Bioinformatics, 38(13), 2022, 3343–3350
https://doi.org/10.1093/bioinformatics/btac180
Advance Access Publication Date: 18 May 2022
Original Paper

Sequence analysis

SPRISS: approximating frequent *k*-mers by sampling reads, and applications

Diego Santoro, Leonardo Pellegrina, Matteo Comin and Fabio Vandin *

Department of Information Engineering, University of Padova, 35131 Padova, Italy

*To whom correspondence should be addressed.

Associate Editor: Can Alkan

Received on December 16, 2021; revised on February 25, 2022; editorial decision on March 21, 2022; accepted on May 16, 2022

Abstract

Motivation: The extraction of *k*-mers is a fundamental component in many complex analyses of large next-generation sequencing datasets, including reads classification in genomics and the characterization of RNA-seq datasets. The extraction of all *k*-mers and their frequencies is extremely demanding in terms of running time and memory, owing to the size of the data and to the exponential number of *k*-mers to be considered. However, in several applications, only *frequent* *k*-mers, which are *k*-mers appearing in a relatively high proportion of the data, are required by the analysis.

Results: In this work, we present SPRISS, a new efficient algorithm to approximate frequent *k*-mers and their frequencies in next-generation sequencing data. SPRISS uses a simple yet powerful reads sampling scheme, which allows to extract a representative subset of the dataset that can be used, in combination with any *k*-mer counting algorithm, to perform downstream analyses in a fraction of the time required by the analysis of the whole data, while obtaining comparable answers. Our extensive experimental evaluation demonstrates the efficiency and accuracy of SPRISS in approximating frequent *k*-mers, and shows that it can be used in various scenarios, such as the comparison of metagenomic datasets, the identification of discriminative *k*-mers, and SNP (single nucleotide polymorphism) genotyping, to extract insights in a fraction of the time required by the analysis of the whole dataset.

Availability and implementation: SPRISS [a preliminary version (Santoro *et al.*, 2021) of this work was presented at RECOMB 2021] is available at https://github.com/VandinLab/SPRISS.

Contact: fabio.vandin@unipd.it

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The study of substrings of length k, or k-mers, is a fundamental task in the analysis of large next-generation sequencing datasets. The extraction of k-mers, and of the frequencies with which they appear in a dataset of reads, is a crucial step in several applications, including the comparison of datasets and reads classification in metagenomics (Wood and Salzberg, 2014), the characterization of variation in RNA-seq data (Audoux *et al.*, 2017), the analysis of structural changes in genomes (Liu *et al.*, 2017; Li and Waterman, 2003), RNA-seq quantification (Patro *et al.*, 2014; Zhang and Wang, 2014), fast search-by-sequence over large high-throughput sequencing repositories (Solomon and Kingsford, 2016), genome comparison (Sims *et al.*, 2009) and error correction for genome assembly (Kelley *et al.*, 2010; Salmela *et al.*, 2016).

k-mers and their frequencies can be obtained with a linear scan of a dataset. However, due to the massive size of the modern datasets and the exponential growth of the k-mers number (with respect to k), the extraction of k-mers is an extremely computationally intensive task, both in terms of running time and memory (Elworth *et al.*, 2020), and several algorithms have been proposed to reduce the running time and memory requirements (see Section 1.2). Nonetheless, the extraction of all k-mers and their frequencies from a reads dataset is still highly demanding in terms of time and memory [e.g. KMC 3 (Kokot *et al.*, 2017), one of the currently best performing tools for k-mer counting, requires more than 2.5 hours, 34 GB of memory and 500 GB of space on disk on a sequence of 729 Gbases (Kokot *et al.*, 2017), and from our experiments more than 30 minutes, 300 GB of memory and 97 GB of disk space for counting k-mers from Mo17 dataset (Using $k = 31$, 32 workers, and maximum RAM of 350 GB. See Supplementary Table S2 for the size of Mo17.)].

While some applications, such as error correction (Kelley *et al.*, 2010; Salmela *et al.*, 2016) or reads classification (Wood and Salzberg, 2014), require to identify *all* k-mers, even the ones that appear only once or few times in a dataset, other analyses, such as the comparison of abundances in metagenomic datasets (Benoit *et al.*, 2016; Danovaro *et al.*, 2017; Dickson *et al.*, 2017; Pellegrina *et al.*,

2020) or the discovery of k-mers discriminating between two datasets (Liu *et al.*, 2017; Ounit *et al.*, 2015), hinge on the identification of *frequent* k-mers, which are k-mers appearing with a (relatively) high frequency in a dataset. For the latter analyses, tools capable of efficiently extracting frequent k-mers only would be extremely beneficial and much more efficient than tools reporting all k-mers (given that a large fraction of k-mers appear with extremely low frequency). However, the efficient identification of frequent k-mers and their frequencies is still relatively unexplored (see Section 1.2).

A natural approach to speed-up the identification of frequent k-mers is to analyze only a *sample* of the data, since frequent k-mers appear with high probability in a sample, while unfrequent k-mers appear with lower probability. A major challenge in sampling approaches is how to rigorously relate the results obtained analyzing the sample and the results that would be obtained analyzing the whole dataset. Tackling such challenge requires to identify a minimum sample size which guarantees that the results on the sample well represent the results to be obtained on the whole dataset. An additional challenge in the use of sampling for the identification of frequent k-mers is due to the fact that, for values of k of interest in modern applications (e.g. $k \in [20, 60]$), even the most frequent k-mers appear in a relatively low portion of the data (e.g. 10^{-7}–10^{-5}). The net effect is that the application of standard sampling techniques to rigorously approximate frequent k-mers results in sample sizes *larger* than the initial dataset.

1.1 Our contributions

In this work, we study the problem of approximating frequent k-mers in a dataset of reads. In this regard, our contributions are:

- We propose SPRISS, SamPling Reads algorIthm to eStimate frequent k-merS (https://vec.wikipedia.org/wiki/Spriss). SPRISS is based on a simple yet powerful read sampling approach, which renders SPRISS very flexible and suitable to be used in combination with *any* k-mer counter. In fact, the read sampling scheme of SPRISS returns a subset of a dataset of reads, which can be used to obtain representative results for down-stream analyses based on frequent k-mers.
- We prove that SPRISS provides rigorous guarantees on the quality of the approximation of the frequent k-mers. In this regard, our main technical contribution is the derivation of the sample size required by SPRISS, obtained through the study of the pseudodimension (Pollard, 1984), a key concept from statistical learning theory, of k-mers in reads.
- We show on several real datasets that SPRISS approximates frequent k-mers with high accuracy, while requiring a fraction of the time needed by approaches that analyze all k-mers in a dataset.
- We show the benefits of using the approximation of frequent k-mers obtained by SPRISS in three applications: the comparison of metagenomic datasets, the extraction of discriminative k-mers and SNP genotyping. In all these applications, SPRISS significantly speeds up the analysis, while providing the same insights obtained by the analysis of the whole data.

1.2 Related works

The problem of exactly counting k-mers in datasets has been extensively studied, with several methods proposed for its solution (Audano and Vannberg, 2014; Kokot *et al.*, 2017; Kurtz *et al.*, 2008; Marçais and Kingsford, 2011; Melsted and Pritchard, 2011; Pandey *et al.*, 2017; Rizk *et al.*, 2013; Roy *et al.*, 2014). Such methods are typically highly demanding in terms of time and memory when analyzing large high-throughput sequencing datasets (Elworth *et al.*, 2020). For this reason, many methods have been recently developed to compute approximations of the k-mers abundances to reduce the computational cost of the task (e.g. Chikhi and Medvedev, 2014; Melsted and Halldórsson, 2014; Mohamadi *et al.*, 2017; Pandey *et al.*, 2017; Sivadasan *et al.*, 2016; Zhang *et al.*, 2014). However, such methods do not provide guarantees on the accuracy of their approximations that are simultaneously valid for all (or the most frequent) k-mers. In recent years, other problems closely related to the task of counting k-mers have been studied, including how to efficiently index (Harris and Medvedev, 2020; Marchet *et al.*, 2020a,b; Pandey *et al.*, 2018), represent (Almodaresi *et al.*, 2018; Chikhi *et al.*, 2014; Dadi *et al.*, 2018; Guo *et al.*, 2021; Holley and Melsted, 2020; Marchet *et al.*, 2019; Rahman and Medvedev, 2020), query (Bradley *et al.*, 2019; Marchet *et al.*, 2021; Solomon and Kingsford, 2016, 2018; Sun *et al.*, 2018; Yu *et al.*, 2018) and store (Hernaez *et al.*, 2019; Hosseini *et al.*, 2016; Numanagić *et al.*, 2016; Rahman *et al.*, 2021) the massive collections of sequences or of k-mers that are extracted from the data. See also Chikhi *et al.* (2021) for a unified presentation of methods to store and query a set of k-mers.

