UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Head Office: Università degli Studi di Padova
Department: Information Engineering

Ph.D. Course: Information Engineering
Curriculum: Information and Communication Technologies (ICT)
Series: XXXV, 2023

Rigorous and Efficient Algorithms for Mining Sequential Data, and Applications to Bioinformatics

Coordinator**:** Prof. Andrea Neviani
Supervisor: Prof. Fabio Vandin

Ph.D. student: Diego Santoro

Abstract

Data mining is the task of extracting meaningful information, i.e., patterns and new knowledge, from massive data. Data mining finds application in several domains, ranging from, e.g., e-commerce, social networks, and Internet of Things, to medicine and biology. In this Thesis we focus on the analysis of a specific type of data, i.e., *sequential data*, that are data composed of elements with an underlying intrinsic sequential nature among them. In particular, we study large datasets of *sequential transactions*, i.e., sequences of sets of items, where the items can represent, e.g., objects purchased by customers, and large datasets of *k-mers*, which are substrings of length k of a biological sequence. In data mining, there are two ways in which a dataset can be considered. In the first scenario, the dataset is considered as a *sample* drawn from the *unknown generative process* underlying the data, and it is studied to gain insights about the generative distribution. In the second scenario, the most common one in data mining, the dataset is studied in order to extract meaningful patterns that reside in it. In this Thesis we consider both knowledge discovery approaches.

For the first scenario, we study datasets of sequential patterns in order to gain insights about their unknown generative process. The problem we consider is to mine the *true frequent sequential patterns*, which are those sequential patterns that are frequently generated by the generative process of the data. Since exact approaches are infeasible, given that the generative distribution is unknown, we propose algorithms to mine rigorous approximations of the true frequent sequential patterns (and their frequencies). Our algorithms are based on theoretical results that guarantee that their outputs are high-quality approximations, leveraging on the study of the *Rademacher complexity*, an advanced tool from statistical learning theory, of sequential patterns. We show the effectiveness of our algorithms with our extensive experimental evaluation on several real world datasets.

For the second scenario, we study datasets representing biological sequences in order to mine meaningful k-mers residing in them. The problem we consider is to mine *frequent k-mers*, which are those k-mers that appear with a relatively high frequency in the dataset. Exact approaches to extract frequent k-mers exist, but they require high computational resources to analyze large datasets. Thus, we propose SPRISS, a sampling-based algorithm to rigorously approximate frequent k-mers, proving rigorous guarantees on the quality of its output. Our sampling scheme is based on the creation of a

sample of reads, which is analyzed in order to gain insights about the original dataset. For our theoretical analysis, which provides the sample size required to obtain accurate estimates from SPRISS, we study the *pseudodimension*, an advanced concept from statistical learning theory, of k-mers in reads. We show the effectiveness of SPRISS with our extensive experimental evaluation on several real world datasets. Finally, we show that SPRISS can be used in several bioinformatics applications, like the comparison of metagenomic datasets, the discovery of discriminative k-mers, and the SNP genotyping, in order to speed-up the down-stream analyses, while achieving high-quality estimations of the exact results.

Acknowledgements

The end of this incredible adventure is here. It's impressive how things may appear impossible at the beginning, and then, step by step, they become possible. I feel really lucky and happy I have had the opportunity to start, almost three years ago, this Ph.D. path. Now, I want to thank the many people who helped me during this journey.

Prima di tutto, vorrei ringraziare Fabio Vandin, mio supervisore di dottorato. Dopo essere stato il mio relatore di tesi di laurea triennale, dopo aver accettato lo stesso ruolo anche per la laurea magistrale, hai avuto il coraggio di sopportarmi per ulteriori tre anni di dottorato! Ti ringrazio per avermi fatto scoprire questa possibile strada, per avermi incoraggiato a intraprenderla, e per i molti consigli che mi hai dato durante tutto il cammino.

Ringrazio Matteo e Andrea per il loro prezioso feedback sul mio lavoro durante le presentazioni di fine anno. Inoltre, volevo ringraziare i miei colleghi Andrea, Antonio, Dario, Davide, Ilie e Leonardo per i numerosi momenti passati insieme, sia durante i group meeting sia durante aperitivi e cene.

Infine, il ringraziamento più speciale va alla mia famiglia. Mamma, Papà, Fabio, Eliana, Gaetano: se ce l'ho fatta, è soprattutto merito vostro. Abbiamo condiviso molti momenti di gioia di questo percorso, ma soprattutto mi avete supportato e aiutato nei momenti difficili, credendo sempre in me. Questa tesi è dedicata a voi.

Contents

1 Introduction **1**

2 Background **7**
2.1 Pattern Mining 7
2.1.1 Frequent Pattern Mining 8
2.1.2 True Frequent Pattern Mining 10
2.2 Maximum Deviation 11
2.3 Statistical Learning Theory 13
2.3.1 VC-dimension 13
2.3.2 Rademacher Complexity 15

3 Mining True Frequent Sequential Patterns with Rademacher Complexity **17**
3.1 Introduction 17
3.1.1 Our Contributions 18
3.1.2 Related Works 19
3.1.3 Organization of the Chapter 20
3.2 Preliminaries 20
3.3 Rademacher Complexity of Sequential Patterns 23
3.3.1 An Efficiently Computable Upper Bound to the Rademacher Complexity of Sequential Patterns 24
3.3.2 Approximating the Rademacher Complexity of Sequential Patterns 34
3.4 Algorithms for True Frequent Sequential Pattern Mining . . . 36
3.5 Experimental Evaluation 38
3.5.1 Implementation, Datasets, Parameters, and Environment 39
3.5.2 True Frequent Sequential Patterns Mining Results . . . 41

4 SPRISS: Approximating Frequent k-mers by Sampling Reads 47
4.1 Introduction . . . 47
4.1.1 Our Contributions . . . 48
4.1.2 Related Works . . . 49
4.1.3 Organization of the Chapter . . . 51
4.2 Preliminaries . . . 51
4.3 Warm-Up: A Simple Algorithm for Approximating Frequent k-mers by Sampling Reads . . . 53
4.4 A First Improvement: A Pseudodimension-based Algorithm for k-mers Approximation by Sampling Reads . . . 56
4.5 SPRISS: Sampling Reads Algorithm to Estimate Frequent k-mers . . . 59
4.6 Experimental Evaluation . . . 65
4.6.1 Implementation, Datasets, Parameters, and Environment . . . 66
4.6.2 Approximation of Frequent k-mers . . . 67

5 Applications of SPRISS 73
5.1 Introduction . . . 73
5.1.1 Our Contributions . . . 74
5.1.2 Related Works . . . 74
5.1.3 Organization of the Chapter . . . 75
5.2 Implementation, Datasets, Parameters, and Environment . . . 75
5.3 Comparing Metagenomic Datasets . . . 78
5.4 Approximation of Discriminative k-mers . . . 83
5.5 SNP Genotyping . . . 86

6 Conclusions 89

Bibliografy 91

Chapter 1

Introduction

Data mining is the task of extracting meaningful information from massive data. Nowadays, the generation of large amount of data is a consolidated practice in many domains: *e-commerce*, for example, has given a lot of importance to the *market basket analysis*, where a huge number of purchasing behaviours of customers can be analyzed to inform retailers on how to raise their sales by adopting ad hoc marketing and recommendations; *Social networks* virtually connect people around the world providing massive data that are useful to understand, for example, how groups of people interact and how people react to information shared on the network; *Internet of Things* (IoT) provides enormous quantity of data about the internet interconnections of devices, which can be analyzed to improve the management and performances of IoT systems; In *medicine*, tons of data about patients are collected, whose statistical and computational analysis can help, for example, to improve medical diagnosis and to adopt specific intervention to prevent or treat a disease; In *biology*, the recent impressive improvement of next-generation sequencing machines has given the possibility to analyze massive datasets storing DNA sequences in order to, for example, identify the mutations that drive cancers. The domains mentioned above are just some examples of the many where data mining techniques find application to discover patterns and new knowledge. The simultaneous growth of the technologies generating big data and of the effort of data mining research communities can truly lead to discover *patterns*, i.e., rules describing the data, and new *knowledge*.

In this Thesis we focus on the analysis of a specific type of data, i.e., *sequential data*, that are data composed of elements with an underlying intrinsic sequential nature among them. In particular, we study two types of

sequential data, i.e., *sequential patterns* and *k-mers*. A sequential pattern is a sequence of sets of items (i.e., itemsets), where items can represent, for example, objects purchased by customers in a website store, or events associated to actions of the users of a website. A k-mer, instead, is a substring of length k of a sequence, as generated, for example, by high-throughput sequencing experiments. Studying sequential patterns helps, for example, to identify customers and users behaviour, while studying k-mers helps, for example, to understand the structure of biological sequences.

In data mining, there are two ways in which a dataset can be considered. In the first scenario, the dataset is considered as a *sample* drawn from the *unknown generative process* underlying the data: the dataset can be studied to gain insights about the unknown distribution that generates the data. In the second scenario, instead, the dataset is studied in order to extract meaningful patterns that reside in it. In this Thesis we consider both knowledge discovery approaches: we study the unknown generative process of sequential patterns to identify meaningful patterns, and we study biological sequences to identify k-mers that are meaningful for the datasets analyzed. The notion of *meaningfulness* for patterns depends on the type of patterns and the specific data mining task. Now we describe the contributions of this Thesis, specifying the measures of meaningfulness for each data mining task and type of patterns considered.

In Chapter 3 we study datasets of sequential patterns in order to gain insights about their unknown generative process. In this scenario, we say that the dataset is composed of *transactions*, that are *independent and identically distributed (i.i.d.)* samples from the unknown generative process underlying the data (thus, transactions are sequential patterns). In this context, we consider the *true frequency* of sequential patterns as the measure of meaningfulness of patterns. Informally, the true frequency of a sequential pattern P is the probability that a transaction sampled from the unknown generative process contains P. The problem we consider is to mine the *true frequent sequential patterns*, which are those sequential patterns that are frequently generated by the generative process, given a user-defined minimum frequency threshold. Exact approaches to mine true frequent sequential patterns (and their true frequency) are infeasible, since the underlying generative process is unknown. In addition, the observed frequencies of the sequential patterns in the dataset only approximately reflect the true frequencies. Consequently, one has to resort to approximation methods by analyzing datasets of transactions sampled from the unknown distribution. In this Thesis we define two

rigorous approximations of the true frequent sequential patterns, one without false negatives and one without false positives. Then, we present algorithms that output rigorous approximations of the set of true frequent sequential patterns containing, with high probability, no false positives or false negatives. Our algorithms are based on theoretical results that guarantee that their outputs are high-quality approximations. Our theoretical analysis is based on the study of the *Rademacher complexity*, an advanced tool from statistical learning theory, of sequential patterns. In particular, we provide an efficient computable upper bound on the Rademacher complexity, together with a strategy to approximate it. Finally, we describe our extensive experimental evaluation studying real world datasets of transactions from several domains, showing that our algorithms provide high-quality approximations of the set of true frequent sequential patterns. To the best of our knowledge, there is no method to approximate true frequent sequential patterns. The contributions described in Chapter 3 appear in [Santoro et al., 2020].

In Chapter 4 we study datasets representing biological sequences. In particular, the dataset $\mathcal{D}$ is a finite bag of *reads*, where a read, which is generated by high-throughput sequencing experiments, represents a portion of a biological sequence. A k-mer K is a substring of length k of a read, and its *frequency* in a dataset $\mathcal{D}$ is the fraction of times K appears in $\mathcal{D}$. In this scenario, we consider the *frequency* of k-mers as the measure of meaningfulness of patterns. The problem we consider in Chapter 4 is to mine *frequent k-mers*, which are those k-mers that appear with a relatively high frequency in the dataset, given a user-defined minimum frequency threshold. Exact approaches to solve this problem exist, but their executions on large datasets are still highly demanding in terms of computational resources. Thus, efficient approximation methods are to be sought. A natural approach to speed-up the computation of frequent k-mers is to analyze only a *sample* of the data, instead of the entire dataset $\mathcal{D}$. This is motivated by the fact that frequent k-mers of $\mathcal{D}$ appear with high probability in the sample, and, instead, infrequent k-mers appear with lower probability. In this Thesis we present SPRISS, a sampling-based algorithm to rigorously approximate frequent k-mers, proving rigorous guarantees on the quality of its output. Our sampling scheme is based on the creation of a sample of reads. To the best of our knowledge, there is no method to rigorously approximate frequent k-mers by sampling reads. Our theoretical analysis provides the sample size which is required to obtain accurate estimates from SPRISS. To prove our theoretical results we study the *pseudodimension*, an advanced concept from statistical

learning theory, of k-mers in reads, showing that less sophisticated tools like Hoeffding's inequality combined with a union bound, and the VC-dimension, fails to provide practical sample sizes. Finally, we describe our extensive experimental evaluation studying real world datasets of reads, showing that SPRISS outputs high-quality approximations of frequent k-mers by only analyzing small samples, while speeding-up the computation with respect to the exact approaches. The contributions described in Chapter 3 appear in [Santoro et al., 2021] as conference version, and in [Santoro et al., 2022] as journal version.

In Chapter 5 we describe our experiments to show how SPRISS can be used in bioinformatics to speed-up the analysis of datasets of biological sequences. Several applications, e.g., comparison of datasets and reads classification in metagenomics, genome comparison, error correction for genome assembly, and many others, heavily depend on the identification of k-mers and their frequencies. In this Thesis, we use SPRISS to speed-up applications that rely on the identification of frequent k-mers. In particular, we evaluate the usage of SPRISS in the comparison of metagenomic datasets, using SPRISS's approximations to estimate abundance based distances between them. Then, we test SPRISS in the discovery of discriminative k-mers between pairs of metagenomic datasets. Finally, we combine the sampling scheme of SPRISS with state of the art genotyping algorithms to approximate SNP genotyping. For all these three applications of SPRISS, our extensive experimental evaluation shows that SPRISS is able to speed-up the downstream analyses, while achieving high-quality estimations of the exact results. The contributions about the application of SPRISS to compare metagenomic datasets and to discover discriminative k-mers appear both in [Santoro et al., 2021] as conference version and in [Santoro et al., 2022] as journal version, while the application of SPRISS to approximate SNP genotyping appear only in the journal version [Santoro et al., 2022].

The rest of this Thesis is organized as follows. In Chapter 2 we introduce some preliminary definitions and concepts used throughout this Thesis. Then, in Chapter 3 we present our theoretical analysis and experimental results of our algorithms to rigorously approximate the true frequent sequential patterns. In Chapter 4 we present our theoretical analysis and experimental results of SPRISS, our algorithm for the rigorous approximation of frequent k-mers by sampling reads. Next, in Chapter 5 we present our extensive experimental evaluation of the application of SPRISS to compare metagenomic datasets, to compute discriminative k-mers, and to SNP genotyping. Finally,

in Chapter 6 we end this Thesis with some final considerations.

Chapter 2

Background

In this chapter we introduce some preliminary definitions and concepts that will be useful for the rest of this Thesis, and we give a preliminary overview about previous works. The detailed presentations of previous works about our novel contributions are reported in the respective chapters. In Section 2.1 we introduce the two types of pattern mining problem studied in this Thesis, i.e., frequent pattern mining and true frequent pattern mining, whose goals are, respectively, to mine patterns that appear with high frequency in a set of data, and to mine patterns that are frequently generated by the process generating the data. Then, in Section 2.2 we introduce the fundamental concept of maximum deviation, over all possible patterns, of the actual frequency and its estimate, describing it for both pattern mining scenarios studied in this work. Finally, in Section 2.3 we introduce the VC-dimension and the Rademacher complexity, two fundamental concepts of statistical learning theory that are useful to obtain probabilistic upper bounds to the maximum deviation and to derived rigorous approximations of the frequent patterns and the true frequent patterns.

2.1 Pattern Mining

The goal of pattern mining is to find *meaningful patterns*, i.e. rules describing patterns in the data, for a given measure of meaningfulness which depends on the specific knowledge discovery task. Pattern mining finds applications in several domains, ranging, e.g., from market basket analysis to network analysis, and from biology to medicine.

Let $p \in \mathbb{U}$ be a generic pattern, where $\mathbb{U}$ is the universe of all patterns. We define the *dataset* $\mathcal{D}$ as a finite bag of $n = |\mathcal{D}|$ points that corresponds to elements of the universe $\mathbb{U}$: $\mathcal{D} = \{s_1, \ldots, s_n\}$, where $s_i \in \mathbb{U}$. In pattern mining strategies, the dataset $\mathcal{D}$ is analyzed to extract those patterns of $\mathbb{U}$ that are of interest for the specific task. For example, two types of patterns that have been extensively studied are *itemsets* (i.e. sets of items) [Agrawal et al., 1993], and *sequential patterns* (i.e. sequences of itemsets) [Agrawal and Srikant, 1995].

Now we introduce two types of pattern mining tasks that are of interest for this Thesis: frequent pattern mining (Section 2.1.1) and true frequent pattern mining (Section 2.1.2) .

2.1.1 Frequent Pattern Mining

Frequent pattern mining is a fundamental task in data mining and knowledge discovery. In the frequent pattern mining scenario, the measure of meaningfulness of patterns is the *frequency* in which they appear in the dataset $\mathcal{D}$. The goal of frequent pattern mining is to identify those patterns that appear in a sufficient high portion of the data, i.e. with high frequency. The frequent pattern mining task has been extensively studied for several types of pattern using different efficient algorithms to mine frequent patterns from large amount of data [Han et al., 2007].

The specific definition of the frequency $f_{\mathcal{D}}(p)$ of a pattern p in $\mathcal{D}$ can differ depending on the specific contexts and applications. However, in this Thesis, if not stated otherwise, the frequency $f_{\mathcal{D}}(p)$ of a pattern p in a dataset $\mathcal{D}$ is defined as follows.

Definition 1. *Given a dataset $\mathcal{D}$ and a pattern p, the* support set $T_{\mathcal{D}}(p)$ *of p in $\mathcal{D}$ is the set of points in $\mathcal{D}$ that contain p. The* support $Supp_{\mathcal{D}}(p)$ *of p in $\mathcal{D}$ is the cardinality of the support set $T_{\mathcal{D}}(p)$: $Supp_{\mathcal{D}}(p) = |T_{\mathcal{D}}(p)|$. The frequency $f_{\mathcal{D}}(p)$ of p in $\mathcal{D}$ is the fraction of points in $\mathcal{D}$ to which p belongs:*

$$f_{\mathcal{D}}(p) = \frac{Supp_{\mathcal{D}}(p)}{|\mathcal{D}|}. \tag{2.1}$$

The frequent pattern mining problem is defined as follows.

Definition 2. *Given a minimum frequency threshold $\theta \in (0, 1]$ and a dataset $\mathcal{D}$, we are interested in finding the set $FP(\mathcal{D}, \theta)$ of frequent patterns in $\mathcal{D}$*

with respect to (w.r.t.) θ, i.e.,

$$FP(\mathcal{D}, \theta) = \{(p, f_{\mathcal{D}}(p)) : p \in \mathbb{U},\ f_{\mathcal{D}}(p) \geq \theta\}. \tag{2.2}$$

Algorithms solving this problem typically requires access to the entire dataset $\mathcal{D}$, which is really impractical in big data contexts where $\mathcal{D}$ can be very large. In addition, the complex structure that forms the patterns makes often difficult to handle them in an efficient way, leading the exact computation of $FP(\mathcal{D}, \theta)$ to be infeasible in practice. Thus, one has to resort to some efficient approaches to compute an approximation of the set $FP(\mathcal{D}, \theta)$ of frequent patterns. Now we formally define the approximations of $FP(\mathcal{D}, \theta)$ that are of interest for this Thesis. In particular, we consider the rigorous approximation of $FP(\mathcal{D}, \theta)$ known as *ε-approximation*, for a given accuracy parameter $\varepsilon \in (0, 1)$. Here we present the definition of *ε-approximation* for general patterns, instead Definition 1 of [Riondato and Upfal, 2015] defines it for itemsets (i.e., sets of items), and Definition 2 of [Servan-Schreiber et al., 2018] defines it for sequential patterns (i.e., sequences of itemsets).

Definition 3. *Given $\varepsilon \in (0, 1)$, an ε-approximation $\mathcal{C}$ of $FP(\mathcal{D}, \theta)$ is defined as a set of pairs (p, f_p):*

$$\mathcal{C} = \{(p, f_p) : p \in \mathbb{U}, f_p \in [0, 1]\} \tag{2.3}$$

that has the following properties:

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

Intuitively, the approximation $\mathcal{C}$ contains all the frequent patterns that are in $FP(\mathcal{D}, \theta)$ (i.e., there are no *false negatives*) and no pattern that has frequency in $\mathcal{D}$ much below θ. In addition, $\mathcal{C}$ provides a good approximation of the actual frequency of the pattern in $\mathcal{D}$, within an error $\varepsilon/2$, arbitrarily small. Since the definition of the approximation set $\mathcal{C}$ in Definition 3 ensures that there are no false negatives, in This thesis, where necessary, we call C a *false negatives free* (FNF) ε-approximation of $FP(\mathcal{D}, \theta)$.

2.1.2 True Frequent Pattern Mining

In addition to frequent pattern mining, several types of pattern mining has been studied by considering different measure of meaningfulness for the patterns: significant pattern mining [Hämäläinen and Webb, 2019], high-utility pattern mining [Fournier-Viger et al., 2019], and true frequent pattern mining [Riondato and Vandin, 2014]. In particular, Riondato and Vandin [Riondato and Vandin, 2014] proposed to mine *true frequent itemsets*, which are sets of items generated with high probability by the unknown generative process underlying the data, by employing the empirical VC-dimension, a fundamental concept of statistical learning theory, of itemsets.

In several applications, the dataset $\mathcal{D}$ is a sample of independent and identically distributed points drawn from an unknown probability distribution π, with $\pi : \mathbb{U} \to [0, 1]$. The *true frequency* $t_\pi(p)$ of pattern p w.r.t. π is defined as follows.

Definition 4. *Consider an unknown probability distribution $\pi : \mathbb{U} \to [0, 1]$. For any pattern $p \in \mathbb{U}$, we define the* true support set *$T(p)$ of p as the set of patterns in $\mathbb{U}$ to which p belongs: $T(p) = \{\tau \in \mathbb{U} : p \in \tau\}$. In the true frequent pattern mining scenario, we define the* true frequency *$t_\pi(p)$ of p w.r.t. π as the probability that a point sampled from π contains p:*

$$t_\pi(p) = \sum_{\tau \in T(p)} \pi(\tau). \tag{2.4}$$

In this scenario, the final goal of the pattern mining process on $\mathcal{D}$ is to gain a better understanding of the process generating the data, that is, of the distribution π, through the true frequencies t_π, which are unknown and only approximately reflected in the dataset $\mathcal{D}$. The *true frequent pattern mining* problem is defined as follows.

Definition 5. *Consider an unknown probability distribution $\pi : \mathbb{U} \to [0, 1]$. Given a minimum frequency threshold $\theta \in (0, 1]$, we are interested in finding the set $TFP(\pi, \theta)$ of true frequent patterns with true frequency t_π at least θ:*

$$TFP(\pi, \theta) = \{(p, t_\pi(p)) : p \in \mathbb{U}, t_\pi(p) \geq \theta\}. \tag{2.5}$$

Note that, given a finite number of random samples from π (e.g., the dataset $\mathcal{D}$), it is not possible to find the exact set $TFP(\pi, \theta)$, and one has to resort to approximations of $TFP(\pi, \theta)$. The definition of specific types

of approximation sets of $TFP(\pi, \theta)$ under a specific type of patterns, i.e., sequential patterns, without false positives or without false negatives, and efficient methods to mine them will be presented as contributions of this Thesis in Chapter 3.

2.2 Maximum Deviation

In the approximate frequent pattern mining and true frequent pattern mining problems we aim to find good estimates of the actual frequencies. Now we introduce the fundamental concept of *maximum deviation*, over all patterns in $\mathbb{U}$, between the actual frequency and its estimate. Note that a small value for the maximum deviation typically leads to high quality approximations of the patterns of interest.

Let $\mathcal{M}$ be a probability distribution over a domain set $\mathcal{Z}$. Let $\mathcal{F}$ be a set of functions that go from $\mathcal{Z}$ to $[-1, 1]$. Given a function $f \in \mathcal{F}$, we define the expectation of f as:

$$\mathbb{E}(f) = \mathbb{E}_{z \sim \mathcal{M}}[f(z)], \tag{2.6}$$

and, given a sample Z of n observations $z_1, \ldots, z_n$ drawn from $\mathcal{M}$, the empirical average of f on Z as:

$$E(f, Z) = \frac{1}{n} \sum_{i=1}^{n} f(z_i). \tag{2.7}$$

The *maximum deviation* $D(\mathcal{F}, Z)$ is defined as the largest difference between the expectation of a function f and its empirical average on sample Z as:

$$D(\mathcal{F}, Z) = \sup_{f \in \mathcal{F}} |\mathbb{E}(f) - E(f, Z)|. \tag{2.8}$$

We now use the maximum deviation to capture quantities of interest for the two mining tasks we consider in this Thesis.

In the frequent pattern mining scenario, we aim to find good estimates for $f_{\mathcal{D}}(p)$ for each pattern p. The frequency $f_{\mathcal{D}}(p)$ is the expectation of a Bernoulli random variable (r.v.) $X_{\mathcal{D}}(p, s)$ which is 1 if the pattern p appears in a point s drawn uniformly at random from $\mathcal{D}$:

$$\mathbb{E}_{s \sim \mathcal{D}}[X_{\mathcal{D}}(p, s)] = \Pr_{s \sim \mathcal{D}}(X_{\mathcal{D}}(p, s) = 1) = Supp_{\mathcal{D}}(p)/|\mathcal{D}| = f_{\mathcal{D}}(p). \tag{2.9}$$

A natural approach to approximate the frequency of patterns of the dataset $\mathcal{D}$ is to only analyze a small portion of it, i.e. a *sample*. Let $\mathcal{S}$ be a sample of points drawn uniformly and independently at random from $\mathcal{D}$. We define the frequency $f_{\mathcal{S}}(p)$ of pattern p in a sample $\mathcal{S}$ as the fraction of points of $\mathcal{S}$ where p appears. Note that $f_{\mathcal{S}}(p)$ is an empirical average (over the points in $\mathcal{S}$) and its expectation is $\mathbb{E}[f_{\mathcal{S}}(p)] = f_{\mathcal{D}}(p)$. Thus, the maximum deviation is:

$$\sup_{p \in \mathbb{U}} |f_{\mathcal{D}}(p) - f_{\mathcal{S}}(p)|. \tag{2.10}$$

In the true frequent pattern mining scenario, we aim to find good estimates for $t_\pi(p)$ for each pattern p. Note that the true frequency $t_\pi(p)$ is the expectation of a Bernoulli r.v. which is 1 if the pattern p appears in a point drawn from π, and that the observed frequency $f_{\mathcal{D}}(p)$ is an empirical average (over the points in $\mathcal{D}$). Moreover, it is easy to prove that the observed frequency $f_{\mathcal{D}}(p)$ of a pattern p in a dataset $\mathcal{D}$ of points drawn from π is an unbiased estimator for $t_\pi(p)$, that is: $\mathbb{E}\left[f_{\mathcal{D}}(p)\right] = t_\pi(p)$. Thus, the maximum deviation is:

$$\sup_{p \in \mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)|. \tag{2.11}$$

Intuitively, small values for the maximum deviations of equations 2.10 and 2.11 imply that, for every pattern p, the actual frequency is well approximated by its estimate. Bounding the maximum deviation $D(\mathcal{F}, Z) \leq \mu$ (for a factor $\mu \in (0, 1)$), which is also known as *uniform convergence*, implies that simultaneously all the estimates are uniformly close to the actual frequencies within the factor μ. As we will see in the next chapters, finding a small upper bound to the maximum deviation is strictly correlated in finding good approximations of frequent patterns and true frequent patterns.

