Mining of Significant Patterns:
Theory and Practice

Dottorando: Fabio Vandin

Supervisore: Ch.mo Prof. Andrea Alberto Pietracaprina

Direttore della Scuola: Ch.mo Prof. Matteo Bertocco

Scuola di Dottorato in Ingegneria dell'Informazione
Indirizzo: Scienza e Tecnologia dell'Informazione
XXII Ciclo

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Padova

2009

Abstract

Recent advances in technology allow for the collection and storage of vast amounts of data in many different areas. Data mining is the process of discovering new and useful information. Many techniques have been developed in recent years for the analysis of large datasets, but the task of assessing the significance of discovered patterns and the validity of forecast based on these discoveries is becoming a major challenge in data intensive applications. The objective of this thesis is the development of rigorous and efficient techniques for mining significant patterns in three different and important scenarios.

The first scenario is the mining of frequent itemsets from transactional datasets. For this problem we first study two primitives: the extraction of top-K frequent closed itemsets, a recently proposed alternative to the extraction of frequent itemsets, that provides a better control on the output size, which is one of the main challenges of the traditional problem; and the use of sampling for the extraction of top-K frequent items/itemsets. The notion of top-K frequent patterns provides a first attempt to enhance the effectiveness of the traditional framework by relating the significance to a frequency based ranking rather than to a mere frequency threshold. For both primitives we develop new algorithms and provide experimental evidence of their effectiveness. We then address the problem of identifying a meaningful frequency threshold such that that the itemsets that are frequent w.r.t. that threshold can be flagged as statistically significant with a small *False Discovery Rate* (FDR), which is defined as the expected ratio of false discoveries among all discoveries. A crucial feature of our approach is that, unlike most previous work, it takes into account the entire dataset rather than individual discoveries. Experimental results are reported which show the effectiveness of our approach.

The second scenario is the mining of patterns, called *motifs*, that repeat frequently, possibly with some errors, in biological sequences. This problem has attracted wide interest in recent years, since sequence similarity is often a necessary condition for functional correlation. We introduce *density*, a simple and flexible measure for bounding the number of errors, modeled thorugh *don't cares*, in a motif. We design a new algorithm to extract maximal dense motifs from a sequence, and provide experimental evidence of the biological significance of the motifs that the algorithm returns. Moreover, we compare the motifs extracted by our algorithm with the ones found by a recently proposed algorithm, showing that our algorithm can identify motifs that are more significant according to z-score, a widely employed measure of significance.

The last problem we consider is the mining of significant patterns from large-scale gene and protein interaction networks, a problem of increasing interest since its importance in cancer studies. For this scenario we define the problem of identifying *significantly mutated pathways* in large scale gene and protein interaction networks. We introduce a computational framework that is the first, to our knowledge, to demonstrate a computationally efficient strategy for *de novo* identification of *statistically significant* mutated subnetworks, and design two algorithms to efficiently extract the significantly mutated pathways. Moreover we test these algorithms on a large human protein-protein interaction network using mutation data from recent studies on two different type of cancers. The results of our tests show that our methods correctly identifies the pathways that are implicated in cancer.

Sommario

I recenti progressi tecnologici permettono la raccolta e la memorizzazione di enormi quantità di dati in molte aree diverse. Il *data mining* è il processo di estrazione di informazione nuova, interessante ed utile. Negli ultimi anni un cospicuo numero di soluzioni sono state sviluppate per l'analisi di grandi moli di dati, ma il processo di valutazione della significatività dei pattern estratti e di validazione delle previsioni basate su questi pattern sta diventando uno dei principali *challenge* nell'ambito delle applicazioni che elaborano enormi quantità di dati. Questa tesi si focalizza sullo sviluppo di tecniche rigorose ed efficienti per l'estrazione di pattern significativi in tre diversi scenari rilevanti.

Il primo scenario considerato è l'estrazione di pattern frequenti, chiamati *itemset*, da dataset transazionali. Inizialmente vengono studiate due primitive molto utilizzate per questo problema: l'estrazione dei K itemset chiusi più frequenti, un problema proposto recentemente come alternativa all'estrazione degli itemset frequenti che fornisce un maggior controllo sulla taglia dell'output, che è una delle principali difficoltà per il problema tradizionale; l'estrazione dei K itemset più frequenti tramite *sampling*. La nozione di K itemset chiusi più frequenti fornisce un primo tentativo di migliorare l'efficacia del framework tradizionale, legando la significatività ad un ordinamento basato sulla frequenza invece che ad un semplice soglia di frequenza. Per entrambe queste primitive vengono sviluppati nuovi algoritmi e viene fornita evidenza sperimentale della loro efficacia. Successivamente viene studiato il problema dell'identificazione di una soglia di supporto significativa tale che gli itemset che risultano frequenti rispetto a tale soglia possono essere contrassegnati come significativi con un basso *False Discovery Rate* (FDR), che è definito come il rapporto atteso tra il numero di scoperte erronee e il numero totale di pattern prodotti in output. Una caratteristica cruciale che distingue il nostro approccio dalla maggior parte dei lavori precedenti è che il nostro framework considera l'intero dataset per valutare la significatività di un pattern. Vengono inoltre forniti i risultati dell'analisi sperimentale che mostrano l'efficacia del nostro approccio.

Il secondo scenario che consideriamo è l'estrazione di pattern, chiamati *motif*, che si ripetono frequentemente, eventualmente con errori, in sequenze biologiche. Questo problema ha attratto molto interesse negli ultimi anni, dato che la similarità a livello di sequenza è spesso una condizione necessaria per avere correlazione a livello funzionale a livello di DNA, RNA o proteine. Per questo problema viene introdotta la nozione di *densità*, una misura semplice e flessibile per limitare il numero di errori, rappresentati tramite *don't cares*, in un motif. Viene sviluppato un nuovo algoritmo per l'estrazione di motif densi massimali da una sequenza, e viene fornita evidenza

sperimentale della significatività biologica dei motivi che l'algoritmo estra. Inoltre, i motivi estratti dal nostro algoritmo vengono confrontati con quelli trovati da un altro algoritmo proposto recentemente, mostrando che il nostro algoritmo identifica motif che risultano più significativi rispetto allo z-score, una misura di significatività molto utilizzata.

L'ultimo scenario che viene considerato è l'estrazione di pattern significativi da grandi reti di interazione fra geni e proteine, un problema di crescente interesse vista la sua importanza negli studi sul cancro. Per questo scenario viene definito il problema dell'identificazione di sottoreti *mutate in maniera significativa.* Viene introdotto il primo framework computazionale, al meglio della nostra conoscenza, che fornisce una strategia computazionale efficiente per l'identificazione *de novo* di sottoreti mutate in maniera *statisticamente significativa* e vengono sviluppati due algoritmi per l'identificazione di tali sottoreti. Tali algoritmi sono valutati utilizzando una grande rete di interazione tra proteine e utilizzando dati di mutazione ottenuti da recenti studi su due tipi di cancro. I risultati di questa valutazione mostrano che i nostri algoritmi identificano correttamente le sottoreti che sono implicate nell'insorgenza del cancro.

Acknowledgments

Three years. A lot of time, or maybe not. Judging from the changes in my life in these last three years, they are a lot. And so I am here to thank the ones have been walking with me in this journey. Since Italian is still the language I know the best, I am going to resort to it whenever possible.

Prima di tutti voglio ringraziare Andrea: da lui ho imparato tantissimo, in ogni aspetto della ricerca. La sua pazienza e i suoi consigli mi hanno fatto crescere da ogni punto di vista. E umanamente è ancora più bravo che come professore, pochi hanno la fortuna di avere un maestro del genere. Sarò sempre in debito con lui. Voglio poi ringraziare Geppino: mi sarei considerato fortunato per aver potuto lavorare con un ottimo professore che è anche una grande persona, averne trovati due ha dell'incredibile. In these three years, I have been abroad (almost) a year and a half, enjoying the environment of Brown University. I have to thank Eli for this opportunity, and for much more. Voglio poi ricordare Lorenzo: di gente come lui ne ho conosciuta veramente poca. R.I.P. I wish to thank all my collaborators, on both sides of the Atlantic Ocean.

Un anno e mezzo l'ho passato con dei grandi compagni di laboratorio : Francesco (quello biondo) è semplicemente una persona meravigliosa, sempre pronto a parlare sia di problemi di ricerca (dai che l'articolo insieme riusciamo almeno a scriverlo, pubblicarlo magari sarà un pò più dura...) che di tutti gli altri problemi; Francesco, Marco, e Paolo, che tra chiaccherate di ricerca, spritz e tramezzini hanno reso la vita nel laboratorio sempre interessante. E non posso certo non ringraziare il "nonno" Alberto e i giovani Michele e Alberto, e Gianfranco, Enoch e Carlo. Tra tutti gli amici con cui ho condiviso questi anni, Carlo e Ciro meritano una menzione particolare: se dopo almeno 25 anni siamo ancora così uniti un motivo ci sarà. My life in Providence would have been sad, and I would not have enjoyed going out for dinners and drinks, without Matteo, Carleton, Olya, Aparna, Aggeliki, Alex, Wenjin, Babis.

Un grazie va alla mia famiglia: Carlo e Cinzia, Floriano e Sandra, Andrea e Mirko, Antonella e Martina e Francesca. Non credo riuscirò mai a ripagarvi per tutto quello che avete fatto per me, mai. Grazie ad Irene, per avermi accolto come un figlio.

La ricerca è stata al centro di questi tre anni e mi ha portato qualche dolore e molta gioia. Ma se posso definirmi felice lo devo solo a Cinzia: grazie, senza te non saprei sorridere.

Questi tre anni di dottorato mi hanno dato molto, dal punto di vista professionale ed umano, ma hanno lasciato anche un grande vuoto. Questa tesi è per un angelo di nome Eufrasia: ti devo tutto. R.I.P.

Contents

Chapter 1

Introduction

We are living in the information era. Recent advances in technology allow for the collection and storage of vast amounts of data in areas ranging from market basket analysis and supply chain management to computational molecular biology and epidemiology. A 2003 study [LV03] reported that between 3 and 6 exabyte (EB, 10^{18} bytes) of newly produced information has been *stored* in 2002, and that the storage of new information has been growing at a rate of more than 30% a year.

Computer science has been dealing with problems related to the ever increasing need for collection and storage of data for decades (e.g., the LZ77 algorithm by Lempel and Ziv has been published in 1977 [ZL77]), developing tools which constitute now a solid, accessible ground for managing large datasets, and the improvement of these tools is still object of research. Given the ubiquity of large datasets, and the need not only to transmit, archive, and compress them, but also to analyze and understand their content, the challenge of our era is the extraction of useful information from overwhelming amounts of data. Even if a huge body of research has been produced on the processing of large datasets, much work remains to be done. It is possible to find a piece of data in a petabyte-size storage system, but analyzing an entire dataset to find correlations and meaningful trends remains challenging. On the one hand there is the need to improve the efficiency of many of the algorithms designed for vast amounts of data, but on the other hand there is the need for novel algorithmic solutions for more effectively extracting significant information from the data.

Data mining is the process of discovering new and useful information. The community of data mining researchers has developed in recent years a set of techniques that has led to great improvement in the analysis of vast amount of data, but the task of analyzing that data is still a major challenge, and in particular assessing the significance of discovered patterns and the validity of forecast based on these discoveries is becoming a major challenge in data intensive applications. The objective of this thesis is the development of rigorous and efficient techniques for mining significant patterns in the context of three specific and important scenarios, as explained below.

First we considered the classical problem of mining frequent itemsets from transactional datasets, a fundamental primitive for market basket analysis and several other commercial and scientific applications. Given a set of transactions, that are subsets of a base set of items, the traditional definition of the problem requires to produce in output all the sets of items (*itemsets*), that appear in at least a fraction f of the transactions, where f is a *frequency* threshold defined by the user. Since the number of transactions is fixed, specifying a minimum frequency threshold f is equivalent to specify a minimum *support* threshold σ, where the support of an itemset is the number of transactions in which the itemset appears. This definition reflects the idea that the significance of an itemset is revealed by its frequency. A huge body of algorithmic studies has been produced for the classical problem. However, the choice of a suitable frequency threshold is usually problematic, and unless specific domain knowledge is available, this choice is often arbitrary. One of the problems of this arbitrariness is that the number of patterns obtained can be either too high or too low, requiring then more iterations of the mining process to obtain a tractable and useful number of patterns in output. Even worse, an arbitrary choice of σ can lead to an high number of false positive or false negative discoveries, that can undermine the correctness of subsequent analyses based on frequent itemset mining.

The set of frequent itemsets usually contains a lot of redundant information. To reduce this redundancy, the mining of frequent *closed* itemset have been proposed. An itemset is closed if any itemset obtained adding an item to it has a lower frequency. The set of frequent closed itemsets is a compact representation of the information contained in the set of frequent itemsets, since from the closed itemsets and their frequency it is possible to recover all frequent itemsets and their frequency. This variation however does not solve the problem of the choice of the minimum threshold σ, which remains problematic.

Recently in [WHLT05] a method has been proposed that does not require a minimum threshold in input, but, rather, extracts the top-K most frequent closed itemsets, that is, the closed itemsets that are frequent w.r.t. a threshold σ_K, defined as the maximum frequency threshold resulting in at least K closed itemsets in output. In this way it is possible to better control the size of the output through the parameter K, even if it is possible that more than K closed itemsets are produced in output. Moreover this approach can be considered as an enhancement of the traditional framework where the significance of an itemset is not merely determined by the comparison of its frequency with an arbitrarily fixed threshold but it is related to its position in a frequency-based ranking of all itemsets.

For the problem of frequent itemset mining, this thesis work contributes the following results:

(i) We study the basic primitive of the extraction of top-K frequent closed itemsets. For the extraction of top-K frequent closed itemsets, we provide the first analytical evidence of its effectiveness, proving a tight upper bound on the ratio between the actual number of closed itemsets returned in output and the input value K. Then, we develop an efficient algorithm for mining top-K frequent closed itemsets in order of decreasing support, which exhibits consistently better performance than the best previously known one, attaining substantial improvements in some cases. A distinctive feature of our algorithm is that it allows the user to dynamically raise the value K with no need to restart the computation from scratch. These results appeared in [PV07].

(ii) We study a second primitive, the use of sampling to extract the top-K frequent itemsets. Traditional methods for the extraction of frequent itemsets work on the entire dataset. Since the size of the dataset can be huge, processing the entire dataset can require too many resources in terms of both space and time, resulting in a mining process computationally too expensive. To overcome this problem one natural approach is to work only on a small sample of the entire dataset. Sampling has been used extensively to extract items/itemsets in the traditional framework, but its use for the extraction of the most frequent itemsets is instead not well studied. We provide a tight bound on the sufficient sample size required to approximate the top-K frequent items/itemsets while giving probabilistic guarantees on the quality of the output. Then, we develop an algorithm to efficiently extract the top-K frequent items/itemsets through sampling. These results have been presented in [PRUV09]

(iii) We develop a novel methodology to identify a meaningful support threshold σ^* for a dataset, such that the number of itemsets with support at least σ^* represents a substantial deviation from what would be expected in a random dataset with the same number of transactions and the same individual item frequencies. The threshold σ^* is chosen in such a way to guarantee that the frequent itemsets with respect to σ^* can then be flagged as statistically significant with a small *False Discovery Rate* (FDR), that is the expected ratio of false discoveries among all discoveries. A crucial feature of our approach is that, unlike most previous work, it takes into account the entire dataset rather than individual discoveries. It is therefore better able to distinguish between significant observations and random fluctuations. These results have been published in [KMP$^+$09a, KMP$^+$09b].

As a second scenario, we considered the mining of patterns, called *motifs*, which occur frequently, possibly with some errors, in biological sequences (e.g., DNA sequences). The discovery of frequent motifs has attracted wide interest in recent years, since sequence similarity in biological molecules (DNA, RNA, amino acids sequence of proteins) is often a necessary condition for functional correlation. The presence of errors in the repetition of a motif are often modeled through the use of the *don't care* character in certain positions, which is a wild card matching all characters of the alphabet. Since the set of frequent motifs contains a lot of redundancy, the notion of *maximal motif* (the analogous of closed itemset for sequences) has been introduced to produce a more compact representation without losing information.

Traditionally the significance of a motif has been assessed using its frequency. However the significance of a motif cannot be exclusively related to its frequency, as the following simple experiment taught us. We extracted the $10,000$ most frequent maximal motifs obtained from *Human Glutamate Metabotropic Receptors* HGMR 1 (410277 bps) and HGMR 5 (91243 bps) sequences, and asked a biologist to verify if there were biologically interesting motifs. The biologist immediately discarded our results as non interesting, since the motifs we reported were either too short or contained too many don't cares. Then the frequency of a motif does not reflect its biological significance, and some of the frequent motifs can be immediately flagged as non significant simply looking at their structure.

For this problem, the thesis contributes the following result:

(i) We develop, analyze and experiment with a new tool, called MADMX, which extracts frequent motifs, possibly including don't care characters, from biological sequences. We introduce *density*, a simple and flexible measure for bounding the number of don't cares in a motif, defined as the ratio of solid (i.e., different from don't care) characters to the total length of the motif. By extracting only *maximal dense motifs*, MADMX reduces the output size and improves performance, while enhancing the quality of the discoveries. The efficiency of our approach relies on a newly defined combining operation, dubbed *fusion*, which allows for the construction of maximal dense motifs in a bottom-up fashion, while avoiding the generation of nonmaximal ones. We provide experimental evidence of the efficiency and the quality of the motifs returned by MADMX, comparing them with the known biological repetitions available in a very popular genomic database, and with the motifs extracted by the recently developed tool VARUN [ACP09] using the same statistical metric employed in [ACP09] for assessing their relative significance. These results have been published in [GPP+09].

Finally, we turned our attention to the mining of significant patterns from large-scale gene and protein interaction networks. This problem is of great interest in the study of cancer, since it is a disease caused mainly by *somatic mutations*, changes in DNA sequence that accumulate during the lifetime of an individual and are not inherited from parents. When a mutation appears in a *gene*, the portion of the DNA that contains the information useful to produce the corresponding *protein*, it can alter the functionality of the protein produced. Proteins are the primary components of living things. Since it is the interaction of the proteins that regulates the activity of a cell and the processes occurring inside it, changes in the functionality of a protein can disrupt the correct functioning of the cell, leading to cancer.

While few of the genes that, when altered, promote the development of malignancies, called *cancer genes*, are mutated at high frequency (e.g. well known cancer genes like TP53 or KRAS), most cancer genes are mutated at much lower frequencies. Thus, the observed frequency of mutation is an inadequate measure of the importance of a gene, particularly with the relatively modest number of samples that are tested in current cancer studies. In fact cancer is a disease of *pathways*, sequences of interactions between proteins that regulate the processes inside the cell. It is hypothesized that somatic mutations target genes in a relatively small number of regulatory and signaling pathways [HW02, VK04]. Thus, the fact that only few genes are mutated in a large number of samples is explained by the fact that there is a huge number of possible combinations of mutations that transform a normal cell into a cancer cell. To understand what are the mechanisms leading to cancer, and what are the genes whose alterations are the cause of malignancies, it is then crucial to find what are the pathways that are significantly mutated.

For this part, this thesis work contributes the following result:

(i) We define the problem of identifying *significantly mutated pathways* in large scale gene and protein interaction networks. We introduce a computational framework that is the first, to our knowledge, to demonstrate a computationally efficient strategy for *de novo* identification of *statistically significant* mutated subnetworks. We propose two algorithms to identify significantly mutated pathways, both based on an *influence* measure between pairs of genes obtained using a diffusion process defined on the interaction network. Moreover, building on the technique we developed in [KMP+09a] we derive a statistical test that identifies significantly mutated pathways and estimates the FDR of the identified subnetworks. We test these algorithms on a large human protein-protein interaction network using mutation data from recent studies on two different type of cancers (glioblastoma multiforme and lung adenocarcinoma). Our methods successfully recover pathways that are known to be important in the considered cancers, and moreover identify additional pathways that have been implicated in cancer but not previously reported as mutated in these samples. These results appeared in [VUR09, VUR10].

The rest of this thesis is organized as follows. Chapter 2 provides the background for the remaining chapters. Chapter 3 presents the results regarding the extraction of top-K frequent closed itemsets, and the use of sampling to extract the top-K frequent items/itemsets. In Chapter 4 the methodology to identify statistically significant frequent itemsets is introduced. Chapter 5 presents our tool MADMX to extract maximal dense motifs in biological sequences. Chapter 6 introduces the framework to discover significantly mutated pathways in biological networks. Chapter 7 ends the thesis with some concluding remarks.

Chapter 2

Background

This thesis proposes novel solutions to discover significant patterns in different scenarios. In this chapter we provide the background related to the problems addressed in this thesis work, and a survey of previous work. The first three sections provides the background for the first part of the thesis. In particular, in Section 2.1 we introduce the problem of frequent itemsets mining, a problem that has attracted a lot of attention in the data mining community, as testified from the huge body of work produced by the researcher in that field, but for which many interesting questions are still open, like, for example, how to efficiently extract the top-K frequent closed itemsets. Another interesting question that is still open is how to employ sampling to extract the top-K frequent itemsets: the background for this problem is presented in Section 2.2. In Section 2.3 we review the approaches that have been proposed to extract the statistically significant frequent itemsets from a dataset, employing measures different from the frequency to measure the significance of an itemsets.

Section 2.4 and Section 2.5 provide the background for the second part of the thesis, where we turn our attention to two problems in computational biology. In particular, in Section 2.4 we introduce the problem of mining motifs in biological sequences, that is one of the fundamental problems in computational biology. In Section 2.5 we instead define the problem of finding significantly mutated pathways in biological networks, a problem for which no efficient solution as been proposed yet, but that is receiving an increasing attention in the biomedical community given the availability of the first data on large-scale tumors sequencing.

While each section presents a survey of previous work, the works that are closely related to our novel contributions will be reviewed in more details in the respective chapters.

2.1 Mining for frequent itemsets: classical setting

The discovery of frequent itemsets is a fundamental primitive which arises in the mining of association rules and in many other mining problems. The problem has been formally introduced in [AIS93], and is the following: given a (multi)set $\mathcal{D} = \{t_1, t_2, \ldots, t_{|\mathcal{D}|}\}$ of transactions, where each transaction t_j is a subset of a base set of items $\mathcal{I}$, and a minimum threshold σ, produce in output the set $\mathcal{F}(\mathcal{D}, \sigma)$ of *frequent itemsets*, that is all of the (nonempty) subsets $X \subseteq \mathcal{I}$ which appear in at least σ transactions. We use $\|\mathcal{D}\|$ to denote the dataset size, that is, $\|\mathcal{D}\| = \sum_{t \in \mathcal{D}} |t|$. For an *itemset* $X \subseteq \mathcal{I}$ we define its *conditional dataset* $\mathcal{D}_X \subseteq \mathcal{D}$ as the (multi)set of transactions $t \in \mathcal{D}$ that contain X. The number of transactions of $\mathcal{D}_X$ is referred to as the *support* of X w.r.t. D, denoted with $\mathrm{s}_{\mathcal{D}}(X)$, while the quantity $\frac{\mathrm{s}_{\mathcal{D}}(X)}{|\mathcal{D}|}$ is referred to as the *frequency* of X, denoted with $f_{\mathcal{D}}(X)$[1].

Since the pioneering work by Agrawal et al. [AIS93] a vast body of works has appeared in the literature presenting novel algorithmic strategies or clever implementations of known strategies, studying foundational issues, and proposing variants of the problem together with efficient algorithmic solutions. Despite this impressive amount of research, many challenging problems are still open [HCXY07].

One of the problems in the mining of frequent itemsets is that the size of the output can be huge, since the number of frequent itemsets can be exponential in the size of the input. It is thus challenging to choose a threshold σ such that the number of frequent itemsets produced in output is not overwhelming, but still large enough to permit significative analyses. However, the set of *all* frequent itemsets usually contains a lot of redundant information which is partly responsible for their large number. In order to eliminate the redundancy, the notion of frequent *maximal* itemsets [Bay98] has been introduced in [Bay98]: a frequent itemset X is *maximal* w.r.t. a support threshold σ if there is no itemset Y, with $X \subset Y \subseteq \mathcal{I}$, such that $\mathrm{s}(Y) \geq \sigma$. From the set of all frequent maximal itemsets and their supports, it is possible to recover the set of all frequent itemsets, but it is not possible to recover their supports without accessing the input database.

Another alternative that has been proposed is the mining of the set $\mathcal{FC}(\mathcal{D}, \sigma)$ of *frequent closed itemsets* [PBTL99]: an itemset X is *closed w.r.t.* $\mathcal{D}$ if there exists no itemset Y, with $X \subset Y \subseteq \mathcal{I}$, such that $\mathrm{s}_{\mathcal{D}}(Y) = \mathrm{s}_{\mathcal{D}}(X)$. In other words, if X is closed, then adding a single item to X decreases its support. Given a support threshold σ, an itemset X is then *closed frequent* if it is frequent w.r.t. σ, and it is closed.

[1] For simplicity, in what follows we will omit explicit reference to $\mathcal{D}$ in the notation for the support and the frequency, if $\mathcal{D}$ is clear from the context.

For any itemset X, its *closure w.r.t. $\mathcal{D}$*, denoted by $\mathrm{Clo}_{\mathcal{D}}(X)$, is the closed itemset $Y \supseteq X$ such that $Y = \bigcap_{t \in \mathcal{D}_X} t$[2]. From the set of frequent closed itemsets and their supports it is possible to recover the set of all frequent itemsets and their supports without accessing the input database.