A natural approach to reduce computational demands is to analyze a small sample instead of the entire dataset. To this end, methods that perform a downsampling of massive datasets have been recently proposed (Brown *et al.*, 2012; Coleman *et al.*, 2019; Wedemeyer *et al.*, 2017). These methods focus on discarding reads of the datasets that are very similar to the reads already included in the sample, computing approximate similarity measures as each read is considered. Such measures (i.e. the Jaccard similarity) are designed to maximize the diversity of the content of the reads in the sample. This approach is well suited for applications where rare k-mers are important, but they are less relevant for analyses, of interest to this work, where the most frequent k-mers carry the major part of the information. Furthermore, these methods have a heuristic nature, and do not provide guarantees on the relation between the accuracy of the analysis performed on the sample w.r.t. the analysis performed on the entire dataset. SAKEIMA (Pellegrina *et al.*, 2020) is the first sampling method that provides an approximation of the set of frequent k-mers (together with their estimated frequencies) with rigorous guarantees, based on counting only a subset of all occurrences of k-mers, chosen at random. SAKEIMA performs a full scan of the entire dataset, in a streaming fashion, and processes each k-mer occurrence according to the outcome of its random choices. SPRISS, the algorithm we present in this work, is instead the first sampling algorithm to approximate frequent k-mers (and their frequencies), with rigorous guarantees, by sampling *reads* from the dataset. In fact, SPRISS does not require to receive in input and to scan the entire dataset, but, instead, it needs in input only a small sample of reads drawn from the dataset, sample that may be obtained, for example, at the time of the physical creation of the whole dataset. While the sampling strategy of SAKEIMA could be analyzed using the concept of *VC dimension* (Vapnik, 1998), the reads-sampling strategy of SPRISS requires the more sophisticated concept of *pseudodimension* (Pollard, 1984), for its analysis.

In this work, we consider the use of SPRISS to speed up the computation of the Bray-Curtis distance between metagenomic datasets, the identification of discriminative k-mers and the SNP genotyping process. Computational tools for these problems have been recently proposed (Benoit *et al.*, 2016; Saavedra *et al.*, 2020; Sun and Medvedev, 2018). These tools are based on exact k-mer counting strategies, and the approach we propose with SPRISS could be applied to such strategies as well.

2 Preliminaries

Let Σ be an alphabet of σ symbols. A dataset $\mathcal{D} = \{r_1, \ldots, r_n\}$ is a bag of $|\mathcal{D}| = n$ *reads*, where, for $i \in \{1, \ldots, n\}$, a read r_i is a string of length n_i built from Σ. For a given integer k, a *k-mer* K is a string of length k on Σ, that is $K \in \Sigma^k$. Given a k-mer K, a read r_i of $\mathcal{D}$ and a position $j \in \{0, \ldots, n_i - k\}$, we define the indicator function $\phi_{r_i,K}(j)$ to be 1 if K *appears* in r_i at position j, that is $K[h] = r_i[j + h]\ \forall h \in \{0, \ldots, k - 1\}$, while $\phi_{r_i,K}(j)$ is 0 otherwise. The size $t_{\mathcal{D},k}$ of the multiset of k-mers that appear in $\mathcal{D}$ is $t_{\mathcal{D},k} = \sum_{r_i \in \mathcal{D}} (n_i - k + 1)$. The average size of the multiset of k-mers that appear in a read of $\mathcal{D}$ is $g_{\mathcal{D},k} = t_{\mathcal{D},k}/n$, while the maximum value

of such quantity is $g_{\max,\mathcal{D},k} = \max_{r_i \in \mathcal{D}}(n_i - k + 1)$. The *support* $o_{\mathcal{D}}(K)$ of k-mer K in dataset $\mathcal{D}$ is the number of distinct positions of $\mathcal{D}$ where k-mer K appears, that is $o_{\mathcal{D}}(K) = \sum_{r_i \in \mathcal{D}} \sum_{j=0}^{n_i-k} \phi_{r_i,K}(j)$. The *frequency* $f_{\mathcal{D}}(K)$ of a k-mer K in $\mathcal{D}$ is the fraction of all positions in $\mathcal{D}$ where K appears, that is $f_{\mathcal{D}}(K) = o_{\mathcal{D}}(K)/t_{\mathcal{D},k}$.

The task of finding *frequent k-mers* (FKs) is defined as follows: given a dataset $\mathcal{D}$, a positive integer k and a *minimum frequency threshold* $\theta \in (0,1]$, find the set $FK(\mathcal{D},k,\theta)$ of all the k-mers whose frequency in $\mathcal{D}$ is at least θ, and their frequencies, that is $FK(\mathcal{D},k,\theta) = \{(K, f_{\mathcal{D}}(K)) : K \in \Sigma^k, f_{\mathcal{D}}(K) \geq \theta\}$.

The set of frequent k-mers can be computed by scanning the dataset and counting the number of occurrences for each k-mers. However, when dealing with a massive dataset $\mathcal{D}$, the exact computation of the set $FK(\mathcal{D},k,\theta)$ requires large amount of time and memory. For this reason, one could instead focus on finding an *approximation* of $FK(\mathcal{D},k,\theta)$ with rigorous guarantees on its quality. In this work, we consider the following approximation, introduced in (Pellegrina *et al.*, 2020).

Definition 1.*Given a dataset $\mathcal{D}$, a positive integer k, a frequency threshold $\theta \in (0,1]$, and an accuracy parameter $\varepsilon \in (0,\theta)$, an ε-approximation $\mathcal{C} = \{(K, f_K) : K \in \Sigma^k, f_K \in [0,1]\}$ of $FK(\mathcal{D},k,\theta)$ is a set of pairs (K, f_K) with the following properties:*

- *$\mathcal{C}$ contains a pair (K, f_K) for every $(K, f_{\mathcal{D}}(K)) \in FK(\mathcal{D},k,\theta)$;*
- *$\mathcal{C}$ contains no pair (K, f_K) such that $f_{\mathcal{D}}(K) < \theta - \varepsilon$;*
- *for every $(K, f_K) \in \mathcal{C}$, it holds $|f_{\mathcal{D}}(K) - f_K| \leq \varepsilon/2$.*

Intuitively, the approximation $\mathcal{C}$ contains no *false negatives* (i.e. all the frequent k-mers in $FK(\mathcal{D},k,\theta)$ are in C) and no k-mer whose frequency in $\mathcal{D}$ is much smaller than θ. In addition, the frequencies in $\mathcal{C}$ are good approximations of the actual frequencies in $\mathcal{D}$, i.e. within a small error $\varepsilon/2$.