In the next section, we see tools from statistical learning theory, e.g., VC-dimension [Vapnik and Chervonenkis, 2015, Mitzenmacher and Upfal, 2017] and Rademacher Complexity [Boucheron et al., 2005, Shalev-Shwartz and Ben-David, 2014], that are useful to compute probabilistic upper bounds to the maximum deviation, i.e., $\Pr(D(\mathcal{F}, Z) \leq \mu) \geq 1 - \delta$, for some confidence parameter $\delta \in (0, 1)$.

2.3 Statistical Learning Theory

Statistical learning theory [Vapnik, 1999] is an important branch of machine learning that provides tools to derive probabilistic guarantees on the performances of learning algorithms. In this section, we introduce fundamental concepts of statistical learning theory, like VC-dimension and Rademacher complexity, that are useful to compute probabilistic upper bounds to the maximum deviation $D(\mathcal{F}, Z) = \sup_{f \in \mathcal{F}} |\mathbb{E}(f) - E(f, Z)|$ (see Equation 2.8).

In this section we present the VC-dimension and Rademacher Complexity in a general setting. Instead, in Chapter 3 we present the Rademacher complexity of sequential patterns in the true frequent pattern mining scenario, and in Chapter 4 we present the VC-dimension and pseudodimension, an advanced statistical learning tool which is be based on the VC-dimension, of k-mers in the frequent pattern mining scenario. The connection between such tools and the probabilistic bounds on the maximum deviations (Equation 2.10 and Equation 2.11) will be presented in Chapter 3 for the true pattern mining scenario and in Chapter 4 for the frequent pattern mining scenario.

As stated in the previous section, a natural approach to approximate frequent patterns of a dataset $\mathcal{D}$ is to only analyze a sample $\mathcal{S}$ of $\mathcal{D}$. *Sampling* is a general fundamental technique of statistical data analysis and machine learning which is useful to estimate proprieties of a domain by just analyze a small portion of it. A crucial challenge in sampling techniques is to identify the *sample complexity* of the problem, i.e., the sample size which is enough to obtain the required quality of the results with rigorous guarantees [Mitzenmacher and Upfal, 2017]. In this Thesis we use sampling techniques for mining frequent patterns, and, in particular, to approximate frequent k-mers from a dataset of reads, by studying the VC-dimension and pseudodimension of k-mers.

2.3.1 VC-dimension

The *VC-dimension* (Vapnik-Chervonenkis dimension) [Vapnik and Chervonenkis, 2015] is a measure of the complexity or expressiveness of a family of indicator functions or, equivalently, of a family of subsets defined on a space of points. The presentation below follows the one in [Mitzenmacher and Upfal, 2017].

The definition of the VC-dimension follows.

Definition 6. *We define $Q = (X, R)$ as a range space, where X is a finite or infinite domain of points, and R is the range set, i.e., a family of subsets of X. The members $r \in R$ of the range set R are called ranges. Given $N \subseteq X$, the* projection *of R on N is $proj_R(N) = \{r \cap N : r \in R\}$. If $|proj_R(N)| = 2^{|N|}$ then we say that N is shattered by R. The VC-dimension $VC(Q)$ of Q is the maximum cardinality of a subset of X shattered by R.*

Note that to prove that $VC(Q) = v$, the following conditions need to hold: (a) there exists a set $N \subseteq X$ of size v that is shattered; (b) every set $N \subseteq X$ of size $v+1$ is not shattered. If there exist arbitrary large subsets of X that can be shattered, then $VC(Q) = \infty$.

The main application of the VC-dimension is to derive the sample size which is enough to approximately *learn* the ranges, as defined below.

Definition 7. *Let X be a finite or infinite domain of points. Given a finite bag $N \subseteq X$, and an accuracy parameter $\varepsilon \in (0, 1]$, a bag B of elements drawn uniformly at random from N is an ε-bag of X if for every range $r \in R$ holds:*

$$\left| \frac{|X \cap r|}{|X|} - \frac{|B \cap r|}{|B|} \right| \leq \varepsilon/2. \tag{2.12}$$

The following theorem from [Mitzenmacher and Upfal, 2017] relates the accuracy parameter ε with the probability that a bag $B \subseteq N$ of size m is an *ε-bag* for a range space Q of VC-dimension $VC(Q)$ at most v.

Theorem 1. *[Mitzenmacher and Upfal, 2017] There is an absolute positive constant c such that if $Q = (X, \mathcal{R})$ is a range space of VC-dimension $VC(Q) \leq v$, $N \subseteq X$ is a finite bag, and $\varepsilon, \delta \in (0, 1)$, then a bag B of m elements drawn with independent random extractions with replacement from N, with*

$$m \geq \frac{c}{\varepsilon^2} \left(d + \ln \frac{1}{\delta} \right), \tag{2.13}$$

is an ε-bag of X with probability $\geq 1 - \delta$.

The universal constant c has been experimentally estimated to be at most 0.5 [Löffler and Phillips, 2009].

As we can see from the previous theorem, the VC-dimension is a fundamental tool of learning theory which provides a way to derive the sample size needed to learn an approximation of the ranges, which are typically associated to patterns and their frequencies. In Chapter 4 we study the

VC-dimension and the pseudodimension, an advanced tool based on the VC-dimension, in the frequent pattern mining scenario.

2.3.2 Rademacher Complexity

A key quantity from statistical learning theory to study and derive an upper bound of the maximum deviation of Equation 2.8 is the *Rademacher complexity* [Boucheron et al., 2005, Shalev-Shwartz and Ben-David, 2014], which is a tool to measure the complexity of a family of real-valued functions. Bounds based on the Rademacher complexity depend on the distribution of the data, differently from the ones based on VC-dimension that are distribution independent.

The Rademacher complexity, which is defined below, is a measure of the expressiveness of a set $\mathcal{Y}$ of real-valued functions.

Definition 8. *Let $\mathcal{D}$ be a dataset of n points $\mathcal{D} = \{s_1, \ldots, s_n\}$. For each $i \in \{1, \ldots, n\}$, let σ_i be an independent Rademacher random variable (r.v.) that takes value 1 or -1, each with probability $1/2$. Let $\mathcal{Y}$ be a set of real-valued functions. The empirical Rademacher complexity $R_{\mathcal{D}}$ on $\mathcal{D}$ is defined as follows:*

$$R_{\mathcal{D}} = \mathbb{E}_{\sigma}\left[\sup_{g \in \mathcal{Y}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i g(s_i)\right], \tag{2.14}$$

where the expectation is taken w.r.t. the Rademacher r.v. σ_i's.

Note that a specific combination of σ's represents a splitting of $\mathcal{D}$ into two random sub-samples $\mathcal{D}_1$ and $\mathcal{D}_{-1}$: $\mathcal{D}_1$ consists of the points of $\mathcal{D}$ for which the corresponding r.v. $\sigma = 1$, while $\mathcal{D}_{-1}$ consists of the points of $\mathcal{D}$ for which the corresponding r.v. $\sigma = -1$. For a function $g \in \mathcal{Y}$, $\sum_{i=1}^{n} \sigma_i g(s_i)/n$ represents the difference between $\mathbb{E}[g]$ over the two random sub-samples $\mathcal{D}_1$ and $\mathcal{D}_{-1}$. By considering the expected value of the supremum of this difference over the set $\mathcal{Y}$, we get the empirical Rademacher complexity. Therefore the intuition is that if $R_{\mathcal{D}}$ is small, the dataset $\mathcal{D}$ is sufficiently large to ensure a good estimate of $\mathbb{E}[g]$ for every $g \in \mathcal{Y}$.

In Chapter 3 we study the Rademacher complexity of sequential patterns, which has not been explored before, providing efficient methods to both approximate and bound it in the true frequent pattern mining scenario. This will be crucial to upper bound the maximum deviation of Equation 2.11 and to provide rigorous approximations of true frequent sequential patterns.

Chapter 3

Mining True Frequent Sequential Patterns with Rademacher Complexity

3.1 Introduction

Sequential pattern mining [Agrawal and Srikant, 1995] is a fundamental task in data mining and knowledge discovery, with applications in several fields, from recommender systems and e-commerce to biology and medicine. In its original formulation, sequential pattern mining requires to identify all *frequent sequential patterns*, that is, sequences of itemsets that appear in a fraction at least θ of all the transactions in a transactional dataset, where each transaction is a sequence of itemsets. The threshold θ is a user-specified parameter and its choice must be, at least in part, be informed by domain knowledge. In general, sequential patterns describe sequences of events or actions that are useful for predictions in many scenarios.

In several applications, the analysis of a dataset is performed to gain insight on the *underlying generative process* of the data. For example, in market basket analysis one is interested in gaining knowledge on the behaviour of all the customers, which can be modelled as a generative process from which the transactions in the dataset have been drawn. In such a scenario, one is not interested in sequential patterns that are frequent *in the dataset*, but in sequential patterns that are frequent *in the generative process*, that is, whose probability of appearing in a transaction generated from the

process is above a threshold θ. Such patterns, called *true frequent patterns*, have been introduced by Riondato and Vandin [Riondato and Vandin, 2014], which provides a Vapnik-Chervonenkis (VC) dimension based approach to mine true frequent itemsets. While there is a relation between the probability that a pattern appears in a transaction generated from the process and its frequency in the dataset, one cannot simply look at patterns with frequency above θ in the dataset to find the ones with probability above θ in the process. Moreover, due to the stochastic nature of the data, one cannot identify the true frequent patterns with certainty, and approximations are to be sought. In such a scenario, relating the probability that a pattern appears in a transaction generated from the process with its frequency in the dataset using standard techniques is even more challenging. Hoeffding inequality and union bounds require to bound the number of all the possible sequential patterns that can be generated from the process. Such bound is infinite if one considers all possible sequential patterns (e.g., does not bound the pattern length). To the best of our knowledge, no method to mine *true frequent sequential patterns* has been proposed.

3.1.1 Our Contributions

We study the *true frequent sequential pattern* mining problem, and we propose efficient algorithms based on the concepts of the Rademacher complexity. In this regard, our contributions are:

- We define rigorous approximations of the set of true frequent sequential patterns. In particular, we define two approximations: one with no *false negatives*, that is, containing all elements of the set; and one with no *false positives*, that is, without any element that is not in the set. Our approximations are defined in terms of a single parameter, which controls the accuracy of the approximation and is easily interpretable.

- We study the Rademacher complexity of sequential patterns, an advanced concept from statistical learning theory that has been used in other mining contexts. We provide the first efficiently computable upper bound to the Rademacher complexity of sequential patterns. We also show how to approximate the Rademacher complexity of sequential patterns. Thus, we provide efficient algorithms both to bound and approximate the Rademacher complexity of sequential patterns.

- We introduce efficient algorithms to obtain rigorous approximations of the true frequent sequential patterns with probability $1 - \delta$, where δ is a confidence parameter set by the user. Our algorithms use the novel bound and approximation of the Rademacher complexity that we have derived, and they allow to obtain accurate approximations of the true frequent sequential patterns, where the accuracy depends on the size of the available data.

- We perform an extensive experimental evaluation analyzing several sequential datasets, showing that our algorithms provide high-quality approximations, even better than guaranteed by their theoretical analysis.

3.1.2 Related Works

Since the introduction of the frequent sequential pattern mining problem [Agrawal and Srikant, 1995], a number of exact algorithms has been proposed for this task, ranging from multi-pass algorithms using the anti-monotonicity property of the frequency function [Srikant and Agrawal, 1996], to prefix-based approaches [Pei et al., 2004], to work focusing on the closed frequent sequences [Wang et al., 2007].

The use of sampling to reduce the amount of data for the mining process while obtaining rigorous approximations of the collection of interesting patterns has been successfully applied in many mining tasks. Raïssi and Poncelet [Raïssi and Poncelet, 2007] provided a theoretical bound on the sample size for a single sequential pattern in a static dataset using Hoeffding concentration inequalities, and they introduced a sampling approach to build a dynamic sample in a streaming scenario using a biased reservoir sampling. Our work is heavily inspired by the work of Riondato and Upfal [Riondato and Upfal, 2015], which introduced advanced statistical learning techniques for the task of frequent itemsets mining. In particular, Riondato and Upfal [Riondato and Upfal, 2015] proposed a progressive sampling approach based on an efficiently computable upper bound on the Rademacher complexity of itemsets. Rademacher complexity has also been used in graph mining [Al Hasan et al., 2007, Corizzo et al., 2019, Cheng et al., 2010], to design random sampling approaches for estimating betweenness centralities in graphs [Riondato and Upfal, 2018], and to bound the family-wise error rate in local causal discovery [Simionato and Vandin, 2022].

To the best of our knowledge, [Riondato and Vandin, 2014] is the only work that considers the extraction of frequent patterns w.r.t. an underlying generative process, based on the concept of empirical VC-dimension of itemsets. While we use the general framework introduced by Riondato and Vandin [Riondato and Vandin, 2014], the solution proposed by them requires to solve an optimization problem that is tailored to itemsets and, thus, not applicable to sequential patterns; in addition, computing the solution of such problem could be relatively expensive.

[Pellegrina et al., 2019] considers the problem of mining significant patterns under a similar framework, making more realistic assumptions on the underlying generative process compared to commonly used tests (e.g., Fisher's exact test). Several works have been proposed to identify statistically significant patterns where the significance is defined in terms of the comparison of patterns statistics. Few methods [Gwadera and Crestani, 2010, Low-Kam et al., 2013, Tonon and Vandin, 2019] have been proposed to mine statistically significant sequential patterns. These methods are orthogonal to our approach, which focuses on finding sequential patterns that are frequent w.r.t. an underlying generative distribution.

3.1.3 Organization of the Chapter

The rest of the Chapter is organized as follows. In Section 3.2 we introduce some preliminary concepts used throughout this work. The study of the Rademacher complexity of sequential patterns is presented in Section 3.3: in Section 3.3.1 we present an efficient strategy to compute an upper bound to the Rademacher complexity of sequential patterns; in Section 3.3.2 we present a strategy to approximate the Rademacher complexity of sequential patterns. Next, in Section 3.4 we describe algorithms to find rigorous approximations to the true frequent sequential patterns. Finally, in Section 3.5 we describe our experimental evaluation to assess the performance of our algorithms to approximate true frequent sequential patterns.

3.2 Preliminaries

We now provide the definitions and concepts used throughout this Chapter. We start by defining the sequential patterns, and then we formally define the problem which is the focus of this work: approximating sequential patterns

that are frequently generated from the underlying generative process.

Let $\mathcal{I} = \{i_1, i_2, \ldots, i_h\}$ be a finite set of elements called *items*. $\mathcal{I}$ is also called the *ground set*. An *itemset* P is a (non-empty) subset of $\mathcal{I}$, that is, $P \subseteq \mathcal{I}$. A *sequential pattern* $p = \langle P_1, P_2, \ldots, P_\ell \rangle$ is a *finite ordered sequence* of itemsets, with $P_i \subseteq \mathcal{I}, 1 \leq i \leq \ell$. A sequential pattern p is also called a *sequence*. The *length* $|p|$ of p is defined as the number of itemsets in p. The *item-length* $||p||$ of p is the sum of the sizes of the itemsets in p, that is,

$$||p|| = \sum_{i=1}^{|p|} |P_i|, \tag{3.1}$$

where $|P_i|$ is the number of items in itemset P_i. A sequence $a = \langle A_1, A_2, \ldots, A_m \rangle$ is a *subsequence* of another sequence $b = \langle B_1, B_2, \ldots, B_n \rangle$, denoted by $a \sqsubseteq b$, if and only if there exist integers $1 \leq i_1 < i_2 < \ldots < i_m \leq n$ such that $A_1 \subseteq B_{i_1}$, $A_2 \subseteq B_{i_2}, \ldots, A_m \subseteq B_{i_m}$. If a is a subsequence of b, then b is called a *super-sequence* of a, denoted by $b \sqsupseteq a$.

Let $\mathbb{U}$ denote the set of all the sequences which can be built with itemsets containing items from $\mathcal{I}$. A *dataset* $\mathcal{D}$ is a finite bag of *(sequential) transactions* where each transaction is a sequence from $\mathbb{U}$. A sequence p *belongs* to a transaction $\tau \in \mathcal{D}$ if and only if $p \sqsubseteq \tau$. For any sequence p, the *support set* $T_{\mathcal{D}}(p)$ of p in $\mathcal{D}$ is the set of transactions in $\mathcal{D}$ to which p belongs: $T_{\mathcal{D}}(p) = \{\tau \in \mathcal{D} : p \sqsubseteq \tau\}$. The *support* $Supp_{\mathcal{D}}(p)$ of p in $\mathcal{D}$ is the cardinality of the set $T_{\mathcal{D}}(p)$, that is the number of transactions in $\mathcal{D}$ to which p belongs: $Supp_{\mathcal{D}}(p) = |T_{\mathcal{D}}(p)|$. Finally, the *frequency* $f_{\mathcal{D}}(p)$ of p in $\mathcal{D}$ is the *fraction* of transactions in $\mathcal{D}$ to which p belongs:

$$f_{\mathcal{D}}(p) = \frac{Supp_{\mathcal{D}}(p)}{|\mathcal{D}|}. \tag{3.2}$$

A sequence p is *closed* w.r.t. $\mathcal{D}$ if for each of its super-sequences $y \sqsupset p$ we have $f_{\mathcal{D}}(y) < f_{\mathcal{D}}(p)$, or, equivalently, none of its super-sequence has support equal to $f_{\mathcal{D}}(p)$. We denote the set of all closed sequences in $\mathcal{D}$ with $CS(\mathcal{D})$.

Example 1. *Consider the following dataset* $\mathcal{D} = \{\tau_1, \tau_2, \tau_3, \tau_4\}$ *as example:*

$$\begin{aligned}
\tau_1 &= \langle \{6,7\}, \{5\}, \{7\}, \{5\} \rangle \\
\tau_2 &= \langle \{1\}, \{2\}, \{6,7\}, \{5\} \rangle \\
\tau_3 &= \langle \{1,4\}, \{3\}, \{2\}, \{1,2,5,6\} \rangle \\
\tau_4 &= \langle \{1\}, \{2\}, \{6,7\}, \{5\} \rangle
\end{aligned}$$

The dataset above has 4 transactions. The first one, $\tau_1 = \langle\{6,7\},\{5\},\{7\},\{5\}\rangle$*, it is a sequence of length* $|\tau_1| = 4$ *and item-length* $||\tau_1|| = 5$*. The frequency* $f_{\mathcal{D}}(\langle\{7\},\{5\}\rangle)$ *of* $\langle\{7\},\{5\}\rangle$ *in* $\mathcal{D}$*, is 3/4, since it is contained in all transactions but* τ_3*. Note that the sequence* $\langle\{7\},\{5\}\rangle$ *occurs three times as a subsequence of* τ_1*, but* τ_1 *contributes only once to the frequency of* $\langle\{7\},\{5\}\rangle$*. The sequence* $\langle\{7\},\{6\},\{5\}\rangle$ *is not a subsequence of* τ_1 *because the order of the itemsets in the two sequences is not the same. Note that from the definitions above, an item can only occur once in an itemset, but it can occur multiple times in different itemsets of the same sequence. Finally, the sequence* $\langle\{6,7\},\{5\}\rangle$*, whose frequency is 3/4, is a* closed *sequence, since its frequency is higher than the frequency of each of its super-sequences.*

Given a minimum frequency threshold $\theta \in (0,1]$, we define the set $FSP(\mathcal{D},\theta)$ as the set of all the sequential patterns (and their frequencies) whose frequency in $\mathcal{D}$ is at least θ, that is

$$FSP(\mathcal{D},\theta) = \{(p, f_{\mathcal{D}}(p)) : p \in \mathbb{U}, f_{\mathcal{D}}(p) \geq \theta\}. \tag{3.3}$$

In this work our aim is to use the transactional dataset $\mathcal{D}$ in order to approximate the true frequent sequential patterns, i.e., the sequential patterns that are frequently generated from the unknown distribution π that generates sequential patterns. Thus, given a minimum frequency threshold $\theta \in (0,1]$, we are interested in finding a rigorous approximation to the set $TFSP(\pi,\theta)$ of true frequent sequential patterns, which is defined as follows:

$$TFSP(\pi,\theta) = \{(p, t_\pi(p)) : p \in \mathbb{U}, t_\pi(p) \geq \theta\}, \tag{3.4}$$

where, recalling from Section 2.1.2, $t_\pi(p)$ is the true frequency of sequential pattern p w.r.t. π.

Now we provide two different definitions of approximation of $TFSP(\pi,\theta)$. The first definition provides an approximation that does not have false negatives, i.e., that does not miss any true frequent sequential pattern, similarly to Definition 3.

Definition 9. *Given* $\mu \in (0,1)$*, a false negatives free (FNF)* μ*-approximation* $\mathcal{E}$ *of* $TFSP(\pi,\theta)$ *is defined as a set of pairs* (p, f_p)*:*

$$\mathcal{E} = \{(p, f_p) : p \in \mathbb{U}, f_p \in [0,1]\} \tag{3.5}$$

that has the following properties:

- $\mathcal{E}$ *contains a pair* (p, f_p) *for every* $(p, t_\pi(p)) \in TFSP(\pi, \theta)$*;*
- $\mathcal{E}$ *contains no pair* (p, f_p) *such that* $t_\pi(p) < \theta - \mu$*;*
- *for every* $(p, f_p) \in \mathcal{E}$*, it holds* $|t_\pi(p) - f_p| \leq \mu/2$*.*

While the previous definition provides guarantees to avoid false negatives, depending on the application instead one may want to avoid false positives, i.e., sequential patterns that are erroneously reported as true frequent. The following definition provides an approximation that does not contain false positives.

Definition 10. *Given* $\mu \in (0, 1)$*, a false positives free (FPF)* μ*-approximation* $\mathcal{G}$ *of* $TFSP(\pi, \theta)$ *is defined as a set of pairs* (p, f_p)*:*

$$\mathcal{G} = \{(p, f_p) : p \in \mathbb{U}, f_p \in [0, 1]\} \tag{3.6}$$

that has the following properties:

- $\mathcal{G}$ *contains no pair* (p, f_p) *such that* $t_\pi(p) < \theta$*;*
- $\mathcal{G}$ *contains all the pairs* (p, f_p) *such that* $t_\pi(p) \geq \theta + \mu$*;*
- *for every* $(p, f_p) \in \mathcal{G}$*, it holds* $|t_\pi(p) - f_p| \leq \mu/2$*.*

In the next section we present the Rademacher complexity of sequential patterns, which is a crucial tool to find an upper bound to the maximum deviation $\sup_{p \in \mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)|$, and then to find rigorous approximations of $TFSP(\pi, \theta)$ accordingly with Definition 9 and Definition 10.

3.3 Rademacher Complexity of Sequential Patterns

In this section we introduce the Rademacher complexity of sequential patterns. We propose a method for finding an efficiently computable upper bound to the empirical Rademacher complexity $R_{\mathcal{D}}$ of sequential patterns (similar to what has been done in [Riondato and Upfal, 2015] for itemsets) and a method for approximating it. In the true frequent pattern mining scenario, these results will be useful to define a quantity which is an upper bound to the maximum deviation $\sup_{p \in \mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)|$ with high probability.

The introduction of the Rademacher complexity of sequential patterns requires the definition of a set of real-valued functions. We define, for each pattern $p \in \mathbb{U}$, the indicator function $\phi_p : \mathbb{U} \to \{0, 1\}$ as:

$$\phi_p(t) = \begin{cases} 1 & if\ p \sqsubseteq t \\ 0 & otherwise \end{cases}, \tag{3.7}$$

where t is a transaction. Given a transaction t of a dataset $\mathcal{D}$ with n transactions, $\phi_p(t)$ is 1 if p appears in t, otherwise it is 0. We define the set of real-valued functions as the family of these indicator functions. The frequency of p in $\mathcal{D}$ can be defined using the indicator function ϕ_p: $f_{\mathcal{D}}(p) = \sum_{t \in \mathcal{D}} \phi_p(t)/n$. The *(empirical) Rademacher complexity* $R_{\mathcal{D}}$ on a given dataset $\mathcal{D}$ is defined as:

$$R_{\mathcal{D}} = \mathbb{E}_{\sigma}\left[\sup_{p \in \mathbb{U}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \phi_p(t_i)\right], \tag{3.8}$$

where the expectation is taken w.r.t. the Rademacher r.v. σ_i, that is, conditionally on the dataset $\mathcal{D}$. The connection between the Rademacher complexity of sequential patterns and the maximum deviation is given by the following theorem, which derives from standard results in statistical learning theory (Thm. 3.2 in [Boucheron et al., 2005]).

Theorem 2. *With probability at least* $1 - \delta$*:*

$$\sup_{p \in \mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq 2R_{\mathcal{D}} + \sqrt{\frac{2 \ln (2/\delta)}{|\mathcal{D}|}} = \frac{\mu_R}{2}. \tag{3.9}$$

The naïve computation of the exact value of $R_{\mathcal{D}}$ is expensive since it requires to mine all patterns from $\mathcal{D}$ and to generate all possible 2^n combination values of the Rademacher variables for the computation of the expectation. In the next sections we present an efficiently computable upper bound on the Rademacher complexity of sequential patterns and a simple method that approximates it, which are useful to find, respectively, an upper bound and an approximation to $\mu_R/2$.

3.3.1 An Efficiently Computable Upper Bound to the Rademacher Complexity of Sequential Patterns

For any pattern $p \in \mathbb{U}$, let us define the following $|\mathcal{D}|$-dimensional vector

$$v_{\mathcal{D}}(p) = (\phi_p(t_1), \ldots, \phi_p(t_{|\mathcal{D}|})) \tag{3.10}$$

and let $V_{\mathcal{D}} = \{v_{\mathcal{D}}(p), p \in \mathbb{U}\}$, where $t_1, t_2, \ldots, t_{|\mathcal{D}|}$ are the $|\mathcal{D}|$ transactions of $\mathcal{D}$. Note that all the infinite sequences of the universe $\mathbb{U}$ which do not appear in $\mathcal{D}$ are associated with the vector $(0, \ldots, 0)$ of $|\mathcal{D}|$ zeros. This implies the finiteness of the size of $V_{\mathcal{D}}$: $|V_{\mathcal{D}}| < \infty$. In addition, defining $|\mathbb{U}(\mathcal{D})|$ as the number of sequential patterns that appear in $\mathcal{D}$, we have that potentially $|V_{\mathcal{D}}| \ll |\mathbb{U}(\mathcal{D})|$, since there may be two or more patterns associated with the same vector $v_{\mathcal{D}} \in V_{\mathcal{D}}$ (i.e., these patterns appear exactly in the same transactions).

The following two theorems derive from known results of statistical learning theory (Thm. 3.3 of [Boucheron et al., 2005]). Both theorems have been used for mining frequent itemsets [Riondato and Upfal, 2015], and can be applied for sequential pattern mining.

Theorem 3. *(Massart's Lemma)*

$$R_{\mathcal{D}} \leq \max_{p \in \mathbb{U}} ||v_{\mathcal{D}}(p)|| \frac{\sqrt{2 \ln |V_{\mathcal{D}}|}}{|\mathcal{D}|} \tag{3.11}$$

where $|| \cdot ||$ *indicates the Euclidean norm.*

The following theorem is a stronger version of the previous one.