It would be impossible to survey here the vast literature on the mining of frequent itemsets, maximal frequent itemsets or frequent closed itemsets. We refer the interested reader to the proceedings of the two recently held editions of the Frequent Itemset Mining Implementations (FIMI) Workshop, which illustrate the state of the art for these problems [GZ03, BGZ04]. Among the many algorithms that have been proposed for extracting frequent maximal or closed itemsets, algorithm LCM, proposed in [UAUA04], is particularly relevant for our purposes. In this work, a conceptual organization of the closed itemsets as nodes of a tree, with support decreasing with increasing depth, is proposed. This organization allows LCM (i) to avoid processing non-closed itemsets, and (ii) to avoid maintaining in memory the frequent closed itemsets discovered before producing them in output, resulting in increased time and space performance. LCM is the first algorithm that exhibited these features. A strategy similar to the one employed by LCM is used in [LOP06].

Although the number of frequent closed itemsets is often much smaller than the number of all frequent itemsets, there are cases when $|\mathcal{FC}(\mathcal{D}, \sigma)|$ is still exponential in $||\mathcal{D}||$. The following example is by Yang [Yan04]: let $\mathcal{I}_{\mathrm{Yang}} = \{a_1, a_2, \ldots, a_n\}$ and let $\mathcal{D}_{\mathrm{Yang}} = \{t_1, t_2, \ldots, t_n\}$ with $t_i = \mathcal{I} - \{a_i\}$, for $1 \leq i \leq n$. Thus, $||\mathcal{D}_{\mathrm{Yang}}|| \in \Theta\left(n^2\right)$. It is easy to see that every itemset $X \subseteq \mathcal{I}_{\mathrm{Yang}}$ is closed and has support $n - |X|$, hence the number of closed itemsets of support at least $\sigma = \lfloor n/2 \rfloor$ is $\sum_{k=1}^{\lceil n/2 \rceil} \binom{n}{k} \in \Omega\left(2^n\right)$.

For a given dataset $\mathcal{D}$ and support threshold σ, it is hard to predict the $|\mathcal{F}(\mathcal{D}, \sigma)|$ or $|\mathcal{FC}(\mathcal{D}, \sigma)|$, and this is a problematic aspect of the classical frequent (closed) itemset mining task. Setting σ too large may exclude interesting itemsets from the output, while setting it too small may yield an impractically large output set. Consequently, a user may have to repeat the mining process several times for different support thresholds until one is found which yields a suitable number of frequent itemsets. To overcome this problem, in [WHLT05] the authors propose to modify the mining task into that of discovering the *top-K frequent closed itemsets*, as defined below.

Definition 2.1. *For a dataset $\mathcal{D}$ and an integer K, define the set of* top-K frequent closed itemsets *(*top-K f.c.i.*, for short) as $\mathcal{FC}_K(\mathcal{D}) = \mathcal{FC}(\mathcal{D}, \sigma_K)$, where σ_K is the maximum value such that $\mathcal{FC}(\mathcal{D}, \sigma_K) \geq K$.*

[2] For simplicity, in what follows the terms closed itemset, and closure will be used without explicit reference to $\mathcal{D}$, if $\mathcal{D}$ is clear from the context.

$\mathcal{D}$	$\mathcal{I}$					
t_1	a_6		a_4			a_1
t_2	a_6		a_4			
t_3	a_6	a_5	a_4	a_3	a_2	
t_4	a_6	a_5				
t_5		a_5		a_3	a_2	
t_6		a_5		a_3		a_1
t_7	a_6	a_5	a_4	a_3	a_2	a_1

(a)

X	s(X)
a_6	5
a_5	5
a_6 a_4	4
a_5 a_3	4
a_6 a_5	3
a_5 a_3 a_2	3
a_1	3

(b)

Figure 2.1: (a) Sample dataset $\mathcal{D}$. (b) Top-5 frequent closed itemsets for $\mathcal{D}$.

The top-5 frequent closed itemsets for a sample dataset $\mathcal{D}$ are shown in Figure 2.1. Note that when mining the top-K frequent closed itemsets the threshold σ_K is not given as part of the input and it is uniquely, although implicitly, defined as a function of K, which sets a more direct constraint on the output size. However, requiring the discovery of all closed itemsets of support at least σ_K may yield many itemsets (of support equal to σ_K) in excess of K, but these extra itemsets are necessary in case other patterns (e.g., association rules) must be derived from the frequent closed itemsets.

2.2 Mining of frequent itemsets through sampling

When dealing with massive datasets, computing the exact set of (maximal/closed) frequent itemsets can be too expensive. If the dataset does not fit completely in main memory, disk accesses may slow down exact algorithms to a point where they become impractical. Algorithms for the standard frequent itemset mining task developed to solve the problem in an exact way must scan the entire dataset, typically several times, which has a considerable impact on performance. It is then necessary to accept a tradeoff between the accuracy of the results and the time needed to compute them, especially if it is possible for the user of the algorithm to specify the maximum decay in the “quality” of the output she is willing to accept.

Sampling is one technique that can be employed to reduce the running time, obtaining approximated results. Almost immediately after the first efficient algorithms had been developed, the data mining community started wondering whether it would be possible to lower the execution time by using only a sample of the dataset and give probabilistic guarantees on the output.

One of the first problems that has been addressed by the community is the determination of a sufficient sample size which would allow the sample to respect some “quality standards”. The authors of [ZPLO97] focused on the use of Chernoff bounds to define these standards in terms of *accuracy*, that is, the ratio between the support of an itemset in the sample and its real support, and of *confidence* of the sample, that is, the probability that the itemsets extracted from the sample have a given accuracy. There are two drawbacks in the approach of [ZPLO97]. First of all, the sample size obtained with their method can be larger than the original dataset; second, their approach is not sound from a statistical point of view since the confidence bound is derived for one individual itemset, rather than the entire output set. A straightforward correction of this problem would result in an even worse sample size.

In [JL96] the use of progressive sampling and learning curves is proposed for data mining tasks. Their article refers principally to classification, but the ideas presented can be adapted to the mining of frequent itemsets. The main idea is the use of learning curves to evaluate whether the distribution of elements in the sample is approximately the same distribution of the elements as in the original dataset. This approach could solve the issue of having a sample size larger than the size of the original dataset. The experimental results presented in that work suggest that using progressive sampling can be more efficient than static sampling since it may yield higher accuracy.

An algorithm inspired by the progressive sampling approach presented in [JL96] is introduced in [CHS02]. The main idea is to derive a small sample that reflects some properties of the entire dataset starting from a large, hence more accurate, sample. The algorithm considers at the beginning a large sample $\mathcal{S}_0$, from which an accurate estimation of the frequent items can be derived. Then a small final sample $\mathcal{S}$ of fixed size n, where n is chosen by the user, is obtained by trimming $\mathcal{S}_0$. The transactions removed in the trimming phase are chosen so that the set of frequent items in $\mathcal{S}$ is close to the set of frequent items in $\mathcal{S}_0$, given a suitable distance function between two sets of frequent items.

Another algorithm that starts from the ideas presented in [JL96] is described in [Par02]. The goal of this algorithm is to identify the knee of the learning curve using basic slope characterization across recently evaluated samples. To this end, progressive sampling is employed: starting from a small sample, larger and larger samples are considered. A self-similarity measure is defined between subsets of frequent itemsets obtained from two different samples and is used to stop the growth of the sample size when it becomes small enough. The subset of frequent itemsets considered for the self-similarity measure is such that the mining process is not too expensive. In that paper the accuracy and confidence proposed method is not assessed analytically, but experimental evidence of its effectiveness is provided.

The authors of [LG04] derive a sufficient sample size based on central limit theorem. The sample sizes derived with this method are smaller than the ones derived using the method of [ZPLO97], but the analysis suffers from the same statistical weaknesses as [ZPLO97].

The question of deriving a sufficient sample size for sampling is not the only one that has been addressed by the data mining community. In [Toi96] the author develops and analyzes an algorithm that with one pass of the entire dataset extracts the entire set of frequent itemsets with probability $1 - \Delta$, where Δ is a user defined parameter. The algorithm uses a sample to extract a set $\mathcal{C}$ of itemsets that represents the candidate set of frequent itemsets w.r.t. the entire dataset, and then one scan of the entire dataset is performed to compute the exact frequencies of itemsets in $\mathcal{C}$. The author shows that if some frequent itemset is not found in the first pass (event that holds with probability Δ), an additional pass is sufficient to complete the identification of all frequent itemsets.

The literature related to the problem of finding the top-K frequent items or itemsets by limiting the access to the dataset is not as rich as the one on the classical problem. Some papers [CCFC04, MAA05, CGK08] appeared in the field of data streams and limited to the case of top-K items, while [WF06] deals with top-K itemsets. In the data stream scenario, the transactions are provided to the algorithm one after the other, and it not possible to maintain all the input dataset in memory, then when a transaction is provided to the algorithm, it must decide whether to store it in memory, having then the possibility to use it for the computation, or not. In the data stream scenario the question of major interest is the total space required to solve the problem, hence the authors of works above were mainly concerned with bounding the space needed to compute a solution to the problem or to one of its relaxed versions, and little attention was given to how much data must be sampled to obtain such a solution, since such a question is less crucial in the data stream setting. However, some of these works are of interest because they formally define an approximation to the set of top-K items/itemsets.

The authors of [CCFC04] present a 1-pass algorithm to estimate the most frequent items in a data stream under the constraint of limited storage space. They present an algorithm, CountSketch, which is proved to solve the problem with probability $1 - \delta$ using $O\left(K \log \frac{n}{\delta}\right)$ space, where n is the total number of elements in the stream (i.e., n is the length of the stream), while to obtain a set of items such the the k most frequent items occur in the set a sample of size $O\left(\frac{\log K}{f_K}\right)$, where f_K is the frequency of the K-th most frequent item, is required with a naïve approach (by keeping a uniform random sample of the elements as a list of items and a count for each of them). Since

$f_K \leq 1/n$, the improvement obtained with the CountSketch algorithm is large. A drawback of the CountSketch algorithm is that the parameters of the data structure employed by the algorithm depend on the distribution of the frequencies of the items, so one must have some prior knowledge about that distribution to correctly apply the method.

The authors of [WF06] use the Chernoff bounds to derive a method to mine the top-K frequent itemsets from a datastream. This method seems promising because it gives a probabilistic lower bound to the frequency in the sample of the K-th most frequent itemset in the dataset. The problem is that the proof of this bound contains a flaw, which leads to the non-correctness of the entire algorithm. In particular the authors derive the lower bound to the frequency of the real top-K frequent itemset using a confidence interval for the frequency in the sample of the K-th most frequent itemset in the dataset, without conditioning on the fact that the itemset used to derive this lower bound is *observed* with a certain frequency in the sample.

To understand why this is not correct, consider a dataset where all the items have the same frequency. Using a ball and bins argument it is easy to show that in a random sample there will be an item with frequency f much higher than expected, such that the probability of observing *that particular* item with frequency f in a random sample is negligible. Then the frequency of this item cannot be used to obtain a probabilistic lower bound to the frequency of the most frequent item.

2.3 Statistically significant frequent itemsets

Of the many problems that remain open concerning the mining of frequent itemsets, assessing the significance of the discovered itemsets, or equivalently, flagging statistically significant discoveries with a limited number of false positive outcomes, is still poorly understood and remains one of the most challenging problems in this area [HCXY07]. Since we are inter

The classical framework requires that the user decide what is significant by specifying the support threshold σ. Unless specific domain knowledge is available, the choice of such a threshold is often arbitrary [HK01, TSK06], and may lead to a large number of spurious discoveries that would undermine the success of subsequent analysis.

A number of works have explored various notions of significant itemsets and have proposed methods for their discovery. Below, we review those most relevant to this thesis work and refer the reader to [HCXY07, Section 3] for further references. The paper [AY98] relates the significance of an itemset X to the quantity

$((1 - v(X))/(1 - \mathbf{E}[v(X)])) \cdot (\mathbf{E}[v(X)]/v(X))$, where $v(X)$ represents the fraction of transactions containing some but not all of the items of X, and $\mathbf{E}[v(X)]$ represents the expectation of $v(X)$ in a random dataset where items occur in transactions independently. This ratio provides an empirical measure of the correlation among the items of X which, according to [AY98], is more effective than absolute support. In [SA96, DuM99, DP01], the significance of an itemset is measured as the ratio R between its actual support and its expected support in a random dataset. In order to make this measure more accurate for small supports, [DuM99, DP01] proposes smoothing the ratio R using an empirical Bayesian approach. Bayesian analysis is also employed in [ST96] to derive subjective measures of significance of patterns (e.g., itemsets) based on how strongly they "shake" a system of established beliefs. In [JS05], the significance of an itemset is defined as the absolute difference between the support of the itemset in the dataset and the estimate of this support made from a Bayesian network with parameters derived from the dataset.

A statistical approach for identifying significant itemsets is presented in [SBM98], where the measure of interest for an itemset is defined as the degree of dependence among its constituent items, which is assessed through a χ^2 test. Unfortunately, as reported in [DuM99, DP01], there are technical flaws in the applications of the statistical test in [SBM98]. In particular, it is reported that the χ^2 distribution used in their approach has one degree of freedom for any length of considered itemsets, while this is true only for itemsets of size 2. Their results are then correct only for itemsets of size 2. Nevertheless, [SBM98] pioneered the quest for a rigorous framework for addressing the discovery of significant itemsets.

A common drawback of the aforementioned works is that they assess the significance of each itemset *in isolation*, rather than taking into account the *global* characteristics of the dataset from which they are extracted. As argued before, if the number of itemsets considered by the analysis is large, even in a purely random dataset some of them are likely to be flagged as significant if considered in isolation. A few works attempt at accounting for the global structure of the dataset in the context of frequent itemset mining. The authors of [GMMT07] propose an approach based on Markov chains to generate a random dataset that has identical transaction lengths and identical frequencies of the individual items as the given real dataset. The work suggests comparing the outcomes of a number of data mining tasks, frequent itemset mining among the others, in the real and the randomly generated datasets in order to establish whether the real datasets exhibit any significant global structure. However, such an assessment is carried out in a purely qualitative fashion without rigorous statistical grounding.

Multi-hypothesis testing

In a simple statistical test, a null hypothesis H_0 is tested against an alternative hypothesis H_1. A test consists of a rejection (critical) region C such that, if the statistic (outcome) of the experiment is in C, then the null hypothesis is rejected, and otherwise the null hypothesis is not rejected. The *significance level* of a test, $\alpha =$ Pr(Type I error), is the probability of rejecting H_0 when it is true (*false positive*). The *power* of the test, $1 -$ Pr(Type II error), is the probability of correctly rejecting the null hypothesis. A “Type II error” is the erroneous non rejection of a null hypothesis (*false negative*). The *p-value* of a test is the probability of obtaining an outcome at least as extreme as the one that was actually observed, under the assumption that H_0 is true.

In a multi-hypothesis statistical test, the outcome of an experiment is used to test simultaneously a number of hypotheses. For example, in the context of frequent itemsets, if we seek significant k-itemsets, we are in principle testing $\binom{n}{k}$ null hypotheses simultaneously, where each null hypothesis corresponds to the support of a given itemset not being statistically significant. The experiment in this case corresponds to the extraction of the k-itemsets and their supports from the datasets. In the context of multi-hypothesis testing, the significance level cannot be assessed by considering each individual hypothesis in isolation. To demonstrate the importance of correcting for multiplicity of hypotheses, consider a simple real dataset of 1,000,000 transactions over 1,000 items, each with frequency 1/1000. Assume that we observed that a pair of items (i, j) appears in at least 7 transactions. Is the support of this pair statistically significant? To evaluate the significance of this discovery we consider a random dataset where each item is included in each transaction with probability 1/1000, independent of all items. The probability that the pair (i, j) is included in a given transaction is 1/1,000,000, thus the expected number of transactions that include this pair is 1. A simple calculation shows that the probability that (i, j) appears in at least 7 transactions is about 0.0001. Thus, it seems that the support of (i, j) in the real dataset is statistically significant. However, each of the 499,500 pairs of items has probability 0.0001 to appear in at least 7 transactions in the random dataset. Thus, even under the assumption that items are placed independently in transactions, the expected number of pairs with support at least 7 is about 50. If there were only about 50 pairs with support at least 7, returning the pair (i, j) as a statistically significant itemset would likely be a false discovery since its frequency would be better explained by random fluctuations in observed data. On the other hand, assume that the real dataset contains 300 disjoint pairs each with support at least 7. By the Chernoff bound [MU05], the probability of that event in the random dataset is less than 2^{-300}. Thus, it is very likely that the support of most of these pairs would be statistically significant. A discovery process that does not return these pairs will result in a large number of false negatives.

A natural generalization of the notion of significance level to multi-hypothesis testing is the *Family Wise Error Rate (FWER)*, which is the probability of incurring at least one Type I error in any of the individual tests. If we are testing simultaneously m hypotheses and we want to bound the FWER by α, then the Bonferroni method tests each individual null hypothesis with significance level α/m. While controlling the FWER, this method is too conservative in that the power of the test is too low, resulting in many false negatives. There are a number of techniques that improve on the Bonferroni method, but for large numbers of hypotheses all of these techniques lead to tests with low power (see [DSB03] for a good review).

The *False Discovery Rate (FDR)* was suggested by Benjamini and Hochberg [BH95] as an alternative, less conservative approach to control errors in multiple tests. Let R be the total number of null hypotheses rejected by the multiple test, and let V be the number of Type I errors among these rejections. Then we define FDR to be the expected ratio of erroneous rejections among all rejections, namely FDR $= E[V/R]$, with $V/R = 0$ when $R = 0$. Designing a statistical test that controls for FDR is not simple, since the FDR is a function of two random variables that depend both on the set of null hypotheses and the set of alternative hypotheses. Building on the work of [BH95], Benjamini and Yekutieli [BY01] developed a general technique for controlling the FDR in any multi-hypothesis test (see Theorem 4.5).

Few works employ the multi-hypothesis testing framework for frequent itemset mining or in the realm of discovering association rules. The problem of spurious discoveries when mining significant patterns is studied in [BHA02]. The paper is concerned with the discovery of significant pairs of items, where significance is measured through the p-value, that is, the probability of occurrence of the observed support in a random dataset. Significant pairs are those whose p-values are below a certain threshold that can be suitably chosen to bound the FWER, or to bound the FDR. The authors compare the relative power of the two metrics through experimental results, but do not provide methods to set a meaningful support threshold. In [HN08], the authors provide a variation of the well-known Apriori strategy for the efficient discovery of a subset $\mathcal{A}$ of association rules with p-value below a given cutoff value, while the results in [MS98] provide the means of evaluating the FDR in $\mathcal{A}$. The FDR metric is also employed in [ZPT04] in the context of discovering significant quantitative rules, a variation of association rules. None of these works is able to establish support thresholds such that the returned discoveries feature small FDR.

2.4 Mining of motifs in biological sequences

All of the genetic information in any living creature is stored in *deoxyribonucleic acid (DNA)* and *ribonucleic acid (RNA)*, which are polymers of four simple nucleic acid units, called nucleotides. The portions of the DNA that really contains the information necessary for the correct functioning of the cell are called *genes.* Each gene codifies the information to produce a *protein*, the final product of *genetic expression.* In particular, the process of genetic expression starts from the DNA sequence of a gene. Using the information coded into the gene, an RNA molecule is produced through the process of *transcription*, and then the *amino acids* sequence that constitute the protein corresponding to the starting gene is produced through *translation* of the RNA molecule. The final step of the genetic expression is the *folding* of the protein into its three-dimensional structure.

The discovery of frequent patterns (*motifs*) in biological sequences has attracted much interest in recent years, due to the understanding that sequence similarity is often a necessary condition for functional correlation. For example since the structure of a protein is determined by the sequence of the corresponding gene, genes that have a similar sequence will likely produce proteins sharing similar structure and thus probably having similar functions.

Among other applications, motif discovery proves an important tool for identifying *regulatory regions* and *binding sites* in the study of functional genomics. Regulatory regions are segments of DNA where proteins that regulates the transcription process binds preferentially, and are thus involved in the control of gene expression. A binding site is a region of a protein, DNA (or RNA) to which specific other molecules form a chemical bond. For example, a *transcription factor binding site* is the portion of DNA to which a protein (called *transcription factor*) binds controlling the transfer of genetic information from DNA to RNA.

From a computational point of view, a major complication for the discovery of motifs is that they may feature some sequence variation without loss of function. The discovery process must therefore target *approximate motifs*, whose occurrences in the input sequence are similar but not necessarily identical. Approximate motifs are often modeled through the use of the *don't care* character in certain positions, which is a wild card matching all characters of the alphabet, called *solid characters* [Par07].

Finding interesting approximate motifs is computationally challenging. As the number of don't cares increases and/or the minimum frequency threshold decreases, the output may explode combinatorially, even if the discovery targets only maximal motifs—a subset of the motifs which succinctly represents the complete set. Moreover, even when the final output is not too large, partial data during the inference of target motifs might lead to memory saturation or to extensive computation during the intermediate steps.

A large body of literature in the last decade has dealt with efficient motif discovery [Par00, AP04, PCGS05, Ukk07, MNU08, AU07, AT08, ACP09, AT07], and an excellent survey of known results can be found in the book [Par07]. In order to alleviate the computational burden of motif extraction and to limit the output to the most promising or interesting discoveries, some works combine the traditional use of a frequency threshold with restrictions on the flexibility of the extracted motifs, often captured by limitations on the number of occurring don't cares.

Traditionally, the significance of a motif has been assessed through its frequency. To understand if there is a direct correlation between frequency and biological significance, we extracted the $10,000$ most frequent motifs obtained from *Human Glutamate Metabotropic Receptors* HGMR 1 (410277 bps) and HGMR 5 (91243 bps) sequences, and asked a biologist to verify if there were biologically interesting motifs. The biologist immediately discarded our results as non interesting, since the motifs we reported were either too short or contained too many don't cares. Then the frequency of a motif does not reflect its biological significance. Other then the frequency, a number of different statistics have been employed to measure the significance of a motif (see [FA07] for a comparison of these measures). However, to find the most significant motifs under one of those measures, the first step is the extraction of all motifs, since there no strategy has been proposed to directly extract significant motifs under those measures.

In a recent work, Apostolico et al. [ACP09] study the extraction of *extensible motifs*, comprising standard don't cares and extensible wild cards. The latter are spacers of variable length that can take different size (within pre-specified limits) in each occurrence of the motif. An efficient tool, called VARUN, is devised in [ACP09] for extracting all maximal extensible motifs (according to a suitable notion of maximality defined in the paper) which occur with frequency above a given threshold σ and with upper limits D on the length of the spacers. VARUN returns the extracted motifs sorted by decreasing z-score, that is the measure of the distance in standard deviations of the outcome of a random variable from its deviation. The authors demonstrate the effectiveness of their approach both theoretically, by proving that each maximal motif features the highest z-score within the class of motifs it represents, and experimentally, by showing that the returned top-scored motifs comprise biologically relevant ones when run on protein families and DNA sequences.

A slightly more general way of limiting the number of don't cares in a motif has been explored in [RF98]. The authors define $\langle L, W \rangle$ motifs, for $L \leq W$, where at least L solid characters must occur in each substring of length W of the motif. They propose a strategy for extracting $\langle L, W \rangle$ motifs which are also maximal, although their notion of maximality is not internal to the class of $\langle L, W \rangle$ motifs. As a consequence, the algorithm is not complete, since it disregards all those $\langle L, W \rangle$ motifs that are subsumed by a maximal non-$\langle L, W \rangle$ one.

2.5 Mining of significantly mutated pathways in biological networks

Cancer is a disease that is largely driven by *somatic mutations*, changes in DNA sequence not inherited from parents that accumulate during the lifetime of an individual. When a mutation appears in a gene, it can alter the three-dimensional structure of the corresponding protein, affecting its functionality. Since it is the interaction of the proteins that regulates the activity of a cell and the processes occurring inside it, changes in the functionality of a protein can disrupt the correct functioning of the cell, leading to cancer.

Decades of experimental work have identified numerous cancer-promoting oncogenes (also called *cancer genes*) and tumor suppressor genes that are mutated in many types of cancer. Recent cancer genome sequencing studies have dramatically expanded our knowledge about somatic mutations in cancer. For example, large projects like The Cancer Genome Atlas (TCGA) [Net08], the Tumor Sequencing Project (TSP) [D$^+$08], and the Cancer Genome Anatomy Project [G$^+$07] have sequenced hundreds of protein coding genes in hundreds of patients with a variety of cancers. Other efforts have taken a global survey of approximately 20,000 genes in a 1-2 dozen patients [W$^+$07, J$^+$08, P$^+$08]. These studies have shown that: (i) tumors harbor on average less than 100 somatic mutations; (ii) different tumors rarely have the same set of mutations; (iii) and thousands of genes are mutated in at least one type of cancer [W$^+$07]. This mutational heterogeneity complicates efforts to distinguish functional mutations, that alter the three-dimensional structure of the protein from sporadic, passenger mutations that do not cause cancer. While a few cancer genes are mutated at high frequency (e.g. well known cancer genes like TP53 or KRAS), most cancer genes are mutated at much lower frequencies. Thus, the observed frequency of mutation is an inadequate measure of the importance of a gene, particularly with the relatively modest number of samples that are tested in current cancer studies.