Definition 2.*Given a dataset $\mathcal{D}$ of n reads, we define a reads sample S of $\mathcal{D}$ as a bag of m reads, sampled independently and uniformly at random, with replacement, from the bag of reads in $\mathcal{D}$.*

A natural way to compute an approximation of the set of frequent k-mers is by processing a *sample*, i.e. a small portion of the dataset $\mathcal{D}$, instead of the whole dataset. While previous work (Pellegrina *et al.*, 2020) considered samples obtained by drawing k-mers independently from $\mathcal{D}$, we consider samples obtained by drawing entire *reads*. Note that the development of an efficient scheme to effectively approximate the frequency of all frequent k-mers by sampling reads is highly non-trivial, due to dependencies among k-mers appearing in the same read. As explained in Section 1.1, our approach has several advantages, including the vfact that it can be combined with any efficient k-mer counting procedure, and that it can be used to extract a *representative* subset of the data on which to conduct down-stream analyses obtaining, in a fraction of the time required to process the whole dataset, the same insights. Such representative subsets could be stored and used for exploratory analyses, with a gain in terms of space and time requirements compared to using the whole dataset. In addition, note that SPRISS can approximate both canonical and non-canonical k-mers.

3 Method and algorithm

In this section, we develop and analyze our algorithm SPRISS, the first efficient algorithm to approximate frequent k-mers by read sampling.

Let $\mathcal{D}$ be a bag of n reads. We define $I_\ell = \{i_1, i_2, \ldots, i_\ell\}$ as a *bag* of ℓ indexes of reads of $\mathcal{D}$ chosen uniformly at random, with replacement, from the set $\{1,\ldots,n\}$. Then we define an *ℓ-reads sample* S_ℓ as a collection of m bags of ℓ reads $S_\ell = \{I_{\ell,1}, \ldots, I_{\ell,m}\}$. Let k be a positive integer. Define the domain X as the set of bags of ℓ indexes of reads of $\mathcal{D}$. Then define the family of real-valued functions $\mathcal{F} = \{f_{K,\ell}, \forall K \in \Sigma^k\}$ where, for every $I_\ell \in X$ and for every $f_{K,\ell} \in \mathcal{F}$, we have $f_{K,\ell}(I_\ell) = \min(1, o_{I_\ell}(K))/(\ell g_{\mathcal{D},k})$, where $o_{I_\ell}(K) = \sum_{i \in I_\ell} \sum_{j=0}^{n_i-k} \phi_{r_i,K}(j)$ counts the number of occurrences of K in all the ℓ reads of I_ℓ. Therefore, $f_{K,\ell}(I_\ell) \in \left\{0, \frac{1}{\ell g_{\mathcal{D},k}}\right\}$ $\forall f_{K,\ell}$ and $\forall I_\ell$. Note that, for a given bag I_ℓ, the functions $f_{K,\ell}$ have value equal to $1/\ell g_{\mathcal{D},k}$ even if K appears more than once in all the ℓ reads of I_ℓ, thus ignoring multiple occurrences of K in the bag. We define the frequency $f_{S_\ell}(K)$ of a k-mer K obtained from the sample S_ℓ of bags of reads as $f_{S_\ell}(K) = \frac{1}{m}\sum_{I_{\ell,i} \in S_\ell} o_{I_\ell}(K)/(\ell g_{\mathcal{D},k})$, which is an unbiased estimator of $f_{\mathcal{D}}(K)$ (i.e. $\mathbb{E}[f_{S_\ell}(K)] = f_{\mathcal{D}}(K)$). While the unbiased estimate $f_{S_\ell}(K)$ is the frequency reported by SPRISS as the estimated frequency of a k-mer K, SPRISS selects the k-mers to produce in output using a different estimate, namely $\hat{f}_{S_\ell}(K) = \frac{1}{m}\sum_{I_{\ell,i} \in S_\ell} f_{K,\ell}(I_{\ell,i})$, which is a 'biased' version of $f_{S_\ell}(K)$ since multiple occurrences of K in a bag are ignored. For the technical motivation to use the biased frequency $\hat{f}_{S_\ell}(K)$, see the analysis in Supplementary Section S3.

Our algorithm SPRISS (Algorithm 1) starts by computing the number m of bags of ℓ reads as in Equation (1), based on the input parameters $k, \theta, \delta, \varepsilon, \ell$ and on the characteristics $(g_{\mathcal{D},k}, g_{\max,\mathcal{D},k}, \sigma)$ of dataset $\mathcal{D}$. It then draws a sample S of exactly $m\ell$ reads, uniformly and independently at random, with replacement, from $\mathcal{D}$. Next, it computes for each k-mer K the number of occurrences $o_S(K)$ of K in sample S, using any exact k-mers counting algorithm. We denote the call of this method by `exact_counting`(S, k), which returns a collection T of pairs $(K, o_S(K))$. The sample S is then randomly partitioned into m bags, where each bag contains exactly ℓ reads. For each k-mer K, SPRISS computes the biased frequency $\hat{f}_{S_\ell}(K)$ and the unbiased frequency $f_{S_\ell}(K)$, reporting in output only k-mers with biased frequency at least $\theta - \varepsilon/2$. Note that, the estimated frequency of a k-mer K reported in output is always given by the unbiased frequency $f_{S_\ell}(K)$.

SPRISS (Algorithm 1) is motivated by our main technical result, Proposition 1, which establishes a rigorous relation between the number m of bags of ℓ reads and the guarantees obtained by approximating the frequency $f_{\mathcal{D}}(K)$ of a k-mer K with its (biased) estimate $\hat{f}_{S_\ell}(K)$ (the full analysis is in Supplementary Section S3—see Supplementary Proposition S13).

Proposition 1. *Let k and ℓ be two positive integers. Consider a sample S_ℓ of m bags of ℓ reads from $\mathcal{D}$. For fixed frequency threshold $\theta \in (0,1]$, error parameter $\varepsilon \in (0,\theta)$ and confidence parameter $\delta \in (0,1)$, if*

$$m \geq \frac{2}{\varepsilon^2}\left(\frac{1}{\ell g_{\mathcal{D},k}}\right)^2\left(\lfloor \log_2 \min(2\ell g_{\max,\mathcal{D},k}, \sigma^k)\rfloor + \ln\left(\frac{1}{\delta}\right)\right) \quad (1)$$

then, with probability at least $1-\delta$:

i. *for any k-mer $K \in FK(\mathcal{D},k,\theta)$ such that $f_{\mathcal{D}}(A) \geq \tilde{\theta} = \frac{g_{\max,\mathcal{D},k}}{g_{\mathcal{D},k}}(1-(1-\ell g_{\mathcal{D},k}\theta)^{1/\ell})$ it holds $\hat{f}_{S_\ell}(K) \geq \theta - \varepsilon/2$;*
ii. *for any k-mer K with $\hat{f}_{S_\ell}(K) \geq \theta - \varepsilon/2$ it holds $f_{\mathcal{D}}(K) \geq \theta - \varepsilon$;*
iii. *for any k-mer $K \in FK(\mathcal{D},k,\theta)$ it holds $f_{\mathcal{D}}(K) \geq \hat{f}_{S_\ell}(K) - \varepsilon/2$;*
iv. *for any k-mer K with $\ell g_{\mathcal{D},k}(\hat{f}_{S_\ell}(K) + \varepsilon/2) \leq 1$ it holds $f_{\mathcal{D}}(K) \leq \frac{g_{\max,\mathcal{D},k}}{g_{\mathcal{D},k}}(1-(1-\ell g_{\mathcal{D},k}(\hat{f}_{S_\ell}(K) + \varepsilon/2))^{(1/\ell)})$.*

SPRISS builds on Proposition 1, and returns the approximation of $FK(\mathcal{D},k,\theta)$ defined by the set $A = \{(K, f_{S_\ell}(K)) : \hat{f}_{S_\ell}(K) \geq \theta - \varepsilon/2\}$. Therefore, with probability at least $1-\delta$ the output of SPRISS provides the guarantees stated in Proposition 1. Note that, given a sample S_ℓ of m bags of ℓ reads from $\mathcal{D}$, with m satisfying the condition of Proposition 1, the set A is *almost* an ε-approximation of $FK(\mathcal{D},k,\theta)$: Proposition 1 ensures that all k-mers in A have frequency $f_{\mathcal{D}}(K) \geq \theta - \varepsilon$ with probability at least $1-\delta$, but it does not guarantee that all k-mers with frequency $\in [\theta, \tilde{\theta})$ will be in output. However, we show in Section 4.2 that, in practice, almost all of them are reported in output by SPRISS. Furthermore, we remark that it is possible to obtain different guarantees on the approximation computed by SPRISS by modifying the criteria used to report k-mers in output; for example, in some

applications, *perfect recall* may be particularly important. To this aim, we note that by reporting all k-mers with upper bound $\geq \theta$ (where the upper bound to $f_{\mathcal{D}}(K)$ is given by (*iv*) in Proposition 1), we obtain that all frequent k-mers are in the approximation, with relaxed guarantees on the precision (i.e. some k-mers with frequency $< \theta - \varepsilon$ may be in the output). Moreover, in applications in which obtaining tight *confidence intervals* on all exact frequencies $f_{\mathcal{D}}(K)$ is important, an approximation scheme based on using multiple values of ℓ, analogous to the one described in Section 3.3 of Pellegrina *et al.* (2020), is directly applicable to SPRISS.