Theorem 4. *Let* $w : \mathbb{R}^+ \to \mathbb{R}^+$ *be the function*

$$w(s) = \frac{1}{s} \ln \sum_{v \in V_{\mathcal{D}}} \exp\left(\frac{s^2 ||v||^2}{2|\mathcal{D}|^2}\right), \tag{3.12}$$

then

$$R_{\mathcal{D}} \leq \min_{s \in \mathbb{R}^+} w(s). \tag{3.13}$$

The upper bound on $R_{\mathcal{D}}$ of Theorem 4 is not directly applicable to sequential pattern mining since it requires to mine every pattern that appear in $\mathcal{D}$ in order to determine the entire set $V_{\mathcal{D}}$. However, the set $V_{\mathcal{D}}$ is related to the set of closed sequential patterns on $\mathcal{D}$. The following two results give us an upper bound to the size of $V_{\mathcal{D}}$ which depends on the number of closed sequential patterns of $\mathcal{D}$.

Lemma 1. *Consider a subset* W *of the dataset* $\mathcal{D}$, $W \subseteq \mathcal{D}$. *Let* $CS_W(\mathcal{D})$ *be the set of closed sequential patterns in* $\mathcal{D}$ *whose support set in* $\mathcal{D}$ *is* W, *that is,* $CS_W(\mathcal{D}) = \{p \in CS(\mathcal{D}) : T_{\mathcal{D}}(p) = W\}$, *with* $C = |CS_W(\mathcal{D})|$. *Then the number* C *of closed sequential patterns in* $\mathcal{D}$ *with* W *as support set satisfies:* $0 \leq C \leq |CS(\mathcal{D})|$.

Proof. The proof is organized in such a way: first, we show that the basic cases $C = 0$ and $C = 1$ hold, second, we prove the cases for $2 \leq C \leq |CS(\mathcal{D})|$.

Let us consider the case where W is a particular subset of $\mathcal{D}$ for which no sequence has W as support set in $\mathcal{D}$. Thus, $CS_W(\mathcal{D})$ is an empty set and $C = 0$. The case $C = 1$ is trivial, since it could happen that only one closed sequential pattern has W as support set in $\mathcal{D}$.

Now, before proving the cases for a generic value of C in $[2, \ldots, |CS(\mathcal{D})|]$, we start by considering the case $C = 2$. Let p_1,p_2 be two sequences with W as support set. Assume that each super-sequence of p_1 but not of p_2 has support lower than the support of p_1, and each super-sequence of p_2 but not of p_1 has support lower than the support of p_2. Now, let us focus on super-sequences of both p_1 and p_2. Let $\tau \in W$ be a transaction of W. We define $\mathbf{y}_\tau = \tau_{p_1,p_2}$ as the subsequence of τ restricted to only the sequences p_1 and p_2, preserving the relative order of their itemsets. For instance, let $p_1 = \langle A, B\rangle$, $p_2 = \langle C, D\rangle$ and $\tau = \langle A, C, F, D, B\rangle$, where A, B, C, D, F are itemsets: thus, $\mathbf{y}_\tau = \langle A, C, D, B\rangle$. Now, if the support set of $\mathbf{y}_\tau$ in W does not coincide with W, that is, $T_W(\mathbf{y}_\tau) \subset W$, then for each transaction $\tau \in W$ we have $|T_W(\mathbf{y}_\tau)| < |T_W(p_1)| = |T_W(p_2)| = |W|$. Note that this could happen because the set of itemsets of p_1 and p_2 may not appear in the same order in all transactions. Hence each super-sequence of both p_1 and p_2 has support lower than the support of p_1 (that is equal to the support of p_2). Thus, each super-sequence of p_i has a lower support compared to the support of p_i, for $i = 1, 2$. This implies that p_1 and p_2 are closed sequences in $\mathcal{D}$ and since their support set is W, they belong to $CS_W(\mathcal{D})$. Thus, the case $C = 2$ could happen.

Now we generalize this concept for a generic number C of closed sequential patterns, where $2 \leq C \leq |CS(\mathcal{D})|$. Let $H = \{p_1, p_2, \ldots, p_C\}$ be a set of C sequential patterns with W as support set. Assume that each super-sequence of p_i but not of p_k has support lower than the support of p_i, for each $i, k \in [1, \ldots, C]$ with $k \neq i$. Let H_p be the power set of H without the empty set and the sets made of only one sequence, that is, $H_p = P(H) \setminus \{\{\emptyset\}, \{p_1\}, \{p_2\}, \ldots, \{p_C\}\}$. So, in H_p there are every possible subset of H of size greater than one. For a transaction $\tau \in W$ and $h_p \in H_p$, we define $\mathbf{y}_\tau(h_p) = \tau_{h_p}$ as the subsequence of τ restricted to h_p, that is, to only the sequences $p \in h_p$, preserving the relative order of their itemsets. If $\forall h_p \in H_p$ there exits a transaction $\tau \in W$ such that the support set of $\mathbf{y}_\tau(h_p)$ in W does not coincide with W, that is, $T_W(\mathbf{y}_\tau(h_p)) \subset W$, then for each transaction $\tau \in W$ we have $|T_W(\mathbf{y}_\tau(h_p))| < |T_W(p_1)| = \cdots = |T_W(p_C)| = |W|$. Hence

each super-sequence made of only sequences of h_p has support lower than the support of p_i, for $i = 1, \ldots, C$. Thus, each super-sequence of p_i has a lower support compared to the support of p_i, for $i = 1, \ldots, C$. This implies that all sequences of H are closed sequence in $\mathcal{D}$ and since their support set is W, they belong to $CS_W(\mathcal{D})$. □

Now we present an example where $C = 2$.

Example 2. *A simple example where $C = 2$ is depicted in Figure 3.1. Note first of all that each super-sequence of x_1 but not of x_2 has support lower than the support of x_1, and each super-sequence of x_2 but not of x_1 has support lower than the support of x_2. Let $\boldsymbol{y}_\tau = \tau_{x_1,x_2}$ be the subsequence of transaction τ restricted to only the sequences x_1 and x_2, preserving the relative order of their itemsets. Then $y_{\tau_1} = y_{\tau_3} \neq y_{\tau_2}$ which implies $|T_W(y_{\tau_1})|$, $|T_W(y_{\tau_2})|$, and $|T_W(y_{\tau_3})|$ be lower than $|T_W(x_1)| = |T_W(x_2)| = |W|$. Therefore each super-sequence of both x_1 and x_2 has support lower than the support of x_1 (i.e. equal to the one of x_2). Thus, x_1 and x_2 are closed sequences in $\mathcal{D}$ with the same support set W.*

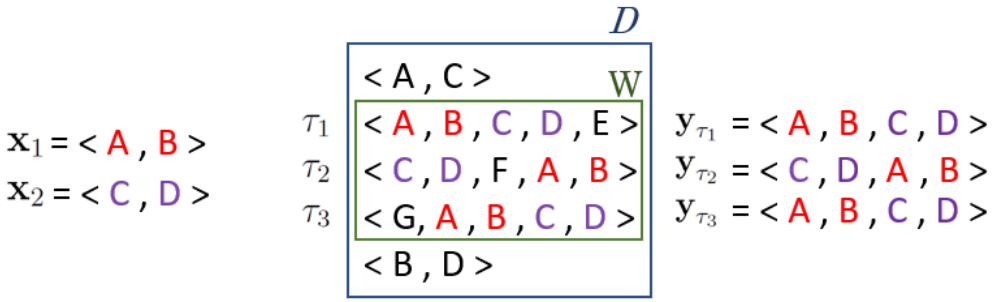


Figure 3.1: Graphical representation of the case $CS_W(\mathcal{D}) = 2$. Sequences x_1 and x_2 are closed sequences in $\mathcal{D}$ with the same support set W.

Note that Lemma 1 represents a sequential patterns version of Lemma 3 of [Riondato and Upfal, 2015] for itemsets, where the upper bound to the number of closed itemsets in $\mathcal{D}$ with W as support set is one (this holds by the nature of the itemsets where the notion of "ordering" is not defined). Lemma 1 is crucial for proving the following lemma which provides a bound on the size of the set $V_\mathcal{D}$ of binary vectors.

Lemma 2. *$V_{\mathcal{D}} = \{v_{\mathcal{D}}(p) : p \in CS(\mathcal{D})\} \cup \{(0,\dots,0)\}$ and $|V_{\mathcal{D}}| \leq |CS(\mathcal{D})| + 1$, that is, each vector of $V_{\mathcal{D}}$ different from $(0,\dots,0)$ is associated with at least one closed sequential pattern in $\mathcal{D}$.*

Proof. Let $V_{\mathcal{D}} = \overline{V}_{\mathcal{D}} \cup \{(0,\dots,0)\}$, where $\overline{V}_{\mathcal{D}} = \{v \in V_{\mathcal{D}} : v \neq (0,\dots,0)\}$. Let $p \in \mathbb{U}$ be a sequence of non-empty support set in $\mathcal{D}$, that is, $v_{\mathcal{D}}(p) \neq (0,\dots,0)$. There are two possibilities: p is or is not a closed sequence in $\mathcal{D}$. If p is not a closed sequence, then there exists a closed super-sequence $\mathbf{y} \sqsupset p$ with support equal to the support of p, so with $v_{\mathcal{D}}(p) = v_{\mathcal{D}}(\mathbf{y})$. Thus, $v_{\mathcal{D}}(p)$ is associated with at least one closed sequence. Combining this with the fact that each vector $v \in \overline{V}_{\mathcal{D}}$ is associated with at least one sequence $p \in \mathbb{U}$ and Lemma 1, then each vector of $V_{\mathcal{D}}$ different from $(0,\dots,0)$ is associated with at least one closed sequential pattern of $\mathcal{D}$. To conclude our proof is sufficient to show that there are no closed sequences associated with the vector $(0,\dots,0)$. Let $SP_\infty = \{p \in \mathbb{U} : v_{\mathcal{D}}(p) = (0,\dots,0)\}$. Note that $|SP_\infty| = \infty$. For each $p \in SP_\infty$, there always exists a super-sequence $\mathbf{y} \sqsupset p$ such that $f_{\mathcal{D}}(p) = f_{\mathcal{D}}(\mathbf{y}) = 0$. This implies that each sequence of SP_∞ is not closed. Thus, $\overline{V}_{\mathcal{D}} = \{v_{\mathcal{D}}(p) : p \in CS(\mathcal{D})\}$ and $|V_{\mathcal{D}}| = |\overline{V}_{\mathcal{D}}| + 1 \leq |CS(\mathcal{D})| + 1$. □

Combining a partitioning of $CS(\mathcal{D})$ with the previous lemma we can define a function $\tilde{w}$, an upper bound to the function w of Theorem 4, which is efficient to compute with a single scan of $\mathcal{D}$. Let $\mathcal{I}$ be the set of items that appear in the dataset $\mathcal{D}$ and $<_o$ be its increasing ordering by their support in $\mathcal{D}$ (ties broken arbitrarily). Given an item a, let $T_{\mathcal{D}}(\langle\{a\}\rangle)$ be its support set on $\mathcal{D}$. Let $<_a$ denote the increasing ordering of the transactions $T_{\mathcal{D}}(\langle\{a\}\rangle)$ by the number of items contained that come after a w.r.t. the ordering $<_o$ (ties broken arbitrarily). Let $CS(\mathcal{D}) = C_1 \cup C_{2+}$, where $C_1 = \{p \in CS(\mathcal{D}) : ||p|| = 1\}$ and $C_{2+} = \{p \in CS(\mathcal{D}) : ||p|| \geq 2\}$. Let us focus on partitioning C_{2+}. Let $p \in C_{2+}$ and let a be the item in p which comes before any other item in p w.r.t. the order $<_o$. Let τ be the transaction containing p which comes before any other transaction containing p w.r.t. the order $<_a$. We assign p to the set $C_{a,\tau}$. Remember that an item can appear multiple times in a sequence. Given a transaction $\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)$, $b_{a,\tau}$ is the number of items in τ (counted with their multiplicity) equal to a or that come after a in $<_o$. Let $z_{a,\tau}$ be the multiplicity of a in τ. For each $b, z \geq 1$, $z \leq b$, let $\varphi_{a,b,z}$ be the number of transactions in $T_{\mathcal{D}}(\langle\{a\}\rangle)$ that contain exactly b items (counted with their multiplicity) equal to a or located after a in the ordering $<_o$, with

exactly z repetitions of a. Let $\chi_a = max\{b : \varphi_{a,b,z} > 0\}$. The following lemma gives us an upper bound to the size of $C_{a,\tau}$.

Lemma 3. *We have*

$$|C_{a,\tau}| \leq 2^{b_{a,\tau}-z_{a,\tau}}(2^{z_{a,\tau}} - 1). \tag{3.14}$$

Proof. $C_{a,\tau}$ represents a subset of the set Φ of all those subsequences of τ that are made of only items equal to a or that come after a in $<_o$, with item-length at least two and with at least one occurrence of a. Let us focus on finding an upper bound to $|\Phi|$. In order to build such a generic subsequence of τ, it is sufficient to select i occurrences of a among the $z_{a,\tau}$ available, with $1 \leq i \leq z_{a,\tau}$, and choose j items among the remaining $b_{a,\tau} - z_{a,\tau}$ items different from a. Note that if $i = 1$, then j must be greater than 0. Thus, using the fact that the sum of $\binom{n}{k}$ for $k = 0, \ldots, n$ is equal to 2^n, we have

$$|\Phi| \leq \binom{z_{a,\tau}}{1} \sum_{j=1}^{b_{a,\tau}-z_{a,\tau}} \binom{b_{a,\tau} - z_{a,\tau}}{j} + \sum_{i=2}^{z_{a,\tau}} \left[\binom{z_{a,\tau}}{i} \sum_{j=0}^{b_{a,\tau}-z_{a,\tau}} \binom{b_{a,\tau} - z_{a,\tau}}{j} \right] \leq$$

$$\leq 2^{b_{a,\tau}-z_{a,\tau}} \sum_{i=1}^{z_{a,\tau}} \binom{z_{a,\tau}}{i} = 2^{b_{a,\tau}-z_{a,\tau}}(2^{z_{a,\tau}} - 1), \tag{3.15}$$

where the first inequality holds because some sequences of Φ are counted more times. Since $|C_{a,\tau}| \leq |\Phi|$, the thesis holds. □

Combining the following partitioning of $CS(\mathcal{D})$ as

$$CS(\mathcal{D}) = C_1 \cup C_{2+} = C_1 \cup \left(\bigcup_{a \in \mathcal{I}} \bigcup_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} C_{a,\tau} \right) \tag{3.16}$$

with the previous lemma, we obtain

$$|CS(\mathcal{D})| \leq |\mathcal{I}| + \sum_{a \in \mathcal{I}} \sum_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} 2^{b_{a,\tau}-z_{a,\tau}}(2^{z_{a,\tau}} - 1). \tag{3.17}$$

Now we are ready to define the function $\tilde{w}$, which can be used to obtain an efficiently computable upper bound to $R_{\mathcal{D}}$. The following lemma represents the analogous of Lemma 5 of [Riondato and Upfal, 2015], adjusted for sequential patterns. Let $\overline{\eta}$ be the average item-length of the transactions of

$\mathcal{D}$, that is, $\overline{\eta} = \sum_{t\in\mathcal{D}} ||t||/n$. Let $\hat{\eta}$ be the maximum item-length of the transactions of $\mathcal{D}$, that is, $\hat{\eta} = \max_{t\in\mathcal{D}} ||t||$. Let η be an item-length threshold, with $\overline{\eta} < \eta \leq \hat{\eta}$. Let $\mathcal{D}(\eta)$ be the bag of transactions of $\mathcal{D}$ with item-length greater than η. Let $V_{\mathcal{D}(\eta)}$ be the set of the $2^{|\mathcal{D}(\eta)|}-1$ binary vectors associated with all possible non-empty sub-bags of $\mathcal{D}(\eta)$.

Lemma 4. *Given an item a in $\mathcal{I}$, we define the following quantity:*

$$q(a,\eta) = 1 + \sum_{b=1}^{\chi_a} \sum_{z=1}^{b} \sum_{j=1}^{\varphi_{a,b,z}} \left(\mathbb{1}(b \leq \eta) 2^{b-z}(2^z - 1) + \mathbb{1}(b > \eta) \sum_{i=1}^{\eta-1} \binom{b-1}{i} \right). \tag{3.18}$$

Let $\tilde{w} : \mathbb{R}^+ \to \mathbb{R}^+$ be the function

$$\tilde{w}(s,\eta) = \frac{1}{s} \ln \sum_{a\in\mathcal{I}} \left(q(a,\eta) e^{\frac{s^2 f_{\mathcal{D}}(\langle\{a\}\rangle)}{2|\mathcal{D}|}} + |V_{\mathcal{D}(\eta)}| e^{\frac{s^2|\mathcal{D}(\eta)|}{2|\mathcal{D}|^2}} + 1 \right). \tag{3.19}$$

Then,

$$R_{\mathcal{D}} \leq \min_{s\in\mathbb{R}^+, \overline{\eta}<\eta\leq\hat{\eta}} \tilde{w}(s,\eta). \tag{3.20}$$

Proof. Let us consider the function w from Theorem 4. For a given value of η, we have that $V_{\mathcal{D}} \subseteq (V_{\mathcal{D}} \setminus V_{\mathcal{D}(\eta)}) \cup V_{\mathcal{D}(\eta)}$, since not all the binary vectors of $V_{\mathcal{D}(\eta)}$ necessarily belong to $V_{\mathcal{D}}$. Thus:

$$w(s) = \frac{1}{s} \ln \sum_{v\in V_{\mathcal{D}}} \exp\left(\frac{s^2||v||^2}{2n^2}\right) \leq$$

$$\leq \frac{1}{s} \ln \left(\sum_{v\in V_{\mathcal{D}}\setminus V_{\mathcal{D}(\eta)}} \exp\left(\frac{s^2||v||^2}{2n^2}\right) + \sum_{v\in V_{\mathcal{D}(\eta)}} \exp\left(\frac{s^2||v||^2}{2n^2}\right) \right), \tag{3.21}$$

where $n = |\mathcal{D}|$. For each binary vector $v \in V_{\mathcal{D}(\eta)}$ the maximum number of 1's is $|\mathcal{D}(\eta)|$. Thus,

$$\sum_{v\in V_{\mathcal{D}(\eta)}} \exp\left(\frac{s^2||v||^2}{2n^2}\right) \leq |V_{\mathcal{D}(\eta)}| \exp\left(\frac{s^2|\mathcal{D}(\eta)|}{2n^2}\right). \tag{3.22}$$

By using the definition of Euclidean norm, we have that, for any sequence $p \in \mathbb{U}$,

$$||v_{\mathcal{D}}(p)|| = \sqrt{\sum_{i=1}^{n} \phi_p(t_i)^2} = \sqrt{n f_{\mathcal{D}}(p)}. \tag{3.23}$$

Note that each closed sequential pattern p with $||p|| > \eta$ can only appear in transactions of $\mathcal{D}(\eta)$ and, consequently, it is associated with a binary vector of $V_{\mathcal{D}(\eta)}$ and not of $V_{\mathcal{D}} \setminus V_{\mathcal{D}(\eta)}$. Thus, defining $CS(\mathcal{D}, \eta)$ as the set of closed sequential patterns of $\mathcal{D}$ with item-length lower or equal to η and using Lemma 2 we can use the sum over $CS(\mathcal{D}, \eta)$ as an upper bound on the sum over $V_{\mathcal{D}} \setminus V_{\mathcal{D}(\eta)}$:

$$\sum_{v \in V_{\mathcal{D}} \setminus V_{\mathcal{D}(\eta)}} \exp\left(\frac{s^2 ||v||^2}{2n^2}\right) \leq \sum_{p \in CS(\mathcal{D},\eta)} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right) + 1. \tag{3.24}$$

Note that the vector $(0, \ldots, 0)$ of $V_{\mathcal{D}} \setminus V_{\mathcal{D}(\eta)}$ provides a $+1$.

Now let us focus on the first term of the sum. The set $CS(\mathcal{D}, \eta)$ can be broken using the Equation 3.16 in the sum over C_1

$$\sum_{p \in C_1} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right) \tag{3.25}$$

plus the sum over $C_{2^+}(\eta)$ (i.e., the set of closed sequential patterns with item-length in $[2, \eta]$)

$$\sum_{a \in \mathcal{I}} \sum_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} \sum_{p \in C_{a,\tau}(\eta)} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right), \tag{3.26}$$

where $C_{a,\tau}(\eta)$ is the set of closed sequential patterns of $C_{a,\tau}$ with item-length in $[2, \eta]$. Since the set of items of the sequences in C_1 is a subset of $\mathcal{I}$, we have

$$\sum_{p \in C_1} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right) \leq \sum_{a \in \mathcal{I}} \exp\left(\frac{s^2 f_{\mathcal{D}}(\langle\{a\}\rangle)}{2n}\right). \tag{3.27}$$

For any $p \in C_{a,\tau}(\eta)$, $f_{\mathcal{D}}(p) \leq f_{\mathcal{D}}(\langle\{a\}\rangle)$ by the anti-monotonicity support property for sequential patterns. An upper bound to the size of $C_{a,\tau}(\eta)$ can be computed in two ways, depending on the value of $b_{a,\tau}$. If $b_{a,\tau} \leq \eta$, we can use Lemma 3:

$$\sum_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} \sum_{p \in C_{a,\tau}(\eta)} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right) \leq$$

$$\leq \sum_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} 2^{b_{a,\tau}-z_{a,\tau}}(2^{z_{a,\tau}}-1)\exp\left(\frac{s^2 f_{\mathcal{D}}(\langle\{a\}\rangle)}{2n}\right). \tag{3.28}$$

If $b_{a,\tau} > \eta$ we have to count the number of possible closed sequential patterns with at least one item equal to a and with item-length in $[2,\eta]$ that we can build from $b_{a,\tau}$ items of τ:

$$\sum_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} \sum_{p \in C_{a,\tau}(\eta)} \exp\left(\frac{s^2 f_{\mathcal{D}}(p)}{2n}\right) \leq$$

$$\leq \sum_{\tau \in T_{\mathcal{D}}(\langle\{a\}\rangle)} \sum_{i=1}^{\eta-1} \binom{b_{a,\tau}-1}{i} \exp\left(\frac{s^2 f_{\mathcal{D}}(\langle\{a\}\rangle)}{2n}\right). \tag{3.29}$$

Finally, using the quantities χ,b,z and φ previously defined and indicator functions we can merge the right-hand sides of the last two inequalities

$$\sum_{b=1}^{\chi_a}\sum_{z=1}^{b}\sum_{j=1}^{\varphi_{a,b,z}} (\mathbb{1}(b \leq \eta)2^{b-z}(2^z-1)+\mathbb{1}(b>\eta)\sum_{i=1}^{\eta-1}\binom{b-1}{i})\exp\left(\frac{s^2 f_{\mathcal{D}}(\langle\{a\}\rangle)}{2n}\right). \tag{3.30}$$

Thus, rearranging all the terms we reach the definition of $\tilde{w}$. Using the above arguments and the best value of η which minimizes the function we have that $w(s) \leq \tilde{w}(s,\eta)$ for any $s \in \mathbb{R}^+$, $\overline{\eta} < \eta \leq \hat{\eta}$. Since $R_{\mathcal{D}} \leq \min_{s\in\mathbb{R}^+} w(s)$ (by Theorem 4), we conclude that $R_{\mathcal{D}} \leq \min_{s\in\mathbb{R}^+,\overline{\eta}<\eta\leq\hat{\eta}} \tilde{w}(s,\eta)$. □

For a given value of η, the function $\tilde{w}$ can be compute with a single scan of the dataset, since it requires to know $\varphi_{a,b,z}$ for each $a \in \mathcal{I}$ and for each b, z, $1 \leq b \leq \chi_a$, $1 \leq z \leq b$. The values $\overline{\eta}$, $\hat{\eta}$, and the support of each item and consequently the ordering $<_o$ are obtained during the dataset creation. Thus, it is sufficient to look at each transaction τ, sorting the items $\mathcal{I}_\tau$ that appear in τ according to $<_o$, and, for each item of $\mathcal{I}_\tau$, keep track of its multiplicity $z_{a,\tau}$, compute $b_{a,\tau}$ and increase by one $g_{a,b_{a,\tau},z_{a,\tau}}$. Finally, since $\tilde{w}$ is convex and has first and second derivatives w.r.t. s everywhere in $\mathbb{R}^+$, its global minimum can be computed using a non-linear optimization solver. This procedure has to be repeated for each possible value of η in $(\overline{\eta},\hat{\eta}]$.

However, one could choose a particular schedule of values of η to be tested, instead of taking into account each possible value, achieving a value of the function $\tilde{w}$ near to its minimum. A possible choice is to look at the restricted interval $[\overline{\eta}+\beta_1, \min(\beta_2,\hat{\eta})]$, given two positive values for β_1 and β_2, instead

of investigating the whole interval $(\overline{\eta}, \hat{\eta}]$. This choice is motivated by the fact that in Lemma 4 the value of η gives us an idea of which term of the summation is dominant (the one based on closed sequential patterns or the one based on binary vectors). If η is close to $\overline{\eta}$ then the number of binary vectors we count could be high, the dominant term is the one based on the set of binary vectors, and we expect the upper bound to be high. Instead, if η is close to $\hat{\eta}$ then the upper bound to the number of closed sequential patterns we count could be high, and the set of binary vectors we take into account is small. In this case, the dominant term is the one based on the closed sequential patterns, and the value of the upper bound could be high (since we count many sequential patterns with item-length greater than η that instead would be associated with a small number of binary vectors). Thus, the best value of η will be the one that is larger than $\overline{\eta}$ and smaller than $\hat{\eta}$, enough to count not too many closed sequential patterns and binary vectors.

The pseudo-code of the algorithm for computing the upper bound to $R_{\mathcal{D}}$ follows.