It is widely accepted that cancer is a disease of *pathways*: a pathway is a sequence of interactions between proteins that can convert one kind of signal or stimulus received from a cell into another (*signaling pathway*) or that can regulate the rates at which other proteins will be produced (*regulatory pathway*). The entire set of (pairwise) interactions between proteins defines the *interaction network* of proteins, and a pathway is a subnetwork of this large interaction network. It is hypothesized that somatic mutations target genes in a relatively small number of regulatory and signaling pathways [HW02, VK04]. Thus, the observed mutational heterogeneity is explained by the fact that there are myriad combinations of mutations that cancer cells can employ to perturb the behavior of these key pathways. The unifying themes of cancer are thus not solely revealed by the individual mutated genes, but by the interactions between these genes. Standard practice in cancer sequencing studies is to assess whether genes that are mutated at sufficiently high frequency significantly overlap known cancer pathways [Net08, D$^+$08, S$^+$06, W$^+$07, P$^+$08, L$^+$07a]. For example, the TCGA study of glioblastoma multiforme (GBM) [Net08] reported that three pathways previously identified as important in GBM were somatically mutated in a large percentage of samples. This result confirms the role of these pathways in GBM, but does not show whether these pathways were the only ones with a surprising pattern of mutation.

Finding significant overlap between mutated genes and genes that are members of known pathways is an important validation of existing knowledge. However, restricting attention to these known pathways does not allow one to detect novel groups of genes that are members of less characterized pathways. Moreover, it is well known that signal components in signal transduction can be shared between between different signaling pathways, and thus responses to a signal inducing condition can activate multiple responses in a cell [ZPZ$^+$09, VK04], a phenomenon called *crosstalk*. Dividing genes into discrete pathway groupings limits the ability to directly detect whether this crosstalk is a target of mutations. An additional source of information about gene and protein interactions is large-scale interaction networks, such as the Human Protein Reference Database (HPRD) [P$^+$09], STRING [J$^+$09], and others [B$^+$01, SMS$^+$04]. These resources incorporate both well-annotated pathways and interactions derived from less specific and accurate methods, like high-throughput experiments, automated literature mining, cross-species comparisons, and other computational predictions. Many researchers have used these interaction networks to analyze gene expression data. Ideker et al. [IOSS02] introduced a method to discover subnetworks of *differentially expressed genes*, that are genes whose expression is different in cancer and normal samples. This idea was later extended in different directions by others [NCTLH07, L$^+$07b, UKS08, KSS09, MLWS07, HLCS09, CLL$^+$07]. Separately, [LACB09] defined metrics that showed clustering of GO annotations [A$^+$00] on an interaction network.

To our knowledge, no algorithm has been hitherto proposed to identify *significantly mutated pathways* – that is connected subnetworks whose genes have more mutations than expected by chance – *de novo* in a large interaction network. This problem differs from the detection of subnetworks of differentially expressed genes in that a relatively small number of genes might be measured, a small subset of genes in a pathway may be mutated, and that a single mutated gene may be sufficient to perturb a pathway.

Chapter 3

Algorithmic Aspects of Basic Mining Primitives

In this chapter we study the algorithmic aspects of two basic mining primitives: the extraction of top-K frequent closed itemsets, and the use of sampling to extract the top-K frequent items/itemsets. These primitives are used in many data mining problems and are the first attempt to overcome the traditional view of the frequency of an itemsets as a direct measure of its significance. In fact, if we are interested in the top-K frequent items/itemsets, we are assuming that the significance of a pattern is not given only by its frequency, but that it is the *ranking* given by the frequency of the itemsets that reflects their significance.

As explained in Chapter 2, the extraction of top-K frequent closed itemsets is a recently proposed alternative to the classical frequent itemset mining, whose purpose is to provide better control on the output size by making the frequency threshold dependent on a parameter K which represents an approximate estimate of the number of returned itemsets, rather than leaving the frequency threshold as an independent input parameter which may be hard to fix.

Sampling is one techniques that can be used to improve the performances of frequent itemset mining problems at the cost of obtaining approximated results, as seen in Chapter 2. In particular, sampling can be use to guarantee certain quality requirements on the output when extracting the top-K frequent items/itemsets.

The chapter is organized as follows. In Section 3.1 we present our work on the discovery of top-K frequent closed itemsets. Our contribution for this problem is twofold. First, we prove a tight upper bound on the ratio between the actual number of closed itemsets returned in output and the input value K, thus providing the first analytical evidence of the effectiveness of the new approach. Second, we develop a new algorithm for mining top-K frequent closed itemsets, which features a tight bound on the number of non-frequent itemsets touched during the mining process, and allows the user to dynamically raise the value K without restarting the computation from scratch. We also report the results of extensive experiments showing that our algorithm exhibits consistently better performance than the best previously known one, attaining substantial improvements in some cases. The results of Section 3.1 were published in [PV07]. In Section 3.2 we discuss the use of sampling to extract top-K frequent items/itemsets. We prove a lower bound for the number of transactions that must be considered by *any* algorithm that employs sampling to extract the top-K frequent items/itemsets and produces in output a set satisfying some quality requirements, providing moreover a family of datasets for which this lower bound is tight. Moreover, we design a new progressive sampling algorithm to efficiently solve the problem. The results of Section 3.2 were presented in preliminary form in [PRUV09].

3.1 Top-K frequent closed itemsets mining

The extraction of *top-K frequent closed itemsets* has been proposed in [WHLT05] to provide the user better control on the size of the output set. For convenience of the reader, we recall the definition of the problem (introduced in Section 2.1). This variation requires that for a given value K, specified as input parameter, all itemsets of support at least σ_K be discovered. σ_K which is uniquely defined by K, is the maximum support threshold that yields at least K frequent closed itemsets. Although one is not guaranteed that top-K frequent closed itemsets are exactly K, it is conceivable that parameter K be more effective than the minimum support threshold in controlling the output size. It is important to remark that the top-K frequent closed itemsets can be employed in every application where frequent closed itemsets are needed.

In [WHLT05] the authors present an efficient algorithm, called TFP, to mine the top-K frequent closed itemsets. The main idea of the algorithm is to use an efficient depth-first mining process starting with an initially low support threshold σ ($\sigma \leq \sigma_K$) which is progressively increased, as the execution proceeds, by means of several effective heuristics, until the final value σ_K is reached. When an itemset is generated it is inserted into a suitable data structure from which it can be removed later and discarded if found to be non-closed or infrequent. TFP has an additional feature which allows the user to specify a minimum length $\min_\ell$ for the closed itemsets to be returned. The authors provide experimental evidence of the efficiency of their algorithm. The main drawbacks of TFP are that no bound is given on the number of non-closed or infrequent itemsets that the algorithm must process, and that an involved itemset closure checking scheme is required. Moreover, TFP does not appear to be able to handle efficiently a dynamic scenario where the user is allowed to raise the value K.

Other works have recently considered different, although somewhat related, problems. In [SM04] the mining the K itemsets of maximum density with respect to a fixed support threshold is studied, where the notion of density relaxes the requirement of strict containment of an itemset in its supporting transactions. The authors propose a priority-queue based approach for solving this problem, which is similar in spirit to the one adopted in our algorithm. The mining of top-K frequent itemsets for every itemset length (i.e., the top-K frequent itemsets of length 1, the top-K frequent itemsets of length 2, and so on) is studied in [FKT00, CF04], and algorithms are proposed based on breadth-first [FKT00] and depth-first [CF04] strategies. A breadth-first algorithm to discover the top-K frequent itemsets without restricting the exploration to the closed ones, is presented in [SSPT98]. The algorithm executes a number of iterations, where in the ℓ-th iteration the K most frequent itemsets of length at most ℓ are discovered.

We contribute the following new results regarding the mining of top-K frequent closed itemsets.

1. We show that the number of top-K frequent closed itemsets can be at most nK, where n is the number of items occurring in the dataset. No such bound was previously known and this provides the first analytical estimate of the effectiveness of parameter K in controlling the output size. We also argue that without the restriction to mining closed itemsets, the ratio between the number of itemsets returned and K can be exponentially large in size of the dataset.

2. We develop a new algorithm, TopKMiner, for discovering top-K frequent closed itemsets, which, unlike algorithm TFP, features a tight bound on the number of itemsets touched during the mining process, and allows the user to dynamically raise the value K without the need to restart the computation from scratch. Also, we experimentally compare the performance of TopKMiner and TFP on both real and synthetic datasets, for different values of K. The results of the experiments show that TopKMiner always exhibits better performance, with substantial improvements in some cases (more than two orders of magnitude). The efficiency of TopKMiner becomes even higher when used in a dynamic scenario where top-K frequent closed itemsets are sought for increasing values of K successively provided by the user.

The rest of the section is organized as follows. Subsection 3.1.1 briefly describes the characteristics of the datasets used in the experiments. The bound on the ratio between the actual number of top-K frequent closed itemsets and the input value K is proved in Subsection 3.1.2. Algorithm TopKMiner is described presenting first its high-level strategy in Subsection 3.1.3 and, then, the most relevant implementation details in Subsection 3.1.4. The results of the experimental comparison between TFP and TopKMiner are reported and discussed in Subsection 3.1.5.

3.1.1 Dataset used in the experiments

The experiments of our work have been conducted on both real and artificially generated datasets available from the FIMI repository[1], which have become standard benchmarks for frequent itemset mining algorithms. In this chapter we report results relative to five of them of large size, which represent the most meaningful and challenging instances for the mining task. These datasets are briefly described below.

T40I10D100K: An artificial dataset obtained using the generator developed in [AS94]. For short, we will often refer to this dataset as T40;

accidents: it is derived from a collection of data relative to traffic accidents;

pos: from Blue-Martini Software Inc., it is derived from several months of click-stream data from e-commerce web sites;

kosarac: it is derived from click-stream data of a hungarian on-line news portal. (In fact, we had to clean up the original instance of the dataset which contained transactions with duplicated item, which is not allowed by the problem's specification.)

webdocs: it is built from a spidered collection of web html documents. More details can be found in [LOPS04].

The table in Figure 3.1 summarizes the main characteristics (number of items, average transaction length, and number of transactions) of the above datasets, while the table in Figure 3.2 reports for each dataset the support threshold σ_K that yields the top-K frequent closed itemsets, for different values of K. For clarity, in the table the frequency value $\sigma_K/|\mathcal{D}|$ rather than the value σ_K is shown.

3.1.2 Tight bound on the output size

In this section we provide the first analytical estimate of the effectiveness of parameter K in controlling the output size when mining the top-K frequent closed itemsets.

[1] `http://fimi.cs.helsinki.fi`

Dataset	#Items	Avg. Trans. Length	# Transactions
T40	1,000	39.5	100,000
accidents	468	33.8	340,183
pos	1,658	7.5	515,597
kosarac	41,270	8.1	990,002
webdocs	5,267,656	177	1,692,082

Figure 3.1: Datasets characteristics

	$\sigma_K/\lvert\mathcal{D}\rvert$		
$\mathcal{D}$	$K = 100$	$K = 1000$	$K = 10000$
T40	0.092	0.027	0.013
accidents	0.820	0.656	0.483
pos	0.036	0.010	0.003
kosarac	0.023	0.006	0.002
webdocs	0.327	0.216	–

Figure 3.2: Values of $\sigma_K/|\mathcal{D}|$ for $K = 100, 1000, 10000$

Since the number of frequent itemsets can be much larger than the number of frequent closed itemsets, when mining the latter it is convenient to avoid processing non-closed itemsets. To this aim, in [UAUA04] the authors propose a conceptual organization of the closed itemsets as nodes of a tree, with support decreasing with increasing depth. Specifically, let $\mathcal{D}$ be a dataset defined over the set of items $\mathcal{I} = \{a_1, a_2, \ldots, a_n\}$ (the indexing of the items is fixed but arbitrary). For an itemset X define its *i-th prefix* as $X(i) = X \cap \{a_j : 1 \le j \le i\}$, for $1 \le i \le n$. The *core index* of a closed itemset X, denoted as $\text{core}_i(X)$, is defined as the minimum i such that $\mathcal{D}_X = \mathcal{D}_{X(i)}$.

Definition 3.1 ([UAUA04]). *A closed itemset X is a* prefix-preserving closure extension (ppc-extension) *of a closed itemset Y if: (1) $X = Clo_{\mathcal{D}}(Y \cup \{a_j\})$, for some $a_i \notin Y$ with $j > core_i(X)$; and (2) $X(j-1) = Y(j-1)$.*

Let $\perp = \text{Clo}_{\mathcal{D}}(\emptyset)$, which is the possibly empty closed itemset consisting of the items occurring in all transactions. The following theorem defines the tree structure over the set of closed itemsets, with $\perp$ being the root of the tree.

Theorem 3.2 ([UAUA04]). *Any closed itemset $X \neq \perp$ is the ppc-extension of exactly one closed itemset Y, and $s(X) < s(Y)$.*

Let $\Delta(n)$ be the family of all datasets $\mathcal{D}$ whose defining set of items $\mathcal{I}$ has size n (we assume that every item in $\mathcal{I}$ occurs in at least one transaction of $\mathcal{D}$). Let also

$$\rho(n, K) = \max_{\mathcal{D} \in \Delta(n)} \frac{\mathcal{FC}_K(\mathcal{D})}{K}.$$

The following theorem establishes the main result of this section.

Theorem 3.3. *For every $n \geq 1$ and $K \geq 1$, we have $\rho(n, K) \leq n$.*

Proof. Consider an arbitrary dataset $\mathcal{D} \in \Delta(n)$ and a value $K \geq 1$. Let $\Phi = \{X_1, X_2, \ldots, X_K\}$ be the set of K most frequent non-empty closed itemsets numbered in decreasing order of support and let $\perp = \mathrm{Clo}_{\mathcal{D}}(\emptyset)$. By Theorem 3.2 we know that any closed itemset $X \notin \Phi$ of support σ_K must be a ppc-extension of some closed itemset $Y \in (\Phi \backslash \{X_K\}) \cup \perp$. The upper bound on $\rho(n, K)$ follows directly from the argument in [BGKM03] which shows that any such itemset Y can generate at most $(n-1)$ ppc-extensions not belonging to Φ. Hence, the number of closed itemsets not included in Φ and of support σ_K is at most $K(n-1)$, which yields $\rho(n, K) \leq n$. □

The lower bound on $\rho(n, K)$ is provided by the dataset described in [Yan04, Section 3.1]. In particular, that dataset shows that $\rho(n, 1) = n$. One may wonder whether for every K it holds that $\rho(n, K) = n$. The following proposition gives a negative answer.

Proposition 3.4. *For any dataset $\mathcal{D} \in \Delta(n)$, if $\mathcal{FC}_K(\mathcal{D})/K = n$ then $K = 1$.*

Proof. Let $\Phi = \{X_1, X_2, \ldots, X_K\}$ be the set of K most frequent non-empty closed itemsets numbered in decreasing order of support and let $\perp = \mathrm{Clo}_{\mathcal{D}}(\emptyset)$. We first show that if $\mathcal{FC}_K(\mathcal{D})/K = n$, then $\Phi \backslash \{X_K\} \subseteq \{a_1\}$. From the proof of Theorem 3.3, to have $\mathcal{FC}_K(\mathcal{D})/K = n$ it necessary that each itemset in $Y \in (\Phi \backslash \{X_K\}) \cup \perp$ generates exactly $n-1$ closed itemsets of support σ_K not in $\Phi \backslash \{X_K\}$ through ppc-extension. Since the ppc-extension is *prefix-preserving*, the only (non-empty) itemset for which this is possible is $\{a_1\}$, that proves $\Phi \backslash \{X_K\} \subseteq \{a_1\}$.

Now we prove that $\{a_1\} \notin \Phi \backslash \{X_K\}$. First of all, notice that it must be $\mathrm{Clo}(\{i_r\}) = \{i_r\}$ for all $i = 1, \ldots, n$, otherwise there would be a closed itemset different from $\{a_1\}$ in $\Phi \backslash \{X_K\}$ that is impossible. (To prove this is sufficient to observe that the intersection of two closed itemsets X, Y is a closed itemset of support $> \max\{s(X), s(Y)\}$, when it is non-empty.)

Since each ppc-extension of $\{a_1\}$ is a superset of at least one $\{a_r\}$ with $r > 1$ and to obtain $\mathcal{FC}_K(\mathcal{D})/K = n$ we need that all the ppc-extensions of $\{a_1\}$ and all the ppc-extension of $\perp$ have frequency σ_K, if $\{a_1\}$ is in $\Phi \backslash \{X_K\}$ we will have two closed itemsets with the same frequency and contained one into the other one, that is impossible. (In particular, the ppc-extension of $\{a_1\}$ using item a_r is a superset of $\{a_r\}$, and these two itemsets cannot have the same frequency.)

This implies that $\Phi \backslash \{X_K\} = \emptyset$, thus $K = 1$. □

		$\mathcal{FC}_K(\mathcal{D})/K$		
$\mathcal{D}$	n	$K = 100$	$K = 1000$	$K = 10000$
T40	1,000	1	1	1.0018
accidents	468	1	1	1
pos	1,658	1	1	1.0003
kosarac	41,270	1	1	1
webdocs	5,267,656	1	1	–

Figure 3.3: Comparison between n and $\mathcal{FC}_K(\mathcal{D})/K$

The proof above moreover implies that if $\mathcal{FC}_K(\mathcal{D})/K = n$, the top-$k$ frequent closed itemsets are $\{a_1\}, \ldots, \{a_n\}$.

The table in Figure 3.3 compares the number of items n against the ratio $\mathcal{FC}_K(\mathcal{D})/K$ for the datasets described in Section 3.1.1 and for different values of K. Note that $\mathcal{FC}_K(\mathcal{D})/K$ is always very close to 1. In fact, we conjecture that when maximized over all datasets over n items, the value $\rho(n, K)$ become a decreasing function of K.

It is important to remark that the result of Theorem 3.3 crucially relies on the fact that the mining task is limited to closed itemsets. Indeed, we could remove the closedness requirement and mine the top-K frequent itemsets, that is, the set $\mathcal{F}_K(\mathcal{D}) = \mathcal{F}(\mathcal{D}, \sigma_K)$, where σ_K, is the maximum value that ensures $|\mathcal{F}(\mathcal{D}, \sigma_K)| \geq K$. In this case, however, the ratio $\mathcal{F}_K(\mathcal{D})/K$ can be exponentially large in the number of itemsets even for non-trivial datasets. To see this, consider the following (nontrivial) example from [UAUA04]. Let $n = 2^d \geq 16$ and let $\mathcal{I}_1$, $\mathcal{I}_2$, and $\mathcal{I}_3$ be three disjoint sets of items of size $n - 2(d+2)$, $d+2$, and $d+2$, respectively. Let also $\mathcal{J}_2$ (resp., $\mathcal{J}_3$) be a family of $n/2 - 1$ distinct subsets of $\mathcal{I}_2$ (resp., $\mathcal{I}_3$) which does not include $\emptyset$ nor $\mathcal{I}_2$ (resp., $\mathcal{I}_3$). Consider the dataset $\mathcal{D}$ over $\mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_3$ comprising the following n transactions:

$$\{\mathcal{I}_1 \cup \mathcal{I}_2 \cup S : S \in \mathcal{J}_3\} \cup \{\mathcal{I}_1 \cup \mathcal{I}_3 \cup S : S \in \mathcal{J}_2\} \cup \{\mathcal{I}_2 \cup \mathcal{I}_3\} \cup \{\mathcal{I}_1\}.$$

$\mathcal{D}$ is non-trivial, in the sense that it contains no duplicated transactions and no item occurs in all transactions. Moreover, it is easy to see that there are $2^{n-2(d+2)} - 1 \in \Theta\left(2^n/n^2\right)$ non-empty itemsets of maximum support $n - 1$, namely all non-empty subsets of $\mathcal{I}_1$. Hence, for $K = 1$, we have $\mathcal{F}_K(\mathcal{D})/K \in \Theta\left(2^n/n^2\right)$.

3.1.3 TopKMiner: main strategy

In this subsection we describe our algorithm *TopKMiner* for mining the top-K frequent closed itemsets from a dataset $\mathcal{D}$, and introduce the algorithm's high-level strategy and its featured characteristics. We let $\mathcal{I} = \{a_1, a_2, \ldots, a_n\}$ denote the set of items and assume that they are ordered by non-decreasing support, that is, item a_n has highest support.

TopKMiner, whose pseudocode is given in Figure 3.4, is based on a conceptually simple strategy, which builds on ideas developed in previous works [PZ03, UAUA04, SM04]. The algorithm receives in input the dataset $\mathcal{D}$ and a value K^* that represents the maximum K for which the user may request the mining of top-K frequent closed itemsets. In other words, the user is allowed to dynamically raise K up to K^*. The algorithm makes use of a priority queue Q whose entries correspond to closed itemsets. Specifically, an entry for a closed itemset Y is a quadruple $(\mathcal{D}_Y, s, i, Y(i-1))$, where $\mathcal{D}_Y$ is the conditional dataset for Y, s its support, i its core index, and $Y(i-1)$ its i-th prefix. Two variables σ and σ' are used to store dynamic approximations from below to σ_{K^*} and σ_K, respectively.

TopKMiner starts by asking the user to provide a first value $K \leq K^*$ (line 1), and by initializing a support threshold σ to be the best approximation from below to σ_{K^*} (line 2). As we will discuss in the next subsection, some heuristics can be used to set σ to a value possibly larger than the trivial lower bound 1. Instead, σ' is initially set equal to σ, and is raised to the final value σ_K as soon as the K-th frequent closed itemset is discovered. The initialization proceeds by determining $\perp = \mathrm{Clo}_{\mathcal{D}}(\emptyset)$ and by inserting into an initially empty priority queue Q entries for all ppc-extensions of $\perp$ of support at least σ (lines 8,9). If $\perp$ is not empty, it is produced in output as the closed itemset of maximum support (lines 5,6). At this point the main loop (lines $10 \div 22$) starts, where in each iteration the entry $(\mathcal{D}_Y, s, i, Y(i-1))$ with maximum support s is extracted, the itemset Y is generated and returned in output (line 13), and for each ppc-extension X of Y with support $s' \geq \sigma$ and core index $j > i$, the entry $(\mathcal{D}_X, s', j, X(j-1))$ is inserted into Q (lines $16 \div 19$). After an insertion into Q, if the number of closed itemsets returned in output so far (variable `extracted`) plus the number of closed itemsets represented by entries in Q is greater than or equal to K^*, the support threshold σ is raised (line 21) to the maximum value for which K^* itemsets of support no less than this value have been seen so far, and all entries in Q corresponding to itemsets of support smaller than the new threshold σ are safely removed from Q (line 22). The loop ends when all top-K frequent closed itemsets have been generated or Q becomes empty. Finally (lines $23 \div 26$) if the user raises K to a new value $K_{\mathrm{new}} \leq K^*$, and more closed itemsets need to be discovered, the main loop is started again resetting σ' equal to σ as a lower bound to $\sigma_{K_{\mathrm{new}}}$.

We remark that an entry $(\mathcal{D}_Y, s, i, Y(i-1))$ in Q for a closed itemset Y does not contain Y itself but only sufficient information to generate the itemset. The actual generation of Y, which is a time-consuming task, is done only when strictly necessary, that is, when the entry $(\mathcal{D}_Y, s, i, Y(i-1))$ is extracted from Q and Y is guaranteed to belong to the output set. In fact, as it will be shown in the following subsection, entries for all ppc-extensions of Y to insert into Q can be produced efficiently without generating the ppc-extensions themselves.

TopKMiner features the following main advantages compared to algorithm TFP by [WHLT05]: (1) only closed itemsets are actually processed (i.e., inserted into Q); (2) every itemset Y extracted from Q surely belongs to the output set and can be immediately returned to the user; (3) the parameter K can be raised dynamically without the need to restart the computation from scratch. Moreover, the upper limit K^* on the value K, although not strictly needed for correctness, is useful to provide a bound on the maximum number of entries inserted into the priority queue Q. This is established by the following theorem.