In practice, in Algorithm 1, the partition of S into m bags and the computation of S_K could be highly demanding in terms of running time and space, since one has to compute and store, for each k-mer K, the exact number S_K of bags where K appears at least once among all reads of the bag. We now describe a much more efficient approach to approximate the values S_K, without the need to explicitly compute the bags. The number of reads in a given bag where K appears is well approximated by a Poisson distribution $Poisson(R[K]/m)$, where $R[K]$ is the number of reads of S where k-mer K appears at least once. Therefore, the number S_K of bags where K appears at least once is approximated by a binomial distribution $Binomial(m, 1 - e^{-R[K]/m})$. Thus, one can avoid to explicitly create the bags and to exactly count S_K, by replacing line '$\hat{f}_{S_\ell}(K) \leftarrow S_K/(m\ell g_{\mathcal{D},k})$' with '$\hat{f}_{S_\ell}(K) \leftarrow Binomial(m, 1 - e^{-R[K]/m})/(m\ell g_{\mathcal{D},k})$'. Corollary 5.11 of Mitzenmacher and Upfal (2017) guarantees that, by using this Poisson distribution to approximate S_K, the output of SPRISS satisfies the properties of Proposition 1 with probability at least $1 - 2\delta$. This leads to the replacement of '$\ln(1/\delta)$' with '$\ln(2/\delta)$' in the computation of m.

However, the approach described above requires to compute, for each k-mer K, the number of reads $R[K]$ of S where K appears at least once. We believe such computation can be obtained with minimal effort within the implementation of most k-mer counters, but we now describe a simple way to approximate $R[K]$. Since most k-mers appear at most once in a read, the number of reads $R[K]$ where a k-mer K appears is well approximated by the number of occurrences $T[K]$ of K in the sample S. Thus, instead of using "$\hat{f}_{S_\ell}(K) \leftarrow Binomial(m, 1 - e^{-R[K]/m})/(m\ell g_{\mathcal{D},k})$" we can replace it with '$\hat{f}_{S_\ell}(K) \leftarrow Binomial(m, 1 - e^{-T[K]/m})/(m\ell g_{\mathcal{D},k})$', which only requires the counts $T[K]$ obtained from the exact counting procedure `exact_counting(S, k)` (see Algorithm S2 in Supplementary Material). Note that approximating $R[K]$ with $T[K]$ leads to overestimating the frequencies of few k-mers who reside in very repetitive sequences, e.g. k-mers composed by the same k consecutive nucleotides, for which $T[K] \gg R[K]$. However, since the majority of k-mers reside in non-repetitive sequences, we can assume $R[K] \approx T[K]$.

Algorithm 1: SPRISS$(\mathcal{D}, k, \theta, \delta, \varepsilon, \ell)$
Data: $\mathcal{D}$, k, $\theta \in (0, 1]$, $\delta \in (0, 1)$, $\varepsilon \in (0, \theta)$, integer $\ell \geq 1$
Result: Approximation A of $FK(\mathcal{D}, k, \theta)$ with probability at least $1 - \delta$

$$\mathrm{m} \leftarrow \left\lceil \frac{2}{\varepsilon^2} \left(\frac{1}{\ell g_{\mathcal{D},k}} \right)^2 \left(\left\lfloor \log_2 \min(2\ell g_{\max,\mathcal{D},k}, \sigma^k) \right\rfloor + \ln\left(\frac{1}{\delta}\right) \right) \right\rceil;$$

$S \leftarrow$ sample of exactly $m\ell$ reads drawn from $\mathcal{D}$;

$T \leftarrow$ `exact_counting(S, k)`;

$S_\ell \leftarrow$ random partition of S into m bags of ℓ reads each;

$A \leftarrow \varnothing$;

for all the $(K, o_S(K)) \in T$ **do**
  $S_K \leftarrow$ number of bags of S_ℓ where K appears;
  $\hat{f}_{S_\ell}(K) \leftarrow S_K/(m\ell g_{\mathcal{D},k})$;
  $f_{S_\ell}(K) \leftarrow o_S(K)/(m\ell g_{\mathcal{D},k})$;
  if $\hat{f}_{S_\ell}(K) \geq \theta - \varepsilon/2$ **then** $A \leftarrow A \cup (K, f_{S_\ell}(K))$

return A;

4 Experimental evaluation

In this section, we present the results of our experimental evaluation. In particular:

- We assess the performance of SPRISS in approximating the set of frequent k-mers from a dataset of reads. In particular, we evaluate the accuracy of estimated frequencies and false negatives in the approximation, and compare SPRISS with the state-of-the-art sampling algorithm SAKEIMA (Pellegrina *et al.*, 2020) in terms of sample size and running time.
- We evaluate SPRISS's performance for the comparison of metagenomic datasets. We use SPRISS's approximations to estimate abundance-based distances (e.g. the Bray-Curtis distance) between metagenomic datasets, and show that the estimated distances can be used to obtain informative clusterings of metagenomic datasets from the Sorcerer II Global Ocean Sampling Expedition (Rusch *et al.*, 2007) (https://www.imicrobe.us) in a fraction of the time required by the exact distances computation (i.e. based on exact k-mers frequencies).
- We test SPRISS to discover discriminative k-mers between pairs of datasets. We show that SPRISS identifies almost all discriminative k-mers from pairs of metagenomic datasets rom (Liu *et al.*, 2017) and the Human Microbiome Project (HMP) (https://hmpdacc.org/HMASM/), with a significant speed-up compared to standard approaches.
- We evaluate SPRISS for approximate SNP genotyping, by combining the sampling scheme of SPRISS with previously proposed genotyping algorithms. We show that we achieve accurate approximations of the most common performance measures (precision, sensitivity and F-measure), obtaining a significant speed-up of the genotyping process.

4.1 Implementation, datasets, parameters and environment

We implemented SPRISS as a combination of C++ scripts, which perform the reads sampling and save the sample on a file, and as a modification of KMC 3 (Kokot *et al.*, 2017) (available at https://github.com/refresh-bio/KMC), a fast and efficient counting k-mers algorithm. We used KMC 3 with the default option to count canonical k-mers. Note that our flexible sampling technique can be combined with any k-mer counting algorithm. [See Supplementary Material for results, e.g. Supplementary Figure S1, obtained using JELLYFISH v. 2.3 (available at https://github.com/gmarcais/Jellyfish) as k-mer counter in SPRISS.] We use the variant of SPRISS that employs the Poisson approximation for computing S_K (see end of Section 3). SPRISS implementation, information about how to retrieve the data used in this work, and scripts for reproducing all results are publicity available (available at https://github.com/VandinLab/SPRISS). We compared SPRISS with the exact k-mer counter KMC and with SAKEIMA (Pellegrina *et al.*, 2020) (available at https://github.com/VandinLab/SAKEIMA), the state-of-the-art sampling-based algorithm for approximating frequent k-mers. In all experiments we fix $\delta = 0.1$ and $\varepsilon = \theta - 2/t_{\mathcal{D},k}$. If not stated otherwise, we considered $k = 31$ and $\ell = \lfloor 0.9/(\theta g_{\mathcal{D},k}) \rfloor$ in our experiments. For SAKEIMA, as suggested in Pellegrina *et al.* (2020) we set the number g_{SK} of k-mers in a bag to be $g_{SK} = \lfloor 0.9/\theta \rfloor$. We remark that a bag of reads of SPRISS contains the same (expected) number