Algorithm 1: RadeBound($\mathcal{D}$): algorithm for bounding the empirical Rademacher complexity of sequential patterns

Data: : a sequential dataset $\mathcal{D}$ built on alphabet $\mathcal{I}$
Result: upper bound $R^{ub}_{\mathcal{D}}$ to $R_{\mathcal{D}}$

$\varphi_{a,b,z} \leftarrow 0, \forall a \in \mathcal{I},\ b, z \in \mathbb{N},\ z \leq b$;
$\chi_a \leftarrow 0, \forall a \in \mathcal{I}$;
/* $\overline{\eta}$, $\hat{\eta}$, and the support of the items are computed during the scan of $\mathcal{D}$ */
for $\tau \in \mathcal{D}$ **do**
    for $a \in \tau$ **do**
        $b_{a,\tau} \leftarrow$ number of items in τ (counted with their multiplicity) equal to a or that come after a in $<_o$;
        $z_{a,\tau} \leftarrow$ number of repetitions of a in τ;
        $g_{a,b_{a,\tau},z_{a,\tau}} += 1$;
        $\chi_a \leftarrow \max(\chi_a, b_{a,\tau})$;
return $R^{ub}_{\mathcal{D}} = \min_{s \in \mathbb{R}^+, \overline{\eta} < \eta \leq \hat{\eta}} \tilde{w}(s, \eta)$;

Finally, we define *ComputeMaxDevRadeBound* as the procedure for computing an upper bound $\mu^{ub}_R/2$ to $\mu_R/2$ where, once the upper bound $R^{ub}_{\mathcal{D}}$ to the Rademacher complexity $R_{\mathcal{D}}$ is computed using Algorithm 1, the upper

bound $\mu_R^{ub}/2$ to $\mu_R/2$ is obtained by

$$\frac{\mu_R^{ub}}{2} = 2R_{\mathcal{D}}^{ub} + \sqrt{\frac{2\ln(2/\delta)}{|\mathcal{D}|}}. \tag{3.31}$$

3.3.2 Approximating the Rademacher Complexity of Sequential Patterns

The previous section presents an efficiently computable upper bound to the Rademacher of sequential patterns, which does not require any extraction of frequent sequences from a given dataset. Here we present a simple method to approximate the Rademacher complexity of sequential patterns, which provides a tighter bound to the maximum deviation compared to $\mu_R^{ub}/2$ of Equation 3.31

In the definition of the Rademacher complexity, a given combination $\overline{\sigma}$ of the Rademacher r.v. σ splits the dataset $\mathcal{D}$ of n transactions in two subsamples $\mathcal{D}_1(\overline{\sigma})$ and $\mathcal{D}_{-1}(\overline{\sigma})$: each transaction associated with 1 and -1 goes respectively into $\mathcal{D}_1(\overline{\sigma})$ and $\mathcal{D}_{-1}(\overline{\sigma})$. For a given sequential pattern $p \in \mathbb{U}$, let $Supp_{\mathcal{D}_1(\overline{\sigma})}(p)$ and $Supp_{\mathcal{D}_{-1}(\overline{\sigma})}(p)$ be respectively the number of transactions of $\mathcal{D}_1(\overline{\sigma})$ and $\mathcal{D}_{-1}(\overline{\sigma})$ in which p appears. Thus, the Rademacher complexity can be rewritten as follows:

$$R_{\mathcal{D}} = \mathbb{E}_\sigma\left[\sup_{p\in\mathbb{U}} \frac{1}{n}\sum_{i=1}^{n}\sigma_i\phi_p(t_i)\right] = \mathbb{E}_\sigma\left[\sup_{p\in\mathbb{U}} \frac{Supp_{\mathcal{D}_1(\sigma)}(p) - Supp_{\mathcal{D}_{-1}(\sigma)}(p)}{n}\right]. \tag{3.32}$$

In our approximation method we generate a single combination $\overline{\sigma}$ of the Rademacher r.v. σ, instead of generating every possible combination and then taking the expectation. Given $\overline{\sigma}$, the approximation $\tilde{R}_{\mathcal{D}}(\overline{\sigma})$ of $R_{\mathcal{D}}$ is

$$\tilde{R}_{\mathcal{D}}(\overline{\sigma}) = \sup_{p\in\mathbb{U}} \frac{Supp_{\mathcal{D}_1(\overline{\sigma})}(p) - Supp_{\mathcal{D}_{-1}(\overline{\sigma})}(p)}{n}. \tag{3.33}$$

The first step of the procedure is to mine frequent sequential patterns from $\mathcal{D}_1(\overline{\sigma})$ and $\mathcal{D}_{-1}(\overline{\sigma})$, given a frequency threshold κ. Let $FSP(\mathcal{D}_1(\overline{\sigma}), \kappa)$ and $FSP(\mathcal{D}_{-1}(\overline{\sigma}), \kappa)$ be the sets of sequential patterns with support greater or equal than κ in $\mathcal{D}_1(\overline{\sigma})$ and $\mathcal{D}_{-1}(\overline{\sigma})$, respectively. Let us define the following quantities:

$$\gamma(p) = Supp_{\mathcal{D}_1(\overline{\sigma})}(p) - Supp_{\mathcal{D}_{-1}(\overline{\sigma})}(p), \tag{3.34}$$

$$\gamma_1 = \sup\{\gamma(p) : p \in FSP(\mathcal{D}_1(\overline{\sigma}), \kappa) \cap FSP(\mathcal{D}_{-1}(\overline{\sigma}), \kappa)\}, \tag{3.35}$$

and

$$\gamma_2 = \sup\{\gamma(p) : p \in FSP(\mathcal{D}_1(\overline{\sigma}), \kappa) \setminus FSP(\mathcal{D}_{-1}(\overline{\sigma}), \kappa)\}. \tag{3.36}$$

If $\max(\gamma_1, \gamma_2)/n \geq \kappa$ then $\tilde{R}_{\mathcal{D}}(\overline{\sigma}) = \max(\gamma_1, \gamma_2)/n$, since each pattern p that is not frequent in both sub-samples has $\gamma(p)/n$ lower than κ. Instead, if $\max(\gamma_1, \gamma_2)/n < \kappa$ the entire procedure is repeated with $\kappa = \max(\gamma_1, \gamma_2)/n$. Note that, since the Rademacher complexity is a non-negative quantity, it is not necessary to look at patterns in $FSP(\mathcal{D}_{-1}(\overline{\sigma}), \kappa) \setminus FSP(\mathcal{D}_1(\overline{\sigma}), \kappa)$ since their $\gamma(p)$'s values are negative. The pseudo-code of the method for finding an approximation of $R_{\mathcal{D}}$ is presented in Algorithm 2. The extraction of frequent sequences from the two sub-samples can be done using one of the many algorithms for mining frequent sequential patterns.

Algorithm 2: RadeApprox($\mathcal{D}, \kappa$): algorithm for approximating the Rademacher complexity of sequential patterns.

```
Data: : dataset D; κ ∈ (0, 1]
Result: approximation R_D^ap to R_D
σ̄ ← combination of σ;
split D into D_1(σ̄) and D_{-1}(σ̄);
found ← false;
γ ← 0;
while !found do
    compute FSP(D_1(σ̄), κ);
    compute FSP(D_{-1}(σ̄), κ);
    if |FSP(D_1(σ̄), κ)| + |FSP(D_{-1}(σ̄), κ)| = 0 then
        κ ← κ/2;
        continue;
    compute γ_1 and γ_2;
    γ ← max(γ_1, γ_2)/|D|;
    if γ ≥ κ then found ← true;
    else κ ← γ;
return R_D^ap = γ;
```

Finally, we define *ComputeMaxDevRadeApprox* as the procedure for computing an approximation $\mu_R^{ap}/2$ of $\mu_R/2$ where, once the approximation $R_{\mathcal{D}}^{ap}$ of the Rademacher complexity $R_{\mathcal{D}}$ is computed using Algorithm 2, the ap-

proximation $\mu_R^{ap}/2$ of $\mu_R/2$ is obtained by:

$$\frac{\mu_R^{ap}}{2} = 2R_{\mathcal{D}}^{ap} + \sqrt{\frac{2\ln(2/\delta)}{|\mathcal{D}|}}. \tag{3.37}$$

3.4 Algorithms for True Frequent Sequential Pattern Mining

In this section, we describe our approach to find rigorous approximations to the true frequent sequential patterns. In particular, given a dataset $\mathcal{D}$, that is a finite bag of n i.i.d. samples from an unknown probability distribution π on $\mathbb{U}$, a minimum frequency threshold $\theta \in (0, 1]$ and a confidence parameter $\delta \in (0, 1)$, we aim to find rigorous approximations of the true frequent sequential patterns w.r.t. θ, defined in Definition 9 and Definition 10, with probability at least $1 - \delta$.

The intuition behind our approach is the following. If we know an upper bound $\mu/2$ on the maximum deviation, that is $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu/2$, we can identify a frequency threshold $\hat{\theta}$ (resp. $\tilde{\theta}$) such that the set $FSP(\mathcal{D}, \hat{\theta})$ is a FPF μ-approximation (resp. $FSP(\mathcal{D}, \tilde{\theta})$ is a FNF μ-approximation) of $TFSP(\pi, \theta)$. The upper bound on the maximum deviation can be computed, as illustrated in the previous sections, with the Rademacher complexity.

We now describe how to identify the threshold $\hat{\theta}$ that allows to obtain a FPF μ-approximation. Suppose that $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu/2$. In such a scenario, we have that every sequential pattern $p^* \notin TFSP(\pi, \theta)$, and so that has $t_\pi(p^*) < \theta$, has a frequency $f_{\mathcal{D}}(p^*) < \theta + \mu/2 = \hat{\theta}$. Hence, the only sequential patterns that can have frequency in $\mathcal{D}$ greater or equal to $\hat{\theta} = \theta + \mu/2$, are those with true frequency at least θ. The intuition is that if we find a μ such that $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu/2$, we know that all the sequences $p \in \mathbb{U}$ that are not true frequent w.r.t θ, cannot be in $FSP(\mathcal{D}, \hat{\theta})$. The following theorem formalizes the strategy to obtain a FPF μ-approximation.

Theorem 5. *Given $\delta \in (0, 1)$, such that $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_{\mathcal{D}}(p)| \leq \mu/2$ with probability at least $1 - \delta$, and given $\theta \in (0, 1]$, the set $FSP(\mathcal{D}, \hat{\theta})$, with $\hat{\theta} = \theta + \mu/2$, is a FPF μ-approximation of the set $TFSP(\pi, \theta)$ with probability at least $1 - \delta$.*

Proof. Suppose that $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_\mathcal{D}(p)| \leq \mu/2$. Thus, we have that for all the sequential patterns $p \in \mathbb{U}$, it results $f_\mathcal{D}(p) \in [t_\pi(p) - \mu/2, t_\pi(p) + \mu/2]$. This also holds for the sequential patterns in $\mathcal{G} = FSP(\mathcal{D}, \hat{\theta})$. Therefore, the set $\mathcal{G}$ satisfies Property 3 of Definition 10. Let p^* be a sequential pattern such that $t_\pi(p^*) < \theta$, that is, it is not a true frequent sequential pattern w.r.t. θ. Then, $f_\mathcal{D}(p^*) < \theta + \mu/2 = \hat{\theta}$, that is, $p^* \notin \mathcal{G}$, which allows us to conclude that $\mathcal{G}$ also has Property 1 from Definition 10. Now, let p' be a sequential pattern such that $t_\pi(p') \geq \theta + \mu$. Then, $f_\mathcal{D}(p') \geq \theta + \mu/2$, that is $p' \in \mathcal{G}$, which allows us to conclude that $\mathcal{G}$ also has Property 2 from Definition 10. Since we know that $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_\mathcal{D}(p)| \leq \mu/2$ with probability at least $1-\delta$, then the set $\mathcal{G}$ is a FPF μ-approximation of $TFSP(\pi, \theta)$ with probability at least $1 - \delta$, which concludes the proof. □

Theorem 5 shows how to compute a corrected threshold $\hat{\theta}$ such that the set $FSP(D, \hat{\theta})$ is a FPF μ-approximation of $TFSP(\pi, \theta)$, that is, $FSP(D, \hat{\theta})$ only contains sequential patterns that are in $TFSP(\pi, \theta)$. It guarantees that with high probability the set $FSP(\mathcal{D}, \hat{\theta})$ does not contain *false positives* but it has not guarantees on the number of *false negatives*, that is, sequential patterns that are in $TFSP(\pi, \theta)$ but not in $FSP(\mathcal{D}, \hat{\theta})$. On the other hand, we might be interested in finding all the true frequent sequential patterns in $TFSP(\pi, \theta)$. The following result shows how to identify a threshold $\tilde{\theta}$ such that the set $FSP(\mathcal{D}, \tilde{\theta})$ contains all the true frequent sequential patterns in $TFSP(\pi, \theta)$ with high probability, that is, $FSP(\mathcal{D}, \tilde{\theta})$ is a FNF μ-approximation of $TFSP(\pi, \theta)$.

Theorem 6. *Given $\delta \in (0, 1)$, such that $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_\mathcal{D}(p)| \leq \mu/2$ with probability at least $1 - \delta$, and given $\theta \in (0, 1]$, the set $FSP(\mathcal{D}, \tilde{\theta})$, with $\tilde{\theta} = \theta - \mu/2$, is a FNF μ-approximation of the set $TFSP(\pi, \theta)$ with probability at least $1 - \delta$.*

Proof. Suppose that $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_\mathcal{D}(p)| \leq \mu/2$. Thus, we have that for all the sequential patterns $p \in \mathbb{U}$, it results $f_\mathcal{D}(p) \in [t_\pi(p) - \mu/2, t_\pi(p) + \mu/2]$. This also holds for the sequential patterns in $\mathcal{E} = FSP(\mathcal{D}, \tilde{\theta})$. Therefore, the set $\mathcal{E}$ satisfies Property 3 of Definition 9. It also means that for all $p \in TFSP(\pi, \theta)$, $f_\mathcal{D}(p) \geq \theta - \mu/2 = \tilde{\theta}$, that is, $p \in \mathcal{E}$, which allows us to conclude that $\mathcal{E}$ also has Property 1 from Definition 9. Now, let p^* be a sequential pattern such that $t_\pi(p^*) < \theta - \mu$. Then, $f_\mathcal{D}(p^*) < \theta - \mu/2$, that is $p^* \notin \mathcal{E}$, which allows us to conclude that $\mathcal{E}$ also has Property 2 from Definition 9. Since we know that $\sup_{p\in\mathbb{U}} |t_\pi(p) - f_\mathcal{D}(p)| \leq \mu/2$ with probability at least $1-\delta$,

then the set $\mathcal{E}$ is a FNF μ-approximation of $TFSP(\pi, \theta)$ with probability at least $1 - \delta$, which concludes the proof. □

Note that while Theorem 6 provides guarantees on false negatives, it does not provide guarantees on the number of false positives in $FSP(\mathcal{D}, \tilde{\theta})$.

Algorithm 3 shows the pseudo-code of the two strategies to mine the true frequent sequential patterns. To compute an upper bound on the maximum deviation, it is possible to use the two procedures *ComputeMaxDevRadeBound* (Equation 3.31) and *ComputeMaxDevRadeApprox* (Equation 3.37) based on the Rademacher complexity. However, the same strategy applies when the upper bound on the maximum deviation is computed with other techniques, e.g., the VC-dimension [Vapnik and Chervonenkis, 1971]. The mining of $\mathcal{D}$ can be performed with any efficient algorithm for the exact mining of frequent sequential patterns.

Algorithm 3: Mining the True Frequent Sequential Patterns.

Data: Dataset $\mathcal{D}$; $\delta \in (0, 1)$; $\theta \in (0, 1]$
Result: Set $\mathcal{G}$ (resp. $\mathcal{E}$) that is a FPF μ-approximation (resp. FNF μ-approximation) to $TFSP(\pi, \theta)$ with probability $\geq 1 - \delta$.

```
1 μ/2 ← ComputeMaxDeviationBound(D, δ);
2 G ← FSP(D, θ + μ/2);     /* resp. E ← FSP(D, θ − μ/2) to obtain
    a FNF μ-approximation */
3 return G;     /* resp. E */
```

3.5 Experimental Evaluation

In this section, we report the results of our experimental evaluation on multiple datasets to assess the performance of the algorithms we proposed in this work. The goals of the evaluation are the following:

- Assess whether a classical algorithm for mining frequent sequential patterns from the datasets provides false positives or false negatives w.r.t. the set of true frequent sequential patterns;
- Assess the performance of our algorithms for mining the true frequent sequential patterns. In particular, to assess whether with probability $1 - \delta$ the set of frequent sequential patterns extracted from the dataset with the corrected threshold does not contain false positives, that is, it is a FPF μ-approximation of $TSFP(\pi, \theta)$, for the first method,

and contains all the TFSPs, that is, it is a FNF μ-approximation of $TSFP(\pi, \theta)$, for the second method. We show the results obtained using the upper bound and the approximation of the Rademacher complexity, which are both used to compute an upper bound to the maximum deviation.

Since no algorithm to mine true frequent sequential patterns have been previously proposed, we do not consider other methods in our experimental evaluation.

3.5.1 Implementation, Datasets, Parameters, and Environment

The code to compute the bound and the approximation to the Rademacher Complexity (resp. Algorithm 1 and Algorithm 2) has been developed in C++. We have performed all our experiments on the same machine with 512 GB of RAM and 2 Intel(R) Xeon(R) CPU E5-2698 v3 @ 2.3GHz. To mine sequential patterns, we used the PrefixSpan [Pei et al., 2004] implementation provided by the SPMF library [Fournier-Viger et al., 2016]. We used NLopt [Johnson, 2014] as non-linear optimization solver. Our open-source implementation and the code developed for the tests, including scripts to reproduce all results, are available online [1].

Now, we describe the datasets we used in our evaluation. All datasets are obtained starting from the following real datasets:

- BIBLE: a conversion of the Bible into sequence where each word is an item;
- BMS1: contains sequences of click-stream data from the e-commerce website Gazelle;
- BMS2: contains sequences of click-stream data from the e-commerce website Gazelle;
- KOSARAK: contains sequences of click-stream data from an Hungarian news portal;

[1] Available at `https://github.com/VandinLab/VCRadSPM`

Table 3.1: Datasets characteristics. For each dataset $\mathcal{D}$, we report the number $|\mathcal{D}|$ of transactions, the total number $|\mathcal{I}|$ of items, the average transaction item-length $||\tau||$ and the maximum transaction item-length $||\tau||$

Dataset $\mathcal{D}$	Size $\|\mathcal{D}\|$	$\|\mathcal{I}\|$	Avg. $\|\|\tau\|\|$	Max. $\|\|\tau\|\|$
BIBLE	36369	13905	21.6	100
BMS1	59601	497	2.5	267
BMS2	77512	3340	4.6	161
KOSARAK	69999	14804	8.0	796
LEVIATHAN	5835	9025	33.8	100
MSNBC	989818	17	4.8	14795

- LEVIATHAN: is a conversion of the novel Leviathan by Thomas Hobbes (1651) as a sequence dataset where each word is an item;
- MSNBC: contains sequences of click-stream data from MSNBC website and each item represents the category of a web page;

All the datasets used are publicly available online [Fournier-Viger et al., 2016]. The characteristics of such datasets are reported in Table 3.1.

To evaluate our algorithms to mine the true frequent sequential patterns, we need to know which are the sequential patterns that are frequently generated from the unknown generative process π. In particular, we need a *ground truth* of the true frequencies of the sequential patterns. We generated pseudo-artificial datasets by taking some of the datasets in Table 3.1 as ground truth for the true frequencies t_π of the sequential patterns. For each ground truth, we created four new datasets by sampling sequential transactions uniformly at random from the original dataset. All the new datasets have the same number of transactions of the respectively ground truth, that is, the respectively original dataset. We used the original datasets as ground truth and we executed our evaluation in the new (sampled) datasets. Therefore, the true frequency of a sequential pattern is its frequency in the original dataset, which is exactly the same that such pattern would have in an hypothetical infinite number of transactions generated by the unknown generative process π.

3.5.2 True Frequent Sequential Patterns Mining Results

In this section, we describe the results of our algorithms for mining the true frequent sequential patterns. In all these experiments, we fixed $\delta = 0.1$.

First of all, for each real dataset we generated 4 pseudo-artificial datasets $\mathcal{D}_i$, $i \in [1, 4]$ from the same ground truth. We mined the set $FSP(\mathcal{D}_i, \theta)$, and we compared it with the true frequent sequential patterns, that is, the set $FSP(\mathcal{D}, \theta)$, where $\mathcal{D}$ is the ground truth. Such experiments aim to verify whether the sets of the frequent sequential patterns extracted from the pseudo-artificial datasets contain false positives and miss some true frequent sequential patterns. Table 3.2 shows the fractions of times that the set $FSP(\mathcal{D}_i, \theta)$ contains false positives and misses true frequent sequential patterns from the ground truth. We ran this evaluation over the four datasets $\mathcal{D}_i$, $i \in [1, 4]$, of the same size from the same ground truth and we reported the average. For each dataset, we report the results with two frequency thresholds θ. In almost all the cases, the frequent sequential patterns mined from the pseudo-artificial datasets contain false positives and miss some true frequent sequential patterns. In particular, with lower frequency thresholds (and, therefore, a larger number of patterns), the fraction of times we find false positives and false negatives usually increases. These results emphasize that, in general, the mining of the frequent sequential patterns is not enough to learn interesting features of the underlying generative process of the data, and techniques like the ones introduced in this work are necessary.

Then, we compute the upper bounds to the maximum deviation introduced in the previous sections, since our strategy to find an approximation to the true frequent sequential patterns hinges on finding a tight upper bound to the maximum deviation. For each pseudo-artificial dataset, we computed the upper bound $\mu_R^{ub}/2$ to the maximum deviation (ComputeMaxDevRadeBound, Equation 3.31) using the upper bound to Rademacher complexity presented in Section 3.3.1, and the upper bound $\mu_R^{ap}/2$ to the maximum deviation (ComputeMaxDevRadeApprox, Equation 3.37) using the Rademacher complexity approximation presented in Section 3.3.2. Table 3.3 shows the values of the upper bound computed with both methods. The method based on the approximation of the Rademacher complexity provides, as expected, tighter upper bounds to the maximum deviation compared to the ones obtained using the method based on the upper bound to the Rademacher complexity. More precisely, for each datasets, the average values for the upper

Table 3.2: The table shows the average fraction of times that $FSP(\mathcal{D}_i, \theta)$ contains false positives and false negatives. For each dataset $\mathcal{D}$: θ is the minimum frequency threshold, |TFSP| is the number of true frequent sequential patterns in the ground truth, Times FPs (%) is the percentage of runs containing at least one false positive (FP), and Times FNs (%) is the percentage of runs containing at least one false negative (FN).

Dataset $\mathcal{D}$	θ	|TFSP|	Times FPs (%)	Times FNs (%)
BIBLE	0.1	174	50	100
	0.05	774	100	100
BMS1	0.025	13	50	0
	0.0225	17	0	25
BMS2	0.025	10	0	0
	0.0225	11	0	0
KOSARAK	0.06	23	100	0
	0.04	41	50	25
LEVIATHAN	0.15	225	75	100
	0.1	651	100	100
MSNBC	0.02	97	75	25
	0.015	143	100	50

bounds $\mu_R^{ub}/2$ and $\mu_R^{ap}/2$ are such that $\mu_R^{ub}/\mu_R^{ap} \in [1.8, 7.2]$.

In our implementation of Algorithm 1 to compute an upper bound to the empirical Rademacher complexity of sequential patterns, we compute several upper bounds associated with different integer values of $\eta \in [\overline{\eta} + \beta_1, \min(\beta_2, \hat{\eta})]$ for fixed values of β_1 and β_2, taking the minimum bound among those computed. In our experiments, we fixed $\beta_1 = 20$ and $\beta_2 = 120$. In practice, by increasing the value of η we observe a decreasing trend of the upper bound value until a minimum value is reached. Then, by increasing again the value of η the value of the upper bound increases until it converges to the one achieved with $\eta = \hat{\eta}$. In addition, for each pseudo-artificial dataset the value of η associated with the minimum value of the upper bound to the maximum deviation is always found in $[\overline{\eta} + \beta_1, \min(\beta_2, \hat{\eta})]$, with $\beta_1 = 20$, $\beta_2 = 120$.

Finally, we evaluated the performance of our two strategies to mine an approximation of the true frequent sequential patterns, the first one with guarantees on the false positives and the second one with guarantees on

Table 3.3: Comparison of the upper bound $\mu/2$ to the maximum deviation achieved by ComputeMaxDevRadeBound ($\mu_R^{ub}/2$), and ComputeMaxDevRadeApprox ($\mu_R^{ap}/2$) for each dataset. We show averages avg, maximum values max, and standard deviations std for each dataset and method over the 4 pseudo-artificial datasets.

Dataset	$\mu_R^{ub}/2$			$\mu_R^{ap}/2$		
	avg	max	std ($\times 10^{-3}$)	avg	max	std ($\times 10^{-3}$)
BIBLE	0.0747	0.0748	0.1	0.0207	0.0223	1.5
BMS1	0.0287	0.0294	0.6	0.0136	0.0153	1.0
BMS2	0.0202	0.0207	0.5	0.0107	0.0115	0.5
KOSARAK	0.0957	0.0972	1.5	0.0145	0.0164	1.5
LEVIATHAN	0.1878	0.1904	1.6	0.0569	0.0636	5.5
MSNBC	0.0252	0.0257	0.9	0.0035	0.0041	0.4

the false negatives, using the tightest upper bounds $\mu_R^{ap}/2$ (from Table 3.3) computed with an approximation of the empirical Rademacher complexity. From each pseudo-artificial dataset, we mined the frequent sequential patterns using $\hat{\theta}$, for the first strategy, and $\tilde{\theta}$, for the second one, respectively computed using Theorem 5 and Theorem 6, and we compared the sequential patterns extracted with the true frequent sequential patterns from the ground truth. Table 3.4 shows the results for both strategies with guarantees on the false positives and false negatives. Using $\mu_R^{ap}/2$ to compute the corrected frequency thresholds $\hat{\theta}$ and $\tilde{\theta}$, our algorithms provide outputs that satisfy, respectively, the guarantees of Theorem 5 and Theorem 6 in all the runs. This means that, using $\hat{\theta}$, the sequential patterns in output are always true frequent sequential patterns (i.e., there are no false positives), and, $\tilde{\theta}$, the output contains all the true frequent sequential patterns (i.e., there are no false negatives). Thus, our algorithms perform even better than guaranteed by their theoretical analysis, which state that there are no false positives or false negatives with probability at least $1 - \delta$. Then, we computed the average fraction $|FSP(\mathcal{D}_i, \hat{\theta})|/|TFSP|$ of true frequent sequential patterns reported in the FPF μ-approximations, that is, the ratio of true frequent sequential patterns captured by our algorithm. We also computed the average fraction $|TFSP|/|FSP(\mathcal{D}_i, \tilde{\theta})|$ of sequential patterns reported in the

FNF μ-approximations that are true frequent sequential patterns, that is, the ratio of reported sequential patterns that are not false positives. Note that the best case for such ratios is to be as close to 1 as possible. However, our algorithms are not designed to provide theoretical guarantees in such terms and, in fact, such ratios are not very high for some datasets, e.g., BMS1, BMS2, and LEVIATHAN (see Table 3.4). Note that LEVIATHAN is a small dataset (only 5835 transactions, see Table 3.1), while BMS1 and BMS2 contain very short transactions (average transaction item-length of 2.5 and 4.6 respectively, see Table 3.1). Instead, the ratios become higher if computed on the other larger (in terms of number of transactions or average transaction item-length) datasets, i.e. BIBLE, KOSARAK, and MSNBC. This is consistent with the fact that our strategy to bound the maximum deviation and then to approximate true frequent sequential patterns strongly depends on the size of the datasets.

To conclude, our experimental evaluation shows that our algorithms to mine true frequent sequential patterns with rigorous guarantees are valid strategies to obtain high-quality approximations, both without false positives or false negatives.

Table 3.4: Results of our algorithms to mine true frequent sequential patterns, with guarantees on false negatives or false positives. The table shows: the dataset $\mathcal{D}$; the minimum frequency threshold θ; the number |TFSP| of true frequent sequential patterns in the ground truth; FNF μ-approx. (%): percentage of FNF μ-approximation obtained in all the runs; $|\text{TFSP}|/|\text{FSP}(\mathcal{D}_i, \tilde{\theta})|$: average fraction of sequential patterns reported in the FNF μ-approximations that are true frequent sequential patterns; FPF μ-approx. (%): percentage of FPF μ-approximation obtained in all the runs; $|\text{FSP}(\mathcal{D}_i, \hat{\theta})|/|\text{TFSP}|$: average fraction of true frequent sequential patterns reported in the FPF μ-approximations.