Theorem 3.5. *For a dataset $\mathcal{D}$ over a set $\mathcal{I}$ of n items, and upper limit K^* on K, algorithm TopKMiner will insert a total of at most nK^* entries into Q during the entire course of the computation*

Proof. Let w be number of itemsets initially inserted into Q (lines $7 \div 8$ of the pseudocode) and let z be the total number of entries extracted from Q. It is easy to see that because of the dynamic update of the threshold σ, if $z \geq K^*$, as soon as the K^*-th entry is extracted from Q, corresponding to some itemset Y, we have $\sigma = \sigma_{K^*} = \mathrm{s}_{\mathcal{D}}(Y)$. Therefore, for this itemset and for all itemsets associated with entries subsequently extracted from Q, no ppc-extension will be generated. This implies that the entries inserted into Q are at most $w + k^* - 1 + \mathcal{T}$, where $w + K^* + 1$ accounts for the w initial entries and the first k^* extracted (the very first one must belong to the w initial ones) and $\mathcal{T}$ accounts for the ppc-extensions of the first $k^* - 1$ extracted. By reasoning as in the proof of Theorem 3.3, we can show that $w \leq n$ and that $\mathcal{T} \leq (k^* - 1)(n - 1)$, hence the total number of entries inserted into Q is at most

$$n + K^* - 1 + (K^* - 1)(n - 1) \leq nK^*.$$

□

As an immediate corollary of the above theorem we observe that if $K^* = K$ the maximum number of entries inserted by TopKMiner into the priority queue Q is nK which is also the maximum size of $\mathcal{FC}_K(\mathcal{D})$. However, if $|\mathcal{FC}_K(\mathcal{D})| = K$ we may still have nK entries inserted into Q, that is a factor n more than $|\mathcal{FC}_K(\mathcal{D})|$. Nevertheless, as reported in the next section, for all of the datasets and values of K we have tested the number of entries inserted into Q has never exceeded $|\mathcal{FC}_K(\mathcal{D})|$ by a factor larger than 3.3. In fact, with a slightly modification of the algorithm it is possible to

Algorithm 3.1: TopKMiner

```
   Input: Dataset $\mathcal{D}$, max value $K^*$ for $K$
   Output: Top-$K$ f.c.i for any $K \leq K^*$ provided by the user
 1 $K \leftarrow$ input from user; /* $K \leq K^*$ */
 2 Initialize $\sigma$ as a lower bound to $\sigma_{K^*}$; $\sigma' \leftarrow \sigma$;
 3 $Q \leftarrow$ empty priority queue; extracted $\leftarrow 0$;
 4 Compute $\bot = \mathrm{Clo}_{\mathcal{D}}(\emptyset)$;
 5 if $\bot \neq \emptyset$ then
 6   Output $\bot$; extracted++;
 7   if $K = 1$ then $\sigma' = |\mathcal{D}|$;
 8 for each ppc-extension $Y$ of $\bot$ of support $s \geq \sigma$ do
 9   $Q$.insert$((\mathcal{D}_Y, s, \mathrm{core}_i(Y), Y(\mathrm{core}_i(Y) - 1)))$;
10 while $(Q \neq \emptyset)$ and $(Q.\max() \geq \sigma')$ do
11   $(\mathcal{D}_Y, s, i, Y(i-1)) \leftarrow Q$.removeMax();
12   extracted++; if extracted $=K$ then $\sigma' = s$;
13   Generate and output closed itemset $Y$;
14   if $s > \sigma$ then
15     for $j \leftarrow i+1$ to $n$ do
16       /* Denote $X = \mathrm{Clo}_{\mathcal{D}}(Y \cup \{j\})$ */
17       Compute $X(j-1)$, $s' = \mathrm{s}_{\mathcal{D}}(X)$, and $\mathcal{D}_X$;
18       if $X(j-1) = Y(j-1)$ and $s' \geq \sigma$ then
19         $Q$.insert$(\mathcal{D}_X, s', j, X(j-1))$ ;
20         if extracted$+|Q| \geq K^*$ then
21           Update $\sigma$ ;
22           Remove from $Q$ all entries of support $< \sigma$;
23 if user wants to raise $K$ then
24   $K \leftarrow$ new input from user;
25   if $K >$ extracted then $\sigma' \leftarrow \sigma$;
26   goto line 8;
```

Figure 3.4: Algorithm TopKMiner: pseudocode

guarantee that the number of itemset inserted in Q will never exceed $nK_{\max}$, where $K_{\max} \leq K^*$ is the maximum K requested by the user. This modification requires that the ppc-extensions of the closed itemsets produced in output after s is set to be equal to σ' in line 12 are not generated, and the itemsets whose ppc-extension are not computed in lines 14–22 are stored in a new queue Q'. If the user wants to raise K, as first step all the ppc-extension of itemsets in Q' will be generated. In this way the ppc-extensions of at most $K_{\max}$ itemsets are computed, leading to the bound above.

3.1.4 TopKMiner: implementation details

For what concerns the implementation of TopKMiner, there are four aspects which have crucial impact on its performance. They are discussed in this subsection.

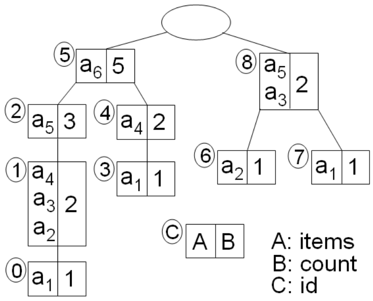


Figure 3.5: Patricia trie for the sample dataset of Figure 2.1 (a). Every node is identified by a unique id shown in a circle to the left of the node

Representation of $\mathcal{D}$: as in [PZ03] the dataset $\mathcal{D}$ is represented through a Patricia trie $T_{\mathcal{D}}$ [Knu73] built on the set of transactions regarded as strings of items. The Patricia trie differs from the standard trie, which is employed by many frequent itemset mining algorithms (see references in [Bod04]), by the fact that chains of nodes with only one child and associated to the same set of transactions are coalesced into a single node. This reduces the overall number of nodes, thus saving space due to node overhead. Each transaction is associated with a distinct path from the root of $T_{\mathcal{D}}$ to some leaf or to some internal node with only one child. Each node v of $T_{\mathcal{D}}$ stores a set of items and a count that indicates how many transactions of $\mathcal{D}$ are associated with paths that contain v. The Patricia trie $T_{\mathcal{D}}$ for the sample dataset of Figure 2.1 (a) is shown in Figure 3.5. It is well known [Bod04] that in order to better exploit the compaction featured by the trie structure, it is convenient to order the items in each transaction by decreasing support. According to our indexing of the items, this requires that for $j > i$ item a_j occur before item a_i. It has been shown both analytically and experimentally in [PZ03] that the Patricia trie provides a space efficient representation for *all* kinds of datasets, and, in particular, for dense ones.

Implementation of Q: the priority queue employed by TopKMiner is implemented as a standard max-heap vector. As mentioned in the previous subsection an entry, corresponding to some closed itemset Y, consists of four components, namely the conditional dataset $\mathcal{D}_Y$, the support s of Y, the core index i of Y, and the prefix $Y(i-1)$, that is the intersection of Y with $\{a_j : 1 \leq j < i\}$. The key for the entry is the support s. While the last three components are stored in a trivial way, a suitable representation of $\mathcal{D}_Y$ is required for both space and time efficiency. We represent $\mathcal{D}_Y$ through a list $L_{\mathcal{D}}(Y)$ of nodes of $T_{\mathcal{D}}$ such that a node v is included in $L_{\mathcal{D}}(Y)$ if and only if v contains the core index item a_i of Y and belongs to a path associated with one or more transactions in $\mathcal{D}_Y$. Let $\mathcal{D}_{Y,v}$ denote the (multi)set of transactions in $\mathcal{D}_Y$ whose associated paths in $T_{\mathcal{D}}$ contain the node v, and let $Z_{Y,v} = \bigcap_{t \in \mathcal{D}_{Y,v}} t$. In the list $L_{\mathcal{D}}(Y)$, together with each node v, we store the prefix $Z_{Y,v}(i-1)$, that is the intersection of all transactions in $\mathcal{D}_{Y,v}$ limited to the items of index less than i. Such a prefix turn out to be useful in the implementation of the while loop described next. Moreover we associate with every node v the number $s_{Y,v}$ of transactions in $\mathcal{D}_Y$ which share this node, that is $s_{Y,v} = |\mathcal{D}_{Y,v}|$.

For very large and sparse datasets, the list $L_{\mathcal{D}}(Y)$ may be very long. If its length exceeds some fixed threshold (5MB in our experiments) the list is stored on disk rather than in main memory. In this fashion we can considerably reduce the amount of main memory required by the algorithm.

Implementation of the while loop: consider an arbitrary iteration of the while loop (lines $10 \div 22$) and suppose that entry $(\mathcal{D}_Y, s, i, Y(i-1))$ is extracted from Q by the first instruction of the iteration. All of the operations prescribed by the iteration can be executed through a simple bottom-up traversal of the sub-trie T' of $T_{\mathcal{D}}$, whose leaves are the nodes in the list $L_{\mathcal{D}}(Y)$ which represents $\mathcal{D}_Y$, as described before. More specifically, the purpose of the traversal is to fill the rows of a *header table* HT, whose j-th row, denoted by HT[j], is associated with item a_j and contains a record with three fields: HT[j].supp, HT[j].pref, and HT[j].list (the contents of these fields will be described below). By using a strategy similar to the one introduced in [PZ03], the subtrie T' can be traversed in such a way to process each node only once. Let $X^{(j)}$ denote the itemset $\mathrm{Clo}_{\mathcal{D}}(Y \cup \{j\})$. During the traversal of T', by percolating upwards the prefixes $Z_{Y,v}(i-1)$ initially stored with the leaves of T' we can update the header table so that, at the end of the traversal, for every $j > i$ we have that:

- HT[j].supp $= \mathrm{s}_{\mathcal{D}}(\mathrm{Clo}_{\mathcal{D}}(Y \cup \{j\}))$;
- HT[j].pref $= X^{(j)}(j-1)$
- HT[j].list is the head of the list of all nodes of T' containing item a_j. Moreover, with each node v in this list we store the count $s_{X^{(j)},v}$ and the prefix $Z_{X^{(j)},v}(j-1)$.

In Figure 3.6 the *HT* filled after a traversal is shown for sample dataset of Figure 2.1 (a). It is easy to see that once the header table is filled as described above, the information stored in its rows is sufficient to fully compute the itemset Y, and to identify each ppc-extension X of Y determining also its support s', its core index j, its prefix $X(j-1)$ and the representation $L_{\mathcal{D}}(X)$ of its conditional dataset. We observe that, at this point, determining for each ppc-extension X of Y all of its constituent items would require an extra non-trivial computation which would be

j	**supp**	**pref**	**list**
6	2	$a_4\ a_1$	5: 2,$\{a_4\ a_1\}$
5	2	$a_3\ a_1$	2: 1,$\{a_4\ a_3\ a_2\ a_1\}$; 8 (1): $\{a_3\ a_1\}$
4	2	a_1	1: 1,$\{a_3\ a_2\ a_1\}$; 4: 1,$\{a_1\}$
3	2	a_1	1: 1,$\{a_2\ a_1\}$; 8: 1,$\{a_1\}$
2	1	a_1	1: 1,$\{a_1\}$
1	-	-	-

Figure 3.6: HT at the end of the traversal of the Patricia trie of Figure 3.5, starting from nodes of $L_{\mathcal{D}}(\{a_1\})$, namely the nodes with id's 0, 3 and 7. For every j and every node v in HT[j].list, its id, $s_{X^{(j)},v}$ and $Z_{X^{(j)},v}(j-1)$ are shown

useless in case X turn out not to belong to the output set. For this reason, TopKMiner postpones the actual determination of a closed itemset X to the time when the entry corresponding to X is extracted from Q, hence ensuring that X belong to the output set.

Update of the threshold σ: at any time during the computation, the threshold σ that TopKMiner uses constitutes an approximation from below of the final value σ_{K^*}. Raising σ allows us to reduce the number of entries inserted into Q, hence to reduce the overall space required by these entries. Moreover, a good estimate σ may allow us to discard infrequent items from the dataset and from the prefixes which are carried along with the representations of the conditional datasets (this optimization, however, has not been implemented in the current version of TopKMiner). Threshold σ can be initially set by using the *closed node heuristic* described in [WHLT05]. At any time during the construction of the Patricia trie, a node v of the Patricia trie is a *closed node* if its support is more than the sum of the supports of its children. This heuristics is based on the fact that, once the construction of the Patricia trie $T_{\mathcal{D}}$ is completed, for each node $v \in T_{\mathcal{D}}$ whose associated count c_v is larger than the sum of the counts of its children, there exists a *different* closed itemset X_v of support at least c_v. If the number of closed nodes is larger than K^*, we can derive a first lower bound $\sigma > 1$. In particular, consider the decreasing sequence of counts $c_1, \ldots, c_{K^*}$ associated with the K^* closed nodes with highest counts. The lower bound derived is then $\sigma = c_{K^*}$.

The subsequent updates of σ (i.e., those performed in line 21 of the pseudocode) can be easily implemented by means of a simple dictionary that maintains for each integer s the number of entries inserted into Q relative to closed itemsets of support s, and provides a method `minSupport()` which, if invoked after that at least K^* entries have been inserted into Q, returns the maximum value s such that K^* among these entries are relative to closed itemsets of support $\geq s$. Clearly, the update of

σ (line 21) can be performed by setting σ to the value returned by `minSupport()`. Finally, all entries of support less than σ which must be removed from Q (line 22) can be identified by maintaining a min-heap Q' whose entries are pointers to the entries of Q together with their supports which are used as keys.

3.1.5 Experimental evaluation

The next two sections present the results of the experiments we performed on the datasets introduced in Section 3.1.1. The experiments have been conduced on an HP Proliant, using a single AMD Opteron 2.2 GHz processor, with 8 GB main memory, 64 KB L1 cache and 1 MB L2 cache. The system's operating system is linux 2.6.5 and the compiler used for the experiments is Intel icc 9.0. The objective of the experiments has been to compare the performance of our algorithm TopKMiner with that of algorithm TFP [WHLT05]. Both TopKMiner and TFP have been coded in `C++` and the source code for TFP has been provided to us by the authors. It must be recalled that TFP has an additional feature which enables the mining of the top-K frequent closed itemsets of length greater than or equal to a minimum value $\min_\ell$ specified in input. We did not implement a similar feature in TopKMiner, hence in the experiments TFP has always been executed with $\min_\ell = 1$.

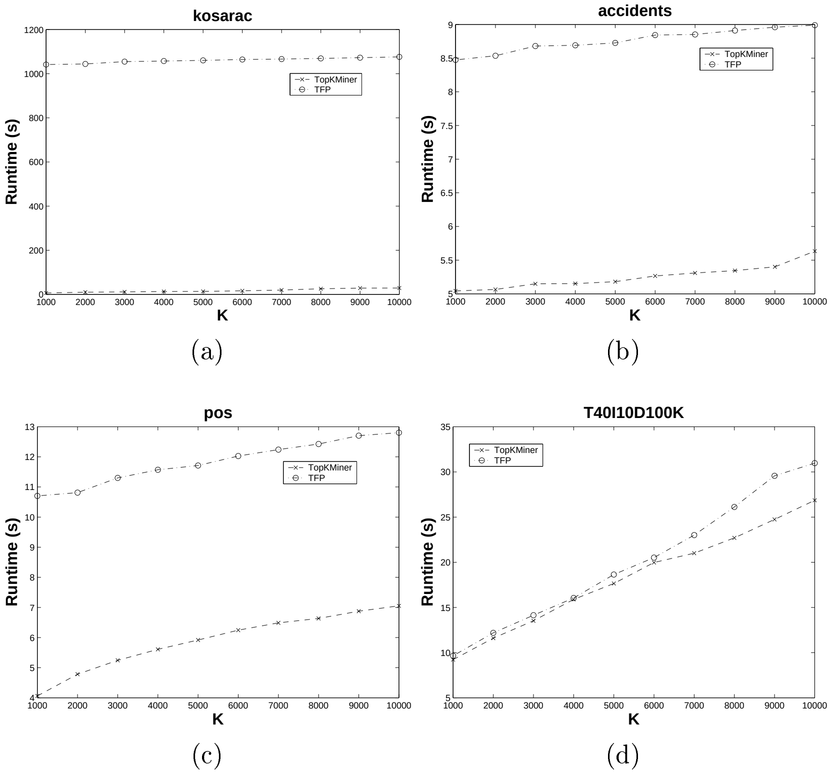
A first set of experiments, compares both running time and memory usage exhibited by the two algorithms on the benchmark datasets for different values of K. In these experiments, the dynamic raising of K featured by TopKMiner has been disabled by always setting $K^* = K$. A critical discussion of the main factors influencing the performance of the two algorithms, besides code optimizations which are hard to account for, is also carried out. A second set of experiments, provides evidence of the effectiveness of the TopKMiner's feature which allows the dynamic raising of K, by simulating a scenario where increasing values of K are input by the user and by comparing the performance of TopKMiner with the one achievable through repeated invocations of TFP. We remark that experiments conducted on several other datasets available in the FIMI repository and for other values of K have given results consistent with those reported here.

3.1.6 Comparing TFP and TopKMiner without dynamic raising of K

We run both TopKMiner and TFP on four of the five datasets described before (i.e., all but webdocs) for values of K ranging from 1000 to 10000 with step 1000. For TopKMiner we imposed $K = K^*$ and for TFP we imposed $\min_\ell = 1$. In this fashion we assessed the relative performance of the two algorithms when focused on the basic task of mining top-K frequent closed itemsets, and with their respective additional features disabled. The running times achieved by the two algorithms are shown in Figure 3.7. It can be seen that TopKMiner runs always faster than TFP, with a performance improvement of more than two orders of magnitude for kosarac. We believe that there are two main reasons that explain the superior performance of TopKMiner. On the one hand, TopKMiner generates only closed itemsets and fully processes itemsets that surely belong to the output set, unlike TFP which may happen to process non-closed and/or infrequent itemsets. On the other hand, TopKMiner features a provable bound on the number of itemsets it touches, while one such bound is not known for TFP. In order to give evidence of this fact, the table in Figure 3.8 reports for the various datasets and for $K = 1000$ and 10000 the number of itemsets touched by the two algorithms. For TopKMiner we consider an itemset X to be touched if an entry for X is inserted into the priority queue, while for TFP we consider an itemset X to be touched if upon its generation it cannot be discarded as non-closed or infrequent and, therefore, it must be stored in a data structure as potential candidate for the output set. We see that TFP touches a number of itemsets which is substantially higher than the number of itemsets touched by TopKMiner. In fact it can be shown that for the artificial dataset $\mathcal{D}_{\text{Yang}}$ defined in [Yan04] and described in Section 2.1 there are several non-constant values of K for which TFP touches a factor n more itemsets than TopKMiner, where n is the number of items.

For dataset webdocs, TFP aborted after a few hours of execution even for $K = 100$ and not because of memory problems. Thus, we compared the running time achieved by TopKMiner with the one achieved by algorithm LCM, [UAUA03], one of the best algorithms at the FIMI'03 competition for mining frequent closed itemsets, feeding LCM with the exact support threshold, which gives a clear advantage to this algorithm in the comparison. As shown in Figure 3.9, TopKMiner surprisingly achieved better performance. In this case, because of the large size of the dataset, it has been crucial for TopKMiner to use external memory to store the conditional dataset representations.

We also compared the memory usage of the two algorithms. While TFP adopts a depth-first mining strategy, which is known to be generally space-efficient, TopKMiner employs a support-driven exploration which may require more space due to the need to store each generated closed itemset until all closed itemsets of higher support have been explored. However, for not too large values of K the actual number of itemsets the TopKMiner must concurrently maintain in the queue is somewhat limited. Figure 3.10 compares the memory usage of TFP and TopKMiner for the same



(a) (b)

(c) (d)

Figure 3.7: Running time for (a) kosarac, (b) accidents, (c) pos, and (d) T40I10D100K for various values of K

	K=1000		K=10000	
Dataset	TopKMiner	TFP	TopKMiner	TFP
T40	1,789	6,091	20,314	78,655
accidents	1,542	2,233	11,057	25,890
pos	2,702	3,597	24,157	42,097
kosarac	2,450	3,798	32,861	56,977

Figure 3.8: Number of itemset “touched” by TopKMiner and TFP

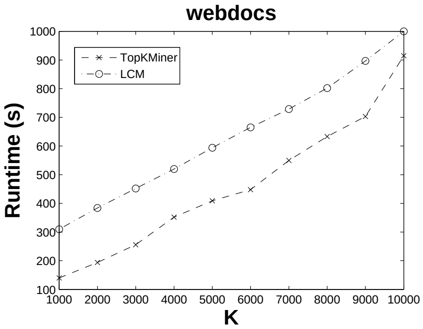


Figure 3.9: Running times of TopKMiner and LCM on webdocs, for various values of K

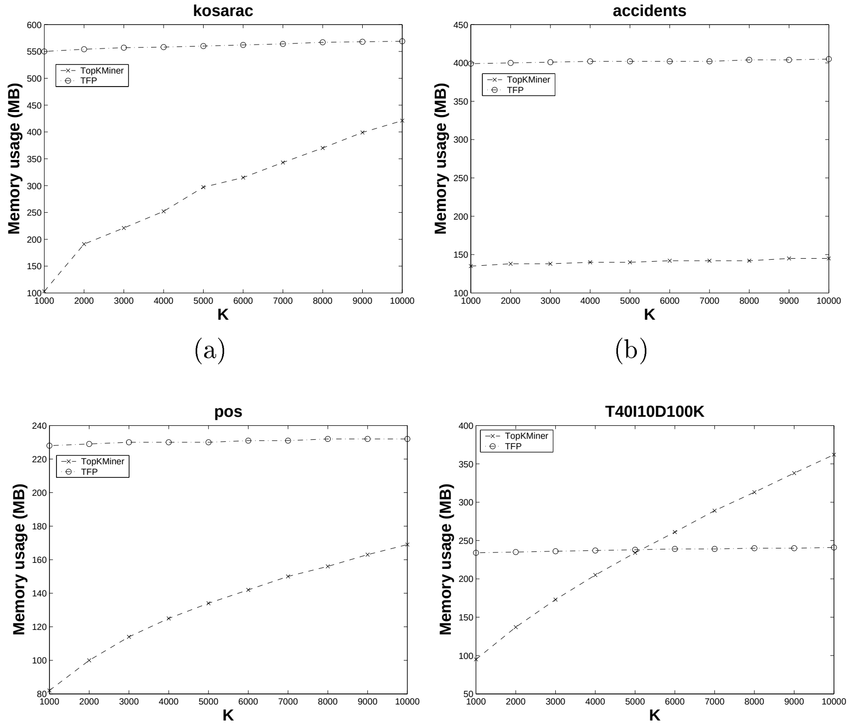
datasets and values of K used when measuring running times. Surprisingly, TopKMiner requires less memory than TFP in all cases except for the artificial dataset T40I10D100K with $K > 5000$ for which it requires more memory (a factor 1.5 for $K = 10000$).

The high memory usage exhibited by TFP can be in part accounted of by the conditional datasets that it creates during execution, while the lower memory usage exhibited by TopKMiner in several cases is due to the efficient representation chosen for the priority queue entries. We remark that although the machine we used for the experiments features a very large RAM (8 GB), in all of the experiments the actual total RAM required never exceeded 450 MB, which is a reasonable quantity even for a low-end PC.

We also profiled the memory usage of TopKMiner separately accounting for the memory required by the priority queue and the rest of the work space. The respective values are shown in Figure 3.11 for the various datasets and for $K = 10000$. We see that, especially for the cases with highest memory usage, a substantial fraction of memory is needed for the priority queue. Since accesses to the priority queue are not the dominant factor in the running time this suggests that the queue could be stored on disk thus reducing considerably the memory usage. This and other space optimizations (e.g., compression of the queue entries) could be exploited when memory is the most important resource.

3.1.7 Comparing TFP and TopKMiner with dynamic raising of K

We tested the effectiveness of the TopKMiner's feature which allows the user to dynamically raise the value K up to a maximum value K^*. To this purpose we



(a)

(b)

(c)

(d)

Figure 3.10: Memory for (a) kosarac, (b) accidents, (c) pos, and (d) T40I10D100K for various values of K

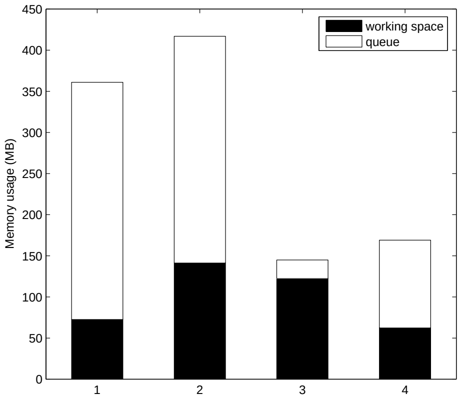
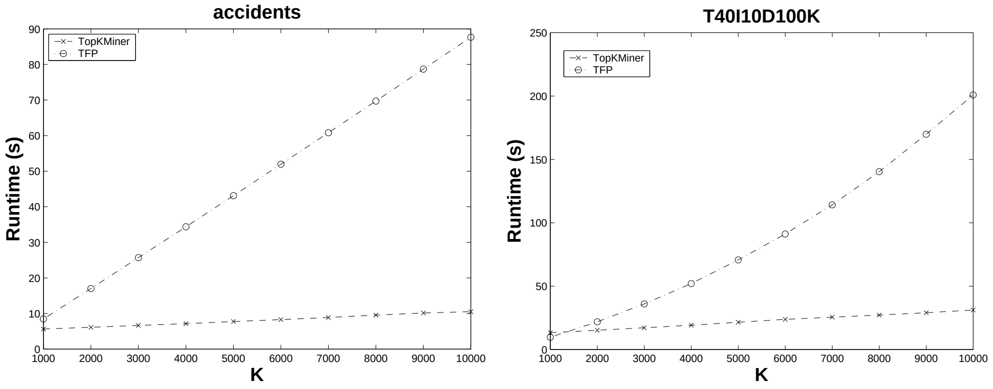


Figure 3.11: Memory used for queue and for computation for $K = 10000$ in (1) T40I10D100K, (2) kosarac, (3) accidents, and (4) pos.

simulated a scenario where K is raised from 1000 to 10000 with step 1000 and run TopKMiner with $K^* = 10000$ measuring the running time after the computation for each value K ended. We compared these running times with those attainable by TFP if used in a similar scenario, by running, for each K, the algorithm from scratch and accumulating the running times of previous executions. The results are shown in Figure 3.12 only for two datasets. (The time for the user's input is not accounted of in the reported times.) Results for the other datasets are similar.

As expected, the time required by TopKMiner for each value of K is considerably lower than the cumulative time required by TFP, which is a clear evidence of the effectiveness of TopKMiner's dynamic feature. Moreover, we remark that the provision of such a feature adds only a negligible slowdown. Indeed, even if the computation is stopped after the first value $K = 1000$, the performance of TopKMiner remains comparable with the one of TFP. This means that the flexibility of TopKMiner (in the raising of K) does not cause a degradation in performance. For the memory usage, the amount used by the two algorithms can be derived from Figure 3.10, since for both TopKMiner and TFP the maximum memory usage with dynamic raising of K (up to K^*) is equal to the maximum memory usage for $K = K^*$.



(a) (b)

Figure 3.12: Running times of TopKMiner and TFP for (a) accidents, and (b) T40I10D100K with dynamic update of K from 1000 to 10000.

3.2 Mining frequent items/itemsets through sampling

When dealing with massive datasets, computing the exact set of top-K (maximal/closed) frequent itemsets can be too expensive. If the dataset does not fit completely in the main memory, the disk access may slow down exact algorithms to a point where they become impractical. Algorithms for the standard frequent itemset mining task developed to solve the problem in an exact way must scan the entire dataset, typically several times, which has a considerable impact on performance. It is then necessary to accept a trade-off between the accuracy of the results and the time needed to compute them, especially if it is possible for the user of the algorithm to specify the maximum decay in the "quality" of the output she is willing to accept. Sampling is one technique that can be employed to reduce the running time, obtaining approximated results.