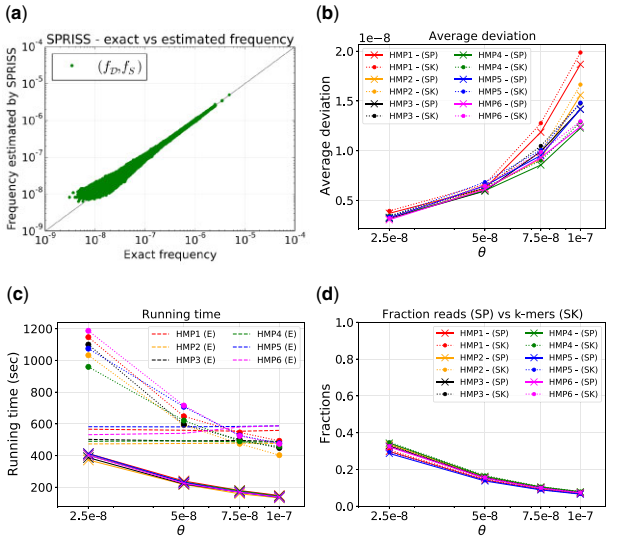

Fig. 1. (**a**) k-mers exact frequency and frequency estimated by SPRISS for dataset SRS024075 and $\theta = 2.5\times10^{-8}$. (**b**) Average deviations between exact frequencies and frequencies estimated by SPRISS (SP) and SAKEIMA (SK), for various datasets and values of θ. (**c**) Running time of SPRISS (SP), SAKEIMA (SK) and the exact computation (E)—see also legend of (b). (**d**) Fraction of the dataset analyzed by SPRISS (SP) and by SAKEIMA (SK)

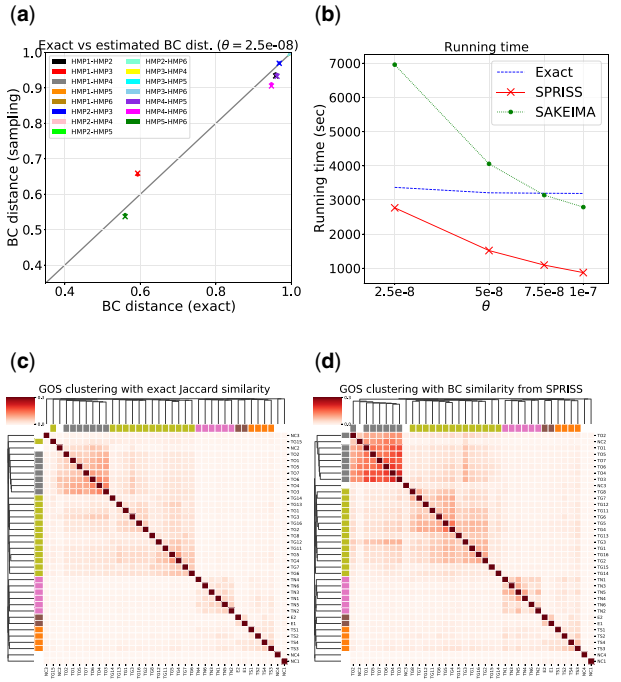

Fig. 2. (**a**) Comparison of the approximations of the Bray-Curtis (BC) distances using approximations of frequent k-mers provided by SPRISS (×) and by SAKEIMA (•), and the exact distances, for $\theta = 2.5\times10^{-8}$. (**b**) Running time to approximate BC distances for all pairs of datasets with SPRISS, with SAKEIMA and the exact approach. (**c**) Average linkage hierarchical clustering of GOS datasets using Jaccard similarity. (**d**) Same as (c), using estimated BC similarity from SPRISS with 50% of the data (see also larger Supplementary Figs S4–S6 for better readability of datasets' labels and computed clusters)

of k-mers positions of a bag of SAKEIMA; this guarantees that both algorithms provide outputs with the same guarantees, thus making the comparison between the two methods fair. To assess SPRISS in approximating frequent k-mers, we considered six large metagenomic datasets from HMP, each with $\approx10^8$ reads and average read length $\approx$100 (see Supplementary Table S1). For the evaluation of SPRISS in comparing metagenomic datasets, we also used 37 small metagenomic datasets from the Sorcerer II Global Ocean Sampling Expedition (Rusch *et al.*, 2007), each with $\approx10^4 - 10^5$ reads and average read length $\approx$1000 (see Supplementary Table S4). For the assessment of SPRISS in the discovery of discriminative k-mers we used two large datasets from (Liu *et al.*, 2017), B73 and Mo17, each with $\approx4\times10^8$ reads and average read length = 250 (see Supplementary Table S2), and we also experimented with the HMP datasets. To evaluate the benefits of using SPRISS for SNP genotyping, we used an Illumina WGS dataset from NA12878, with $\approx1.55\times10^9$ reads and average read length = 148 (see Supplementary Table S3), available from the Genome In A Bottle (GIAB) consortium (Zook *et al.*, 2014). All experiments have been performed on a machine with 512 GB of RAM and 2 Intel(R) Xeon(R) CPU E5-2698 v3 at 2.3 GHz, with one worker, if not stated otherwise. All reported results are averages over five runs.

4.2 Approximation of frequent k-mers

In this section, we first assess the quality of the approximation of $FK(\mathcal{D}, k, \theta)$ provided by SPRISS, and then compare SPRISS with SAKEIMA.

We use SPRISS to extract approximations of frequent k-mers on six datasets from HMP for values of the minimum frequency threshold $\theta \in \{2.5\times10^{-8}, 5\times10^{-8}, 7.5\times10^{-8}, 10^{-7}\}$. The output of SPRISS satisfied the guarantees from Proposition 1 for all five runs of every combination of dataset and θ. In all cases the estimated frequencies provided by SPRISS are close to the exact ones (see Fig. 1a for an example). In fact, the average (across all reported k-mers) absolute deviation of the estimated frequency w.r.t. the true frequency is always small, i.e. one order of magnitude smaller than θ (Fig. 1b), and the maximum deviation is very small as well (Supplementary Fig. S2B). In addition, even if the values of $\tilde{\theta}$ [see *(i)* in Proposition 1] are always between 4.15×10^{-6} and 1.81×10^{-5}, SPRISS results in a very low false negative rate (i.e. fraction of k-mers of $FK(\mathcal{D}, k, \theta)$ not reported by SPRISS), which is always been below 0.012 in our experiments.

In terms of running time, SPRISS required at most 64% of the time required by the exact approach KMC (Fig. 1c). In addition, SPRISS used at most 30% of the RAM memory required by the exact approach KMC. This is due to SPRISS requiring to analyze at most 34% of the entire dataset (Fig. 1d). Note that the use of collections of bags of reads is crucial to achieve useful sample size, i.e. lower than the whole dataset. In fact, the sample sizes obtained from less sophisticated statistical tools, e.g. Hoeffding's inequality combined with union bound (see Supplementary Section S1), and pseudodimension without collections of bags (see Supplementary Section S2), are much greater than the dataset size:$\approx10^{16}$ and $\approx10^{15}$, respectively, which are useless sample sizes for datasets of $\approx10^8$ reads. These results show that SPRISS obtains very accurate approximations of frequent k-mers in a fraction of the time required by exact counting approaches.