Dataset $\mathcal{D}$	θ	|TFSP|	FNF μ-approx. (%)	$\frac{\|\text{TFSP}\|}{\|\text{FSP}(\mathcal{D}_i, \tilde{\theta})\|}$	FPF μ-approx. (%)	$\frac{\|\text{FSP}(\mathcal{D}_i, \hat{\theta})\|}{\|\text{TFSP}\|}$
BIBLE	0.1	174	100	0.63	100	0.68
	0.05	774	100	0.33	100	0.47
BMS1	0.025	13	100	0.21	100	0.48
	0.0025	17	100	0.19	100	0.43
BMS2	0.025	10	100	0.32	100	0.20
	0.0025	11	100	0.19	100	0.18
KOSARAK	0.06	23	100	0.64	100	0.73
	0.04	41	100	0.49	100	0.74
LEVIATHAN	0.15	225	100	0.30	100	0.41
	0.1	651	100	0.13	100	0.30
MSNBC	0.02	97	100	0.77	100	0.77
	0.015	143	100	0.65	100	0.76

Chapter 4

SPRISS: Approximating Frequent k-mers by Sampling Reads

4.1 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 many applications, e.g., the comparison of datasets and reads classification in metagenomics [Wood and Salzberg, 2014], error correction for genome assembly [Kelley et al., 2010, Salmela et al., 2016], and several others (see Section 5.1.2).

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 4.1.2). 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[1]).

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 [Ounit et al., 2015, Liu et al., 2017], 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 4.1.2).

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.

4.1.1 Our Contributions

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

[1]Using $k = 31$, 32 workers, and maximum RAM of 350 GB. See Supplemental Table 5.1 for the size of `Mo17`.

k-mer$\underline{\text{S}}$ [2]. 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.

4.1.2 Related Works

The problem of exactly counting k-mers in datasets has been extensively studied, with several methods proposed for its solution [Kurtz et al., 2008, Marçais and Kingsford, 2011, Melsted and Pritchard, 2011, Rizk et al., 2013, Audano and Vannberg, 2014, Roy et al., 2014, Kokot et al., 2017, Pandey et al., 2017]. 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, [Melsted and Halldórsson, 2014, Sivadasan et al., 2016, Mohamadi et al., 2017, Chikhi and Medvedev, 2013, Zhang et al., 2014, Pandey et al., 2017]). 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 [Pandey et al., 2018, Harris and Medvedev, 2020, Marchet et al., 2020b, Marchet et al., 2020a], represent [Chikhi et al., 2014, Dadi et al., 2018, Almodaresi et al., 2018, Guo et al., 2019, Marchet et al., 2019b, Holley and Melsted, 2020, Rahman and Medvedev, 2020], query [Solomon and Kingsford, 2016, Solomon and Kings-

[2] `https://vec.wikipedia.org/wiki/Spriss`

ford, 2018, Yu et al., 2018, Sun et al., 2018, Bradley et al., 2019, Marchet et al., 2019a], and store [Hosseini et al., 2016, Numanagić et al., 2016, Hernaez et al., 2019, Rahman et al., 2020] 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, Wedemeyer et al., 2017, Coleman et al., 2019]. 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 maximise 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.

4.1.3 Organization of the Chapter

The rest of the Chapter is organized as follows. In Section 4.2 we introduce some preliminary concepts used throughout this work. A first, simple, warm-up approach to approximate frequent k-mers is presented in Section 4.3. Next, a first improvement to the warm-up approach is presented in Section 4.4. Then, in Section 4.5 we describe SPRISS, our rigorous and efficient algorithm to estimate frequent k-mers. Finally, in Section 4.6 we present our experimental evaluation where we assess the quality of the approximation of the frequent k-mers provided by SPRISS, and compare SPRISS with SAKEIMA.

4.2 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], which represents Definition 3 where the patterns are k-mers.

Definition 11. *Given a dataset $\mathcal{D}$, a positive integer k, a frequency threshold $\theta \in (0, 1]$, and an accuracy parameter $\varepsilon \in (0, \theta)$, a FNF* 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 FNF 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 12. *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 nontrivial, due to dependencies among k-mers appearing in the same read. As explained in Section 4.1.1, our approach has several advantages, including the fact 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. Additionally, note that SPRISS can approximate both canonical or non-canonical k-mers.

In the next sections, we develop and analyze algorithms to approximate $FK(\mathcal{D}, k, \theta)$ by read sampling, starting from a straightforward, but inefficient, approach (Section 4.3), then showing how pseudodimension can be used to improve the sample size required by such approach (Section 4.4), and culminating in our algorithm SPRISS, the first efficient algorithm to approximate frequent k-mers by read sampling (Section 4.5).

4.3 Warm-Up: A Simple Algorithm for Approximating Frequent k-mers by Sampling Reads

A first, simple approach to approximate the set $FK(\mathcal{D}, k, \theta)$ of frequent k-mers consists in taking a sample S of m reads, with m large enough, and report in output the set $FK(S, k, \theta - \varepsilon/2)$ of k-mers that appear with frequency at least $\theta - \varepsilon/2$ in the sample S. This strategy is motivated by Proposition 4, obtained by combining Hoeffding's inequality [Mitzenmacher and Upfal, 2017] and a union bound, which provides an upper bound to the number m of reads required to have guarantees on the quality of the approximation. Before stating and proving Proposition 4, we need to introduce and prove some preliminary results.

Proposition 1. *The expectation $\mathbb{E}[t_{S,k}]$ of the size of the multiset of k-mers that appear in S is $mg_{\mathcal{D},k}$.*

Proof. Let $X(r_i) = n_i - k + 1$ be the number of starting positions for k-mers in read r_i sampled uniformly at random form $\mathcal{D}$, $i \in \{1, \ldots, n\}$. $\mathbb{E}[X(r_i)] = \sum_{r_i \in \mathcal{D}} \frac{1}{n}(n_i - k + 1) = g_{\mathcal{D},k}$. Combining this with the linearity of the expectation, we have:

$$\begin{aligned}\mathbb{E}[t_{S,k}] &= \mathbb{E}\left[\sum_{r_i \in S}(n_i - k + 1)\right] = \sum_{r_i \in S}\mathbb{E}[n_i - k + 1] \\ &= m\mathbb{E}[X(r_i)] = mg_{\mathcal{D},k}.\end{aligned}$$

□

Given a k-mer K, its support $o_S(K)$ in S is $o_S(K) = \sum_{r_i \in S}\sum_{j=0}^{n_i-k}\phi_{r_i,K}(j)$. We define the frequency of K in S as $f_S(K) =$

$o_S(K)/(mg_{\mathcal{D},k})$, that is the ratio between the support of K and the expectation $\mathbb{E}[t_{S,k}] = mg_{\mathcal{D},k}$ of the size of the multiset of k-mers that appear in S. This definition of $f_S(K)$ gives us an unbiased estimator for $f_{\mathcal{D}}(K)$.

Proposition 2. *The frequency $f_S(K) = o_S(K)/(mg_{\mathcal{D},k})$ is an unbiased estimator for $f_{\mathcal{D}}(K) = o_{\mathcal{D}}(K)/t_{\mathcal{D},k}$.*

Proof. Let $X_{r_i}(K) = \sum_{j=0}^{n_i-k} \phi_{r_i,K}(j)$ be the number of distinct positions where k-mer K appears in read r_i sampled uniformly at random form $\mathcal{D}$, $i \in \{1, \ldots, n\}$. $E[X_{r_i}(K)] = \sum_{r_i \in \mathcal{D}} \left(\frac{1}{n} \sum_{j=0}^{n_i-k} \phi_{r_i,K}(j) \right) = o_{\mathcal{D}}(K)/n$. Combining this with the linearity of the expectation, we have:

$$\mathbb{E}[f_S(K)] = \frac{E[o_S(K)]}{mg_{\mathcal{D},k}} = \frac{\mathbb{E}[\sum_{r_i \in S} \sum_{j=0}^{n_i-k} \phi_{r_i,K}(j)]}{mg_{\mathcal{D},k}} =$$

$$= \frac{\mathbb{E}[X_{r_i}(K)]}{g_{\mathcal{D},k}} = \frac{o_{\mathcal{D}}(K)}{ng_{\mathcal{D},k}} = \frac{o_{\mathcal{D}}(K)}{t_{\mathcal{D},k}} = f_{\mathcal{D}}(K).$$

□

By using the sampling framework based on reads and Hoeffding's inequality [Mitzenmacher and Upfal, 2017], we prove the following bound on the probability that $f_S(K)$ is not within $\varepsilon/2$ from $f_{\mathcal{D}}(K)$, for an arbitrary k-mer K.

Proposition 3. *Consider a sample S of m reads from $\mathcal{D}$. Let $g_{\max,\mathcal{D},k} = \max_{r_i \in \mathcal{D}}(n_i - k + 1)$. Let $K \in \Sigma^k$ be an arbitrary k-mer. For a fixed accuracy parameter $\varepsilon \in (0, 1)$ we have:*

$$\Pr\left(|f_S(K) - f_{\mathcal{D}}(K)| \geq \frac{\varepsilon}{2} \right) \leq 2 \exp\left(-\frac{1}{2} m\varepsilon^2 \left(\frac{g_{\mathcal{D},k}}{g_{\max,\mathcal{D},k}} \right)^2 \right).$$

Proof. The frequency $f_S(K) = o_S(K)/(mg_{\mathcal{D},k})$ of K in S can be rewritten as:

$$f_S(K) = \frac{\sum_{r_i \in S} \sum_{j=0}^{n_i-k} \phi_{r_i,K}(j)}{mg_{\mathcal{D},k}}$$

$$= \sum_{r_i \in S} \sum_{j=0}^{n_i-k} \frac{\phi_{r_i,K}(j)}{mg_{\mathcal{D},k}} = \sum_{r_i \in S} \hat{\phi}_K(r_i),$$

where the random variable (r.v.) $\hat{\phi}_K(r_i) = \sum_{j=0}^{n_i-k} \frac{\phi_{r_i,K}(j)}{mg_{\mathcal{D},k}}$ is the number of times K appears in read r_i divided by $mg_{\mathcal{D},k}$. Thus, $f_S(K)$ can be rewritten as a sum of m independent r.v. that take values in $[0, \frac{g_{\max,\mathcal{D},k}}{mg_{\mathcal{D},k}}]$. Combining this fact with Proposition 2, and by applying Hoeffding's inequality [Mitzenmacher and Upfal, 2017] we have:

$$\begin{aligned}\Pr(|f_S(K) - f_{\mathcal{D}}(K)| \geq \frac{\varepsilon}{2}) &\leq 2\exp\left(\frac{-2(\varepsilon/2)^2}{m\left(\frac{g_{\max,\mathcal{D},k}}{mg_{\mathcal{D},k}}\right)^2}\right) \\ &= 2\exp\left(-\frac{1}{2}m\varepsilon^2\left(\frac{g_{\mathcal{D},k}}{g_{\max,\mathcal{D},k}}\right)^2\right).\end{aligned}$$

□

Since the maximum number of k-mers is σ^k, by combining the result above with the union bound we have the following result.

Proposition 4. *Consider a sample S of m 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*

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

then, with probability $\geq 1-\delta$, $FK(S,k,\theta-\varepsilon/2)$ is a FNF ε-approximation of $FK(\mathcal{D},k,\theta)$.

Proof. Let E_K be the event "$|f_S(K) - f_{\mathcal{D}}(K)| \leq \frac{\varepsilon}{2}$" for a k-mer K. By the choice of m and Proposition 3 we have that the probability of the complementary event $\overline{E}_K$ of E_K is

$$\begin{aligned}\Pr(\overline{E}_K) &= \Pr\left(|f_S(K) - f_{\mathcal{D}}(K)| \geq \frac{\varepsilon}{2}\right) \\ &= 2\exp\left(-\frac{1}{2}m\varepsilon^2\left(\frac{g_{\mathcal{D},k}}{g_{\max,\mathcal{D},k}}\right)^2\right) \leq \frac{\delta}{\sigma^k}.\end{aligned}$$

Now, by applying the union bound, the probability that for at least one k-mer K of Σ^k the event $\overline{E}_K$ holds is bounded by $\sum_{K\in\Sigma^k}\Pr(\overline{E}_K) \leq \delta$. Thus, the probability that events E_K simultaneously hold for all k-mers K in Σ^k is at least $1-\delta$.

Now we prove that, with probability at least $1-\delta$, $FK(S,k,\theta-\varepsilon/2)$ is a FNF ε-approximation of $FK(\mathcal{D},k,\theta)$, when, with probability at least $1-\delta$, "$|f_S(K)-f_{\mathcal{D}}(K)| \leq \frac{\varepsilon}{2}$" for all k-mers K. Note that the third property of Definition 11 is already satisfied. Let K be a k-mer of $FK(\mathcal{D},k,\theta)$, that is $f_{\mathcal{D}}(K) \geq \theta$. Given that $f_S(K) \geq f_{\mathcal{D}}(K)-\varepsilon/2$, we have $f_S(K) \geq \theta-\varepsilon/2$ and the first property of Definition 11 holds. Combining $f_{\mathcal{D}}(K) \geq f_S(K)-\varepsilon/2$ and $f_S(K) \geq \theta-\varepsilon/2$, we have $f_{\mathcal{D}}(K) \geq \theta-\varepsilon$ and the second property of Definition 11 holds. □

Proposition 4 gives us the following simple procedure for approximating the set of frequent k-mers with guarantees on the quality of the solution: draw a sample S of $m \geq \frac{2}{\varepsilon^2}\left(\frac{g_{\max,\mathcal{D},k}}{g_{\mathcal{D},k}}\right)^2\left(\ln\left(2\sigma^k\right)+\ln\left(\frac{1}{\delta}\right)\right)$ reads from $\mathcal{D}$, and output the set $FK(S,k,\theta-\varepsilon/2)$ which is a FNF ε-approximation of $FK(\mathcal{D},k,\theta)$ with probability at least $1-\delta$.

While Proposition 4 provides a first bound to the number m of reads required to obtain a rigorous approximation of the frequent k-mers, it typically results in a sample size m larger than $|\mathcal{D}|$, making the sampling approach useless. This is due to the need for ε to be small in order to obtain meaningful approximations, since the frequencies of k-mers we are estimating are small (see Section 4.6.2). Thus, in the next sections we propose advanced methods to reduce the sample size m.

4.4 A First Improvement: A Pseudodimension-based Algorithm for k-mers Approximation by Sampling Reads

In this section we introduce the notion of pseudodimension and we use it to improve the bound on the sample size m of Proposition 4.

Let $\mathcal{F}$ be a class of real-valued functions from a domain X to $[a,b] \subset \mathbb{R}$. Consider, for each $f \in \mathcal{F}$, the subset of $X' = X \times [a,b]$ defined as $R_f = \{(x,t) : t \leq f(x)\}$, and call it *range*. Let $\mathcal{F}^+ = \{R_f, f \in \mathcal{F}\}$ be a *range set* on X', and its corresponding *range space* Q' be $Q' = (X', \mathcal{F}^+)$. We say that a subset $D \subset X'$ is *shattered* by $\mathcal{F}^+$ if the size of the *projection set* $proj_{\mathcal{F}^+}(D) = \{r \cap D : r \in \mathcal{F}^+\}$ is equal to $2^{|D|}$. The *VC dimension* $VC(Q')$ of Q' is the maximum size of a subset of X' shattered by $\mathcal{F}^+$. The *pseudodimension* $PD(X,\mathcal{F})$ is then defined as the VC dimension of Q': $PD(X,\mathcal{F}) = VC(Q')$.

Let π be the uniform distribution on X, and let S be a sample of X of size $|S| = m$, with every element of S sampled independently and uniformly at random from X. We define, $\forall f \in \mathcal{F}$, $f_S = \frac{1}{m}\sum_{x \in S} f(x)$ and $f_X = \mathbb{E}_{x\sim\pi}[f(x)]$. Note that $\mathbb{E}[f_S] = f_X$. The following result relates the accuracy and confidence parameters ε,δ and the pseudodimension with the probability that the expected values of the functions in $\mathcal{F}$ are well approximated by their averages computed from a finite random sample.

Proposition 5. *([Talagrand, 1994, Long, 1999])*

Let X be a domain and $\mathcal{F}$ be a class of real-valued functions from X to $[a, b]$. Let $PD(X, \mathcal{F}) = VC(Q') \leq v$. There exist an absolute positive constant c such that, for fixed $\varepsilon, \delta \in (0, 1)$, if S is a random sample of m samples drawn independently and uniformly at random from X with $m \geq \frac{c(b-a)^2}{\varepsilon^2}\left(v + \ln\left(\frac{1}{\delta}\right)\right)$ then, with probability $\geq 1 - \delta$, it holds simultaneously $\forall f \in \mathcal{F}$ that $|f_S - f_X| \leq \varepsilon$.

The universal constant c has been experimentally estimated to be at most 0.5 [Löffler and Phillips, 2009].

We now define the range space associated to k-mers, derive an upper bound to its pseudodimension, and use the result above to derive an improved bound on the number m of reads to be sampled in order to obtain a rigorous approximation of the frequent k-mers. Let k be a positive integer and $\mathcal{D}$ be a bag of n reads. Define the domain X as the set of integers $\{1, \ldots, n\}$, where every $i \in X$ corresponds to the i-th read of $\mathcal{D}$. Then define the family of real-valued functions $\mathcal{F} = \{f_K, \forall K \in \Sigma^k\}$ where, for every $i \in X$ and for every $f_K \in \mathcal{F}$, the function $f_K(i)$ is the number of distinct positions in read r_i where k-mer K appears divided by the average size of the multiset of k-mers that appear in a read of $\mathcal{D}$: $f_K(i) = \sum_{j=0}^{n_i-k} \frac{\phi_{r_i,K}(j)}{g_{\mathcal{D},k}}$. Therefore $f_K(i) \in [0, \frac{g_{\max,\mathcal{D},k}}{g_{\mathcal{D},k}}]$. For each $f_K \in \mathcal{F}$, the subset of $X' = X \times [0, \frac{g_{\max,\mathcal{D},k}}{g_{\mathcal{D},k}}]$ defined as $R_{f_K} = \{(i, t) : t \leq f_K(i)\}$ is the associated range. Let $\mathcal{F}^+ = \{R_{f_K}, f_K \in \mathcal{F}\}$ be the range set on X', and its corresponding range space Q' be $Q' = (X', \mathcal{F}^+)$.

A trivial upper bound to $PD(X, \mathcal{F})$ is given by $PD(X, \mathcal{F}) \leq \lfloor \log_2 |\mathcal{F}| \rfloor = \lfloor \log_2 \sigma^k \rfloor$. Before proving a tighter bound to $PD(X, \mathcal{F})$, we first state a technical Lemma (Lemma 3.8 from [Riondato and Upfal, 2018]).

Lemma 5. *Let $B \subseteq X'$ be a set that is shattered by $\mathcal{F}^+$. Then B does not contain any element in the form $(i, 0)$, for any $i \in X$.*

The following result provides an improved upper bound to $PD(X, \mathcal{F})$.

Proposition 6. *Let $\mathcal{D}$ be a bag of n reads, k a positive integer, $X = \{1, \ldots, n\}$ be the domain, and let the family $\mathcal{F}$ of real-valued functions be $\mathcal{F} = \{f_K, \forall K \in \Sigma^k\}$. Then the pseudodimension $PD(X, \mathcal{F})$ satisfies $PD(X, \mathcal{F}) \leq \lfloor \log_2(g_{\max,\mathcal{D},k}) \rfloor + 1$.*

Proof. From the definition of pseudodimension we have $PD(X, \mathcal{F}) = VC(Q')$, therefore showing $VC(Q') = v \leq \lfloor \log_2(g_{\max,\mathcal{D},k}) \rfloor + 1$ is sufficient for the proof. An immediate consequence of Lemma 5 is that for all elements (i, t) of any set B that is shattered by $\mathcal{F}^+$ it holds $t \geq 1/g_{\mathcal{D},k}$. Now we denote an integer v and suppose that $VC(Q') = v$. Thus, there must exist a set $B \subseteq X'$ with $|B| = v$ which needs to be shattered by $\mathcal{F}^+$. This means that 2^v subsets of B must be in projection of $\mathcal{F}^+$ on B. If this is true, then every element of B needs to belong to exactly 2^{v-1} such sets. This means that for a given (i, t) of B, all the projections of 2^{v-1} elements of $\mathcal{F}^+$ contain (i, t). Since $t \geq 1/g_{\mathcal{D},k}$, there need to exist 2^{v-1} distinct k-mers appearing at least once in the read r_i. More formally, it needs to hold $n_i - k + 1 \geq 2^{v-1}$, that implies $v \leq \lfloor \log_2(n_i - k + 1) \rfloor + 1$, $\forall (i, t) \in B$. Since $n_i - k + 1 \leq g_{\max,\mathcal{D},k}$ for each $(i, t) \in B$, then $v \leq \lfloor \log_2(g_{\max,\mathcal{D},k}) \rfloor + 1$, and the thesis holds. □

Combining Proposition 5 and Proposition 6, we derive the following.

Proposition 7. *Let S be a sample of m reads from $\mathcal{D}$. For fixed threshold $\theta \in (0, 1]$, error parameter $\varepsilon \in (0, \theta)$, and confidence parameter $\delta \in (0, 1)$, if $m \geq \frac{2}{\varepsilon^2} \left(\frac{g_{\max,\mathcal{D},k}}{g_{\mathcal{D},k}} \right)^2 \left(\lfloor \log_2 \min(2g_{\max,\mathcal{D},k}, \sigma^k) \rfloor + \ln\left(\frac{1}{\delta}\right) \right)$ then, with probability $\geq 1 - \delta$, $FK(S, k, \theta - \varepsilon/2)$ is a FNF ε-approximation of $FK(\mathcal{D}, k, \theta)$.*

Proof. Let consider the domain X and the class of real-valued functions $\mathcal{F}$ previously defined. For a given function $f \in \mathcal{F}$ (so for a given k-mer K), we have for $f_X = \mathbb{E}_{x \sim \pi}[f(x)]$ that

$$f_X = \mathbb{E}_{r_i \sim \mathcal{D}}[f_K(i)] = \mathbb{E}_{r_i \sim \mathcal{D}} \left[\sum_{j=0}^{n_i - k} \frac{\phi_{r_i,K}(j)}{g_{\mathcal{D},k}} \right]$$

$$= \frac{1}{g_{\mathcal{D},k}} \sum_{r_i \in \mathcal{D}} \frac{1}{n} \sum_{j=0}^{n_i - k} \phi_{r_i,K}(j) = \frac{o_D(K)}{n g_{\mathcal{D},k}} = f_{\mathcal{D}}(K),$$

and for $f_S = \frac{1}{m}\sum_{x\in S} f(x)$ that

$$f_S = \frac{1}{m}\sum_{r_i\in S} f_K(i) = \frac{1}{m}\sum_{r_i\in S}\sum_{j=0}^{n_i-k}\frac{\phi_{r_i,K}(j)}{g_{\mathcal{D},k}} = \frac{o_S(K)}{m g_{\mathcal{D},k}} = f_S(K).$$

Combining the trivial bound $PD(X,\mathcal{F}) \leq \lfloor \log_2 \sigma^k \rfloor$ with Propositions 5 and 6 we have that, with probability at least $1-\delta$, $|f_S(K) - f_{\mathcal{D}}(K)| \leq \varepsilon/2$ simultaneously holds for every k-mer K.

Now, as for Proposition 4, we prove that, with probability at least $1-\delta$, $FK(S,k,\theta-\varepsilon/2)$ is a FNF ε-approximation of $FK(\mathcal{D},k,\theta)$, when, with probability at least $1-\delta$, "$|f_S(K) - f_{\mathcal{D}}(K)| \leq \frac{\varepsilon}{2}$" for all k-mers K. Note that the third property of Definition 11 is already satisfied. Let K be a k-mer of $FK(\mathcal{D},k,\theta)$, that is $f_{\mathcal{D}}(K) \geq \theta$. Given that $f_S(K) \geq f_{\mathcal{D}}(K) - \varepsilon/2$, we have $f_S(K) \geq \theta - \varepsilon/2$ and the first property of Definition 11 holds. Combining $f_{\mathcal{D}}(K) \geq f_S(K) - \varepsilon/2$ and $f_S(K) \geq \theta - \varepsilon/2$, we have $f_{\mathcal{D}}(K) \geq \theta - \varepsilon$ and the second property of Definition 11 holds. □

This bound significantly improves on the one in Proposition 4, since the factor $\ln(2\sigma^k)$ is reduced to $\lfloor \log_2 \min(2g_{\max,\mathcal{D},k}, \sigma^k) \rfloor$. However, even the bound from Proposition 7 results in a sample size m larger than $|\mathcal{D}|$ (see Section 4.6.2). In the next section we propose a method to further reduce the sample size m, which results in a practical sampling approach.

4.5 SPRISS: Sampling Reads Algorithm to Estimate Frequent k-mers

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 $S_\ell = \{I_{\ell,1}, \ldots, I_{\ell,m}\}$. Now we need to formulate a new definition of range space $Q' = (X', \mathcal{F}^+)$ associated to k-mers, which requires to define a new domain X and a new class of real-valued functions $\mathcal{F}$. 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 $f_{K,\ell}(I_\ell) \in \{0, \frac{1}{\ell g_{\mathcal{D},k}}\}\ \forall f_{K,\ell}$ 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. For each $f_{K,\ell} \in \mathcal{F}$, the subset of $X' = X \times \{0, \frac{1}{\ell g_{\mathcal{D},k}}\}$ defined as $R_{f_{K,\ell}} = \{(I_\ell, t) : t \leq f_{K,\ell}(I_\ell)\}$ is the associated range. Let $\mathcal{F}^+ = \{R_{f_{K,\ell}}, f_{K,\ell} \in \mathcal{F}\}$ be the range set on X', and its corresponding range space Q' be $Q' = (X', \mathcal{F}^+)$. We now prove an upper bound to the pseudodimension $PD(X, \mathcal{F})$.

Proposition 8. *Let $\mathcal{D}$ be a bag of n reads, and k a positive integer. Let the domain X be the set of bags of ℓ indexes of reads of $\mathcal{D}$, and $\mathcal{F} = \{f_{K,\ell}, \forall K \in \Sigma^k\}$ be the family of real-valued functions. Then the pseudodimension $PD(X, \mathcal{F})$ satisfies*

$$PD(X, \mathcal{F}) \leq \lfloor log_2(\ell g_{\max,\mathcal{D},k}) \rfloor + 1. \tag{4.1}$$

Proof. From the definition of pseudodimension we have $PD(X, \mathcal{F}) = VC(Q')$, therefore showing $VC(Q') = v \leq \lfloor \log_2(\ell g_{\max,\mathcal{D},k}) \rfloor + 1$ is sufficient for the proof. Since Lemma 5 is also valid for the new definition of the range space $Q' = (X', \mathcal{F}^+)$, an immediate consequence is that for all elements (i, t) of any set B that is shattered by $\mathcal{F}^+$ it holds $t \geq 1/(\ell g_{\mathcal{D},k})$. Now we denote an integer v and suppose that $VC(Q') = v$. Thus, there must exist a set $B \subseteq X'$ with $|B| = v$ which needs to be shattered by $\mathcal{F}^+$. This means that 2^v subsets of B must be in projection of $\mathcal{F}^+$ on B. If this is true, then every element of B needs to belong to exactly 2^{v-1} such sets. This means that for a given (I_ℓ, t) of B, all the projections of 2^{v-1} elements of $\mathcal{F}^+$ contain (I_ℓ, t). Since $t \geq 1/(\ell g_{\mathcal{D},k})$, there need to exist 2^{v-1} distinct k-mers appearing at least once in the bag of ℓ reads associated with I_ℓ. More formally, it needs to hold $\sum_{i \in I_\ell}(n_i - k + 1) \geq 2^{v-1}$, that implies $v \leq \lfloor \log_2 \sum_{i \in I_\ell}(n_i - k + 1) \rfloor + 1$, $\forall (I_\ell, t) \in B$. Since $n_i - k + 1 \leq g_{\max,\mathcal{D},k}$ for each $(I_\ell, t) \in B$ and $i \in I_\ell$, then $v \leq \lfloor \log_2(\ell g_{\max,\mathcal{D},k}) \rfloor + 1$, and the thesis holds. □

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. The technical motivation to use the biased frequency $\hat{f}_{S_\ell}(K)$ can be found in the rest of this section.

