics, 2019. 35(4): p. 671-673.
12. Berlin, K., et al., Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. Nat Biotechnol, 2015. 33(6): p. 623-30.
13. Broder, A.Z., *On the resemblance and containment of documents*. Compression and Complexity of Sequences 1997 - Proceedings, 1998: p. 21-29.

14. Ondov, B.D., et al., *Mash Screen: High-throughput sequence containment estimation for genome discovery*. bioRxiv, 2019: p. 557314.

15. Indyk, P. and R. Motwani, *Approximate nearest neighbors: towards removing the curse of dimensionality*, in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 1998, ACM: Dallas, Texas, USA. p. 604-613.

16. Shakya, M., et al., *Comparative metagenomic and rRNA microbial diversity characterization using archaeal and bacterial synthetic communities*. Environ Microbiol, 2013. 15(6): p. 1882-99.

17. NCBI. *RefSeq Growth Statistics*. [cited 2019 August 8th, 2019 ]; Available from: https://www.ncbi.nlm.nih.gov/refseq/statistics/.

18. Bradley, P., et al., *Ultrafast search of all deposited bacterial and viral genomic data*. Nat Biotechnol, 2019. 37(2): p. 152-159.

19. Jain, C., et al., *High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries*. Nat Commun, 2018. 9(1): p. 5114.

20. Fisher, R.A. and F. Yates, *Statistical tables for biological, agricultural and medical research*. Statistical tables for biological, agricultural and medical research. 1938, Oxford, England: Oliver & Boyd. 90-90.

21. Fan, H., et al., *An assembly and alignment-free method of phylogeny reconstruction from next-generation sequencing data*. BMC Genomics, 2015. 16: p. 522.

22. Yi, H. and L. Jin, *Co-phylog: an assembly-free phylogenomic approach for closely related organisms*. Nucleic Acids Res, 2013. 41(7): p. e75.

8 Figure Legends
Figure 1 Accuracy of kssd, mash and bindash. (A) Pearson correlation-coefficients on closely-related group and (B) distantly-related group. y-axis is scaled to $-\log(1 - r)$ for plotting clarity, where $r$ is the Pearson correlation-coefficient between the ground truth- and the estimated- mutation rates. The decimal above the highest data point at a sketch-size is the maximal $r$ value of all the three methods of all $k$ settings with that sketch-size. The default $k$-mer lengths $k$ for kssd, bindash and mash are 16, 21 and 21, respectively; to match $k$, we also run mash and bindash with $k = 16$ in addition to the default $k$ settings, but kssd takes only even $k$, so we also run with $k = 20$ to variate $k$. Due to different sketching mechanism, mash and bindash take as parameter the sketch-size of continuous integers and multiples of 64, respectively; but sketch-size is not a parameter of kssd and can only counted from the sketch file. To match sketch-sizes as closely as possible, we first sketched the reference using kssd with dimensionality-reduction levels $n = \{4, 3, 2, 1, 0\}$ and obtained the sketch-sizes $s_k = \{84, 1268, 21077, 337277, 5236120\}$, respectively; we got the nearest multiples of 64 of $s_k$ (the parenthesized values) and interpolated with their 2-, 4- and 8-fold sketch-sizes to obtain the sketch-sizes parameter $s_b$ for bindash; and we merged $s_k$ and the interpolated points of $s_b$ to obtain the sketch-size parameter $s_m$ for mash. Mash with $k=20, 21$ at sketch-size 5236120 is not shown due to the running error. Bindash with $k=20, 21$ at sketch-size 64 and 128 on distantly-related group are not shown due to the estimates lacking of variation for correlation.
Figure 2 accuracy of containment-measurements. $x$-axis indicates the minimum reference-to-constituents distance computed by kssd (1st row or blue) or mash (2nd row or red), and $y$-axis indicates the reference-to-mixture containment-measurement (aaf-distance or containment-score for kssd or mash screen, respectively). The decimals below the data points are the correlation-coefficients of the plot controlling $x < 0.15$. The 1st and 2nd columns are the plots of the simulated and the real shakya datasets, respectively. On the real shakya datasets, the data points biased from expect due to low abundance constitutes are circled in purple, and those biased due to two different contaminations are circled in orange and red.
**Figure 3 Computational efficiencies.** Assessments of CPU time (A), Elapse time (B) of the three methods were performed on a 32Gb,12-cores machine using a test dataset consisting of 100k bacteria genomes.

**Figure 4 The main idea of kssd.** (A) The set of blue dots $S(A)$ and the set of orange dots $S(B)$ are the $k$-mer sets randomly drawn from sequence $A$ and $B$, respectively, and the set of red dots is $S(A) \cap S(B)$. (B) The set of all red dots is the $k$-mer subspace $s$ that sampling from the $k$-mer space $S$, where $S = \text{green dots} \cup \text{red dots}$. 
Figure 5 kssd algorithm overview. (A) k-mer substring space shuffling. First, a k-mer substring selection pattern (kssp) \( p = '000010010000' \) is pre-determined for the 12-mer analysis, so the length of \( p \)-selected-substring is 2 and the k-mer substring space has dimensionality \( D = 4^2 = 16 \) (\(|Σ| = 4\) for nucleotide sequence, 1st and 2nd column). This 16-dimensions space is shuffled and partitioned into \( N \) subspaces of equal size (3rd column, \( N = 4 \) here), and the dimensions in each subspace are recoded by thelexically-ordered strings of length \( l = \log_4 \frac{D}{N} \) (4th column, \( l = 1 \) here). We use one subspace \( s \) (3rd and 4th column, the red subspace here) for sequence sketching. (B) Sequence sketching. First the k-mers with \( p \)-selected-substring (green substring in 1st column) belonging to the red subspace \( s \) are selected (2nd column), where the \( p \)-selected-substrings are recoded by thelexically-ordered dimension (3rd column), thus each of the selected k-mers is rearranged in such a way that the recoded \( p \)-selected-substring suffixes the rest substring (4th column). (C) Kssd distance. Once all the sequences are sketched, the Jaccard- and containment-coefficients could be estimated by \( \tilde{J} = \frac{|S(A) ∩ S(B)|}{|S(A) ∪ S(B)|} \) and \( \tilde{C} = \frac{|S(A) ∩ S(B)|}{\min(|S(A)|, |S(B)|)} \), respectively.