We then compared SPRISS with SAKEIMA. In terms of quality of approximation, SPRISS reports approximations with an average deviation lower than SAKEIMA's approximations, while SAKEIMA's approximations have a lower maximum deviation. However, the ratio between the maximum deviation of SPRISS and the one of SAKEIMA are always below 2. Overall, the quality of the approximation provided by SPRISS and SAKEIMA are, thus, comparable. In terms of running time, SPRISS significantly improves over SAKEIMA (Fig. 1c), and processes slightly smaller portions of the dataset compared to SAKEIMA (Fig. 1d). Summarizing, SPRISS is able to report most of the frequent k-mers and estimate their frequencies with small errors, by analyzing small samples of the datasets and with significant improvements on running times compared to exact approaches and to state-of-the-art sampling algorithms.

4.3 Comparing metagenomic datasets

We evaluated SPRISS to compare metagenomic datasets by computing an approximation to the Bray-Curtis (BC) distance between pairs of datasets of reads, and using such approximations to cluster datasets.

Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two datasets of reads. Let $\mathcal{F}_1 = FK(\mathcal{D}_1, k, \theta)$ and $\mathcal{F}_2 = FK(\mathcal{D}_2, k, \theta)$ be the set of frequent k-mers, respectively, of $\mathcal{D}_1$ and $\mathcal{D}_2$, where θ is a minimum frequency threshold. The *BC distance* between $\mathcal{D}_1$ and $\mathcal{D}_2$ considering only frequent k-mers is defined as $BC(\mathcal{D}_1, \mathcal{D}_2, \mathcal{F}_1, \mathcal{F}_2) = 1 - 2I/U$, where $I = \sum_{K \in \mathcal{F}_1 \cap \mathcal{F}_2} \min\{o_{\mathcal{D}_1}(K), o_{\mathcal{D}_2}(K)\}$ and $U = \sum_{K \in \mathcal{F}_1} o_{\mathcal{D}_1}(K) + \sum_{K \in \mathcal{F}_2} o_{\mathcal{D}_2}(K)$. Conversely, the *BC similarity* is defined as $1 - BC(\mathcal{D}_1, \mathcal{D}_2, \mathcal{F}_1, \mathcal{F}_2)$.

We considered six datasets from HMP, and estimated the BC distances among them by using SPRISS to approximate the sets of frequent k-mers $\mathcal{F}_1 = FK(\mathcal{D}_1, k, \theta)$ and $\mathcal{F}_2 = FK(\mathcal{D}_2, k, \theta)$ for the values of θ as in Section 4.2. We compared such estimated distances with the exact BC distances and with the estimates obtained using SAKEIMA. Both SPRISS and SAKEIMA provide accurate estimates of the BC distances (Fig. 2a and Supplementary Fig. S3), which can be used to assess the relative similarity of pairs of datasets. However, to obtain such approximations SPRISS requires at most 40% of the time required by SAKEIMA and usually 30% of the time required by the exact computation with KMC (Fig. 2b). Therefore SPRISS provides accurate estimates of metagenomic distances in a fraction of time required by other approaches.

As an example of the impact in accurately estimating distances among metagenomic datasets, we used the sampling approach of SPRISS to approximate all pairwise BC distances among 37 small datasets from the Sorcerer II Global Ocean Sampling Expedition (GOS) (Rusch *et al.*, 2007), and used such distances to cluster the datasets using average linkage hierarchical clustering. The k-mer-based clustering of metagenomic datasets is often performed by using *presence-based* distances, such as the Jaccard distance (Ondov *et al.*, 2016), which estimates similarities between two datasets by computing the fraction of k-mers in common between the two datasets. Abundance-based distances, such as the BC distance (Benoit *et al.*, 2016; Danovaro *et al.*, 2017; Dickson *et al.*, 2017), provide more detailed measures based also on the k-mers abundance, but are often not used due to the heavy computational requirements to extract all k-mers counts. However, the sampling approach of SPRISS can significantly speed-up the computation of all BC distances, and, thus, the entire clustering analysis. In fact, for this experiment, the use of SPRISS reduces the time required to analyze the datasets (i.e. obtain k-mers frequencies, compute all pairwise distances and obtain the clustering) by 62%.

We then compared the clustering obtained using the Jaccard distance (Fig. 2c) and the clustering obtained using the estimates of the BC distances (Fig. 2d) obtained using only 50% of reads in the GOS datasets, which are assigned to groups and macro-groups according to the origin of the sample (Rusch *et al.*, 2007). Even if the BC distance is computed using only a sample of the datasets, while the Jaccard distance is computed using the entirety of all datasets, the use of approximate BC distances leads to a better clustering in terms of correspondence of clusters to groups, and to the correct cluster separation for macro-groups. In addition, the similarities among datasets in the same group and the dissimilarities among datasets in different groups are more accentuated using the approximated BC distance. In fact, the ratio between the average BC similarity among datasets in the same group and the analogous average Jaccard is in the interval $[1.25, 1.75]$ for all groups. In addition, the ratio between (i) the difference of the average BC similarity within the tropical macro-group and the average BC similarity between the tropical and temperate groups, and (ii) the analogous difference using the Jaccard similarity is ≈1.53. These results tell us the approximate BC-distances, computed using only half of the reads in each dataset, increase by ≈50% the similarity signal inside all groups defined by the original study (Rusch *et al.*, 2007), and the dissimilarities between the two macro-groups (tropical and temperate).

To conclude, the estimates of the BC similarities obtained using the sampling scheme of SPRISS allows to better cluster metagenomic datasets than using the Jaccard similarity, while requiring less than 40% of the time needed by the exact computation of BC similarities, even for fairly small metagenomic datasets.

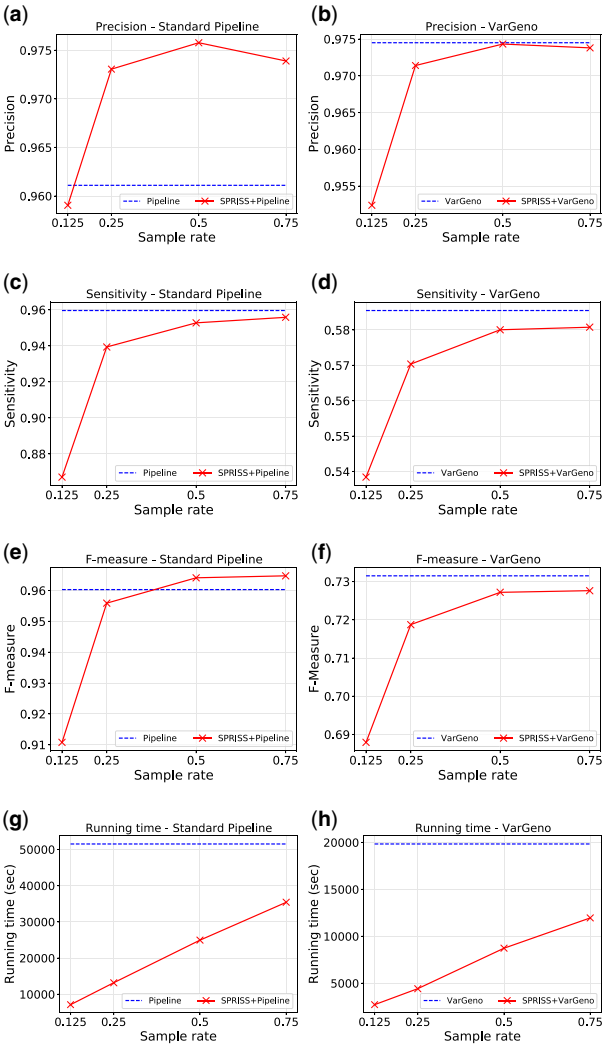

Fig. 3. As function of the sample rate, experimental results of combining SPRISS with VarGeno and the standard pipeline in the SNP genotyping process: VarGeno's precision (**a**), sensitivity (**c**) F-measure (**e**), running time (**g**) and standard pipeline's precision (**b**), sensitivity (**d**) F-measure (**f**), running time (**h**)