Before showing an improved bound on the sample size, we need some additional but necessary results. In order to find a relation between $\mathbb{E}[\hat{f}_{S_\ell}(K)]$ and $f_{\mathcal{D}}(K)$, we need the following proposition.

Proposition 9. *Let* $\tilde{f}_{\mathcal{D}}(K) = \sum_{r_i \in \mathcal{D}} \mathbb{1}(K \in r_i)/n$ *and* $f_{\mathcal{D}}(K) = o_{\mathcal{D}}(K)/t_{\mathcal{D},k}$. *It holds that:*

$$\frac{g_{\mathcal{D},k}}{g_{\max,\mathcal{D},k}} f_{\mathcal{D}}(K) \leq \tilde{f}_{\mathcal{D}}(K) \leq g_{\mathcal{D},k} f_{\mathcal{D}}(K).$$

Proof. Let us rewrite $\tilde{f}_{\mathcal{D}}(K)$:

$$\tilde{f}_{\mathcal{D}}(K) = \sum_{r_i \in \mathcal{D}} \frac{\mathbb{1}(K \in r_i)}{n}.$$

Since $\mathbb{1}(K \in r_i) \leq o_{r_i}(K)$ for every $i \in \{1, \ldots, n\}$, we have

$$\tilde{f}_{\mathcal{D}}(K) \leq \sum_{r_i \in \mathcal{D}} \frac{o_{r_i}(K)}{n} = g_{\mathcal{D},k} \sum_{r_i \in \mathcal{D}} \frac{o_{r_i}(K)}{g_{\mathcal{D},k} n} = g_{\mathcal{D},k} f_{\mathcal{D}}(K).$$

Then, since $\mathbb{1}(K \in r_i) \geq o_{r_i}(K)/g_{\max,\mathcal{D},k}$ for every $i \in \{1, \ldots, n\}$, we have

$$\tilde{f}_{\mathcal{D}}(K) \geq \sum_{r_i \in \mathcal{D}} \frac{o_{r_i}(K)}{g_{\max,\mathcal{D},k} n} = \frac{g_{\mathcal{D},k}}{g_{\max,\mathcal{D},k}} \sum_{r_i \in \mathcal{D}} \frac{o_{r_i}(K)}{g_{\mathcal{D},k} n} = \frac{g_{\mathcal{D},k}}{g_{\max,\mathcal{D},k}} f_{\mathcal{D}}(K).$$

□

Now we show a relation between $\mathbb{E}[\hat{f}_{S_\ell}(K)]$ and $f_{\mathcal{D}}(K)$.

Proposition 10. *Let* $\tilde{f}_{\mathcal{D}}(K) = \sum_{r_i \in \mathcal{D}} \mathbb{1}(K \in r_i)/n$ *and* $f_{\mathcal{D}}(K) = o_{\mathcal{D}}(K)/t_{\mathcal{D},k}$. *Let* S_ℓ *be a bag of* m *bags of* ℓ *reads drawn from* $\mathcal{D}$. *Then:*

$$\mathbb{E}[\hat{f}_{S_\ell}(K)] \geq \frac{1}{\ell g_{\mathcal{D},k}} \left(1 - \left(1 - \frac{g_{\mathcal{D},k}}{g_{\max,\mathcal{D},k}} f_{\mathcal{D}}(K)\right)^{\ell}\right).$$

Proof. Let us rewrite $\mathbb{E}[\hat{f}_{S_\ell}(K)]$:

$$\mathbb{E}[\hat{f}_{S_\ell}(K)] = \frac{1}{\ell g_{\mathcal{D},k}} \mathbb{E}[\min(1, o_{I_\ell}(K))] = \frac{1}{\ell g_{\mathcal{D},k}} \Pr(o_{I_\ell}(K) > 0).$$

Then, we have

$$\mathbb{E}[\hat{f}_{S_\ell}(K)] = \frac{1}{\ell g_{\mathcal{D},k}} \Pr(o_{I_\ell}(K) > 0) = \frac{1}{\ell g_{\mathcal{D},k}}(1 - \Pr(o_{I_\ell}(K) = 0))$$
$$= \frac{1}{\ell g_{\mathcal{D},k}}(1 - \prod_{i \in I_\ell} \Pr(o_{r_i}(K) = 0)) = \frac{1}{\ell g_{\mathcal{D},k}}(1 - (1 - \tilde{f}_{\mathcal{D}}(K))^\ell),$$

and since $\tilde{f}_{\mathcal{D}}(K) \geq \frac{g_{\mathcal{D},k}}{g_{\max,\mathcal{D},k}} f_{\mathcal{D}}(K)$ by Proposition 9, the thesis holds. □

Let θ be a minimum frequency threshold. Using the previous proposition, if

$$f_{\mathcal{D}}(K) \geq \frac{g_{\max,\mathcal{D},k}}{g_{\mathcal{D},k}}(1 - (1 - \ell g_{\mathcal{D},k}\theta)^{1/\ell})$$

with $\ell \leq 1/(g_{\mathcal{D},k}\theta)$, then $\mathbb{E}[\hat{f}_{S_\ell}(K)] \geq \theta$.

SPRISS (Algorithm 4) is motivated by our main technical result, Proposition 11, 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 $\hat{f}_{S_\ell}(K)$.

Proposition 11. *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*

$$m \geq \frac{2}{\varepsilon^2}\left(\frac{1}{\ell g_{\mathcal{D},k}}\right)^2 \left(\lfloor \log_2 \min(2\ell g_{\max,\mathcal{D},k}, \sigma^k) \rfloor + \ln\left(\frac{1}{\delta}\right)\right) \tag{4.2}$$

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 $f_{\mathcal{D}}(K) \leq \frac{g_{\max,\mathcal{D},k}}{g_{\mathcal{D},k}}(1 - (1 - \ell g_{\mathcal{D},k}(\hat{f}_{S_\ell}(K) + \varepsilon/2))^{(1/\ell)})$.

Proof. Let us consider $\hat{f}_{S_\ell}(K) = \frac{1}{m}\sum_{I_{\ell,i}\in S_\ell} f_{K,\ell}(I_{\ell,i})$ and its expectation $E[\hat{f}_{S_\ell}(K)] = \mathbb{E}[f_{K,\ell}(I_{\ell,i})]$, which is taken with respect to the uniform distribution over bags of ℓ reads. By using Proposition 5, Proposition 8, and by the choice of m, we have that with probability at least $1-\delta$ it holds $|\mathbb{E}[\hat{f}_{S_\ell}(K)] - \hat{f}_{S_\ell}(K)| \leq \varepsilon/2$ for every k-mer K, which implies $\hat{f}_{S_\ell}(K) \geq \mathbb{E}[\hat{f}_{S_\ell}(K)] - \varepsilon/2$. Using Proposition 10, when $f_{\mathcal{D}}(K) \geq \frac{g_{\max,\mathcal{D},k}}{g_{\mathcal{D},k}}(1-(1-\ell g_{\mathcal{D},k}\theta)^{1/\ell})$, then $\mathbb{E}[\hat{f}_{S_\ell}(K)] \geq \theta$ and the first part holds.

By the definitions of $\hat{f}_{S_\ell}(K)$ and $f_{S_\ell}(K)$ we have $E[\hat{f}_{S_\ell}(K)] \leq E[f_{S_\ell}(K)] = f_{\mathcal{D}}(K)$. From the proof of the first part we have $|\mathbb{E}[\hat{f}_{S_\ell}(K)] - \hat{f}_{S_\ell}(K)| \leq \varepsilon/2$ for every k-mer K. If we consider a k-mer K with $f_{\mathcal{D}}(K) < \theta - \varepsilon$ we have $\hat{f}_{S_\ell}(K) \leq \mathbb{E}[\hat{f}_{S_\ell}(K)] + \varepsilon/2 \leq f_{\mathcal{D}}(K) + \varepsilon/2 < \theta - \varepsilon/2$ and the second part holds.

Since $f_{\mathcal{D}}(K) \geq E[\hat{f}_{S_\ell}(K)]$ and $|\mathbb{E}[\hat{f}_{S_\ell}(K)] - \hat{f}_{S_\ell}(K)| \leq \varepsilon/2$ for every k-mer K, we have $\mathbb{E}[\hat{f}_{S_\ell}(K)] \geq \hat{f}_{S_\ell}(K) - \varepsilon/2$ and the third part holds.

By Proposition 10 we have $f_{\mathcal{D}}(K) \leq \frac{g_{\max,\mathcal{D},k}}{g_{\mathcal{D},k}}(1-(1-\ell g_{\mathcal{D},k}E[\hat{f}_{S_\ell}(K)])^{(1/\ell)})$. Using the fact that $E[\hat{f}_{S_\ell}(K)] \leq \hat{f}_{S_\ell}(K) + \varepsilon/2$ for every k-mer K, the last part holds. □

SPRISS builds on Proposition 11, 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 11. Note that, given a sample S_ℓ of m bags of ℓ reads from $\mathcal{D}$, with m satisfying the condition of Proposition 11, the set A is *almost* a FNF ε-approximation of $FK(\mathcal{D}, k, \theta)$: Proposition 11 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 4.6.2 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 11) 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).

Now, let us describe in details our algorithm SPRISS (Algorithm 4). SPRISS starts by computing the number m of bags of ℓ reads as in Equa-

tion 4.2, 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 $f_{S_\ell}(K)$.

Algorithm 4: 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 $1 - \delta$

$m \leftarrow \lceil \frac{2}{\varepsilon^2} \left(\frac{1}{\ell g_{\mathcal{D},k}} \right)^2 \left(\lfloor \log_2 \min(2\ell g_{\max,\mathcal{D},k}, \sigma^k) \rfloor + \ln\left(\frac{1}{\delta}\right) \right) \rceil$;
$S \leftarrow$ 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;
$A \leftarrow \emptyset$;
forall $(K, o_S(K)) \in T$ **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;

In practice, in Algorithm 4, 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 Pois-

son 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 11 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 5). 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 $R[K] \approx T[K]$.

4.6 Experimental Evaluation

In this section we present the results of our experimental evaluation of SPRISS. In Section 4.6.2 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.

Algorithm 5: 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 $1-2\delta$

$m \leftarrow \lceil \frac{2}{\varepsilon^2} \left(\frac{1}{\ell g_{\mathcal{D},k}}\right)^2 \left(\lfloor \log_2 \min(2\ell g_{\max,\mathcal{D},k}, \sigma^k) \rfloor + \ln\left(\frac{2}{\delta}\right)\right) \rceil$;
$S \leftarrow$ sample of exactly $m\ell$ reads drawn from $\mathcal{D}$;
$T \leftarrow exact\_counting(S, k)$;
$A \leftarrow \emptyset$;
forall *k-mers* $K \in T$ **do**
    $\hat{f}_{S_\ell}(K) \leftarrow Binomial(m, 1 - e^{-T[K]/m})/(m\ell g_{\mathcal{D},k})$;
    $f_{S_\ell}(K) \leftarrow T[K]/(m\ell g_{\mathcal{D},k})$;
    if $\hat{f}_{S_\ell}(K) \geq \theta - \varepsilon/2$ **then**
        $A \leftarrow A \cup (K, f_{S_\ell}(K))$
return A;

4.6.1 Implementation, Datasets, Parameters, and Environment

We implemented SPRISS as a combination of C++ programs, which perform the reads sampling and save the sample on a file, and as a modification of KMC 3 [Kokot et al., 2017][3], 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 Figure 4.1 for results obtained using JELLYFISH v. 2.3[4] as k-mer counter in SPRISS). In our experiments we used the variant of SPRISS that employs the Poisson approximation for computing S_K (see Algorithm 5). SPRISS implementation, information about how to retrieve the data used in this work, and scripts for reproducing all results are publicity available[5]. We compared SPRISS with the exact k-mer counter KMC and with SAKEIMA [Pellegrina et al., 2020][6], 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 $k = 31$

[3] Available at `https://github.com/refresh-bio/KMC`
[4] Available at `https://github.com/gmarcais/Jellyfish`
[5] Available at `https://github.com/VandinLab/SPRISS`
[6] Available at `https://github.com/VandinLab/SAKEIMA`

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 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 6 large metagenomic datasets from the Human Microbiome Project (HMP)[7], each with $\approx 10^8$ reads and average read length ≈ 100. The characteristics of the HMP datasets are reported in Table 4.1.

dataset	label	$t_{\mathcal{D},k}$	$\lvert\mathcal{D}\rvert$	$\max_{n_i}$	avg_{n_i}
SRS024075(s)	HMP1	$8.82 \cdot 10^9$	$1.38 \cdot 10^8$	95	93.88
SRS024388(s)	HMP2	$7.92 \cdot 10^9$	$1.20 \cdot 10^8$	101	96.21
SRS011239(s)	HMP3	$8.13 \cdot 10^9$	$1.24 \cdot 10^8$	101	95.69
SRS075404(t)	HMP4	$7.75 \cdot 10^9$	$1.22 \cdot 10^8$	101	93.51
SRS043663(t)	HMP5	$9.15 \cdot 10^9$	$1.31 \cdot 10^8$	100	100.00
SRS062761(t)	HMP6	$8.26 \cdot 10^9$	$1.18 \cdot 10^8$	100	100.00

Table 4.1: HMP datasets for our experimental evaluation. For each dataset $\mathcal{D}$ the table shows: the dataset name and site ((s) for stool, (t) for tongue dorsum); its corresponding label on figures; the total number $t_{\mathcal{D},k}$ of k-mers ($k = 31$) in $\mathcal{D}$; the number $\lvert\mathcal{D}\rvert$ of reads it contains; the maximum read length $\max_{n_i} = \max_i\{n_i | r_i \in \mathcal{D}\}$; the average read length $\text{avg}_{n_i} = \sum_{i=1}^{n} n_i/n$.

4.6.2 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 6 datasets from HMP for values of the minimum frequency threshold $\theta \in \{2.5 \cdot 10^{-8}, 5 \cdot 10^{-8}, 7.5 \cdot 10^{-8}, 10^{-7}\}$. The output of SPRISS satisfied the guarantees from Proposition 11 for all 5 runs of every combination of dataset and θ. In all cases the estimated frequencies provided by SPRISS are close to the exact ones (see Figure 4.2a 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 (Figure 4.2b), and

[7] https://hmpdacc.org/HMASM/

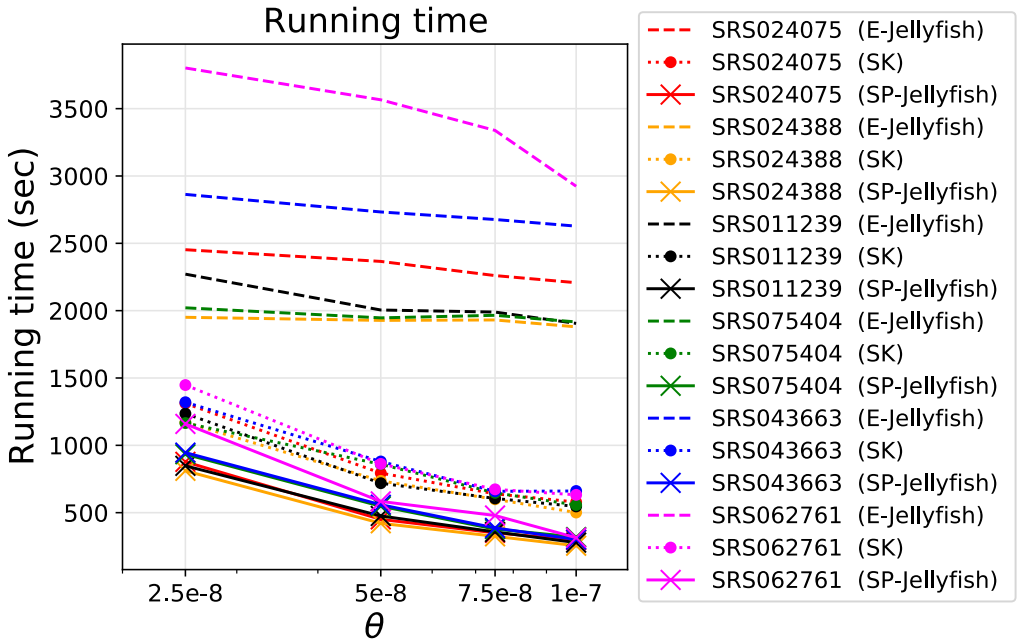


Figure 4.1: As function of θ and for each dataset $\mathcal{D}$ (see Table 4.1), running times to approximate $FK(\mathcal{D}, k, \theta)$ with SPRISS using JELLYFISH (SP-JELLYFISH), with the state-of-the-art sampling algorithm SAKEIMA (SK), and for exactly computing $FK(\mathcal{D}, k, \theta)$ with JELLYFISH (E-JELLYFISH).

the maximum deviation is very small as well (Figure 4.3b). In addition, even if the values of $\tilde{\theta}$ (see *i)* in Proposition 11) are always between $4.15 \cdot 10^{-6}$ and $1.81 \cdot 10^{-5}$, 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 (see Figure 4.3a).

In terms of running time, SPRISS required at most 64% of the time required by the exact approach KMC (Figure 4.2c). 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 (Figure 4.2d). 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 Section 4.3), and pseudodimension without collections of bags (see Section 4.4), are much greater than the

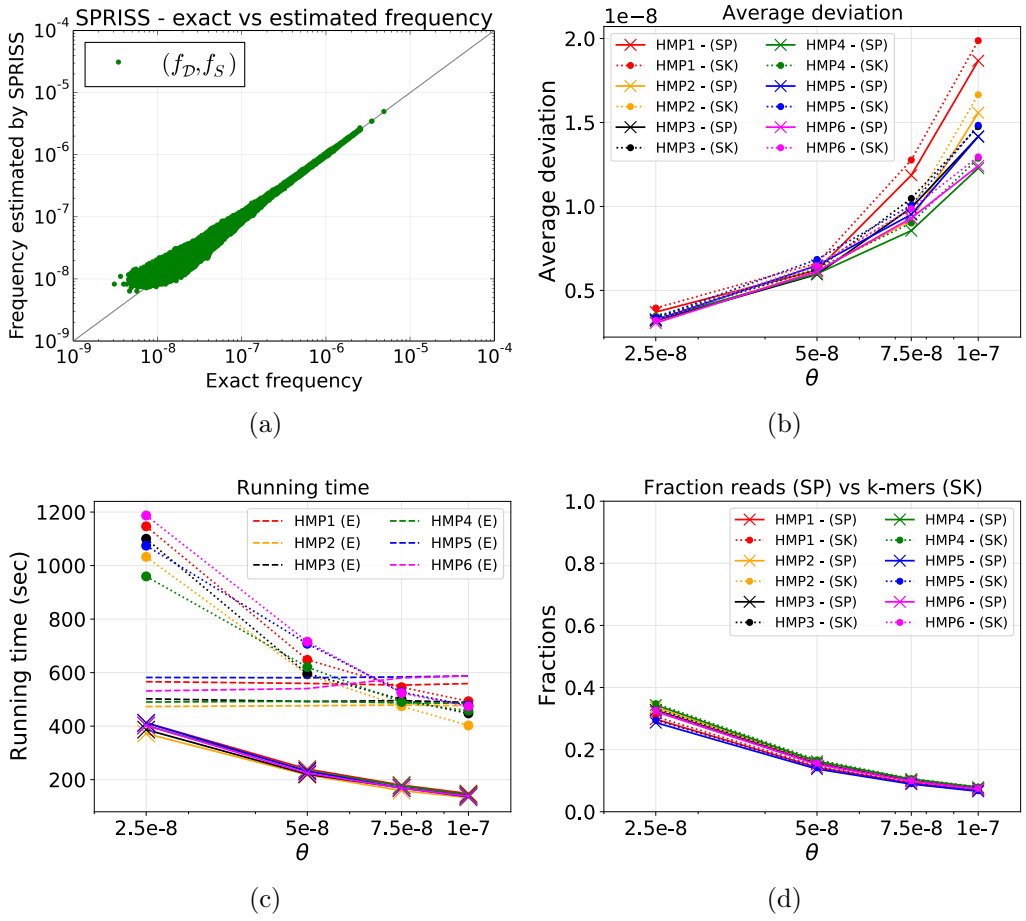
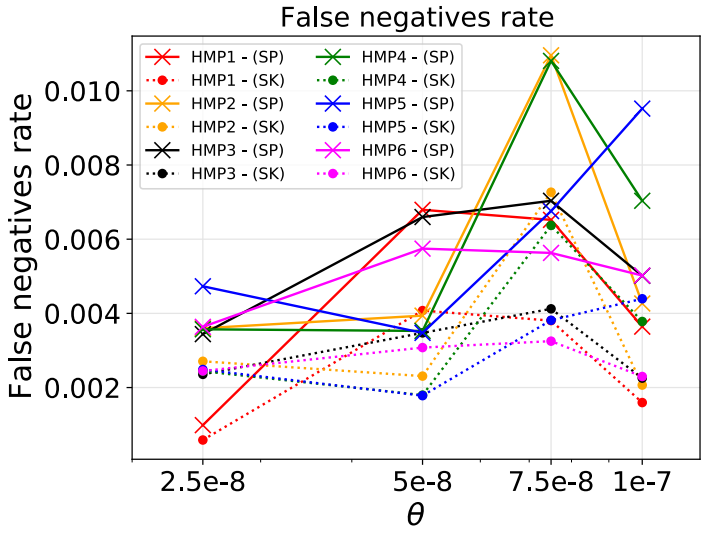


Figure 4.2: (a) k-mers exact frequency and frequency estimated by SPRISS for dataset SRS024075 and $\theta = 2.5 \cdot 10^{-8}$. (b) Average deviations between exact frequencies and frequencies estimated by SPRISS (SP) and SAKEIMA (SK), for various datasets and values of θ. (c) Running time of SPRISS (SP), SAKEIMA (SK), and the exact computation (E) - see also legend of panel 4.2b). (d) Fraction of the dataset analyzed by SPRISS (SP) and by SAKEIMA (SK).

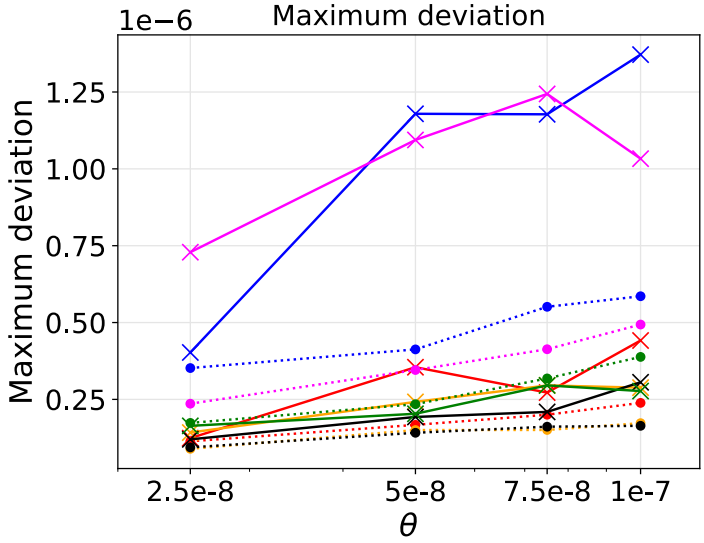
dataset size:$\approx 10^{16}$ and $\approx 10^{15}$, respectively, which are useless sample sizes for datasets of $\approx 10^8$ reads. These results show that SPRISS obtains very accurate approximations of frequent k-mers in a fraction of the time required by exact counting approaches.

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 (Figure 4.2c), and processes slightly smaller portions of the dataset compared to SAKEIMA (Figure 4.2d).

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.



(a)



(b)

Figure 4.3: As function of θ and for every dataset $\mathcal{D}$: (a) False negatives rates, i.e. the fraction of k-mers of $FK(\mathcal{D}, k, \theta)$ not reported by the approximation sets, obtained using SPRISS (SP) and SAKEIMA (SK); (b) Maximum deviations between exact and unbiased observed frequencies provided by the approximations sets of SPRISS (SP) and SAKEIMA (SK).

Chapter 5

Applications of SPRISS

5.1 Introduction

The analysis of k-mers (and of their frequencies) of a dataset of reads is of fundamental importance in several applications, e.g., in [Wood and Salzberg, 2014, Audoux et al., 2017, Li and Waterman, 2003, Liu et al., 2017, Patro et al., 2014, Zhang and Wang, 2014, Solomon and Kingsford, 2016, Kelley et al., 2010, Salmela et al., 2016] (see Section 5.1.2). As stated in Section 4.1, some applications like 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 [Ounit et al., 2015, Liu et al., 2017] require to identify the k-mers that appear in the dataset with relatively high frequency, i.e. frequent k-mers. Since the exact identification of frequent k-mers is highly demanding (see Section 4.6), a valid strategy to speed up such applications is to consider an efficiently computable rigorous approximation of the frequent k-mers obtained from a sample of the whole dataset, which can be computed, e.g., with our algorithm SPRISS described in Chapter 4. Thus, in this Chapter 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.

5.1.1 Our Contributions

In this Chapter we present several applications of SPRISS. In particular:

- 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][1] 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)[2], 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 on an llumina WGS dataset from the Genome In A Bottle (GIAB) consortium [Zook et al., 2014].

5.1.2 Related Works

Given a dataset of reads, the identification of k-mers (and of the frequencies) is crucial 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 [Li and Waterman, 2003, Liu et al., 2017], 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

[1] `https://www.imicrobe.us`
[2] `https://hmpdacc.org/HMASM/`

et al., 2016]. In particular, for a given minimum frequency threshold, frequent k-mers are important for comparing abundances in metagenomic datasets [Benoit et al., 2016, Danovaro et al., 2017, Dickson et al., 2017, Pellegrina et al., 2020] and the discovery of k-mers discriminating between two datasets [Ounit et al., 2015, Liu et al., 2017]. 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.

5.1.3 Organization of the Chapter

Section 5.2 is dedicated to give information about the implementation of the applications of SPRISS, the environment used, the values of the parameters and the datasets used in our tests. In Section 5.3 we evaluate the benefit of using SPRISS for the comparison of metagenomic datasets. Then, in Section 5.4 we test SPRISS in the identification of discriminative k-mers. Finally, in Section 5.5 we evaluate the usage of SPRISS for approximate SNP genotyping.

5.2 Implementation, Datasets, Parameters, and Environment

For all the applications presented in this Chapter, in order to speed up the analyses, we used our algorithm SPRISS to approximate frequent k-mers. The environment used, the implementation of SPRISS and the values used for its parameters are the same as described in Section 4.6.1. Scripts to reproduce all results and information about how to retrieve the data are publicly available[3].

For the evaluation of SPRISS in comparing metagenomic datasets we used the HMP datasets (Table 4.1), which we have also used for the experimental evaluation of SPRISS in Section 4.6, and 37 small metagenomic datasets from the Sorcerer II Global Ocean Sampling Expedition [Rusch et al., 2007], each

[3]Available at `https://github.com/VandinLab/SPRISS`

with $\approx 10^4$-10^5 reads and average read length ≈ 1000 (see Table 5.3). 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 $\approx 4 \cdot 10^8$ reads and average read length = 250 (see Table 5.1), 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 $\approx 1.55 \cdot 10^9$ reads and average read length = 148 (see Table 5.2), available from the Genome In A Bottle (GIAB) consortium [Zook et al., 2014]. All reported results are averages over 5 runs.