The rest of the section is organized as follows. In Subsection 3.2.1 we formally introduce the problem of extracting top-K frequent itemsets through sampling, providing a tight bound on the sufficient sample size in Subsection 3.2.2. In Subsection 3.2.3 we present an algorithm to solve this problem, and Subsection 3.2.4 proves the correctness of the algorithm. In Subsection 3.2.5 we show how to improve the space requirements of the method using a *count-min* filter, and prove the correctness of the approach in Subsection 3.2.6. In Subsection 3.2.8 we show that our algorithms can be used to obtain an approximation of top-K frequent itemsets with guarantee on the quality of the estimated frequencies. Finally, Subsection 3.2.7 provides the results of the experimental assessment of our algorithms.

3.2.1 Mining (approximated) top-K frequent itemset

Consider a dataset $\mathcal{D}$ of transactions over the set $\mathcal{I}$ of n items. For convenience, we fix a *canonical ordering* of the itemsets built on $\mathcal{I}$ by decreasing frequency, ties broken arbitrarily. Let $m = 2^n - 1$, we suppose the itemsets to be labeled $X_1, X_2, \cdots, X_m$ according to this order. For a given K, with $1 \leq K \leq m$, we denote $f_{\mathcal{D}}^{(K)} = f_{\mathcal{D}}(X_K)$. For convenience, we use $\mathrm{TOPK}(\mathcal{D}, \mathcal{I}, K)$ to indicate the set of top-K frequent itemsets.

We aim at efficiently mining the following approximation to the set $\mathrm{TOPK}(\mathcal{D}, \mathcal{I}, K)$.

Definition 3.6. *Let $\varepsilon \in (0,1)$ be a real-valued parameter. A set $W \subseteq 2^{\mathcal{I}}$ is an* ε-approximation *to* $\mathrm{TOPK}(\mathcal{D}, \mathcal{I}, K)$ *if and only if the following two properties hold:*

P1: *for each $X \in W$, $f_{\mathcal{D}}(X) \geq f_{\mathcal{D}}^{(K)} - \varepsilon$;*

P2: *for each $X \notin W$, $f_{\mathcal{D}}(X) < f_{\mathcal{D}}^{(K)} + \varepsilon$.*

A similar approximation is defined in [CCFC04], but requires only **P1** to hold, thus providing only a guarantee that itemsets with frequency well below $f^{(k)}$ are not produced in output. The same approximation (i.e., requiring only **P1** to hold) is considered in [VV09] for the problem of mining the top-K frequent items. The authors of that work define different approximations, with different properties, of the set of the top-K frequent items and present algorithms to mine them. Other than the one considered in [CCFC04], one of interest for our work requires in addition an approximation of the frequencies of the itemsets in output. Moreover, they present an approximation of the set such that the ranking of the output set is approximately correct with regards to the relative ranking in the dataset of the output items. The authors provide bounds on the sufficient sample size required to obtain the desired approximations. The stricter bounds are based on the idea of *lumping* small frequency items, i.e., aggregating two or more items with frequencies smaller than some threshold to form a meta-item whose frequency is the sum of the frequencies of the items that form the meta-item. This is done iteratively until none or one (meta-)item is left with frequency smaller than the threshold. The goal of this lumping process is to bound the size of the set of elements to be considered, in order to obtain better bounds on the sufficient sample size. However, their results do not apply to the problem of approximating top-K frequent itemsets. Moreover, the stricter bounds related to the problems of interest to our work require the knowledge of the exact distribution of frequencies of the items, which is not available in real cases.

3.2.2 Bound on sufficient sample size

The following theorem shows that the set of *top-K frequent itemsets* mined from a sample[2] of $\mathcal{D}$ of suitable size constitutes an ε-approximation to $\mathrm{TOPK}(\mathcal{D}, \mathcal{I}, K)$, with a certain probability.

Theorem 3.7. *For fixed $\varepsilon, \delta \in (0,1)$, consider a random sample $\mathcal{S} \subseteq \mathcal{D}$ containing*

$$t = \frac{2}{\varepsilon^2} \ln \frac{2K(m-K)}{\delta}$$

transactions of $\mathcal{D}$, and let $W = \mathrm{TOPK}(\mathcal{S}, \mathcal{I}, K)$. Then, W is an ε-approximation to $\mathrm{TOPK}(\mathcal{D}, \mathcal{I}, K)$ with probability at least $1 - \delta$.

[2] In this work we consider sampling with replacement.

Proof. We define the following four sets:

$$\begin{array}{rcl} L & = & \left\{X \in 2^{\mathcal{I}} \,:\, f_{\mathcal{D}}(X) \geq f_{\mathcal{D}}^{(K)} + \varepsilon\right\} \\ V & = & \left\{X \in 2^{\mathcal{I}} \,:\, f_{\mathcal{D}}(X) < f_{\mathcal{D}}^{(K)} - \varepsilon\right\} \\ M & = & \left\{X_i \in 2^{\mathcal{I}} \,:\, 1 \leq i \leq K\right\} \\ Z & = & \left\{X_i \in 2^{\mathcal{I}} \,:\, K+1 \leq i \leq m\right\}, \end{array}$$

where the indices of the itemsets in the last two sets are consistent with the canonical ordering mentioned above. Notice that $L \subset M$ and $V \subseteq Z$. For an itemset X, let $f_{\mathcal{S}}(X)$ denote its frequency in the sample. Define the events

E_1: "for all pairs $(X, Y) \in L \times Z$ we have $f_{\mathcal{S}}(X) \geq f_{\mathcal{S}}(Y)$"

E_2: "for all pairs $(X, Y) \in M \times V$ we have $f_{\mathcal{S}}(X) > f_{\mathcal{S}}(Y)$".

We first show that if E_1 and E_2 occur then W is an ε-approximation to TOPK$(\mathcal{D}, \mathcal{I}, K)$. Assume that E_1 and E_2 occur. We have to prove that Properties **P1** and **P2** in Definition 3.6 hold for W. Since E_2 occurs and $|M| = K$, no element of V can be included in W, hence Property **P1** follows. As for Property **P2**, consider an itemset $X \notin W$ and suppose, by contradiction, that $f_{\mathcal{D}}(X) \geq f_{\mathcal{D}}^{(K)} + \varepsilon$, thus $X \in L$. Since $|W| \geq K$ and $|\{X \in \mathcal{I} \,:\, f_{\mathcal{D}}(X) > f_{\mathcal{D}}^{(K)}\}| < K$, there must exists $Y \in W$ that is also in Z (if no element of Z is in W, we have that $W = M \supset L$). Then, there is a pair $(X, Y) \in L \times Z$ with $f_{\mathcal{S}}(X) < f_{\mathcal{S}}(Y)$, which is impossible since E_1 occurs.

We complete the proof by showing that if the sample size is t chosen as stated, then both E_1 and E_2 occur with probability at least $1 - \delta$. Consider a pair (x, y) in $L \times Z$, and let t be the number of transactions in $\mathcal{S}$. Since $f_{\mathcal{D}}(X) - f_{\mathcal{D}}(Y) \geq \varepsilon$, by the Azuma bound we have

$$\Pr(f_{\mathcal{S}}(Y) > f_{\mathcal{S}}(X)) \leq 2e^{-\frac{\varepsilon^2}{2}t}.$$

The same bound applies to an arbitrary pair $(X, Y) \in M \times V$. We now apply the union bound. Notice that the same pair (X, Y) can be in both $L \times Z$ and $M \times V$. However, since $L \subset M$, $V \subseteq Z$, and the sets M, Z are disjoint, we have that the total number of pairs that we have to consider in the union bound is $\leq |M| \times |Z| < K \cdot (M - k)$. When for all pairs (X, Y) in $(L \times Z) \cup (M \times V)$ we have $f_{\mathcal{S}}(X) > f_{\mathcal{S}}(Y)$, both E_1 and E_2 occur. Then, the probability that at least one event between E_1 and

E_2 does not occur is at most

$$K(m-K)2e^{-\frac{\varepsilon^2}{2}t} \leq \delta.$$

□

3.2.3 Algorithm

We now describe an efficient algorithm which discovers an ε-approximation to TOPK$(\mathcal{D}, \mathcal{I}, K)$ by mining progressively larger samples of the dataset $\mathcal{D}$ until the sample size established in Theorem 3.7 is reached, or a certain stopping condition is met. When the algorithm stops it returns, as output, the set of top-K frequent itemsets with respect to the last processed sample. Such a set will constitute an ε-approximation to TOPK$(\mathcal{D}, \mathcal{I}, K)$ with probability at least $1-\delta$. For $j \geq 0$, define

$$t_j = \frac{8}{\varepsilon^2}\left(\ln \frac{8aK}{\delta} + bj\right),$$

where $a \geq 1$ and $b \geq 1$ are suitable parameters. Let also $j_{\max} \geq 0$ be the smallest index such that $t_{j_{\max}} \geq \min\{|\mathcal{D}|, 2/(\varepsilon^2)\ln(2K(m-K)/\delta)\}$. The algorithm performs a sequence of phases. Specifically, in Phase j, for $j \geq 0$ and $j < j_{\max}$, the algorithm processes a random sample of t_j transactions. In Phase $j_{\max}$, if $t_{j_{\max}} \geq |\mathcal{D}|$ the algorithm processes $\mathcal{D}$ to extract TOPK$(\mathcal{D}, \mathcal{I}, K)$, otherwise it considers a random sample of $t_{j_{\max}}$ transactions. The algorithm stops when $j = j_{\max}$, or $j < j_{\max}$ and a suitable stopping condition (specified below) holds.

Consider Phase j and let $\mathcal{S}$ be the random sample of size t_j processed in the phase. Define

$$\sigma_j = aK\left(\frac{e}{2}\right)^{bj}.$$

For $i \geq 0$, define also

$$s_j(i) = \lfloor (2\sigma_j)^{(i+1)^2}/2 \rfloor,$$

and

$$S_j(i) = \sum_{\ell=0}^{i} s_j(\ell).$$

For notational convenience, we assume $S_j(-1) = 0$ and use $h(j)$ as the largest index such that $S_j(h(j)-1)+1 \leq m$. Consider an ordering of the itemsets by decreasing frequency w.r.t. $\mathcal{S}$, and let $f_{\mathcal{S}}^{(\ell)}$ be the frequency in $\mathcal{S}$ of the ℓ-th itemset in this ordering. The stopping condition for Phase j is

$$f_{\mathcal{S}}^{(K)} - f_{\mathcal{S}}^{(S_j(i-1)+1)} > (i+1)\varepsilon \quad \text{for} \quad 1 \le i \le h(j).$$

3.2.4 Analysis

For Phase j of the algorithm we define $B_j(i)$, with $0 \le i \le h(j)$, as the set of $s_j(i)$ itemsets whose rank in the canonical ordering (w.r.t. the original dataset $\mathcal{D}$) is in the interval $[S_j(i-1)+1, S_j(i)]$.

Lemma 3.8. *With probability at least $1-\delta$ the following property holds: for every Phase j of the algorithm, for every $0 \le i \le h(j)$, and for every itemset $X \in B_j(i)$*

$$|f_{\mathcal{S}}(X) - f_{\mathcal{D}}(X)| < (i+1)\frac{\varepsilon}{2},$$

where $\mathcal{S}$ is the sample processed in Phase j.

Proof. Let us focus on an arbitrary Phase j. By the Azuma bound, we have that for any $X \in B_j(i)$

$$\Pr(|f_{\mathcal{S}}(X) - f_{\mathcal{D}}(X)| \ge (i+1)\frac{\varepsilon}{2}) \le 2e^{-\varepsilon^2(i+1)^2 t_j/8}.$$

Hence the probability that there exists an itemset X (belonging to any $B_j(i)$) for which the stated bound does not hold is upper bounded by:

$$\sum_{i=0}^{h(j)} s_j(i) 2e^{-\varepsilon^2(i+1)^2 t_j/8} \le \sum_{i=0}^{h(j)} \left(2\sigma_j e^{-\varepsilon^2 t_j/8}\right)^{(i+1)^2} = \sum_{i=0}^{h(j)} \left(\frac{\delta}{2^{j+2}}\right)^{(i+1)^2} \le \frac{\delta}{2^{j+1}}.$$

The lemma follows by applying the union bound over all phases (i.e., $j = 0, 1, \ldots$). □

The following theorem establishes a probabilistic guarantee on the correctness of the algorithm.

Theorem 3.9. *The algorithm returns an ε-approximation to* $\mathrm{TOPK}(\mathcal{D}, \mathcal{I}, K)$ *with probability at least $1-\delta$.*

Proof. We consider two cases, depending on when the algorithm stops. If the algorithm stops at Phase $j = j_{\max}$, then the output is correct with probability at least $1-\delta$, since it is the set $\mathrm{TOPK}(\mathcal{D}, \mathcal{I}, K)$ if $t_{j_{\max}} \ge |\mathcal{D}|$, and we can resort to Theorem 3.7 if $t_{j_{\max}} < |\mathcal{D}|$. Suppose instead that the algorithm stops at an earlier phase $j < j_{\max}$ because the stopping condition is met. By Lemma 3.8, for every

$0 \le i \le h(j)$, and for every itemset $X \in B_j(i)$, we have $|f_{\mathcal{S}}(X) - f_{\mathcal{D}}(X)| < (i+1)\frac{\varepsilon}{2}$. Let W the set of itemsets returned by the algorithm, namely the set of top-K frequent itemsets with respect to the random sample processed in Phase j.

We first show that $W \subseteq B_0$. By contradiction, assume that an itemset $X \in W$ belongs to B_i, for some $i > 0$. Hence, $f_{\mathcal{D}}(X) \le f_{\mathcal{D}}^{(S_j(i-1)+1)}$ and

$$f_{\mathcal{S}}^{(K)} \le f_{\mathcal{S}}(X) \le f_{\mathcal{D}}(X) + (i+1)\frac{\varepsilon}{2} \le f_{\mathcal{D}}^{(S_j(i-1)+1)} + (i+1)\frac{\varepsilon}{2}. \tag{3.1}$$

Observe that all itemsets whose rank in the canonical ordering (w.r.t. $\mathcal{D}$) is not larger than $S_j(i-1)+1$ belong to sets B_ℓ with $\ell \le i$. By Lemma 3.8, for each such itemset Z, we have that

$$f_{\mathcal{S}}(Z) \ge f_{\mathcal{D}}(Z) - (i+1)\frac{\varepsilon}{2} \ge f_{\mathcal{D}}^{(S_j(i-1)+1)} - (i+1)\frac{\varepsilon}{2}.$$

Hence, since there are $S_j(i-1)+1$ of these itemsets, it follows that

$$f_{\mathcal{S}}^{(S_j(i-1)+1)} \ge f_{\mathcal{D}}^{(S_j(i-1)+1)} - (i+1)\frac{\varepsilon}{2}. \tag{3.2}$$

By combining Equations 3.1 and 3.2 we obtain that

$$f_{\mathcal{S}}^{(K)} - f_{\mathcal{S}}^{(S_j(i-1)+1)} \le (i+1)\varepsilon,$$

which contradicts the stopping condition. Thus, $W \subseteq B_0$. Now, if we consider any of the first K itemsets in the canonical ordering, say X_ℓ, for some $1 \le \ell \le K$, which belongs to B_0 by construction, we have that $f_{\mathcal{S}}(X_\ell) \ge f_{\mathcal{D}}(X_\ell) - \frac{\varepsilon}{2} \ge f_{\mathcal{D}}^{(K)} - \frac{\varepsilon}{2}$. Hence, $f_{\mathcal{S}}^{(K)} \ge f_{\mathcal{D}}^{(K)} - \frac{\varepsilon}{2}$. Therefore, for each $X \in W$ we have

$$f_{\mathcal{D}}(X) \ge f_{\mathcal{S}}(X) - \frac{\varepsilon}{2} \ge f_{\mathcal{S}}^{(K)} - \frac{\varepsilon}{2} \ge f_{\mathcal{D}}^{(K)} - \varepsilon,$$

which establishes Property **P1**. As for Property **P2**, note that W must contain an itemset Z such that $f_{\mathcal{D}}(Z) \le f_{\mathcal{D}}^{(K)}$. As argued before, $Z \in B_0$, hence

$$f_{\mathcal{S}}(Z) \le f_{\mathcal{D}}(Z) + \frac{\varepsilon}{2} \le f_{\mathcal{D}}^{(K)} + \frac{\varepsilon}{2}.$$

Since $f_{\mathcal{S}}(Z) \ge f_{\mathcal{S}}^{(K)}$, we have that

$$f_{\mathcal{S}}^{(K)} \le f_{\mathcal{D}}^{(K)} + \frac{\varepsilon}{2} \tag{3.3}$$

Consider an itemset $Y \notin W$. If $Y \in B_i$ with $i > 0$ then by definition of B_i its real frequency is at most $f_{\mathcal{D}}^{(K)}$, hence it cannot be greater than or equal to $f_{\mathcal{D}}^{(K)} + \varepsilon$.

If instead $Y \in B_0$ we have

$$f_{\mathcal{D}}(Y) \leq f_{\mathcal{S}}(Y) + \frac{\varepsilon}{2} < f_{\mathcal{S}}^{(K)} + \frac{\varepsilon}{2} \leq f_{\mathcal{D}}^{(K)} + \varepsilon,$$

where the last inequality follows from Equation 3.3, and Property **P2** follows. □

3.2.5 A Count-Min Filter Based Algorithm

The algorithm presented in the previous section has a major issue: it needs m counters to keep the support counts for all the itemsets in order to be able to find the $f_{\mathcal{S}}^{(i)}$ for all i's.

We now present an improved version of the algorithm which uses count-min filters, a variation of Bloom filters, to save space. A count-min filter B consists of c counters, and uses k_B hash functions. The counters are split into k_B disjoint groups of size c/k_B. (We assume that k_B divides c evenly.) The k_B hash functions map itemsets into counters, so for each hash function $H_i, 1 \leq i \leq k_B$ we have $H_i : 2^{\mathcal{I}} \rightarrow [0, c/k_B - 1]$. A more detailed description of count-min filters and their properties can be found in [MU05], Sec. 13.4.

Using count-min filters, we can provide a ε-approximation to TOPK($\mathcal{D}, \mathcal{I}, K$). Given a set of transactions $\mathcal{S}$, we use a count-min filter B to keep track of an approximation of the supports of the itemsets. Initially, all counters are set to 0. For each transaction $t \in \mathcal{S}$ and each itemset $X \subseteq t$, we increment by one the k_B counters associated with X.

Definition 3.10. *The* count-min support *of an itemset X is the value of the minimum of the k_B counters associated with X in B, and is denoted with $s_B(X)$.*

Definition 3.11. *The* count-min frequency *of X is*

$$f_B(X) = \frac{s_B(X)}{|\mathcal{S}|}.$$

(The notation for count-min support and count-min frequency does not include any reference to $\mathcal{S}$ because the set of transactions on which the count-min filter is built will be clear from the contest.)

Given a set of transactions $\mathcal{S}$, let the length (as number of items) of a transaction $t \in \mathcal{S}$ be denoted as $|t|$. The number of itemsets of non zero length in a transaction

t is $2^{|t|} - 1$. We denote the sum of the number of itemsets in the transactions as $C_{\mathcal{S}}$:

$$C_{\mathcal{S}} = \sum_{t \in \mathcal{S}} (2^{|t|} - 1).$$

The following theorem shows that we can obtain a good approximation of the frequencies of the itemsets using a count-min filter.

Theorem 3.12. *Given $\delta_B > 0$, $\varepsilon_B > 0$, and a set of transactions $\mathcal{S}$, let $\varepsilon_C = \frac{\varepsilon_B |\mathcal{S}|}{C_{\mathcal{S}}}$ and $\delta_c = \delta_B / m$. If B is a count-min filter of parameters*

$$k_B = \left\lceil \ln \frac{1}{\delta_c} \right\rceil \tag{3.4}$$

$$c = \left\lceil \ln \frac{1}{\delta_c} \right\rceil \cdot \left\lceil \frac{e}{\varepsilon_c} \right\rceil \tag{3.5}$$

then

$$\Pr(\exists X | f_B(X) \geq f_{\mathcal{S}}(X) + \varepsilon_B) \leq \delta_B.$$

Proof. A known result for count-min filters (see [MU05], Theorem 13.12) states that if the sum of the counts of the elements inserted in a count-min filter is L, then with probability $1 - (k_B/(c\varepsilon_C))^{k_B}$ for any given element X we have

$$s_B(X) \leq s_{\mathcal{S}}(X) + \varepsilon_C L,$$

where $s_{\mathcal{S}}(X)$ is the support of X in $\mathcal{S}$.

In our case we have that $L = C_{\mathcal{S}}$, thus for any given itemsets X we have that

$$f_B(X) = \frac{s_B(X)}{|\mathcal{S}|} \leq \frac{s_{\mathcal{S}}(X)}{|\mathcal{S}|} + \frac{\varepsilon_C C_{\mathcal{S}}}{|\mathcal{S}|} = f_{\mathcal{S}}(X) + \varepsilon_B$$

with probability

$$1 - \left(\frac{k_B}{c\varepsilon_c}\right)^{k_B} \geq 1 - e^{-\ln(1/\delta_c)} = 1 - \frac{\delta_B}{m}.$$

The thesis follow applying the union bound on the complementary events. □

Now let $K > 0$ and $\mathcal{S}$ be a set of transactions. For given $\delta_B > 0$, $\varepsilon_B > 0$, we store the support counts of the itemsets using a count-min filter B with parameters c and k_B as in Theorem 3.12. From Theorem 3.12 we can obtain a lower bound to the frequency of the K-th most frequent itemset in $\mathcal{S}$.

Corollary 3.13. *Let $X_1^B, X_2^B, \cdots$ be a labeling of the itemsets following the decreasing order of their frequency in the count-min filter B, ties broken arbitrarily. Let*

$f_B^{(i)} = f_B(X_i^B)$ and $r = f_B^{(K)} - \varepsilon_B$. Then, with probability at least $1 - \delta_B$, all the top-K FI's of $\mathcal{S}$ have a frequency in B greater or equal to r.

Proof. Suppose Theorem 3.12 holds, which happens with probability at least $1-\delta_B$. By definition of $f_B^{(k)}$ there are at least k itemsets with a count-min frequency $\geq f_B^{(k)}$. We now consider a subset X with size k of these itemsets. Since Theorem 3.12 holds, none of these itemsets has a frequency in $\mathcal{S}$ less than r. Suppose now that all itemsets in X are among the top-k FI's of $\mathcal{S}$. Then at least one of them has a frequency in $\mathcal{S}$ equal to $f_{\mathcal{S}}^{(k)}$. If the size of the set of the top-k FI's of $\mathcal{S}$ is exactly k, then the thesis follows immediately from Theorem 3.12, the definition of r and the properties of the count-min filter. If the size of the sets of the top-k FI's of $\mathcal{S}$ is greater than k, then there is at least one of such itemsets that is not in X. Let y be one of these itemsets. By definition, $f_{\mathcal{S}}(y) \geq f_{\mathcal{S}}^{(k)}$. Since there is at least one itemset in X with frequency in $\mathcal{S}$ equal to $f_{\mathcal{S}}^{(k)}$, then from Theorem 3.12, from the definition of r, and from the properties of the count-min filter, we have

$$r \leq f_{\mathcal{S}}^{(k)} \leq f_{\mathcal{S}}(y) \leq f_B(y)$$

This holds for any y which belongs to the set of the top-k FI's of $\mathcal{S}$ but not to X, so the thesis follows.

Suppose now that not all itemsets in X are among the top-k FI's of $\mathcal{S}$. Then there is at least one itemset in X such that its frequency in $\mathcal{S}$ is less than $f_{\mathcal{S}}^{(k)}$. Let w be one of such itemsets, and z be any of the top-k FI's of $\mathcal{S}$, we have

$$r \leq f_{\mathcal{S}}(w) < f_{\mathcal{S}}(z) \leq f_B(z)$$

which prove the thesis. □

In the following, we develop and analyze an algorithm to find an ε-approximation of TOPK($\mathcal{D}$,$\mathcal{I}$,K) with probability $1 - \delta$.

As before, the algorithm requires in input a dataset $\mathcal{D}$ and three parameters $K > 0$, $\varepsilon, \delta \in (0, 1)$.

Let $\delta_1, \delta_2 > 0$ such that $(1 - \delta_1)(1 - \delta_2) = 1 - \delta$. We define t_j similarly to Section 3.2.3. The algorithm performs a sequence of phases, and in Phase j, for $j \geq 0$ and $j < j_{\max}$, the algorithm processes a random sample of t_j transactions, as it was for the algorithm in Section 3.2.3.

The algorithm stops when $j = j_{max}$, or $j < j_{max}$ and a suitable stopping condition (specified below) holds. Consider Phase j and let $\mathcal{S}$ be the random sample of size t_j processed in the phase. Define $\sigma_j, s_j(i)$, and $S_j(i)$ as in Section 3.2.3.

Let $\mathcal{S}$ be the sample analyzed by the algorithm at phase j. The algorithm will use a count-min filter B with parameters c, k_B tuned in such a way that $\Pr(\exists X | f_B(X) \geq f_{\mathcal{S}}(X) + \varepsilon_B) \leq \delta_2$ (see Theorem 3.12). Note that ε_B is not given in input by the user. Consider an ordering of the itemsets by decreasing count-min frequency w.r.t. B, and let $f_B^{(\ell)}$ be the count-min frequency of the ℓ-th itemset in this ordering. Let $r = f_B^{(K)} - \varepsilon_B$. The stopping condition for phase j is

$$r - f_B^{(S_j(i-1)+1)} > (i+1)\varepsilon \text{ for } 1 \leq i \leq h(j).$$

(Note that the choice of ε_B influences the stopping conditions, since $r = f_B^{(K)} - \varepsilon_B$.)