4.4 Approximation of discriminative *k*-mers

In this section, we assess SPRISS for approximating discriminative k-mers in metagenomic datasets. In particular, we consider the following definition of discriminative k-mers (Liu *et al.*, 2017). Given two datasets $\mathcal{D}_1, \mathcal{D}_2$, and a minimum frequency threshold θ, we define the set $DK(\mathcal{D}_1, \mathcal{D}_2, k, \theta, \rho)$ of $\mathcal{D}_1$-discriminative k-mers as the collection of k-mers K for which the following conditions both hold: (i) $K \in FK(\mathcal{D}_1, k, \theta)$; (ii) $f_{\mathcal{D}_1}(K) \geq \rho f_{\mathcal{D}_2}(K)$, with $\rho = 2$. Note that the computation of $DK(\mathcal{D}_1, \mathcal{D}_2, k, \theta, \rho)$ requires to extract $FK(\mathcal{D}_1, k, \theta)$ and $FK(\mathcal{D}_2, k, \theta/\rho)$. SPRISS can be used to approximate the set $DK(\mathcal{D}_1, \mathcal{D}_2, k, \theta, \rho)$, by computing approximations $\overline{FK}(\mathcal{D}_i, k, \theta)$ of the sets $FK(\mathcal{D}_i, k, \theta)$, $i = 1, 2$, of frequent k-mers in $\mathcal{D}_1, \mathcal{D}_2$, and then reporting a k-mer K as $\mathcal{D}_1$-discriminative if the following conditions both hold: (i) $K \in \overline{FK}(\mathcal{D}_1, k, \theta)$; (ii) $K \notin \overline{FK}(\mathcal{D}_2, k, \theta)$, or $f_{S_\ell^1}(K) \geq \rho f_{S_\ell^2}(K)$ when $K \in \overline{FK}(\mathcal{D}_2, k, \theta)$.

To evaluate such approach, we considered two datasets from (Liu *et al.*, 2017), and $\theta = 2 \times 10^{-7}$ and $\rho = 2$, which are the parameters used in (Liu *et al.*, 2017). We used the sampling approach of SPRISS with $\ell = \lfloor 0.02/(\theta g_{\mathcal{D},k}) \rfloor$ and $\ell = \lfloor 0.04/(\theta g_{\mathcal{D},k}) \rfloor$, resulting in

analyzing of 5% and 10% of all reads, to approximate the sets of discriminative $\mathcal{D}_1$-discriminative and of $\mathcal{D}_2$-discriminative k-mers. When 5% of the reads are used, the false negative rate is < 0.028, while when 10% of the reads are used, the false negative rate is < 0.018. The running times are $\approx$1130 and $\approx$1970 s, respectively, while the exact computation of the discriminative k-mers with KMC requires $\approx 10^4$ s (we used 32 workers for both SPRISS and KMC). Similar results are obtained when analyzing pairs of HMP datasets, for various values of θ (Supplementary Fig. S7). These results show that SPRISS can identify discriminative k-mers with small false negative rates while providing a remarkable improvement in running time compared to the exact approach.

4.5 SNP genotyping

In this section, we evaluate SPRISS for approximate SNP genotyping. In particular, we assess the use of SPRISS in combination with previously proposed algorithms for SNP genotyping in terms of precision, sensitivity and F-measure. The genotyping algorithms we used are the standard pipeline [BWA (Li and Durbin, 2009) as aligner, and BCFtools (Li, 2011) as variant caller], and VarGeno (Sun and Medvedev, 2018). We considered hg19 as reference genome, and dbSNP (Sherry, 2001) as reference SNP database. We used the gold standard of NA12878 individual provided by the Genome In A Bottle (GIAB) consortium (Zook *et al.*, 2014). The Illumina WGS dataset $\mathcal{D}$ of reads from NA12878 we used has a coverage of $\approx$75x. We used the sampling scheme of SPRISS to create samples of 12.5%, 25%, 50% and 75% of reads of $\mathcal{D}$. The standard pipeline was run with 64 threads. When evaluating the running time, we do not include the time to obtain the sample, since once the sample is created it can be reused several times. Moreover, the time to obtain the sample is always a small fraction of the overall execution time (e.g. even for a sample containing 75% of reads of $\mathcal{D}$ the required time is < 3000 s).

The performance measures of the standard pipeline on $\mathcal{D}$ are the following: 0.961 of precision, 0.959 of sensitivity and 0.960 of F-measure. Figure 3 describes the running times and the performance measures of the standard pipeline using samples of $\mathcal{D}$ from SPRISS. Considering a sample of just 25% of reads of $\mathcal{D}$, the sensitivity and the F-measure decrease, respectively, by 0.02 and 0.004, while the precision increases by 0.012. The increment of the precision is due to a decrement in the number of false positive calls, which is achieved by the reads sampling of SPRISS that filters out low coverage regions and erroneous k-mers. The speed-up of using a sample of 25% of reads of $\mathcal{D}$ instead of the entire dataset $\mathcal{D}$ is $\approx$3.9x.

VarGeno achieves on $\mathcal{D}$ 0.974 of precision, 0.585 of sensitivity and 0.731 of F-measure. With a sample from SPRISS of just 25% of reads of $\mathcal{D}$, we obtain a decrement of the performance of VarGeno of 0.003 in precision, 0.015 in sensitivity, 0.013 in F-measure and a speed-up of $\approx$4.5x with respect to the time required to analyze the entire dataset $\mathcal{D}$. The results for the other sample sizes are described in Figure 3.

To conclude, the sampling scheme of SPRISS is very useful to remarkably speed up genotyping algorithms, while achieving very small decrements in the performance measures, and even improving the precision in some cases.

5 Discussion

We presented SPRISS, an efficient algorithm to compute rigorous approximations of frequent k-mers and their frequencies by sampling reads. SPRISS builds on pseudodimension, an advanced concept from statistical learning theory. Our extensive experimental evaluation shows that SPRISS provides high-quality approximations and can be employed to speed-up exploratory analyses in various applications, such as the analysis of metagenomic datasets, the identification of discriminative k-mers and SNP genotyping. Overall, the sampling approach used by SPRISS provides an efficient way to obtain a representative subset of the data that can be used to perform complex analyses more efficiently than examining the whole data, while obtaining representative results.

Funding

Part of this work was supported by the Italian Ministry of Education, University and Research (MIUR), under PRIN Project No. 20174LF3T8 AHeAD (Efficient Algorithms for HArnessing Networked Data) and the initiative 'Departments of Excellence' (Law 232/2016); and University of Padova under project SEED 2020 RATED-X.

Conflict of Interest: none declared.