Table 5.1: B73 and Mo17 datasets for the discriminative k-mers discovery experiments. For each dataset $\mathcal{D}$ the table shows: the dataset name; the total number $t_{\mathcal{D},k}$ of k-mers ($k = 31$) in $\mathcal{D}$; the number $|\mathcal{D}|$ of reads it contains; the maximum read length $\max_{n_i} = \max_i\{n_i | r_i \in \mathcal{D}\}$; the average read length $\text{avg}_{n_i} = \sum_{i=1}^{n} n_i/n$.

dataset	$t_{\mathcal{D},k}$	$\lvert\mathcal{D}\rvert$	$\max_{n_i}$	avg_{n_i}
`B73`	$9.92 \cdot 10^{10}$	$4.50 \cdot 10^8$	250	250
`Mo17`	$9.97 \cdot 10^{10}$	$4.45 \cdot 10^8$	250	250

Table 5.2: The dataset used for the SNP genotyping experiments. The table shows: the dataset name (we call it *data75x*); the total number $t_{\mathcal{D},k}$ of k-mers ($k = 31$) in $\mathcal{D}$; the number $|\mathcal{D}|$ of reads it contains; the maximum read length $\max_{n_i} = \max_i\{n_i | r_i \in \mathcal{D}\}$; the average read length $\text{avg}_{n_i} = \sum_{i=1}^{n} n_i/n$; the coverage.

dataset	$t_{\mathcal{D},k}$	$\lvert\mathcal{D}\rvert$	$\max_{n_i}$	avg_{n_i}	coverage
`data75x`	$1.83 \cdot 10^{11}$	$1.55 \cdot 10^9$	148	148	$\approx$ 75x

Table 5.3: GOS datasets for our experimental evaluation. For each dataset $\mathcal{D}$: the dataset name; its corresponding label for clustering results of Section 5.3; the total number $t_{\mathcal{D},k}$ of k-mers ($k = 21$) in $\mathcal{D}$; the number $|\mathcal{D}|$ of reads it contains; the maximum read length $\max_{n_i} = \max_i\{n_i | r_i \in \mathcal{D}\}$; the average read length $\text{avg}_{n_i} = \sum_{i=1}^{n} n_i/n$. Prefix IDs of the GOS datasets: TO = Tropical Open ocean, TG = Tropical Galapagos, TN = Temperate North, TS = Temperate South, E = Estuary, NC = Non-Classified.

dataset	label	$t_{\mathcal{D},k}$	$\lvert\mathcal{D}\rvert$	$\max_{n_i}$	avg_{n_i}
GS02	TN1	$1.26 \cdot 10^8$	$1.21 \cdot 10^5$	1349	1058.98
GS03	TN2	$6.56 \cdot 10^7$	$6.16 \cdot 10^4$	1278	1086.07
GS04	TN3	$5.58 \cdot 10^7$	$5.29 \cdot 10^4$	1309	1074.83
GS05	TN4	$6.47 \cdot 10^7$	$6.11 \cdot 10^4$	1242	1079.37
GS06	TN5	$6.34 \cdot 10^7$	$5.96 \cdot 10^4$	1260	1082.71
GS07	TN6	$5.44 \cdot 10^7$	$5.09 \cdot 10^4$	1342	1087.30
GS08	TS1	$1.35 \cdot 10^8$	$1.29 \cdot 10^5$	1444	1062.24
GS09	TS2	$8.27 \cdot 10^7$	$7.93 \cdot 10^4$	1342	1063.35
GS10	TS3	$8.08 \cdot 10^7$	$7.83 \cdot 10^4$	1402	1052.62
GS11	E1	$1.30 \cdot 10^8$	$1.24 \cdot 10^5$	1283	1070.84
GS12	E2	$1.33 \cdot 10^8$	$1.26 \cdot 10^5$	1349	1078.62
GS13	TS4	$1.46 \cdot 10^8$	$1.38 \cdot 10^5$	1300	1079.50
GS14	TG1	$1.37 \cdot 10^8$	$1.28 \cdot 10^5$	1353	1085.58
GS15	TO1	$1.35 \cdot 10^8$	$1.27 \cdot 10^5$	1412	1083.79
GS16	TO2	$1.34 \cdot 10^8$	$1.27 \cdot 10^5$	1328	1081.48
GS17	TO3	$2.76 \cdot 10^8$	$2.57 \cdot 10^5$	1354	1091.92
GS18	TO4	$1.53 \cdot 10^8$	$1.42 \cdot 10^5$	1309	1096.20
GS19	TO5	$1.43 \cdot 10^8$	$1.35 \cdot 10^5$	1325	1081.93
GS20	NC1	$3.09 \cdot 10^8$	$2.96 \cdot 10^5$	1325	1063.42
GS21	TG2	$1.40 \cdot 10^8$	$1.31 \cdot 10^5$	1334	1088.44
GS22	TG3	$1.28 \cdot 10^8$	$1.21 \cdot 10^5$	1288	1077.40
GS23	TO6	$1.40 \cdot 10^8$	$1.33 \cdot 10^5$	1304	1079.48
GS25	NC2	$1.27 \cdot 10^8$	$1.20 \cdot 10^5$	1288	1075.49
GS26	TO7	$1.06 \cdot 10^8$	$1.02 \cdot 10^5$	1337	1061.74
GS27	TG4	$2.32 \cdot 10^8$	$2.22 \cdot 10^5$	1259	1068.65
GS28	TG5	$2.01 \cdot 10^8$	$1.89 \cdot 10^5$	1295	1084.40
GS29	TG6	$1.41 \cdot 10^8$	$1.31 \cdot 10^5$	1356	1093.46
GS30	TG7	$3.84 \cdot 10^8$	$3.59 \cdot 10^5$	1359	1090.61
GS31	TG8	$4.52 \cdot 10^8$	$4.36 \cdot 10^5$	1341	1057.90
GS32	NC3	$1.50 \cdot 10^8$	$1.48 \cdot 10^5$	1366	1035.96
GS33	NC4	$7.15 \cdot 10^8$	$6.92 \cdot 10^5$	1361	1054.10
GS34	TG11	$1.39 \cdot 10^8$	$1.34 \cdot 10^5$	1308	1058.44
GS35	TG12	$1.49 \cdot 10^8$	$1.40 \cdot 10^5$	1321	1078.30
GS36	TG13	$8.42 \cdot 10^7$	$7.75 \cdot 10^4$	1423	1106.00
GS37	TG14	$6.73 \cdot 10^7$	$6.56 \cdot 10^4$	1244	1045.40
GS47	TG15	$6.70 \cdot 10^7$	$6.60 \cdot 10^4$	1304	1035.09
GS51	TG16	$1.37 \cdot 10^8$	$1.28 \cdot 10^5$	1349	1089.27

5.3 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 $\mathcal{FK}_1 = FK(\mathcal{D}_1, k, \theta)$ and $\mathcal{FK}_2 = FK(\mathcal{D}_2, k, \theta)$ 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 $BC(\mathcal{D}_1, \mathcal{D}_2, \mathcal{FK}_1, \mathcal{FK}_2) = 1 - 2I/U$, where $I = \sum_{K \in \mathcal{FK}_1 \cap \mathcal{FK}_2} \min\{o_{\mathcal{D}_1}(K), o_{\mathcal{D}_2}(K)\}$ and $U = \sum_{K \in \mathcal{FK}_1} o_{\mathcal{D}_1}(K) + \sum_{K \in \mathcal{FK}_2} o_{\mathcal{D}_2}(K)$. Conversely, the *BC similarity* is defined as $1 - BC(\mathcal{D}_1, \mathcal{D}_2, \mathcal{FK}_1, \mathcal{FK}_2)$.

We considered 6 datasets from HMP, and estimated the BC distances among them by using SPRISS to approximate the sets of frequent k-mers $\mathcal{FK}_1 = FK(\mathcal{D}_1, k, \theta)$ and $\mathcal{FK}_2 = FK(\mathcal{D}_2, k, \theta)$ for the values of θ as in Section 4.6.2. 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 (see Figure 5.1), 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 (Figure 5.2). 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,

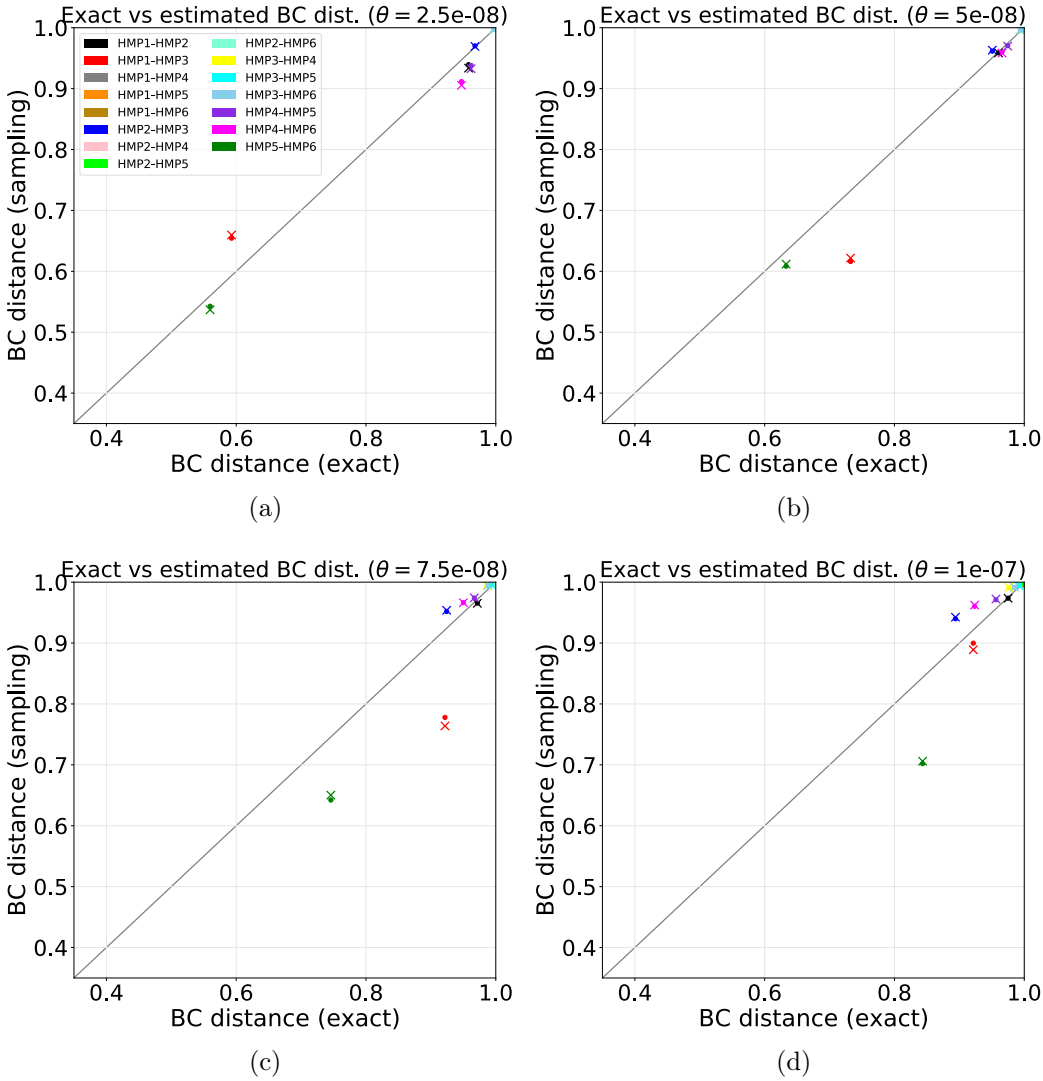
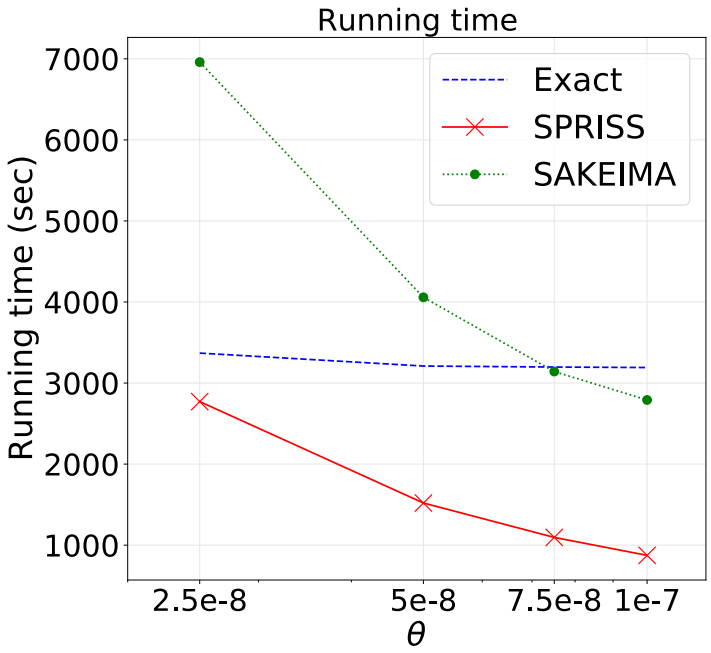


Figure 5.1: Comparison of the approximations of the Bray-Curtis distances using approximations of frequent k-mers sets provided by SPRISS ($\times$) and by SAKEIMA ($\bullet$) with the exact distances, for: (a) $\theta = 2.5 \cdot 10^{-8}$; (b) $\theta = 5 \cdot 10^{-8}$; (c) $\theta = 7.5 \cdot 10^{-8}$; (d) $\theta = 1 \cdot 10^{-7}$.

and, thus, the entire clustering analysis. In fact, for this experiment, the use of the sampling scheme of SPRISS reduces the time required to analyze the datasets (i.e., obtain k-mers frequencies, compute all pairwise distances, and



(a)

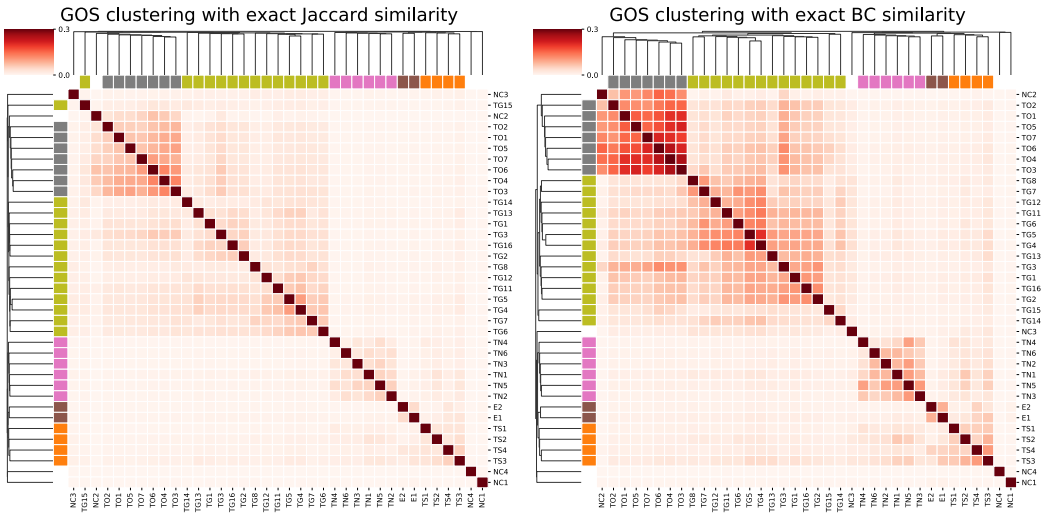
Figure 5.2: Running time to approximate BC distances cumulative for all pairs of HMP datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ in {HMP1,HMP2,HMP3,HMP4,HMP5,HMP6} (see Table 4.1) with SPRISS, with SAKEIMA, and the exact approach.

obtain the clustering) by 62%.

We then compared the clustering obtained using the exact Jaccard distance (Figure 5.3a), the exact BC distance (Figure 5.3b), and the estimates of the BC distance (Figure 5.3c) 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 estimated 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 approx-

imate 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 approximate BC similarity within the tropical macro-group and the average approximate 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 $\approx 50\%$ 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.



(a) (b)

(c)

Figure 5.3: Average linkage hierarchical clustering of GOS datasets using: (a) exact Jaccard similarity; (b) exact Bray-Curtis (BC) similarity; (c) estimates of the BC similarity obtained using the sampling scheme of SPRISS with 50% of the data. Prefix IDs of the GOS datasets: TO=Tropical Open ocean, TG=Tropical Galapagos, TN=Temperate North, TS=Temperate South, E=Estuary, NC=Non-Classified.

5.4 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: 1. $K \in FK(\mathcal{D}_1, k, \theta)$; 2. $f_{\mathcal{D}_1}(K) \geq \rho f_{\mathcal{D}_2}(K)$, 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: 1. $K \in \overline{FK}(\mathcal{D}_1, k, \theta)$; 2. $K \notin \overline{FK}(\mathcal{D}_2, k, \theta)$, or $f_{S^1_\ell}(K) \geq \rho f_{S^2_\ell}(K)$ when $K \in \overline{FK}(\mathcal{D}_2, k, \theta)$.

To evaluate such approach, we considered two datasets from [Liu et al., 2017], and $\theta = 2{\cdot}10^{-7}$ 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 ≈ 1130 sec. and ≈ 1970 sec., respectively, while the exact computation of the discriminative k-mers with KMC requires $\approx 10^4$ sec. (we used 32 workers for both SPRISS and KMC). Similar results are obtained when analyzing pairs of HMP datasets, for various values of θ (Figure 5.4 and Figure 5.5).

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.

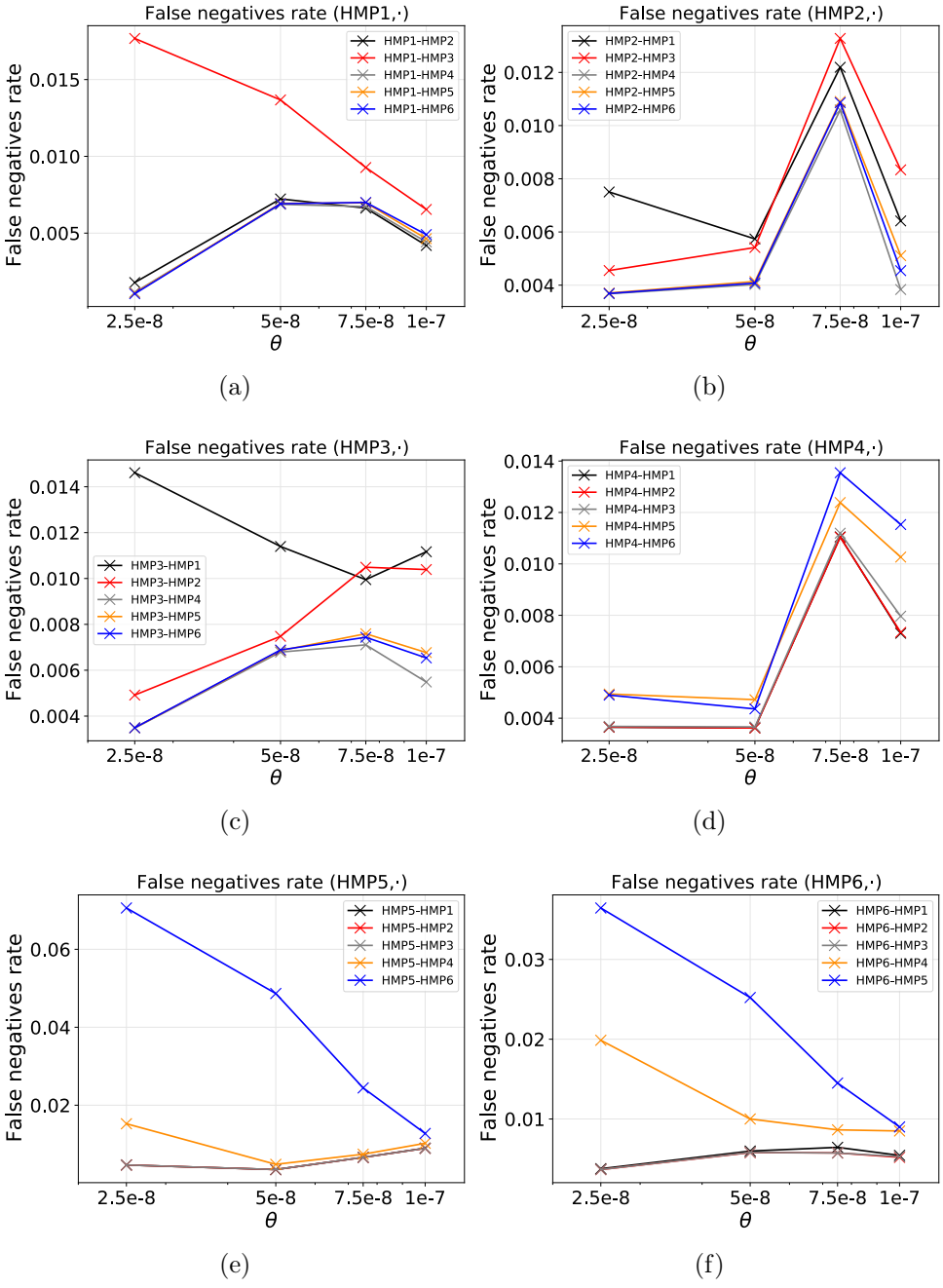
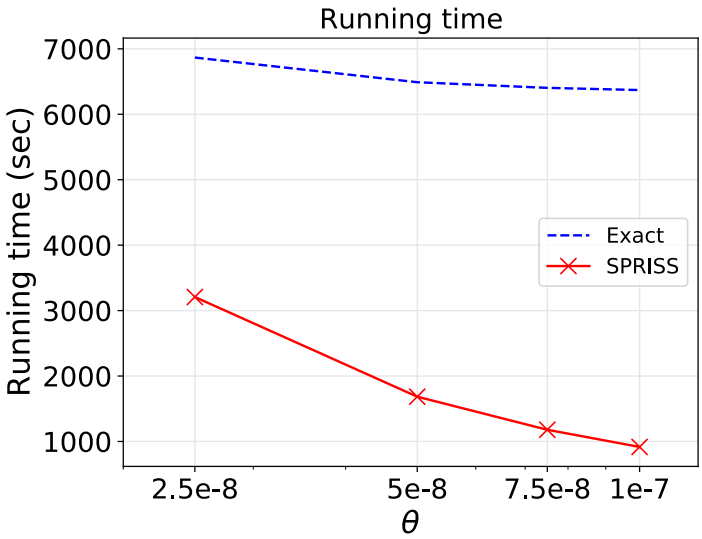


Figure 5.4: As function of θ, false negatives rate, i.e. the fraction of k-mers of $DK(\mathcal{D}_1, \mathcal{D}_2, k, \theta, \rho)$ not included in its approximation $\overline{DK}(\mathcal{D}_1, \mathcal{D}_2, k, \theta, \rho)$, obtained using SPRISS for all pairs of HMP datasets (see Table 4.1).



(a)

Figure 5.5: Running times to compute $\overline{DK}(\mathcal{D}_1, \mathcal{D}_2, k, \theta, \rho)$ using SPRISS against the one required to compute the exact set $DK(\mathcal{D}_1, \mathcal{D}_2, k, \theta, \rho)$, cumulative for all pairs of HMP datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ in {HMP1,HMP2,HMP3,HMP4,HMP5,HMP6} (see Table 4.1).

5.5 SNP Genotyping

In this section we evaluate SPRISS for approximate SNP genotyping. In particular, we assess the use of the sampling scheme of SPRISS in combination with previously proposed algorithms for SNP genotyping in terms of precision, sensitivity, and F-measure.

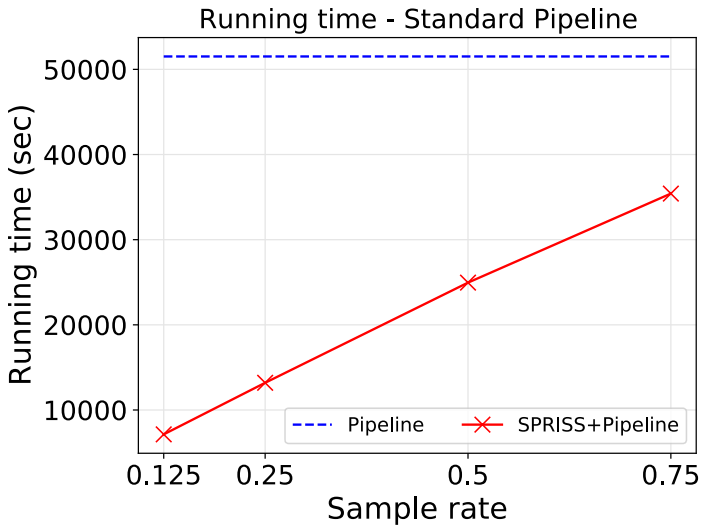
The genotyping algorithms we used are the standard pipeline (BWA [Li and Durbin, 2009] as aligner, and BCFtools [Li, 2011] as variant caller), and VarGeno [Sun and Medvedev, 2018]. We considered hg19 as reference genome, and dbSNP [Sherry, 2001] as reference SNP database. We used the gold standard of 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 contaning 75% of reads of $\mathcal{D}$ the required time is < 3000 sec).

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 5.6 and Figure 5.7 describe 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.

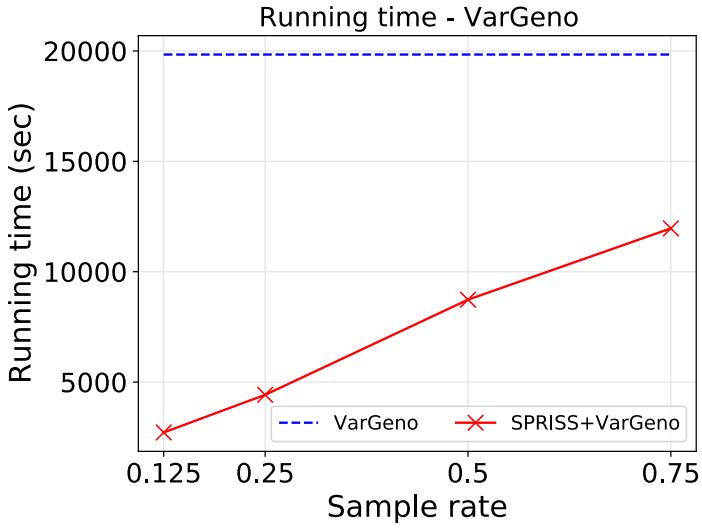
VarGeno achieves on $\mathcal{D}$ 0.974 of precision, 0.585 of sensitivity, and 0.731 of F-measure. With a sample from SPRISS of just 25% of reads of $\mathcal{D}$, we obtain a decrement of the performance of VarGeno of 0.003 in precision, 0.015 in sensitivity, 0.013 in F-measure, and a speed-up of $\approx$ 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 5.6 and Figure 5.7.

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.



(a)



(b)

Figure 5.6: As function of the sample rate, running time of combining the sampling scheme of SPRISS with the standard pipeline (a) and VarGeno (b) in the SNP genotyping process.