When the algorithm stops, it computes the exact frequencies in the sample of the itemsets $\{X : f_B(X) \geq r\} = \mathcal{B}$. Let $\tilde{f}_{\mathcal{S}}^{(K)}$ the frequency in the sample of the K-th most frequent itemset in $\mathcal{B}$. The output of the algorithm is thus the set of itemsets $W = \left\{X \in \mathcal{B} : f_{\mathcal{S}}(X) \geq \tilde{f}_{\mathcal{S}}^{(K)}\right\}$.

3.2.6 Analysis of Count-Min Filter Based Algorithm

First of all, note that the definition of t_j, σ_j, and $S_j(i)$ are the same as in Section 3.2.3, but for the replacement of δ with δ_1. Thus Lemma 3.8 holds with probability at least $1 - \delta_1$.

The following theorem relates the stopping condition of the count-min filter based algorithm to the stopping condition of the algorithm presented in Section 3.2.3.

Theorem 3.14. *With probability at least $1 - \delta_2$, when the stopping condition of the count-min filter based algorithm is met, the stopping condition of the algorithm in Section 3.2.3 holds.*

Proof. Assume that Corollary 3.13 holds, then $r \leq f_{\mathcal{S}}^{(k)}$. For the properties of the count-min filter we have $\forall i, f_B^{(i)} \geq f_{\mathcal{S}}^{(i)}$. Then,

$$f_{\mathcal{S}}^{(k)} - f_{\mathcal{S}}^{(S_j(i-1)+1)} \geq r - f_B^{(S_j(i-1)+1)} \text{ for } 1 \leq i \leq h(j).$$

Hence, if the stopping condition for the count-min filter based algorithm is satisfied, then also the stopping condition for algorithm of Section 3.2.3 must be satisfied. Since Corollary 3.13 holds with probability at least $1 - \delta_2$, we obtain the theorem. □

We are now ready to prove the main theorem.

Theorem 3.15. *The count-min filter based algorithm returns an ε-approximation of* $\mathrm{TOPK}(\mathcal{D}, \mathcal{I}, K)$ *with probability at least* $1 - \delta = (1 - \delta_1)(1 - \delta_2)$.

Proof. We consider two cases, depending on when the algorithm stops. If the algorithm stops at phase $j = j_{\max}$, then the output is correct with probability at least $1-\delta$, since it is correct with probability 1 if the algorithm considers $\mathcal{D}$ in phase $j_{\max}$, otherwise it is correct with probability at least $1 - \delta$ by virtue of Theorem 3.7.

Suppose instead that the algorithm stops at an earlier phase $j < j_{max}$ because the stopping condition is met. From now on we assume that Lemma 3.8 and Theorem 3.12 hold (this happens with probability at least $(1-\delta_1)(1-\delta_2) = 1-\delta$: in each iteration Theorem 3.12 holds with probability at least $1 - \delta_2$, since the quality of approximation of the frequencies in the sample provided by the count-min filter does not depend on previous iterations and on the frequencies in $\mathcal{D}$). Let W be the set of itemsets given as output. Since Theorem 3.12 holds, then Corollary 3.13 also holds, and Theorem 3.14 too. Thus $\mathcal{B}$ is a superset of the set of itemsets W' that algorithm of Section 3.2.3 run with parameters ε, δ would have returned. Thus W is equal to W', and it is an ε-approximation to $\mathrm{TOPK}(\mathcal{D}, \mathcal{I}, K)$ with probability at least $1 - \delta = (1 - \delta_1)(1 - \delta_2)$. □

3.2.7 Experiments

We run a preliminary set of experiments to evaluate the performances of the algorithm described in Section 3.2.3. We run the experiments on two datasets presented in Section 3.1.1: kosarak.dat (999002 transactions) and webdocs.dat (512 transactions). Our choice for the parameters were fixed to the following values: $\varepsilon = 0.02$, $\delta = 0.1$, $a = 1$, $b = 1$, and we asked our algorithm to extract the k most frequent itemsets of length at most l, for different values of k and l, for kosarak, and the k most frequent items in webdocs. We run our algorithm 10 times, and for all executions the output satisfied both properties **P1** and **P2** of Definition 3.6.

For kosarak and $l = 1$, the stopping size was always equal to the theoretical bound given in Theorem 3.7. The results for $l = 2, 3$ the results are reported in Figure 3.13 and Figure 3.14. Figure 3.15 reports instead the results for the extraction of top-K items from webdocs.

We can observe that when the parameter l, that is the maximum size of itemsets to be extracted, increases, the gap between the number of transactions that our algorithm needs to produce the output and the number of transactions implied from the theoretical bound widens. Since we expect the number of potential itemsets in a

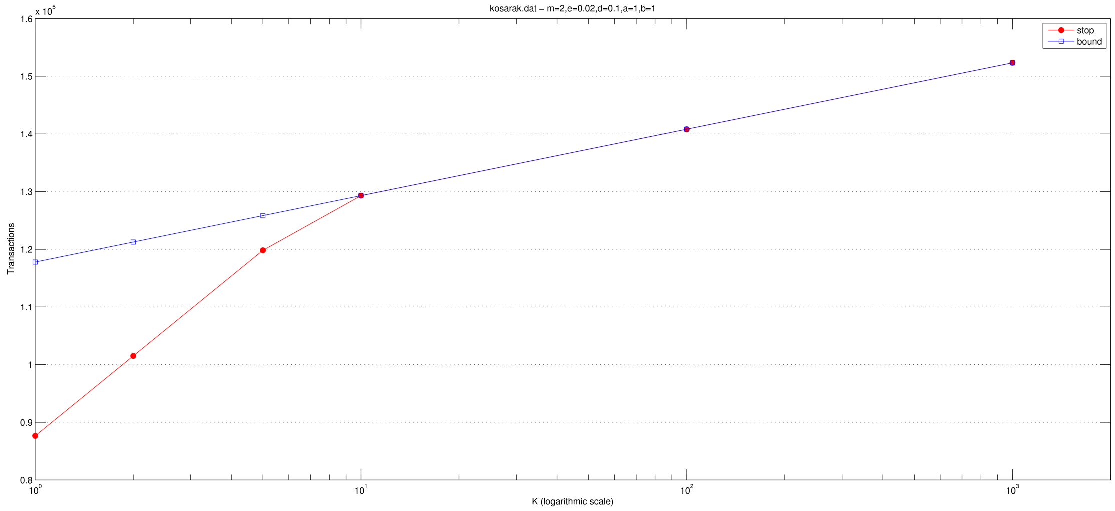


Figure 3.13: Results of algorithm of Section 3.2.3 with dataset Kosarak, for itemsets of length at most $\ell = 2$.

real, enormous dataset to be huge, we believe that this experiments provides a first indication of the possible effectiveness of our algorithm. However, a more in depth and accurate experimental study is required to understand in which scenarios our algorithm can provide good performance. Moreover, the experimental evaluation of algorithm proposed in Section 3.2.5 is still open.

3.2.8 Approximating Top-K Frequent Itemset with Frequencies

A stricter approximation of the set may require a confidence on the frequencies of each itemset in the output:

Definition 3.16. *Let $\varepsilon \in (0,1)$ be a real-valued parameter. An ε-approximation* with frequencies *to* TOPK$(\mathcal{D},\mathcal{I},K)$ *is a set W of K or more ordered pairs (X,f) such that $X \in 2^{\mathcal{I}}$ and $f \in [0,1]$ and for which the following properties hold:*

P1: *for each $(X,f) \in W$, $f_{\mathcal{D}}(X) \geq f_{\mathcal{D}}^{(K)} - \varepsilon$;*

P2: *for each $(X,f) \notin W$, $f_{\mathcal{D}}(X) < f_{\mathcal{D}}^{(K)} + \varepsilon$.*

P3: *for each $(X,f) \in W$, $|f - f_{\mathcal{D}}(X)| \leq \varepsilon$.*

Our algorithms provides a ε-approximation *with frequencies* to TOPK$(\mathcal{D},\mathcal{I},K)$ with probability at least $1-\delta$.

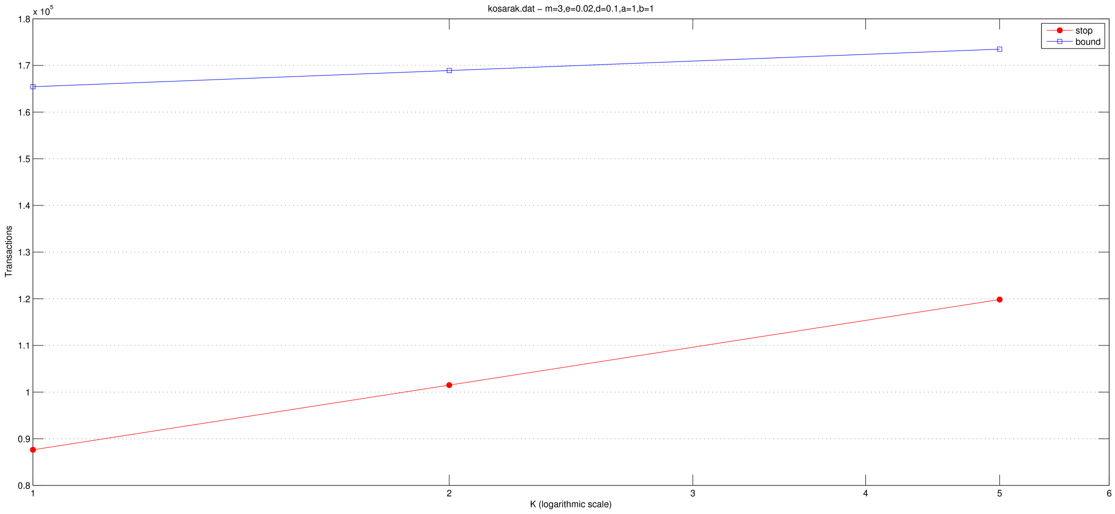


Figure 3.14: Results of algorithm of Section 3.2.3 with dataset Kosarak, for itemsets of length at most $\ell = 3$.

Theorem 3.17. *Let $\mathcal{S}$ be the sample for which our algorithm stops, and let* $W = \{(X, f_{\mathcal{S}}(X)) : X \in \mathrm{TOPK}(\mathcal{S}, \mathcal{I}, K)\}$*. If* $|W| \leq K(m-K)$*, with probability at least* $1-\delta$*, W is a ε-approximation* with frequencies *to* $\mathrm{TOPK}(\mathcal{D}, \mathcal{I}, K)$*.*

Proof. **P1** and **P2** are satisfied by the output of our algorithm, so we only need to consider **P3**.

When our algorithm stops at Phase $j < j_{\max}$, with probability at least $1-\delta$ we have that Lemma 3.8 holds. Since the itemsets returned by our algorithm are always a subset of B_0, for each itemset X in output we have:

$$|f_{\mathcal{S}}(X) - f_{\mathcal{D}}(X)| < \frac{\varepsilon}{2}.$$

If the algorithm stops at Phase $j = j_{\max}$ and the algorithm uses $\mathcal{D}$ to extract $\mathrm{TOPK}(\mathcal{D}, \mathcal{I}, K)$, **P3** trivially holds. If the algorithm stops at Phase $j = j_{\max}$ and the algorithm does not use $\mathcal{D}$, for each itemset X the Azuma bound gives:

$$\begin{aligned} \Pr(|f_{\mathcal{S}}(X) - f_{\mathcal{D}}(X)| \geq \varepsilon) &\leq 2e^{-\varepsilon^2 t_{\max}/2} && (3.6) \\ &\leq \frac{\delta}{K(m-K)}. && (3.7) \end{aligned}$$

Since $|W| \leq K(m-K)$, the union bound gives the desired result.

□

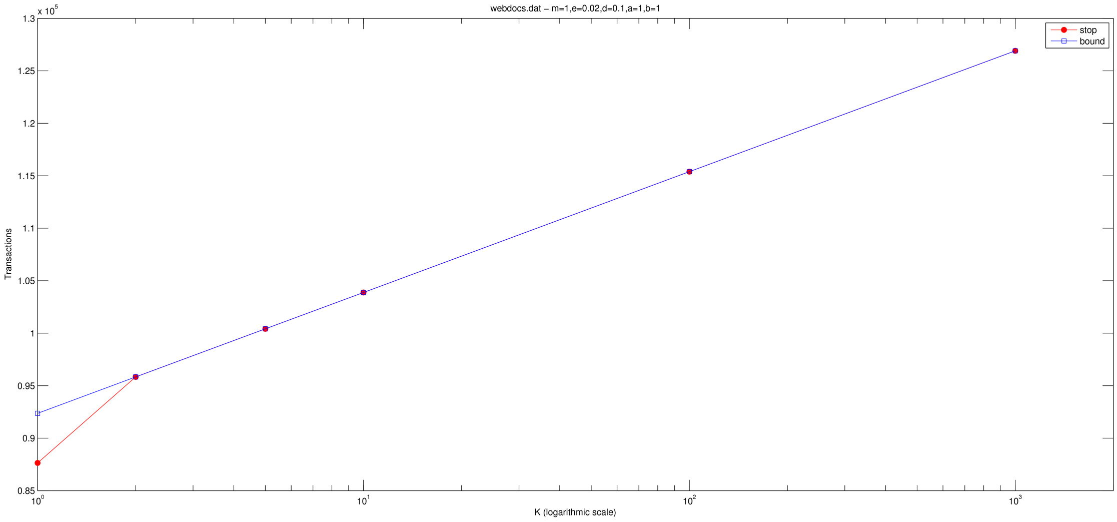


Figure 3.15: Results of algorithm of Section 3.2.3 with dataset webdocs, for itemsets of length at most $\ell = 1$.

Chapter 4

Finding Statistically Significant Frequent Itemsets

In this chapter we address the classical problem of mining frequent itemsets with respect to a certain minimum support threshold, and provide a rigorous methodology to establish a threshold that can guarantee, in a statistical sense, that the returned family of frequent itemsets contains significant ones with a limited false discovery rate. The results presented in the chapter appeared in [KMP$^+$09a, KMP$^+$09b]. Our methodology crucially relies on the following Poisson approximation result, which is the main theoretical contribution for the problem.

Consider a dataset $\mathcal{D}$ of t transactions on a set $\mathcal{I}$ of n items and let $\hat{\mathcal{D}}$ be a corresponding random dataset according to the a random model which will described in Section 4.1. Let $Q_{k,\sigma}$ be the number of itemsets of size k with support at least σ with respect to $\mathcal{D}$, and let $\hat{Q}_{k,\sigma}$ be the corresponding random variable for $\hat{\mathcal{D}}$. We show that there exists a minimum support value $\sigma_{\min}$ (which depends on the parameters of $\mathcal{D}$ and on k), such that for all $\sigma \geq \sigma_{\min}$ the distribution of $\hat{Q}_{k,\sigma}$ is well approximated by a Poisson distribution. Our result is based on a novel application of the Chen-Stein Poisson approximation method [AGG90].

The minimum support $\sigma_{\min}$ provides the grounds to devise a rigorous method for establishing a support threshold for mining significant itemsets, both reducing the overall complexity and improving the accuracy of the discovery process. Specifically, for a fixed itemset size k, we test a small number of support thresholds $\sigma \geq \sigma_{\min}$, and, for each such threshold, we measure the p-value corresponding to the null hypothesis H_0 that the observed value $Q_{k,\sigma}$ comes from a Poisson distribution of suitable expectation. From the tests we can determine a threshold σ^* such that, with user-defined confidence level α, the number of itemsets with support at least σ^* is not sampled from a Poisson distribution and is therefore statistically significant. The fact that the number of itemsets with support at least σ^* is statistically significant does not imply necessarily that each of the itemsets is significant. However, our test is also able to guarantee a user-defined upper bound β on the False Discovery Rate (FDR). We remark that our approach works for any fixed itemset size k, unlike traditional frequent itemset mining, where itemsets of all sizes are extracted for a given threshold.

To grasp the intuition behind the above approach, recall that a Poisson distribution models the number of occurrences among a large set of possible events, where the probability of each event is small. In the context of frequent itemset mining, the Poisson approximation holds when the probability that an individual itemset has support at least $\sigma_{\min}$ in $\hat{\mathcal{D}}$ is small, and thus the existence of such an event in $\mathcal{D}$ is likely to be statistically significant. We stress that our technique discovers statistically significant itemsets among those of relatively high support. In fact, if the expected supports of individual itemsets vary in a large range, there may exist itemsets with very low expected supports in $\hat{\mathcal{D}}$ which may have statistically significant supports in $\mathcal{D}$. These itemsets would not be discovered by our strategy. However, any mining strategy aiming at discovering significant, low-support itemsets is likely to incur high costs due to the large (possibly exponential) number of candidates to be examined, although only a few of them would turn out to be significant.

We validate our theoretical results by mining significant frequent itemsets from a number of real datasets that are standard benchmarks in this field. Also, we compare the effectiveness of our methodology to a standard multi-hypothesis approach based on [BY01], and provide evidence that the latter often returns fewer significant itemsets, which indicates that our method has considerably higher power.

The rest of the chapter is structured as follows. Section 4.1 introduces the random model employed in our approach. Section 4.2 presents the Poisson approximation result for the random variable $\hat{Q}_{k,\sigma}$. The methodology for establishing the support threshold σ^* is presented in Section 4.3, and experimental results are reported in Section 4.4.

4.1 The model

The significance of a discovery in our framework is assessed based on its deviation from what would be expected in a random dataset in which individual items are placed in transactions independently. Formally, let $\mathcal{D}$ denote the input dataset and n the number of items occurring in $\mathcal{D}$. Among all possible $\binom{n}{k}$ itemsets of size k (*k-itemsets*) we are interested in statistically significant ones, that is, those k-itemsets whose supports in $\mathcal{D}$ are significantly higher, in a statistical sense, than their expected supports in a corresponding random dataset.

As in [SBM98], we consider a probability space of datasets with the same number of transactions t, on the same set of items $\mathcal{I}$ as $\mathcal{D}$, and in which item i, of frequency f_i in $\mathcal{D}$, is included in any given transaction with probability f_i, independent of all other items and all other transactions. A similar model is used in [PVGG04] and [SVGP05] to evaluate the running time of algorithms for frequent itemsets mining. Let $\hat{\mathcal{D}}$ denote a random dataset from this probability space. For a given itemset X, the null hypothesis H_0 is that its support $s_{\mathcal{D}}(X)$ in $\mathcal{D}$ is drawn from the distribution of its support $s_{\hat{\mathcal{D}}}(X)$ in $\hat{\mathcal{D}}$. The alternative hypothesis H_1 is that $s_{\mathcal{D}}(X)$ is not drawn from that distribution, and in particular that there is a positive correlation between the occurrences of the individual items in X.

An alternative probability space of datasets, proposed in [GMMT07], considers all arrangements of n items to m transactions which match the exact item frequencies and transaction lengths as $\mathcal{D}$. Conceivably, the technique presented in this chapter could be adapted to this latter model as well.

4.2 Poisson approximation for $\hat{Q}_{k,\sigma}$

The Chen-Stein method [AGG90] is a powerful tool for bounding the error in approximating probabilities associated with a sequence of dependent events by a Poisson distribution. To apply the method to our case, we fix parameters k and σ, and define a collection of Bernoulli random variables $\{Z_X \mid X \subset \mathcal{I},\ |X| = k\}$, such that $Z_X = 1$ if the itemset X appears in at least σ transactions in the random dataset $\hat{\mathcal{D}}$, and $Z_X = 0$ otherwise. Also, let $p_X = \Pr(Z_X = 1)$. We are interested in the distribution of $\hat{Q}_{k,\sigma} = \sum_{X:|X|=k} Z_X$.

For each set X we define the *neighborhood set* of X,

$$I(X) = \{X' \mid X \cap X' \neq \emptyset, |X'| = |X|\}.$$

If $Y \notin I(X)$ then Z_Y and Z_X are independent. Adapting [AGG90, Theorem 1] to our case we have:

Theorem 4.1. *Let U be a Poisson random variable such that $\mathbf{E}[U] = \mathbf{E}[\hat{Q}_{k,\sigma}] = \lambda < \infty$. The variation distance between the distributions $\mathcal{L}(\hat{Q}_{k,\sigma})$ of $\hat{Q}_{k,\sigma}$ and $\mathcal{L}(U)$ of U*

is such that

$$\left\|\mathcal{L}(\hat{Q}_{k,\sigma}) - \mathcal{L}(U)\right\| = \sup_A |\Pr(\hat{Q}_{k,\sigma} \in A) - \Pr(U \in A)| \leq b_1 + b_2,$$

where

$$b_1 = \sum_{X:|X|=k} \sum_{Y \in I(X)} p_X p_Y$$

and

$$b_2 = \sum_{X:|X|=k} \sum_{X \neq Y \in I(X)} \mathbf{E}[Z_X Z_Y].$$

We can derive analytic bounds for b_1 and b_2 in many situations. Specifically, suppose that we generate t transactions in the following way. For each item x, we sample a random variable $R_x \in [0,1]$ independently from some distribution R. Conditioned on the R_x's, each item x occurs independently in each transaction with probability R_x. In what follows, we provide specific bounds for this situation that depend on the moment $\mathbf{E}[R^{2\sigma}]$ of the random variable R.

As a warm-up, we first consider the specific case where each R_x is a fixed value $p = \gamma/n$ for some constant γ for all x. That is, each item appears in each transaction with a fixed probability p, and the expected number of items per transaction is constant. The more general case follows the same approach, albeit with a few more technical difficulties.

Theorem 4.2. *Consider an asymptotic regime where as $n \to \infty$, we have that $k, \sigma = O(1)$ with $\sigma \geq 2$, each item appears in each transaction with probability $p = \gamma/n$ for some constant γ, and $t = O(n^c)$ for some positive constant $0 < c \leq (k-1)(1-1/\sigma)$. Let U be a Poisson random variable such that $\mathbf{E}[U] = \mathbf{E}[\hat{Q}_{k,\sigma}] = \lambda < \infty$. Then the variation distance between the distributions $\mathcal{L}(\hat{Q}_{k,\sigma})$ of $\hat{Q}_{k,\sigma}$ and $\mathcal{L}(U)$ of U satisfies*

$$\left\|\mathcal{L}(\hat{Q}_{k,\sigma}) - \mathcal{L}(U)\right\| = O(1/n^{2\sigma-2}).$$

Proof. For a given set X of k items, let $p_{X,i}$ be the probability that X appears in exactly i transactions, so that $p_X = \sum_{i=\sigma}^{t} p_{X,i}$ and

$$p_{X,i} = \binom{t}{i}\left(\frac{\gamma}{n}\right)^{ki}\left(1 - \left(\frac{\gamma}{n}\right)^k\right)^{t-i}.$$

Applying Theorem 4.1 gives

$$\left\|\mathcal{L}(\hat{Q}_{k,\sigma}) - \mathcal{L}(U)\right\| \leq b_1 + b_2$$

where

$$b_1 = \sum_{X:|X|=k} \sum_{Y\in I(S)} p_X p_Y$$

and

$$b_2 = \sum_{X:|S|=k} \sum_{Y\neq X\in I(S)} \mathbf{E}[Z_X Z_Y].$$

We now evaluate b_1 and b_2. A direct calculation easily gives the value for b_1 given in the statement of the theorem. For the asymptotic analysis, we write

$$\begin{aligned}
&\left(\binom{n}{k}^2 - \binom{n}{k}\binom{n-k}{k}\right)\\
&= \binom{n}{k}^2 \left(1 - \frac{\binom{n-k}{k}}{\binom{n}{k}}\right)\\
&= \binom{n}{k}^2 \left(1 - \prod_{i=0}^{k-1} \frac{n-k-i}{n-i}\right)\\
&= \Theta(n^k)^2 \cdot \Theta(1/n) = \Theta(n^{2k-1})
\end{aligned}$$

and

$$\begin{aligned}
p_{X,\sigma} &= \binom{t}{\sigma}\left(\frac{\gamma}{n}\right)^{k\sigma}\left(1 - \left(\frac{\gamma}{n}\right)^k\right)^{t-\sigma}\\
&= \Theta(t^\sigma)\cdot\Theta(n^{-k\sigma})\cdot(1+o(1)) = \Theta\left(t^\sigma n^{-k\sigma}\right),
\end{aligned}$$

where we have used the fact that $t = o(n^k)$ to obtain the asymptotics for the third term. Also, we note that for any $1 \leq i < t$

$$\frac{p_{X,i+1}}{p_{X,i}} = \frac{t-i}{i+1}\left(\frac{\gamma}{n}\right)^k \left(1 - \left(\frac{\gamma}{n}\right)^k\right)^{-1}$$

and so

$$\max_{i\in\{\sigma,\sigma+1,\ldots,t-1\}} \frac{p_{X,i+1}}{p_{X,i}} = O(tn^{-k}) = O(1/n).$$

Using a geometric series, it follows that

$$p_X = \sum_{i=\sigma}^{t} p_{X,i} = p_{X,\sigma}(1+o(1)) = \Theta\left(t^\sigma n^{-k\sigma}\right).$$

Thus, we obtain

$$\begin{aligned} b_1 &= \Theta(n^{2k-1}) \cdot \Theta\left(t^{\sigma} n^{-k\sigma}\right)^2 \\ &= \Theta(t^{2\sigma} n^{2k(1-\sigma)-1}) = \Theta(n^{2c\sigma+2k(1-\sigma)-1}). \end{aligned}$$

We now turn our attention to b_2. Consider sets $X \neq Y$ of k items, let $g = |X \cap Y|$, and suppose that $g > 0$. Then if $Z_X Z_Y = 1$, there exist disjoint subsets $A, B, C \in \{1, \ldots, t\}$ such that $0 \leq |A| \leq \sigma$, $|B| = |C| = \sigma - |A|$, all of the transactions in A contain both X and Y, all of the transactions in B contain X, and all of the transactions in C contain Y.