References

Almodaresi,F. *et al.* (2018) A space and time-efficient index for the compacted colored de Bruijn graph. *Bioinformatics*, **34**, i169–i177.
Audano,P. and Vannberg,F. (2014) Kanalyze: a fast versatile pipelined k-mer toolkit. *Bioinformatics*, **30**, 2070–2072.
Audoux,J. *et al.* (2017) De-kupl: exhaustive capture of biological variation in RNA-seq data through k-mer decomposition. *Genome Biol.*, **18**, 243.
Benoit,G. *et al.* (2016) Multiple comparative metagenomics using multiset k-mer counting. *PeerJ Comput. Sci.*, **2**, e94.
Bradley,P. *et al.* (2019) Ultrafast search of all deposited bacterial and viral genomic data. *Nat. Biotechnol.*, **37**, 152–159.
Brown,C.T. *et al.* (2012) A reference-free algorithm for computational normalization of shotgun sequencing data. *arXiv preprint arXiv:1203.4802*.
Chikhi,R. and Medvedev,P. (2014) Informed and automated k-mer size selection for genome assembly. *Bioinformatics*, **30**, 31–37.
Chikhi,R. *et al.* (2014) On the representation of de Bruijn graphs. In: *International Conference on Research in Computational Molecular Biology, RECOMB 2014*. Springer, Pittsburgh, PA, pp. 35–55.
Chikhi,R. *et al.* (2021) Data structures to represent a set of k-long DNA sequences. *ACM Comput. Surv*, **54**, 17.
Coleman,B. *et al.* (2019) Diversified race sampling on data streams applied to metagenomic sequence analysis. *bioRxiv*, p. 852889.
Dadi,T.H. *et al.* (2018) Dream-yara: an exact read mapper for very large databases with short update time. *Bioinformatics*, **34**, i766–i772.
Danovaro,R. *et al.* (2017) A submarine volcanic eruption leads to a novel microbial habitat. *Nat. Ecol. Evol.*, **1**, 0144.
Dickson,L.B. *et al.* (2017) Carryover effects of larval exposure to different environmental bacteria drive adult trait variation in a mosquito vector. *Sci. Adv.*, **3**, e1700585.
Elworth,R.L. *et al.* (2020) To petabytes and beyond: recent advances in probabilistic and signal processing algorithms and their application to metagenomics. *Nucleic Acids Res.*, **48**, 5217–5234.
Guo,H. *et al.* (2021) degsm: memory scalable construction of large scale de Bruijn graph. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **18**, 2157–2166.
Harris,R.S. and Medvedev,P. (2020) Improved representation of sequence bloom trees. *Bioinformatics*, **36**, 721–727.
Hernaez,M. *et al.* (2019) Genomic data compression. *Annu. Rev. Biomed. Data Sci.*, **2**, 19–37.
Holley,G. and Melsted,P. (2020) Bifrost: highly parallel construction and indexing of colored and compacted de Bruijn graphs. *Genome Biol.*, **21**, 1–20.
Hosseini,M. *et al.* (2016) A survey on data compression methods for biological sequences. *Information*, **7**, 56.
Kelley,D.R. *et al.* (2010) Quake: quality-aware detection and correction of sequencing errors. *Genome Biol.*, **11**, R116.
Kokot,M. *et al.* (2017) Kmc 3: counting and manipulating k-mer statistics. *Bioinformatics*, **33**, 2759–2761.
Kurtz,S. *et al.* (2008) A new method to compute k-mer frequencies and its application to annotate large repetitive plant genomes. *BMC Genomics*, **9**, 517.
Li,H. (2011) A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, **27**, 2987–2993.
Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, **25**, 1754–1760.
Li,X. and Waterman,M.S. (2003) Estimating the repeat structure and length of DNA sequences using ℓ-tuples. *Genome Res.*, **13**, 1916–1922.
Liu,S. *et al.* (2017) Unbiased k-mer analysis reveals changes in copy number of highly repetitive sequences during maize domestication and improvement. *Sci. Rep.*, **7**, 42444.
Marçais,G. and Kingsford,C. (2011) A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, **27**, 764–770.
Marchet,C. *et al.* (2019). Indexing de Bruijn graphs with minimizers. *bioRxiv*, p. 546309.

Marchet,C. *et al.* (2020a). Reindeer: efficient indexing of k-mer presence and abundance in sequencing datasets. *Bioinformatics*, **36**(Suppl. 1), i177–i185.
Marchet,C. *et al.* (2020b) A resource-frugal probabilistic dictionary and applications in bioinformatics. *Discrete Appl. Math.*, **274**, 92–102.
Marchet,C. *et al.* (2021) Data structures based on k-mers for querying large collections of sequencing datasets. *Genome Res.*, **31**, 1–12.
Melsted,P. and Halldórsson,B.V. (2014) Kmerstream: streaming algorithms for k-mer abundance estimation. *Bioinformatics*, **30**, 3541–3547.
Melsted,P. and Pritchard,J.K. (2011) Efficient counting of k-mers in DNA sequences using a bloom filter. *BMC Bioinformatics*, **12**, 333.
Mitzenmacher,M. and Upfal,E. (2017) *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, New York.
Mohamadi,H. *et al.* (2017) ntcard: a streaming algorithm for cardinality estimation in genomics data. *Bioinformatics*, **33**, 1324–1330.
Numanagić,I. *et al.* (2016) Comparison of high-throughput sequencing data compression tools. *Nat. Methods*, **13**, 1005–1008.
Ondov,B.D. *et al.* (2016) Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol.*, **17**, 132.
Ounit,R. *et al.* (2015) Clark: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics*, **16**, 236.
Pandey,P. *et al.* (2017) Squeakr: an exact and approximate k-mer counting system. *Bioinformatics*, **34**, 568–575.
Pandey,P. *et al.* (2018) Mantis: a fast, small, and exact large-scale sequence--search index. *Cell Syst.*, **7**, 201–207.
Patro,R. *et al.* (2014) Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nat. Biotechnol.*, **32**, 462–464.
Pellegrina,L. *et al.* (2020) Fast approximation of frequent k-mers and applications to metagenomics. *J. Comput. Biol.*, **27**, 534–549.
Pollard,D. (1984) *Convergence of Stochastic Processes*. Springer-Verlag, New York.
Rahman,A. and Medvedev,P. (2020) Representation of *k*-mer sets using spectrum-preserving string sets. In: *International Conference on Research in Computational Molecular Biology, RECOMB 2020*. Springer, Padua, Italy, pp. 152–168.
Rahman,A. *et al.* (2021) Disk compression of k-mer sets. *Algorithms Mol. Biol.*, **16**, 10.
Rizk,G. *et al.* (2013) Dsk: k-mer counting with very low memory usage. *Bioinformatics*, **29**, 652–653.
Roy,R.S. *et al.* (2014) Turtle: identifying frequent k-mers with cache-efficient algorithms. *Bioinformatics*, **30**, 1950–1957.
Rusch,D.B. *et al.* (2007) The sorcerer II global ocean sampling expedition: northwest Atlantic through eastern tropical pacific. *PLoS Biol.*, **5**, 1–34.
Saavedra,A. *et al.* (2020) Mining discriminative k-mers in DNA sequences using sketches and hardware acceleration. *IEEE Access*, **8**, 114715–114732.
Salmela,L. *et al.* (2016) Accurate self-correction of errors in long reads using de Bruijn graphs. *Bioinformatics*, **33**, 799–806.
Santoro,D. *et al.* (2021) Spriss: Approximating frequent k-mers by sampling reads, and applications. *arXiv preprint arXiv:2101.07117*.
Sherry,S.T. *et al.* (2001) dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res.*, **29**, 308–311.
Sims,G.E. *et al.* (2009) Alignment-free genome comparison with feature frequency profiles (FFP) and optimal resolutions. *Proc. Natl. Acad. Sci. USA*, **106**, 2677–2682.
Sivadasan,N. *et al.* (2016) Kmerlight: fast and accurate k-mer abundance estimation. *arXiv preprint arXiv:1609.05626*.
Solomon,B. and Kingsford,C. (2016) Fast search of thousands of short-read sequencing experiments. *Nat. Biotechnol.*, **34**, 300–302.
Solomon,B. and Kingsford,C. (2018) Improved search of large transcriptomic sequencing databases using split sequence bloom trees. *J. Comput. Biol.*, **25**, 755–765.
Sun,C. and Medvedev,P. (2018) Toward fast and accurate SNP genotyping from whole genome sequencing data for bedside diagnostics. *Bioinformatics*, **35**, 415–420.
Sun,C. *et al.* (2018) Allsome sequence bloom trees. *J. Comput. Biol.*, **25**, 467–479.
Vapnik,V. (1998). *Statistical Learning Theory*. Wiley, New York.
Wedemeyer,A. *et al.* (2017) An improved filtering algorithm for big read datasets and its application to single-cell assembly. *BMC Bioinformatics*, **18**, 324.
Wood,D.E. and Salzberg,S.L. (2014) Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol.*, **15**, R46.
Yu,Y. *et al.* (2018) Seqothello: querying RNA-seq experiments at scale. *Genome Biol.*, **19**, 167.
Zhang,Q. *et al.* (2014) These are not the k-mers you are looking for: efficient online k-mer counting using a probabilistic data structure. *PLoS One*, **9**, e101271.
Zhang,Z. and Wang,W. (2014) RNA-skim: a rapid method for RNA-seq quantification at transcript level. *Bioinformatics*, **30**, i283–i292.
Zook,J.M. *et al.* (2014) Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol.*, **32**, 246–251.