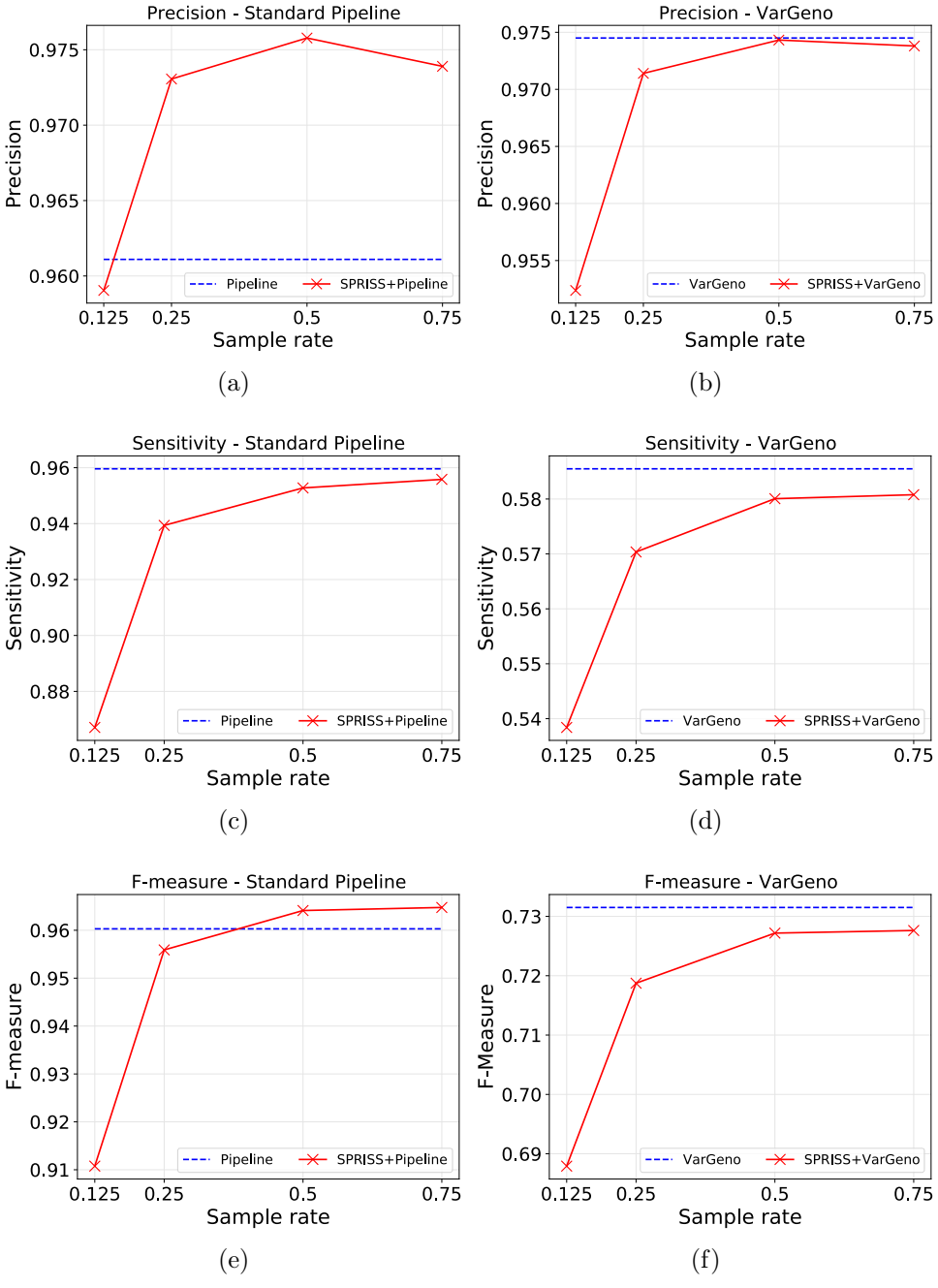


Figure 5.7: As function of the sample rate: precision, sensitivity, and F-measure resulting by combining the sampling scheme of SPRISS with the standard pipeline (resp. (a),(c),(e)) and with VarGeno (resp. (b),(d),(f)) in SNP genotyping.

Chapter 6

Conclusions

In this Chapter we end this Thesis by summarizing our contributions and discussing some possible future directions. In this Thesis we present novel efficient and rigorous approximation algorithms for mining interesting patterns from sequential data, leveraging on strong theoretical guarantees that we proved using tools from statistical learning theory.

In Chapter 3, we studied the task of mining true frequent sequential patterns. We defined rigorous approximations and designed efficient algorithms to extract such approximations with high confidence using an advanced concept from statistical learning theory, i.e., the Rademacher complexity. In particular, we proved the first efficient computable upper bound of the Rademacher complexity of sequential patterns, and we also derived a strategy to approximate it. Both of them, i.e., the upper bound and the approximation of the Rademacher complexity of sequential patterns, are useful to upper bound the maximum deviation between the true frequencies of sequential patterns and their estimates. Our extensive experimental evaluation shows that our algorithms obtain high-quality approximations, even better than guaranteed by their theoretical analyses. In addition, our evaluation shows that the upper bound on the maximum deviation computed using the approximation of the Rademacher complexity allows to obtain better results compared to the ones obtained using the upper bound of the Rademacher complexity. In this scenario of mining true frequent sequential patterns, a possible future direction could be the application of recent results on sharp and uniformly valid confidence bounds based on the Monte Carlo empirical Rademacher complexity [Pellegrina, 2020, Pellegrina et al., 2022] in order to make the upper bound on the maximum deviation sharper and, consequently,

to improve the quality of the approximations.

In Chapter 4, we studied the task of mining frequent k-mers. We presented SPRISS, an efficient algorithm to compute rigorous approximations of frequent k-mers and their frequencies by sampling reads. SPRISS builds on the pseudodimension, an advanced concept from statistical learning theory. In particular, we proved an upper of the pseudodimension of k-mers in reads, which is useful to provide a sample size that is required to obtain high-quality approximations. In addition, we showed that less sophisticated tools like Hoeffding's inequality combined with a union bound, and the VC-dimension, are not sufficient to provide practical sample sizes. Our extensive experimental evaluation shows that SPRISS outputs high-quality estimates of the frequent k-mers, while vastly speeding-up exact approaches by analyzing only a sample of the entire dataset. In Chapter 5, we presented several applications of SPRISS in bioinformatics. In particular, we applied SPRISS to speed-up the comparison of metagenomic datasets, the computation of discriminative k-mers, and the SNP genotyping, showing that we achieve high-quality estimates of the results that would be obtained using exact approaches (i.e., analyzing the entire datasets). In this context of mining frequent k-mers, a possible future direction could be to investigate if the study of the Rademacher complexity, which provides data-dependent bounds, of k-mers in datasets of reads helps in approximating frequent k-mers with better quality and sharper theoretical guarantees. In addition, SPRISS could be used to speed-up several bioinformatic tools that rely on the identification of frequent or discriminative k-mers, e.g., CLARK [Ounit et al., 2015], a tool for the classification of metagenomic and genomic sequences using discriminative k-mers.

Bibliography

[Agrawal et al., 1993] Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216.

[Agrawal and Srikant, 1995] Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In *Proceedings of the Eleventh International Conference on Data Engineering*, pages 3–14. IEEE.

[Al Hasan et al., 2007] Al Hasan, M., Chaoji, V., Salem, S., Besson, J., and Zaki, M. J. (2007). Origami: Mining representative orthogonal graph patterns. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 153–162. IEEE.

[Almodaresi et al., 2018] Almodaresi, F., Sarkar, H., Srivastava, A., and Patro, R. (2018). A space and time-efficient index for the compacted colored de bruijn graph. *Bioinformatics*, 34(13):i169–i177.

[Audano and Vannberg, 2014] Audano, P. and Vannberg, F. (2014). Kanalyze: a fast versatile pipelined k-mer toolkit. *Bioinformatics*, 30(14):2070–2072.

[Audoux et al., 2017] Audoux, J., Philippe, N., Chikhi, R., Salson, M., Gallopin, M., Gabriel, M., Le Coz, J., Drouineau, E., Commes, T., and Gautheret, D. (2017). De-kupl: exhaustive capture of biological variation in rna-seq data through k-mer decomposition. *Genome Biology*, 18(1):243.

[Benoit et al., 2016] Benoit, G., Peterlongo, P., Mariadassou, M., Drezen, E., Schbath, S., Lavenier, D., and Lemaitre, C. (2016). Multiple comparative metagenomics using multiset k-mer counting. *PeerJ Computer Science*, 2:e94.

[Boucheron et al., 2005] Boucheron, S., Bousquet, O., and Lugosi, G. (2005). Theory of classification: A survey of some recent advances. *ESAIM: Probability and Statistics*, 9:323–375.

[Bradley et al., 2019] Bradley, P., Den Bakker, H. C., Rocha, E. P., McVean, G., and Iqbal, Z. (2019). Ultrafast search of all deposited bacterial and viral genomic data. *Nature Biotechnology*, 37(2):152–159.

[Brown et al., 2012] Brown, C. T., Howe, A., Zhang, Q., Pyrkosz, A. B., and Brom, T. H. (2012). A reference-free algorithm for computational normalization of shotgun sequencing data. *arXiv preprint arXiv:1203.4802*.

[Cheng et al., 2010] Cheng, J., Fu, A. W.-c., and Liu, J. (2010). K-isomorphism: privacy preserving network publication against structural attacks. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, pages 459–470.

[Chikhi et al., 2021] Chikhi, R., Holub, J., and Medvedev, P. (2021). Data structures to represent a set of k-long dna sequences. *ACM Comput. Surv.*, 54(1).

[Chikhi et al., 2014] Chikhi, R., Limasset, A., Jackman, S., Simpson, J. T., and Medvedev, P. (2014). On the representation of de bruijn graphs. In *International Conference on Research in Computational Molecular Biology*, pages 35–55. Springer.

[Chikhi and Medvedev, 2013] Chikhi, R. and Medvedev, P. (2013). Informed and automated k-mer size selection for genome assembly. *Bioinformatics*, 30(1):31–37.

[Coleman et al., 2019] Coleman, B., Geordie, B., Chou, L., Elworth, R. L., Treangen, T. J., and Shrivastava, A. (2019). Diversified race sampling on data streams applied to metagenomic sequence analysis. *bioRxiv*, page 852889.

[Corizzo et al., 2019] Corizzo, R., Pio, G., Ceci, M., and Malerba, D. (2019). Dencast: distributed density-based clustering for multi-target regression. *Journal of Big Data*, 6(1):43.

[Dadi et al., 2018] Dadi, T. H., Siragusa, E., Piro, V. C., Andrusch, A., Seiler, E., Renard, B. Y., and Reinert, K. (2018). Dream-yara: An exact

read mapper for very large databases with short update time. *Bioinformatics*, 34(17):i766–i772.

[Danovaro et al., 2017] Danovaro, R., Canals, M., Tangherlini, M., Dell'Anno, A., Gambi, C., Lastras, G., Amblas, D., Sanchez-Vidal, A., Frigola, J., Calafat, A. M., et al. (2017). A submarine volcanic eruption leads to a novel microbial habitat. *Nature Ecology & Evolution*, 1(6):0144.

[Dickson et al., 2017] Dickson, L. B., Jiolle, D., Minard, G., Moltini-Conclois, I., Volant, S., Ghozlane, A., Bouchier, C., Ayala, D., Paupy, C., Moro, C. V., et al. (2017). Carryover effects of larval exposure to different environmental bacteria drive adult trait variation in a mosquito vector. *Science Advances*, 3(8):e1700585.

[Elworth et al., 2020] Elworth, R. L., Wang, Q., Kota, P. K., Barberan, C., Coleman, B., Balaji, A., Gupta, G., Baraniuk, R. G., Shrivastava, A., and Treangen, T. J. (2020). To petabytes and beyond: recent advances in probabilistic and signal processing algorithms and their application to metagenomics. *Nucleic Acids Research*, 48(10):5217–5234.

[Fournier-Viger et al., 2016] Fournier-Viger, P., Lin, J. C.-W., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., and Lam, H. T. (2016). The spmf open-source data mining library version 2. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 36–40. Springer.

[Fournier-Viger et al., 2019] Fournier-Viger, P., Lin, J. C.-W., Nkambou, R., Vo, B., and Tseng, V. S. (2019). High-utility pattern mining. *Cham: Springer*.

[Guo et al., 2019] Guo, H., Fu, Y., Gao, Y., Li, J., Wang, Y., and Liu, B. (2019). degsm: memory scalable construction of large scale de bruijn graph. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

[Gwadera and Crestani, 2010] Gwadera, R. and Crestani, F. (2010). Ranking sequential patterns with respect to significance. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 286–299. Springer.

[Hämäläinen and Webb, 2019] Hämäläinen, W. and Webb, G. I. (2019). A tutorial on statistically sound pattern discovery. *Data Mining and Knowledge Discovery*, 33(2):325–377.

[Han et al., 2007] Han, J., Cheng, H., Xin, D., and Yan, X. (2007). Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86.

[Harris and Medvedev, 2020] Harris, R. S. and Medvedev, P. (2020). Improved representation of sequence bloom trees. *Bioinformatics*, 36(3):721–727.

[Hernaez et al., 2019] Hernaez, M., Pavlichin, D., Weissman, T., and Ochoa, I. (2019). Genomic data compression. *Annual Review of Biomedical Data Science*, 2:19–37.

[Holley and Melsted, 2020] Holley, G. and Melsted, P. (2020). Bifrost: highly parallel construction and indexing of colored and compacted de bruijn graphs. *Genome Biology*, 21(1):1–20.

[Hosseini et al., 2016] Hosseini, M., Pratas, D., and Pinho, A. J. (2016). A survey on data compression methods for biological sequences. *Information*, 7(4):56.

[Johnson, 2014] Johnson, S. G. (2014). The nlopt nonlinear-optimization package.

[Kelley et al., 2010] Kelley, D. R., Schatz, M. C., and Salzberg, S. L. (2010). Quake: quality-aware detection and correction of sequencing errors. *Genome Biology*, 11(11):R116.

[Kokot et al., 2017] Kokot, M., Długosz, M., and Deorowicz, S. (2017). Kmc 3: counting and manipulating k-mer statistics. *Bioinformatics*, 33(17):2759–2761.

[Kurtz et al., 2008] Kurtz, S., Narechania, A., Stein, J. C., and Ware, D. (2008). A new method to compute k-mer frequencies and its application to annotate large repetitive plant genomes. *BMC Genomics*, 9(1):517.

[Li, 2011] Li, H. (2011). A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, 27(21):2987–2993.

[Li and Durbin, 2009] Li, H. and Durbin, R. (2009). Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, 25(14):1754–1760.

[Li and Waterman, 2003] Li, X. and Waterman, M. S. (2003). Estimating the repeat structure and length of dna sequences using ℓ-tuples. *Genome Research*, 13(8):1916–1922.

[Liu et al., 2017] Liu, S., Zheng, J., Migeon, P., Ren, J., Hu, Y., He, C., Liu, H., Fu, J., White, F. F., Toomajian, C., et al. (2017). Unbiased k-mer analysis reveals changes in copy number of highly repetitive sequences during maize domestication and improvement. *Scientific Reports*, 7:42444.

[Löffler and Phillips, 2009] Löffler, M. and Phillips, J. M. (2009). Shape fitting on point sets with probability distributions. In *European Symposium on Algorithms*, pages 313–324. Springer.

[Long, 1999] Long, P. M. (1999). The complexity of learning according to two models of a drifting environment. *Machine Learning*, 37(3):337–354.

[Low-Kam et al., 2013] Low-Kam, C., Raïssi, C., Kaytoue, M., and Pei, J. (2013). Mining statistically significant sequential patterns. In *2013 IEEE 13th International Conference on Data Mining*, pages 488–497. IEEE.

[Marçais and Kingsford, 2011] Marçais, G. and Kingsford, C. (2011). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770.

[Marchet et al., 2019a] Marchet, C., Boucher, C., Puglisi, S. J., Medvedev, P., Salson, M., and Chikhi, R. (2019a). Data structures based on k-mers for querying large collections of sequencing datasets. *bioRxiv*, page 866756.

[Marchet et al., 2020a] Marchet, C., Iqbal, Z., Gautheret, D., Salson, M., and Chikhi, R. (2020a). Reindeer: efficient indexing of k-mer presence and abundance in sequencing datasets. *bioRxiv*.

[Marchet et al., 2019b] Marchet, C., Kerbiriou, M., and Limasset, A. (2019b). Indexing de bruijn graphs with minimizers. *BioRxiv*, page 546309.

[Marchet et al., 2020b] Marchet, C., Lecompte, L., Limasset, A., Bittner, L., and Peterlongo, P. (2020b). A resource-frugal probabilistic dictionary and applications in bioinformatics. *Discrete Applied Mathematics*, 274:92–102.

[Melsted and Halldórsson, 2014] Melsted, P. and Halldórsson, B. V. (2014). Kmerstream: streaming algorithms for k-mer abundance estimation. *Bioinformatics*, 30(24):3541–3547.

[Melsted and Pritchard, 2011] Melsted, P. and Pritchard, J. K. (2011). Efficient counting of k-mers in dna sequences using a bloom filter. *BMC Bioinformatics*, 12(1):333.

[Mitzenmacher and Upfal, 2017] Mitzenmacher, M. and Upfal, E. (2017). *Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis*. Cambridge University Press.

[Mohamadi et al., 2017] Mohamadi, H., Khan, H., and Birol, I. (2017). ntcard: a streaming algorithm for cardinality estimation in genomics data. *Bioinformatics*, 33(9):1324–1330.

[Numanagić et al., 2016] Numanagić, I., Bonfield, J. K., Hach, F., Voges, J., Ostermann, J., Alberti, C., Mattavelli, M., and Sahinalp, S. C. (2016). Comparison of high-throughput sequencing data compression tools. *Nature Methods*, 13(12):1005–1008.

[Ondov et al., 2016] Ondov, B. D., Treangen, T. J., Melsted, P., Mallonee, A. B., Bergman, N. H., Koren, S., and Phillippy, A. M. (2016). Mash: fast genome and metagenome distance estimation using minhash. *Genome Biology*, 17(1):132.

[Ounit et al., 2015] Ounit, R., Wanamaker, S., Close, T. J., and Lonardi, S. (2015). Clark: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics*, 16(1):1–13.

[Pandey et al., 2018] Pandey, P., Almodaresi, F., Bender, M. A., Ferdman, M., Johnson, R., and Patro, R. (2018). Mantis: A fast, small, and exact large-scale sequence-search index. *Cell Systems*, 7(2):201–207.

[Pandey et al., 2017] Pandey, P., Bender, M. A., Johnson, R., and Patro, R. (2017). Squeakr: an exact and approximate k-mer counting system. *Bioinformatics*, 34(4):568–575.

[Patro et al., 2014] Patro, R., Mount, S. M., and Kingsford, C. (2014). Sailfish enables alignment-free isoform quantification from rna-seq reads using lightweight algorithms. *Nature Biotechnology*, 32(5):462.

[Pei et al., 2004] Pei, J., Han, J., Mortazavi-Asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., and Hsu, M.-C. (2004). Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1424–1440.

[Pellegrina, 2020] Pellegrina, L. (2020). Sharper convergence bounds of monte carlo rademacher averages through self-bounding functions. arxiv preprint arxiv.2010.12103.

[Pellegrina et al., 2022] Pellegrina, L., Cousins, C., Vandin, F., and Riondato, M. (2022). Mcrapper: Monte-carlo rademacher averages for poset families and approximate pattern mining. *ACM Trans. Knowl. Discov. Data*, 16(6).

[Pellegrina et al., 2020] Pellegrina, L., Pizzi, C., and Vandin, F. (2020). Fast approximation of frequent k-mers and applications to metagenomics. *Journal of Computational Biology*, 27(4):534–549.

[Pellegrina et al., 2019] Pellegrina, L., Riondato, M., and Vandin, F. (2019). Spumante: Significant pattern mining with unconditional testing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1528–1538.

[Pollard, 1984] Pollard, D. (1984). *Convergence of Stochastic Processes*. Springer-Verlag.

[Rahman et al., 2020] Rahman, A., Chikhi, R., and Medvedev, P. (2020). Disk compression of k-mer sets. In *20th International Workshop on Algorithms in Bioinformatics (WABI 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[Rahman and Medvedev, 2020] 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*, pages 152–168. Springer.

[Raïssi and Poncelet, 2007] Raïssi, C. and Poncelet, P. (2007). Sampling for sequential pattern mining: From static databases to data streams. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 631–636. IEEE.

[Riondato and Upfal, 2015] Riondato, M. and Upfal, E. (2015). Mining frequent itemsets through progressive sampling with rademacher averages. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1005–1014. ACM.

[Riondato and Upfal, 2018] Riondato, M. and Upfal, E. (2018). Abra: Approximating betweenness centrality in static and dynamic graphs with rademacher averages. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(5):61.

[Riondato and Vandin, 2014] Riondato, M. and Vandin, F. (2014). Finding the true frequent itemsets. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 497–505. SIAM.

[Rizk et al., 2013] Rizk, G., Lavenier, D., and Chikhi, R. (2013). Dsk: k-mer counting with very low memory usage. *Bioinformatics*, 29(5):652–653.

[Roy et al., 2014] Roy, R. S., Bhattacharya, D., and Schliep, A. (2014). Turtle: Identifying frequent k-mers with cache-efficient algorithms. *Bioinformatics*, 30(14):1950–1957.

[Rusch et al., 2007] Rusch, D. B., Halpern, A. L., Sutton, G., Heidelberg, K. B., Williamson, S., Yooseph, S., Wu, D., Eisen, J. A., Hoffman, J. M., Remington, K., Beeson, K., Tran, B., Smith, H., Baden-Tillson, H., Stewart, C., Thorpe, J., Freeman, J., Andrews-Pfannkoch, C., Venter, J. E., Li, K., Kravitz, S., Heidelberg, J. F., Utterback, T., Rogers, Y.-H., Falcón, L. I., Souza, V., Bonilla-Rosso, G., Eguiarte, L. E., Karl, D. M., Sathyendranath, S., Platt, T., Bermingham, E., Gallardo, V., Tamayo-Castillo, G., Ferrari, M. R., Strausberg, R. L., Nealson, K., Friedman, R., Frazier, M., and Venter, J. C. (2007). The sorcerer ii global ocean sampling expedition: Northwest atlantic through eastern tropical pacific. *PLOS Biology*, 5(3):1–34.

[Saavedra et al., 2020] Saavedra, A., Lehnert, H., Hernández, C., Carvajal, G., and Figueroa, M. (2020). Mining discriminative k-mers in dna sequences using sketches and hardware acceleration. *IEEE Access*, 8:114715–114732.

[Salmela et al., 2016] Salmela, L., Walve, R., Rivals, E., and Ukkonen, E. (2016). Accurate self-correction of errors in long reads using de bruijn graphs. *Bioinformatics*, 33(6):799–806.

[Santoro et al., 2022] Santoro, D., Pellegrina, L., Comin, M., and Vandin, F. (2022). SPRISS: approximating frequent k-mers by sampling reads, and applications. *Bioinformatics*, 38(13):3343–3350.

[Santoro et al., 2021] Santoro, D., Pellegrina, L., and Vandin, F. (2021). Spriss: Approximating frequent k-mers by sampling reads, and applications. arxiv preprint arxiv.2101.07117.

[Santoro et al., 2020] Santoro, D., Tonon, A., and Vandin, F. (2020). Mining sequential patterns with vc-dimension and rademacher complexity. *Algorithms*, 13(5).

[Servan-Schreiber et al., 2018] Servan-Schreiber, S., Riondato, M., and Zgraggen, E. (2018). Prosecco: Progressive sequence mining with convergence guarantees. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 417–426. IEEE.

[Shalev-Shwartz and Ben-David, 2014] Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

[Sherry, 2001] Sherry, S. T. (2001). dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, 29(1):308–311.

[Simionato and Vandin, 2022] Simionato, D. and Vandin, F. (2022). Bounding the family-wise error rate in local causal discovery using rademacher averages. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2022)*.

[Sims et al., 2009] Sims, G. E., Jun, S.-R., Wu, G. A., and Kim, S.-H. (2009). Alignment-free genome comparison with feature frequency profiles (ffp) and optimal resolutions. *Proceedings of the National Academy of Sciences*, 106(8):2677–2682.

[Sivadasan et al., 2016] Sivadasan, N., Srinivasan, R., and Goyal, K. (2016). Kmerlight: fast and accurate k-mer abundance estimation. *arXiv preprint arXiv:1609.05626*.

[Solomon and Kingsford, 2016] Solomon, B. and Kingsford, C. (2016). Fast search of thousands of short-read sequencing experiments. *Nature Biotechnology*, 34(3):300.

[Solomon and Kingsford, 2018] Solomon, B. and Kingsford, C. (2018). Improved search of large transcriptomic sequencing databases using split sequence bloom trees. *Journal of Computational Biology*, 25(7):755–765.

[Srikant and Agrawal, 1996] Srikant, R. and Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. In *International Conference on Extending Database Technology*, pages 1–17. Springer.

[Sun et al., 2018] Sun, C., Harris, R. S., Chikhi, R., and Medvedev, P. (2018). Allsome sequence bloom trees. *Journal of Computational Biology*, 25(5):467–479.

[Sun and Medvedev, 2018] Sun, C. and Medvedev, P. (2018). Toward fast and accurate SNP genotyping from whole genome sequencing data for bedside diagnostics. *Bioinformatics*, 35(3):415–420.

[Talagrand, 1994] Talagrand, M. (1994). Sharper bounds for gaussian and empirical processes. *The Annals of Probability*, pages 28–76.

[Tonon and Vandin, 2019] Tonon, A. and Vandin, F. (2019). Permutation strategies for mining significant sequential patterns. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1330–1335. IEEE.

[Vapnik, 1998] Vapnik, V. (1998). *Statistical learning theory.* Wiley, New York.

[Vapnik, 1999] Vapnik, V. (1999). *The nature of statistical learning theory.* Springer science & business media.

[Vapnik and Chervonenkis, 1971] Vapnik, V. N. and Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280.

[Vapnik and Chervonenkis, 2015] Vapnik, V. N. and Chervonenkis, A. Y. (2015). On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of Complexity*, pages 11–30. Springer.

[Wang et al., 2007] Wang, J., Han, J., and Li, C. (2007). Frequent closed sequence mining without candidate maintenance. *IEEE Transactions on Knowledge and Data Engineering*, 19(8):1042–1056.

[Wedemeyer et al., 2017] Wedemeyer, A., Kliemann, L., Srivastav, A., Schielke, C., Reusch, T. B., and Rosenstiel, P. (2017). An improved filtering algorithm for big read datasets and its application to single-cell assembly. *BMC Bioinformatics*, 18(1):324.

[Wood and Salzberg, 2014] Wood, D. E. and Salzberg, S. L. (2014). Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology*, 15(3):R46.

[Yu et al., 2018] Yu, Y., Liu, J., Liu, X., Zhang, Y., Magner, E., Lehnert, E., Qian, C., and Liu, J. (2018). Seqothello: querying rna-seq experiments at scale. *Genome Biology*, 19(1):167.

[Zhang et al., 2014] Zhang, Q., Pell, J., Canino-Koning, R., Howe, A. C., and Brown, C. T. (2014). These are not the k-mers you are looking for: efficient online k-mer counting using a probabilistic data structure. *PloS One*, 9(7):e101271.

[Zhang and Wang, 2014] Zhang, Z. and Wang, W. (2014). Rna-skim: a rapid method for rna-seq quantification at transcript level. *Bioinformatics*, 30(12):i283–i292.

[Zook et al., 2014] Zook, J. M., Chapman, B., Wang, J., Mittelman, D., Hofmann, O., Hide, W., and Salit, M. (2014). Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nature Biotechnology*, 32(3):246–251.