Therefore,

$$\mathbf{E}[Z_X Z_Y] \leq \sum_{i=0}^{\sigma} \binom{t}{i; \sigma - i; \sigma - i} \left(\frac{\gamma}{n}\right)^{(2k-g)i+2k(\sigma-i)},$$

where the notation $\binom{m}{x;y;z}$ is a shorthand for $\binom{m}{x}\binom{m-x}{y}\binom{m-x-y}{z}$.

It follows that

$$\begin{aligned} b_2 &\leq \sum_{g=1}^{k-1} \binom{n}{g; k-g; k-g} \sum_{i=0}^{\sigma} \binom{t}{i; \sigma - i; \sigma - i} \left(\frac{\gamma}{n}\right)^{(2k-g)i+2k(\sigma-i)} \\ &= \sum_{g=1}^{k-1} \binom{n}{g; k-g; k-g} \left(\frac{\gamma}{n}\right)^{2k\sigma} \sum_{i=0}^{\sigma} \binom{t}{i; \sigma - i; \sigma - i} \left(\frac{n}{\gamma}\right)^{gi} \\ &= \sum_{g=1}^{k-1} \binom{n}{g; k-g; k-g} \left(\frac{\gamma}{n}\right)^{2k\sigma} \sum_{i=0}^{\sigma} \binom{t}{i; \sigma - i; \sigma - i} \left(\frac{n}{\gamma}\right)^{gi} \\ &= \sum_{g=1}^{k-1} \Theta(n^{2k-g+2c\sigma}) \left(\frac{\gamma}{n}\right)^{2k\sigma} \sum_{i=0}^{\sigma} n^{-ic} \left(\frac{n}{\gamma}\right)^{gi} \\ &= \Theta(n^{2k(1-\sigma)+2c\sigma}) \sum_{g=1}^{k-1} n^{-g} \sum_{i=0}^{\sigma} \gamma^{-gi} n^{(g-c)i} \\ &= \Theta(n^{2k(1-\sigma)+2c\sigma}) \sum_{g=1}^{k-1} n^{-g} \begin{cases} \Theta(1) & g \leq c \\ \Theta(n^{(g-c)\sigma}) & g > c \end{cases} \\ &= \Theta(n^{2k(1-\sigma)+2c\sigma}) \cdot \Theta(n^{-(k-1)+(k-1-c)\sigma}) \\ &= \Theta(n^{2k(1-\sigma)+\sigma(k-1+c)-k+1}) \end{aligned}$$

Note that, in the summation where there are two cases depending on whether $g \leq c$ or $g > c$, we have used the assumption that $c \leq (k-1)(1 - 1/\sigma)$ to ensure the next equality. Finally, it is simple to check that both b_1 and b_2 are $O(1/n^{2\sigma-2})$

if $c \leq (k-1)(1-1/\sigma)$.

□

We now provide the more general theorem.

Theorem 4.3. *Consider an asymptotic regime where as $n \to \infty$, we have that $k, \sigma = O(1)$ with $\sigma \geq 2$, $\mathbf{E}[R^{2\sigma}] = O(n^{-a})$ for some constant $2 < a \leq 2\sigma$, and $t = O(n^c)$ for some positive constant c. Let U be a Poisson random variable such that $\mathbf{E}[U] = \mathbf{E}[\hat{Q}_{k,\sigma}] = \lambda < \infty$. If*

$$c \leq \frac{(k-1)(a-2) + \min(2a-6, 0)}{2\sigma},$$

then the variation distance between the distributions $\mathcal{L}(\hat{Q}_{k,\sigma})$ of $\hat{Q}_{k,\sigma}$ and $\mathcal{L}(U)$ of U satisfies

$$\left\|\mathcal{L}(\hat{Q}_{k,\sigma}) - \mathcal{L}(U)\right\| = O(1/n).$$

Proof. Applying Theorem 4.1 gives

$$\left\|\mathcal{L}(\hat{Q}_{k,\sigma}) - \mathcal{L}(U)\right\| \leq b_1 + b_2$$

where

$$b_1 = \sum_{X:|X|=k} \sum_{Y \in I(X)} p_X p_Y$$

and

$$b_2 = \sum_{X:|X|=k} \sum_{Y \neq X \in I(X)} \mathbf{E}[Z_X Z_Y].$$

We now evaluate b_1 and b_2. Letting $\vec{R}$ denote the vector of the R_x's, we have that for any set X of k items

$$\Pr(Z_X = 1 \mid \vec{R}) \leq \binom{t}{\sigma} \prod_{x \in X} R_x^{\sigma}.$$

Since the R_x's are independent with common distribution R,

$$p_X = \mathbf{E}[\Pr(Z_X = 1 \mid \vec{R})] \leq \binom{t}{\sigma} \mathbf{E}[R^{\sigma}]^k.$$

Using Jensen's inequality, we now have

$$\begin{aligned}
b_1 &= \sum_{X:|X|=k} \sum_{Y \in I(X)} p_X p_Y \\
&\leq \left(\binom{n}{k}^2 - \binom{n}{k}\binom{n-k}{k} \right) \binom{t}{\sigma}^2 \mathbf{E}[R^\sigma]^{2k} \\
&\leq \binom{n}{k}^2 \left(1 - \frac{\binom{n-k}{k}}{\binom{n}{k}} \right) \binom{t}{\sigma}^2 \mathbf{E}[R^{2\sigma}]^k \\
&= \binom{n}{k}^2 \left(1 - \prod_{i=0}^{k-1} \frac{n-k-i}{n-i} \right) \binom{t}{\sigma}^2 \mathbf{E}[R^{2\sigma}]^k \\
&= \Theta(n^k)^2 \cdot \Theta(1/n) \cdot O(n^{2c\sigma}) \cdot O(n^{-ka}) \\
&= O(n^{k(2-a)+2c\sigma-1})
\end{aligned}$$

We now turn our attention to b_2. Consider sets $X \neq Y$ of k items, and suppose $g = |X \cap Y| > 0$. If $Z_X Z_Y = 1$, there exist disjoint subsets $A, B, C \in \{1, \ldots, t\}$ such that $0 \leq |A| \leq \sigma$, $|B| = |C| = \sigma - |A|$, all of the transactions in A contain both X and Y, all of the transactions in B contain X, and all of the transactions in C contain Y. Therefore,

$$\begin{aligned}
\mathbf{E}[Z_X Z_Y \mid \vec{R}] &\leq \sum_{i=0}^{\sigma} \binom{t}{i; \sigma - i; \sigma - i} \left(\prod_{x \in X \cup Y} R_x^i \right) \left(\prod_{x \in X} R_x^{\sigma - i} \right) \left(\prod_{y \in Y} R_y^{\sigma - i} \right) \\
&= \sum_{i=0}^{\sigma} \binom{t}{i; \sigma - i; \sigma - i} \left(\prod_{x \in X \cap Y} R_x^{2\sigma - i} \right) \left(\prod_{x \in X - Y} R_x^{\sigma} \right) \left(\prod_{y \in Y - X} R_y^{\sigma} \right).
\end{aligned}$$

Applying independence of the R_x's and Jensen's inequality gives

$$\begin{aligned}
\mathbf{E}[Z_X Z_Y] &= \mathbf{E}[\mathbf{E}[Z_X Z_Y \mid \vec{R}]] \\
&\leq \sum_{i=0}^{\sigma} \binom{t}{i;\sigma-i;\sigma-i} \mathbf{E}[R^{2\sigma-i}]^{g} \mathbf{E}[R^{\sigma}]^{2(k-g)} \\
&\leq \sum_{i=0}^{\sigma} t^{2\sigma-i} \mathbf{E}[R^{2\sigma}]^{\frac{g(2\sigma-i)}{2\sigma}} \mathbf{E}[R^{2\sigma}]^{k-g} \\
&= \sum_{i=0}^{\sigma} t^{2\sigma-i} \mathbf{E}[R^{2\sigma}]^{k-ig/2\sigma} \\
&\leq O(1) \sum_{i=0}^{\sigma} n^{(2\sigma-i)c-a(k-ig/2\sigma)} \\
&= O(n^{2\sigma c-ak}) \sum_{i=0}^{\sigma} n^{i\left(\frac{ag}{2\sigma}-c\right)} \\
&= O\left(n^{2\sigma c-ak+\max\left\{0,\sigma\left(\frac{ag}{2\sigma}-c\right)\right\}}\right)
\end{aligned}$$

It follows that

$$\begin{aligned}
b_2 &\leq \sum_{g=1}^{k-1} \binom{n}{g;k-g;k-g} O\left(n^{2\sigma c-ak+\max\left\{0,\sigma\left(\frac{ag}{2\sigma}-c\right)\right\}}\right) \\
&= O(n^{2k+2\sigma c-ak}) \sum_{g=1}^{k-1} n^{-g} O\left(n^{\max\left\{0,\sigma\left(\frac{ag}{2\sigma}-c\right)\right\}}\right)
\end{aligned}$$

Now, for $2\sigma c/a < g < k$, we have (using the fact that $a \geq 2$)

$$n^{-g} n^{\max\left\{0,\sigma\left(\frac{ag}{2\sigma}-c\right)\right\}} = n^{g(\frac{a}{2}-1)-\sigma c} \leq n^{(k-1)(\frac{a}{2}-1)-\sigma c}.$$

Thus

$$b_2 = O(n^{2k+\sigma c-ak+(k-1)(\frac{a}{2}-1)}).$$

(Here we are using the fact that our choice of c satisfies $c \leq (k-1)(a-2)/2\sigma$ to ensure that $n^{(k-1)(\frac{a}{2}-1)-c\sigma} = \Omega(1)$.)

Now, we have

$$b_1 = O(1/n)$$

since

$$c \leq \frac{(k-1)(a-2)}{2\sigma} \leq \frac{k(a-2)}{2\sigma},$$

and

$$b_2 = O(1/n)$$

since

$$c \leq \frac{k(a-2)+(a-4)}{2\sigma}.$$

Thus

$$b_1 + b_2 = O(1/n).$$

□

It is easy to see that for fixed k, the quantities b_1 and b_2 defined in Theorem 4.1 are both decreasing in σ. In the following, we will use the notation $b_1(\sigma)$ and $b_2(\sigma)$ to indicate explicitly the dependence on σ. Therefore, for a chosen ϵ, with $0 < \epsilon < 1$, we can define

$$\sigma_{\min} = \min\{\sigma \geq 1 \ : \ b_1(\sigma) + b_2(\sigma) \leq \epsilon\}. \tag{4.1}$$

It immediately follows that for every σ in the range $[\sigma_{\min}, \infty)$, the variation distance between the distribution of $\hat{Q}_{k,\sigma}$ and the distribution of a Poisson variable with the same expectation is less than ϵ. In other words, for every $\sigma \geq \sigma_{\min}$ the number of k-itemsets with support at least σ is well approximated by a Poisson variable. Theorems 4.2 and 4.3, proved above, establish the existence of meaningful ranges of σ for which the Poisson approximation holds, under certain constraints on the individual item frequencies in the random dataset and on the other parameters.

4.2.1 A Monte Carlo method for determining $\sigma_{\min}$

While the analytical results of the previous subsection require that the individual item frequencies in the random dataset be drawn from a given distribution, in what follows we give experimental evidence that the Poisson approximation for the distribution of $\hat{Q}_{k,\sigma}$ holds also when the item frequencies are fixed arbitrarily, as is the case of our reference random model. More specifically, we present a method which approximates the support threshold $\sigma_{\min}$ defined by Equation 4.1, based on a simple Monte Carlo simulation which, given in input the parameters t and n of the input dataset $\mathcal{D}$, the vector $\vec{f}$ of item frequencies, k, Δ, and ϵ, returns estimates of $b_1(\sigma)$ and $b_2(\sigma)$. This approach is also convenient in practice since it avoids the inevitable slack due to the use of asymptotics in Theorem 4.3.

For a given configuration of item frequencies and number of transactions, let $\tilde{\sigma}$

be the maximum expected support of any k-itemset in a random dataset sampled according to that configuration, that is, the product of the k largest item frequencies. Conceivably, the value $b_1(\tilde{\sigma})$ is rather large, hence it makes sense to search for an $\sigma_{\min}$ larger than $\tilde{\sigma}$. For an integral parameter Δ (a suitable choice for Δ will be given below) we generate Δ random datasets and from each such dataset we mine all of the k-itemsets of support at least $\tilde{\sigma}$. Let W be the set of itemsets extracted in this fashion from all of the generated datasets. For each $\sigma \geq \tilde{\sigma}$ we can estimate $b_1(\sigma)$ and $b_2(\sigma)$ by computing for each $X \in W$ the empirical probability p_X of the event $Z_X = 1$, and for each pair $X, Y \in W$, with $X \cap Y \neq \emptyset$, the empirical probability $p_{X,Y}$ of the event $(Z_X = 1) \wedge (Z_Y = 1)$. The empirical probability of the event $Z_X = 1$ estimated with Δ (independent) random trials (in our case, generations of random datasets) is given by the ratio between the number of trials for which $Z_X = 1$ over Δ. The empirical probability $p_{X,Y}$ of the event $(Z_X = 1) \wedge (Z_Y = 1)$ is analogous. Once p_X and $p_{X,Y}$ have been estimated for all itemsets X, Y, we can estimate $b_1(\sigma)$ and $b_2(\sigma)$ with the formulas given in Theorem 4.1.

Note that for itemsets not in W these probabilities are estimated as 0. If it turns out that $b_1(\tilde{\sigma}) + b_2(\tilde{\sigma}) > \epsilon/4$, then we let $\hat{\sigma}_{\min}$ be the minimum $\sigma > \tilde{\sigma}$ such that $b_1(\sigma) + b_2(\sigma) \leq \epsilon/4$. Otherwise, if $b_1(\tilde{\sigma}) + b_2(\tilde{\sigma}) \leq \epsilon/4$, we repeat the above procedure starting from $\tilde{\sigma}/2$. (Based on the above considerations this latter case will be unlikely.) Algorithm 1 implements the above ideas.

The following theorem provides a bound on the probability that $\hat{\sigma}_{\min}$ be a conservative estimate of $\sigma_{\min}$, that is, $\hat{\sigma}_{\min} \geq \sigma_{\min}$.

Theorem 4.4. *If $\Delta = O\left(\log(1/\delta)/\epsilon\right)$, the output $\hat{\sigma}_{\min}$ of the Monte-Carlo process satisfies*

$$\Pr(b_1(\hat{\sigma}_{\min}) + b_2(\hat{\sigma}_{\min}) \leq \epsilon) \geq 1 - \delta.$$

Proof. Let assume $b_1(\hat{\sigma}_{\min}) + b_2(\hat{\sigma}_{\min}) > \epsilon$. Note that $b_1(\hat{\sigma}_{\min}) \leq b_2(\hat{\sigma}_{\min})$, therefore we have $b_2(\hat{\sigma}_{\min}) > \epsilon/2$. Let B be the random variable corresponding to Δ times the estimate of $b_2(\hat{\sigma}_{\min})$ obtained with Algorithm 1. Thus $E[B] > \Delta\epsilon/2$. Since Algorithm 1 returns $\hat{\sigma}_{\min}$ as estimate of $\sigma_{\min}$, we have that $B \leq \Delta\epsilon/4$. Let

$$\Delta = \frac{8\log(1/\delta)}{\epsilon},$$

and $c < 1$ be such that:

$$(1 - c)E[B] = \Delta\epsilon/4.$$

Since $E[B] > \Delta\epsilon/2$, we have $c \geq 1/2$. Using Chernoff bound, we have that:

$$\begin{aligned}\Pr(B \leq \Delta\epsilon/4) &\leq e^{-\frac{c^2 E[B]}{2}} \\ &\leq e^{-\frac{1}{4}\frac{8\log(1/\delta)}{2}} \leq \delta.\end{aligned}$$

Thus $\Pr(b_1(\hat{\sigma}_{\min}) + b_2(\hat{\sigma}_{\min}) > \epsilon) \leq \delta$. □

Algorithm 4.1: FindPoissonThreshold

```
   Input: $t$, $n$, vector $\vec{f}$ of item frequencies, $k$, $\Delta$, $\varepsilon$
   Output: Estimate $\hat{\sigma}_{\min}$ of $\sigma_{\min}$
 1 $\tilde{\sigma} \leftarrow$ highest expected support of a $k$-itemset;
 2 $\sigma_{\max} \leftarrow 0$;
 3 $W \leftarrow \emptyset$;
 4 for $i \leftarrow 1$ to $\Delta$ do
 5     $\hat{\mathcal{D}}_i \leftarrow$ random dataset with parameters $n$,$t$,$\vec{f}$;
 6     $W \leftarrow W \cup \left\{\text{frequent } k\text{-itemsets in } \hat{\mathcal{D}}_i \text{ w.r.t. } \tilde{\sigma}\right\}$;
 7 if $W = \emptyset$ then
 8     $\tilde{\sigma} \leftarrow \tilde{\sigma}/2$;
 9     goto 4;
10 if ($\sigma_{\max} = 0$) then
11     $\sigma_{\max} \leftarrow \max_{X \in W, \hat{\mathcal{D}}_i} \left\{\text{support of } X \text{ in } \hat{\mathcal{D}}_i\right\} + 1$;
12 for $\sigma \leftarrow \tilde{\sigma}$ to $\sigma_{\max}$ do
13     for all $X \in W$ do
14         $p_X(\sigma) \leftarrow$ empirical probability of $\{Z_X = 1\}$;
15     for all $X, Y \in W : X \cap Y \neq \emptyset$ do
16         $p_{X,Y}(\sigma) \leftarrow$ empirical probability of $\{Z_{X,Y} = 1\}$;
17     $b_1(\sigma) \leftarrow \sum_{X,Y \in W; Y \in I(X)} p_X(\sigma) p_Y(\sigma)$;
18     $b_2(\sigma) \leftarrow \sum_{X,Y \in W; X \neq Y \in I(X)} p_{X,Y}(\sigma)$;
19 if $b_1(\tilde{\sigma}) + b_2(\tilde{\sigma}) \leq \varepsilon/4$ then
20     $\sigma_{\max} \leftarrow \tilde{\sigma}$;
21     $\tilde{\sigma} \leftarrow \tilde{\sigma}/2$;
22     goto 3;
23 $\sigma_{\min} \leftarrow \min\{\sigma > \tilde{\sigma} : b_1(\sigma) + b_2(\sigma) \leq \varepsilon/4\}$;
24 return $\sigma_{\min}$;
```

4.3 Procedures for the Discovery of High-Support Significant Itemsets

For a given itemset size k, the value $\sigma_{\min}$ identifies a region of (relatively high) supports where we concentrate our quest for statistically significant k-itemsets. In this section we develop procedures to identify a family of k-itemsets (among those of support greater than or equal to $\sigma_{\min}$) which are statistically significant with a controlled FDR. More specifically, in Subsection 4.3.1 we show that a family with the desired properties can be obtained as a subset of the frequent k-itemsets with respect to $\sigma_{\min}$, selected based on a standard multi-comparison test. However, this procedure may incur in a large number of false negatives. To achieve higher effectiveness, in Subsection 4.3.2 we devise a more sophisticated procedure which identifies a support threshold $\sigma^* \geq \sigma_{\min}$ such that *all* frequent k-itemsets with respect to σ^* are statistically significant with a controlled FDR. In the next section we will provide experimental evidence that in many cases the latter procedure yields much fewer false negatives.

4.3.1 A procedure based on a standard multi-comparison test

We present a first, simple procedure to discover significant itemsets with controlled FDR, based on the following well-established result in multi-comparison testing. The following test can be used for any choice of the minimum support σ.

Theorem 4.5 ([BY01]). *Assume that we are testing for m null hypotheses. Let $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$ be the ordered observed p-values of the m tests. For a given parameter β, with $0 < \beta < 1$, define*

$$\ell = \max\left\{ i \geq 0 : p_{(i)} \leq \frac{i}{m \sum_{j=1}^{m} \frac{1}{j}} \beta \right\}, \tag{4.2}$$

and reject the null hypotheses corresponding to tests $(1), \ldots, (\ell)$. Then, the FDR for the set of rejected null hypotheses is upper bounded by β.

Let $\mathcal{D}$ denote an input dataset consisting of t transactions over n items, and let k be the fixed itemset size. First, we mine from $\mathcal{D}$ the set of frequent k-itemsets $\mathcal{F}_{(k)}(\sigma)$. Then, for each $X \in \mathcal{F}_{(k)}(\sigma)$, we test the null hypothesis H_0^X that the observed support of X in $\mathcal{D}$ is drawn from a Binomial distribution with parameters t and f_X (the product of the individual frequencies of the items of X), setting the rejection threshold as specified by condition (4.2), with parameters β and $m = \binom{n}{k}$.

Based on Theorem 4.5, the itemsets of $\mathcal{F}_{(k)}(\sigma)$ whose associated null hypothesis is rejected can be returned as significant, with FDR upper bounded by β. Since we are interested in itemsets whose supports is $\geq \sigma_{\min}$, we extract from $\mathcal{D}$ only the itemsets of support $\geq \sigma_{\min}$. The pseudocode Procedure 1 implements the strategy described above.

Procedure 1

```
Input: Dataset $\mathcal{D}$ of $t$ transactions over $n$ items, vector $\vec{f}$ of item frequencies, $k$,
    $\beta \in (0,1)$;
Output: Family of significant $k$-itemsets with FDR $\leq \beta$;
Determine $\sigma_{\min}$ and compute $\mathcal{F}_{(k)}(\sigma_{\min})$ from $\mathcal{D}$;
for all $X \in \mathcal{F}_{(k)}(\sigma_{\min})$ do
    $\sigma_X \leftarrow$ support of $X$ in $\mathcal{D}$;
    $f_X \leftarrow \Pi_{i \in X} f_i$;
    $p^{(X)} \leftarrow \Pr(\mathrm{Bin}(t, f_X) \geq \sigma_X)$;
Let $p_{(1)}, p_{(2)}, \ldots,$ be the sorted sequence of the values $p^{(X)}$, with $X \in \mathcal{F}_{(k)}(\sigma_{\min})$;
$m \leftarrow \binom{n}{k}$;
$\ell = \max\left\{0, i : p_{(i)} \leq \frac{i}{m \sum_{j=1}^{m} \frac{1}{j}} \beta\right\}$;
return $\left\{X \in \mathcal{F}_{(k)}(\sigma_{\min}) : p^{(X)} = p_{(i)}, 1 \leq i \leq \ell\right\}$;
```

4.3.2 Establishing a support threshold for significant frequent itemsets

Let α and β be two constants in $(0,1)$. We seek a threshold σ^* such that, with confidence $1-\alpha$, the k-itemsets in $\mathcal{F}_{(k)}(\sigma^*)$ can be flagged as statistically significant with FDR at most β. The threshold σ^* is determined through a robust statistical approach which ensures that the number $Q_{k,\sigma^*} = |\mathcal{F}_{(k)}(\sigma^*)|$ deviates significantly from what would be expected in a random dataset, and that the magnitude of the deviation is sufficient to guarantee the bound on the FDR.

Let $\sigma_{\min}$ be the minimum support such that the Poisson approximation for the distribution of $\hat{Q}_{k,\sigma}$ holds for $\sigma \geq \sigma_{\min}$, and let $\sigma_{\max}$ be the maximum support of an item (hence, of an itemset) in $\mathcal{D}$. Our procedure will performs h comparisons associated to supports $\sigma_i, 0 \leq i < h$, with $\sigma_{\min} \leq \sigma_i \leq \sigma_{\max}$. In the i-th comparison, with $0 \leq i < h$, we test the null hypothesis H_0^i that the observed value Q_{k,σ_i} is drawn from the same Poisson distribution as $\hat{Q}_{k,\sigma_i}$. We choose as σ^* the minimum of the σ_i's, if any, for which the null hypothesis H_0^i is rejected. If no null hypothesis is rejected, we set $\sigma^* = \infty$.

For the correctness of the above procedure, it is crucial to specify a suitable rejection condition for each H_0^i. Assume first that, for $0 \leq i < h$, we reject the null hypothesis H_0^i when the p-value of the observed value Q_{k,σ_i} is smaller than α_i, where the α_i's are chosen so that $\sum_{i=0}^{h-1} \alpha_i = \alpha$. Then, the union bound shows that the probability of rejecting any true null hypothesis is less than α. However, this approach does not yield a bound on the FDR for the set $\mathcal{F}_{(k)}(\sigma^*)$. In fact, some itemsets in $\mathcal{F}_{(k)}(\sigma^*)$ are likely to occur with high support even under H_0^i, hence they would represent false discoveries. The impact of this phenomenon can be contained by ensuring that the FDR is below a specified level β. To this purpose, we must strengthen the rejection condition, as explained below.

Fix suitable values $\beta_0, \beta_1, \ldots, \beta_{h-1}$ such that $\sum_{i=0}^{h-1} \beta_i^{-1} \leq \beta$. For $0 \leq i < h$, let $\lambda_i = E[\hat{Q}_{k,\sigma_i}]$. We now reject H_0^i when the p-value of Q_{k,σ_i} is smaller than α_i, *and* $Q_{k,\sigma_i} \geq \beta_i \lambda_i$. The following theorem establishes the correctness of this approach.

Theorem 4.6. *With confidence* $1-\alpha$, $\mathcal{F}_{(k)}(\sigma^*)$ *is a family of statistically significant frequent k-itemsets with FDR at most* β*.*

Proof. Observe that since $\sum_{i=0}^{h-1} \alpha_i \leq \alpha$, we have that all rejections are correct, with probability at least $1-\alpha$. Let E_i be the event *"H_0^i is rejected"* or equivalently, *"the p-value of Q_{k,σ_i} is smaller than α_i and $Q_{k,\sigma_i} \geq \beta_i \lambda_i$".* Suppose that H_0^i is the first rejected null hypothesis, for some index i, whence $\sigma^* = \sigma_i$. In this case, Q_{k,σ_i} itemsets are flagged as significant. We denote by V_i the number of false discoveries among these Q_{k,σ_i} itemsets. It is easy to argue that the expectation of V_i is upper bounded by $E[X_i|E_i, \bar{E}_{i-1}, \ldots, \bar{E}_0]$, where X_i is a Poisson variable with expectation λ_i. Since $Q_{k,\sigma_i} \geq \beta_i \lambda_i$ when H_0^i is rejected, by the law of total probability we have

$$\begin{aligned}
FDR &\leq \sum_{i=0}^{h-1} E\left[\frac{V_i}{Q_{k,\sigma_i}}\right] \Pr(E_i, \bar{E}_{i-1}, \ldots, \bar{E}_0) \\
&\leq \sum_{i=0}^{h-1} \frac{E\left[V_i\right]}{\beta_i \lambda_i} \Pr(E_i, \bar{E}_{i-1}, \ldots, \bar{E}_0) \\
&\leq \sum_{i=0}^{h-1} \frac{E[X_i \mid E_i \bar{E}_{i-1}, \ldots, \bar{E}_0]}{\beta_i \lambda_i} \Pr(E_i, \bar{E}_{i-1}, \ldots, \bar{E}_0) \\
&= \sum_{i=0}^{h-1} \frac{\sum_{j \geq 0} j \Pr(X_i = j, E_i, \bar{E}_{i-1}, \ldots, \bar{E}_0)}{\beta_i \lambda_i} \\
&\leq \sum_{i=0}^{h-1} \frac{\lambda_i}{\beta_i \lambda_i} = \sum_{i=0}^{h-1} \frac{1}{\beta_i} \leq \beta.
\end{aligned}$$

□

The method above needs the values h and $\sigma_i, 0 \leq i < h$ to be specified. Note that h influences how low a p-value must be to reject the corresponding null hypothesis. An high value of h would require very low p-value to reject an hypothesis, reducing the power of the method. We then choose to consider a number of hypothesis logarithmic in the difference $\sigma_{\max} - \sigma_{\min}$, and to set the corresponding σ_i with exponentially increasing steps. In our opinion this choice gives a good tradeoff between the number of tested supports and the diversity between the tested hypotheses, since we are testing more hypothesis for lower supports, where the number of itemsets is higher. In particular, we set $h = \lfloor \log_2(\sigma_{\max} - \sigma_{\min}) \rfloor + 1$ and $\sigma_0 = \sigma_{\min}$ and $\sigma_i = \sigma_{\min} + 2^i$, for $1 \leq i < h$.

The pseudocode Procedure 4.3 specifies more formally our approach to determine the support threshold σ^*. Note that estimates for the λ_i's needed in the for-loop of Lines 7-9 can be obtained from the same random datasets generated in Algorithm 4.1, which are used there for the estimation of $\sigma_{\min}$. In fact, since λ_i is the expected number of k-itemsets of support at least σ_i in a random dataset $\hat{D}$, we can estimate λ_i counting for each of the Δ random datasets generated by Algorithm 4.1 how many k-itemsets appears with support $\geq \sigma_i$.

Procedure 2

```
   Input: Dataset $\mathcal{D}$ of $t$ transactions over $n$ items, vector $\vec{f}$ of item frequencies, $k$,
          $\alpha, \beta \in (0,1)$;
   Output: $\sigma^*$ such that, with confidence $1-\alpha$, $\mathcal{F}_{(k)}(\sigma^*)$ is a family of significant
           $k$-itemsets with $FDR \leq \beta$
 1 Determine $\sigma_{\min}$ and compute $\mathcal{F}_{(k)}(\sigma_{\min})$ from $\mathcal{D}$;
 2 $i \leftarrow 0$;
 3 $\sigma_0 \leftarrow \sigma_{\min}$;
 4 $h \leftarrow \lfloor \log_2(\sigma_{\max} - \sigma_{\min}) \rfloor + 1$;
 5 Fix $\alpha_0, \dots, \alpha_{h-1} \in (0,1)$ s.t. $\sum_{i=0}^{h-1} \alpha_i = \alpha$;
 6 Fix $\beta_0, \dots, \beta_{h-1} \in (0,1)$ s.t. $\sum_{i=0}^{h-1} \beta_i^{-1} = \beta$;
 7 for $i \leftarrow 0$ to $h-1$ do
 8     Compute $\lambda_i = \mathbf{E}[\hat{Q}_{k,\sigma_i}]$;
 9 while $i < h$ do
10     Compute $Q_{k,\sigma_i}$;
11     $p_{\sigma_i} \leftarrow \Pr(\text{Poisson}(\lambda_i) \geq Q_{k,\sigma_i})$;
12     if ($p_{\sigma_i} \leq \alpha_i$) and $Q_{k,\sigma_i} \geq \beta_i \lambda_i$ then
13         return $\sigma^* \leftarrow \sigma_i$;
14     $\sigma_{i+1} \leftarrow \sigma_{\min} + 2^{i+1}$;
15     $i \leftarrow i + 1$;
16 return $\sigma^* \leftarrow \infty$ ;
```

Dataset	n	$[f_{\min}; f_{\max}]$	m	t
Retail	16470	[1.13e-05 ; 0.57]	10.3	88162
Kosarak	41270	[1.01e-06 ; 0.61]	8.1	990002
Bms1	497	[1.68e-05 ; 0.06]	2.5	59602
Bms2	3340	[1.29e-05 ; 0.05]	5.6	77512
Bmspos	1657	[1.94e-06 ; 0.60]	7.5	515597
Pumsb*	2088	[2.04e-05 ; 0.79]	50.5	49046

Table 4.1: Parameters of the benchmark datasets: n is the number of items; $[f_{\min}, f_{\max}]$ is the range of frequencies of the individual items; m is the average transaction length; and t is the number of transactions.

4.4 Experimental Results

In order to validate the methodology, a number of experiments have been performed on datasets which are standard benchmarks in the context of frequent itemsets mining. The main characteristics of the datasets we use are summarized in Table 4.1. A description of the datasets not introduced in Chapter 3 can be found in the FIMI Repository (`http://fimi.cs.helsinki.fi/data/`), where they are available for download.

First of all, we applied the Monte Carlo method of Subsection 4.2.1 to determine $\sigma_{\min}$: the ranges for which the Poisson approximation holds are reported in Subsection 4.4.1. We then applied our methodology to the benchmark datasets of Table 4.1: our findings are presented in Subsection 4.4.2. In Subsection 4.4.3, we compare the sets of significant itemsets reported by our methodology against those returned by the standard procedure to bound the FDR described in Subsection 4.3.1.

4.4.1 Range of σ for Poisson Approximation

For each dataset $\mathcal{D}$ of Table 4.1 and for itemset sizes $k = 2, 3, 4$, we applied Algorithm 4.1 setting $\Delta = 1,000$ and $\epsilon = 0.01$. The values of $\hat{\sigma}_{\min}$ we obtained are reported in Table 4.2 (we added the prefix "Rand" to each dataset name, to denote the fact that the dataset is random and features the same parameters as the corresponding real one).

4.4.2 Experiments on benchmark datasets

For each benchmark dataset in Table 4.1 and for $k = 2, 3, 4$, we apply Procedure 4.3 with $\alpha = \beta = 0.05$, and $\alpha_i = \beta_i^{-1} = 0.05/h$. The results are displayed in Table 4.3, where, for each dataset and for each value of k, we show: the support σ^* returned

	$\hat{\sigma}_{\min}$		
Dataset	$k = 2$	$k = 3$	$k = 4$
RandRetail	9237	4366	784
RandKosarak	273266	100543	20120
RandBms1	268	23	5
RandBms2	168	13	4
RandBmspos	76672	15714	2717
RandPumsb*	29303	21893	16265

Table 4.2: Values of $\hat{s}_{\min}$ for $\epsilon = 0.01$ and for $k = 2, 3, 4$, in random datasets with the same values of n, t, and with the same frequencies of the items as the corresponding benchmark datasets.

by Procedure 4.3, the number Q_{k,σ^*} of k-itemsets with support at least σ^*, and the expected number $\lambda(\sigma^*)$ of itemsets with support at least σ^* in a corresponding random dataset.

	$k = 2$			$k = 3$			$k = 4$		
Dataset	σ^*	Q_{k,σ^*}	$\lambda(\sigma^*)$	σ^*	Q_{k,σ^*}	$\lambda(\sigma^*)$	σ^*	Q_{k,σ^*}	$\lambda(\sigma^*)$
Retail	∞	0	0	∞	0	0	848	6	0.01
Kosarak	∞	0	0	∞	0	0	21144	12	0.01
Bms1	276	56	0.19	23	258859	0.06	5	27M	0.05
Bms2	168	429	0.73	13	36112	0.25	4	714045	0.01
Bmspos	∞	0	0	16226	22	0.01	2717	891	0.38
Pumsb*	29303	29	0.05	21893	406	0.35	16265	6293	1.37

Table 4.3: Results obtained by applying Procedure 4.3 with $\alpha = 0.05, \beta = 0.05$ and $k = 2, 3, 4$ to the benchmark datasets of Table 4.1.

We observe that for most pairs (dataset,k) the number of significant frequent k-itemsets obtained is rather small, but, in fact, at support σ^* in random instances of those datasets, less than two (often much less than one) frequent k-itemsets would be expected. These results provide evidence that our methodology not only defines significance on statistically rigorous grounds, but also provides the mining task with suitable support thresholds that avoid explosion of the output size (the widely recognized “Achilles’ heel” of traditional frequent itemset mining). This feature crucially relies on the identification of a region of “rare events” provided by the Poisson approximation. The discovery of significant itemsets with low support (not returned by our method) would require the extraction of a large (possibly exponential) number of itemsets, that would make any strategy aiming to discover these itemsets unfeasible. Instead, we provide an efficient method to identify, with high confidence level, the family of most frequent itemsets that are statistically significant without overwhelming the user with a huge number of discoveries.

There are, however, a few cases where the number of itemsets returned is still considerably high. Their large number may serve as a sign that the results call for further analysis, possibly using clustering techniques [XHYC05] or limiting the search to closed itemsets. For example, consider dataset Bms1 with $k = 4$ and the corresponding value $\sigma^* = 5$ from Table 4.3. Extracting the closed itemsets of support greater or equal to σ^* in that dataset revealed the presence of a closed itemset of cardinality 154 with support greater than 7 in the dataset. This itemset, whose occurrence by itself represents an extremely unlikely event in a random dataset, accounts for more than 22M non-closed subsets with the same support among the 27M reported as significant.

It is interesting to observe that the results obtained for dataset Retail provide further evidence for the conclusions drawn in [GMMT07], which suggested random behavior for this dataset (although the random model in that work is slightly different from ours, in that the family of random datasets also maintains the same transaction lengths as the real one). Indeed, no support threshold σ^* could be established for mining significant k-itemsets with $k = 2, 3$, while the support threshold σ^* identified for $k = 4$ yielded as few as 6 itemsets. However, the conclusion drawn in [GMMT07] was based on a qualitative assessment of the discrepancy between the numbers of frequent itemsets in the random and real datasets, while our methodology confirms the findings on a statistically sound and rigorous basis.

Observe also that for some other pairs (dataset,k) our procedure does not identify any support threshold useful for mining statistically significant itemsets. This is an evidence that, for the specific k and for the high supports considered by our approach, these datasets do not present a significant deviation from the corresponding random datasets.

Finally, in order to assess its robustness, we applied our methodology to random datasets. Specifically, for each benchmark dataset of Table 4.1 and for $k = 2, 3, 4$, we generated 100 random instances with the same parameters as those of the benchmark, and applied Procedure 4.3 to each instance, searching for a support threshold σ^* for mining significant itemsets. In Table 4.4 we report the number of times Procedure 4.3 was successful in returning a finite value for σ^*. As expected, the procedure returned $\sigma^* = \infty$, in *all cases* but for 2 of the 100 instances of the random dataset with the same parameters as dataset Pumsb* with $k = 2$. However, in these two latter cases, mining at the identified support threshold only yielded a very small number of significant itemsets (one and two, respectively).

	$\sigma^* < \infty$		
Dataset	$k = 2$	$k = 3$	$k = 4$
RandomRetail	0	0	0
RandomKosarak	0	0	0
RandomBms1	0	0	0
RandomBms2	0	0	0
RandomBmspos	0	0	0
RandomPumsb*	2	0	0

Table 4.4: Results for Procedure 4.3 with $\alpha = 0.05, \beta = 0.05$ for random versions of benchmark datasets; each entry reports the number of times, out of 100 trials, the procedure returned a finite value for σ^*.

4.4.3 Relative effectiveness of Procedures 4.2 and 4.3

In order to assess the relative effectiveness of the two procedures presented in the previous section, we applied them to the benchmark datasets of Table 4.1. Specifically, we compared the number of itemsets extracted using the threshold σ^* provided by Procedure 4.3, with the number of itemsets flagged as significant using the more standard method based on Benjamini and Yekutieli's technique (Procedure 4.2), imposing the same upper bound $\beta = 0.05$ on the FDR.

The results are displayed in Table 4.5, where for each pair (dataset,k), we report the cardinality of the family $\mathcal{R}$ of k-itemsets flagged as significant by Procedure 4.2, and the ratio $r = Q_{k,\sigma^*}/|\mathcal{R}|$, where Q_{k,σ^*} is the number of k-itemsets of support at least σ^*, which are returned as significant with the methodology of Subsection 4.3.2.

We observe that in all cases where Procedure 4.3 returned a finite value of σ^* the ratio r is greater than or equal to 1 (except for dataset Bms1 and $k = 2$, and dataset Bmspos and $k = 3$, where r is however very close to 1). Moreover, in some cases the ratio r is rather large. Since both methodologies identify significant k-itemsets among all those of support at least σ_{min}, these results provide evidence that the methodology of Subsection 4.3.2 is often more (sometimes much more) effective. The methodology succeeds in identifying more significant itemsets, since it evaluates the significance of the *entire* set $\mathcal{F}_{(k)}(\sigma^*)$ by comparing Q_{k,σ^*} to $\hat{Q}_{k,\sigma^*}$. In contrast, Procedure 4.2 must implicitly test considerably more hypotheses (corresponding to the significance all possible k-itemsets), thus the power of the test (1-Pr(Type-II error)) is significantly smaller.

Observe that the cases where $r = 0$ in Table 4.5 correspond to pairs (dataset,k) for which Procedure 4.3 returned $\sigma^* = \infty$, that is, the procedure was not able to identify a threshold for mining significant k-itemsets. Note, however, that in all of these cases the number of significant k-itemsets returned by Procedure 4.2 is extremely small (between 1 and 3). Hence, for these pairs, both methodologies indicate that there is very little significant information to be mined at high supports.

	$k = 2$		$k = 3$		$k = 4$	
Dataset	$\|\mathcal{R}\|$	r	$\|\mathcal{R}\|$	r	$\|\mathcal{R}\|$	r
Retail	3	0	3	0	6	1.0
Kosarak	1	0	1	0	12	1.0
Bms1	60	0.933	64367	4.441	219706	122.9
Bms2	429	1.0	25906	1.394	60927	11.72
Bmspos	2	0	23	0.957	891	1.0
Pumsb*	29	1.0	406	1.0	6288	1.001

Table 4.5: Results using Test 4.2 to bound the FDR with $\beta = 0.05$ for itemsets of support $\geq \sigma_{\min}$.

Chapter 5

Maximal Dense Motif in Biological Sequences

This chapter focuses on the discovery of *rigid motifs*, which contain blocks of solid characters (solid blocks) separated by one or more don't cares. A rigid motif is different from an extensible motif in that the latter can contain *spacers*, special characters that correspond to possibly more than one character of the input string. On the opposite, each don't care character correspond to a single character of the input string, so all occurrences of a rigid motif in the input string have the same length.

As discussed in Chapter 1, the significance of a motif has been traditionally assessed through its frequency, but the biological significance of a motif cannot be exclusively related to its frequency. In particular, some very frequently occurring motifs can deemed as non significant because of certain aspects of their structure, such as, for example, an excessive number of errors in their occurrences. A strategy that returns frequent motifs with a moderate number of don't cares can then presumably provide a more significant set of motifs.

We propose a novel approach for controlling the number of don't cares in rigid motifs. Specifically, we introduce the notion of *dense motif*, a frequent pattern where the fraction of solid characters is above a given threshold. Our density notion is more flexible and general than the one considered in [Par07, ACP09], since it allows for arbitrarily long runs of don't cares as long as the fraction of solid characters in the pattern is above the threshold. We define a natural notion of *maximality* for dense patterns and devise an efficient algorithm, called MADMX (pronounced *Mad Max*), which performs complete MAximal Dense Motif eXtraction from an input sequence, with respect to user-specified frequency and density thresholds.

The key technical result at the core of our extraction strategy is a closure property which affords the complete generation of all maximal dense motifs in a breadth-first fashion, through an *apriori*-like strategy [AS94], starting from a relatively small set of solid blocks, and then repeatedly applying a suitable combining operator, called *fusion*, to pairs of previously generated motifs. In this fashion, our strategy avoids the generation and consequent storage of intermediate patterns which are not in the output set, which ensures time and space complexities polynomial in the combined input and output sizes.

We performed a number of experiments on MADMX to assess the biological significance of maximal dense motifs and to compare MADMX against its most recent and close competitor VARUN [ACP09]. For the first objective, we used MADMX to extract maximal dense motifs from a number of human DNA fragments. We compared the motifs extracted against those in RepBase [JKP+05], the largest repository of repetitive patterns for eukaryotic species, using REPEATMASKER [SHG04], a popular tool for masking repetitive DNA. The experiments show that all of our returned motifs are occurrences of patterns in RepBase, and *fully* characterize the family of SINE/ALU repeats (and partially the LINE/L1 family). This provides evidence that the notion of density, when applied to rigid motifs, captures biological significance.

Next we compared the motifs produced by MADMX with the ones returned by VARUN using the z-score measure. We ran both algorithms on several families of DNA fragments, limiting VARUN to the generation of rigid motifs and setting the parameters so as to obtain comparable output sizes, with motifs listed by decreasing z-score. The experiments show that the top-m highest-ranking motifs returned by MADMX almost always feature higher z-scores than the corresponding top-m ones returned by VARUN, even for large values of m, with only a modest increase in running time, which may be partly due to the fact that coding of MADMX is yet to be optimized. In fairness, we must remark that VARUN deals also with extensible motifs while MADMX only targets rigid motifs.

This chapter is organized as follows. In Section 5.1 several technical definitions and properties of motifs with don't cares are given. Section 5.2 proves the closure property at the base of MADMX and provides a high-level description of the algorithm. In Section 5.3, the experimental validation of MADMX is presented.

5.1 Preliminary definitions and properties

Let Σ be an alphabet of m characters and let $s = s[0]s[1]\dots s[n-1]$ be a string of length n over Σ. We denote the length of s with $|s|$. We use $s[i\dots j]$ to denote the substring $s[i]\ s[i+1]\ \cdots\ s[j]$ of s, for $i \leq j$. Characters in Σ are also called *solid characters*. We use $\circ \notin \Sigma$ to denote a distinguished character called *wild card* or *don't care* character. Let ϵ denote the empty string. A *pattern* x is a string in $\{\epsilon\} \cup \Sigma \cup \Sigma(\Sigma \cup \{\circ\})^*\Sigma$. However, whenever necessary, we will assume that patterns are implicitly padded to their left and right with arbitrary sequences of don't care characters.

Given two patterns x, y we say that y is *more specific* than x, and write $x \preceq y$, iff for every $i \geq 0$ either $x[i] = y[i]$ or $x[i] = \circ$. Given two patterns x, y we say that *x occurs in y at position ℓ* iff $x \preceq y[\ell \ldots \ell + |x| - 1]$: we also say that *$y$ contains x*. For a string s, the *location list* $\mathcal{L}_x$ of a pattern x in s is the complete set of positions at which x occurs in s. We refer to $f(x) = |\mathcal{L}_x|$ as the *frequency* of pattern x in s. (Note that $f(\epsilon) = n$.) As in [Ukk07], the *translated representation* of the location list $\mathcal{L}_x = \{l_0, l_1, l_2, \ldots, l_k\}$ is $\tau(\mathcal{L}_x) = \{l_1 - l_0, l_2 - l_0, \ldots, l_k - l_0\}$. Given two patterns x, y, we say that *y subsumes x* in s if $f(x) = f(y)$ and y contains x. As a consequence, if y subsumes x then $\tau(\mathcal{L}_x) = \tau(\mathcal{L}_y)$. A pattern x is *maximal* if it is not subsumed by any other pattern y. (We observe that this notion of maximality coincides with that of [PCGS05].) Given a pattern x, its *maximal extension* $\mathcal{M}(x)$ is the maximal pattern that subsumes x, which can be shown to be unique [PCGS05].

In what follows, we call *solid block* a string in Σ^+ and a *don't care block* a string in $\circ^+$. Furthermore, given a pattern x, $\mathrm{dc}(x)$ denotes the number of don't care characters contained in x, while $\mathrm{sc}(x)$ denotes the number of solid characters in x.

Definition 5.1. *The* density $\delta(x)$ *of x is: $\delta(x) = sc(x)/|x|$. Given a (density) threshold ρ, $0 < \rho \leq 1$, we say that a pattern x is* dense *if $\delta(x) \geq \rho$.*

Note that a solid block is a dense pattern with respect to every threshold ρ. It is reasonable to concentrate the attention on dense patterns that are not subsumed by any other dense pattern, since they are the most interesting dense representatives in the equivalence classes induced by "sharing" the same translated representation; these representatives are defined below.

Definition 5.2. *A dense pattern x is a* maximal dense pattern *in s if it is not subsumed by any other dense pattern $x' \neq x$.*

Observe that a maximal dense pattern x needs not be a maximal pattern in the general sense, since $\mathcal{M}(x)$ might be a nondense pattern. However, every dense pattern x is subsumed by *at least* one maximal dense pattern. In fact, all of the maximal dense patterns that subsume x are dense substrings of $\mathcal{M}(x)$, namely, those that contain x and are not substrings of any other dense substring of $\mathcal{M}(x)$. We want to stress that there might be several maximal dense patterns that subsume x. As an example, for $\rho = 2/3$, the dense pattern $x = \texttt{B}$ in the string $S = \texttt{AdBeCfAgBhC}$

is subsumed by maximal dense patterns $\mathtt{A} \circ \mathtt{B}$ and $\mathtt{B} \circ \mathtt{C}$ that are not maximal patterns, while $\mathcal{M}(x) = \mathtt{A} \circ \mathtt{B} \circ \mathtt{C}$ is not dense.

Definition 5.3. *Given a frequency threshold σ and a density threshold ρ, a pattern x is a* dense maximal motif *in s if x is a maximal dense pattern in s with respect to ρ, and $f(x) \geq \sigma$. A dense maximal motif for $\rho = 1$ is also referred to as* maximal solid block. *In the rest of the chapter, we will omit referencing the input string s when clear from the context.*

The problem we tackle is then the following: we are given an input string s, a frequency threshold σ, and a density threshold ρ, and we want to find all the maximal dense motifs in s. We restrict our attention to dense motifs because the notion of density provides a more general way to control the number of don't cares that appear in a motif, and the number of don't cares in a motif is related to its biological significance.

An important property of maximal dense patterns, which we will exploit in our mining strategy, is that all of their solid blocks are maximal solid blocks. This property is stated in the following proposition.

Proposition 5.4. *Let x be a maximal dense pattern with respect to a density threshold ρ, and let $b = x[i \dots j]$ be a solid block in x such that $x[i-1] = x[j+1] = \circ$ and $j \geq i$. Then, b is a maximal solid block.*

Proof. For the sake of contradiction, assume that b is not a maximal solid block. Consider $\mathcal{M}(x)$ and let $\tilde{x} = \mathcal{M}(x)[\ell_1 \dots \ell_2]$ be the shortest substring of $\mathcal{M}(x)$ subsuming x made of complete solid blocks, that is, such that with $\mathcal{M}(x)[\ell_1 - 1] = \mathcal{M}(x)[\ell_2 + 1] = \circ$. By known results [Ukk07, Pis02], all complete solid blocks in $\mathcal{M}(x)$, hence in $\tilde{x}$, are maximal solid blocks. Thus $\tilde{x}$ contains more solid characters than x, and no more don't cares than x. This implies that $\tilde{x}$ is strictly denser that x. This contradicts the hypothesis that x is maximal dense with respect to ρ. □

5.2 An algorithm for MAximal Dense Motif eXtraction

In this section we describe an algorithm, called MADMX (pronounced *Mad Max*), for MAximal Dense Motif eXtraction. The algorithm adopts a breadth-first *apriori*-like strategy [AS94], similar in spirit to the one developed in [ACP09], using maximal solid blocks as building blocks by virtue of Proposition 5.4. MADMX operates by repeatedly combining together, in a suitable fashion, pairs of maximal dense motifs, and extracting from the combinations less frequent maximal dense motifs.

A key notion for the algorithm, underlying the aforementioned combining operations, is the *fusion* of characters/patterns.

Definition 5.5. *Given three characters $c, c_1, c_2 \in \Sigma \cup \{\circ\}$, we say that c is the fusion of c_1 and c_2, and write $c = c_1 \bigtriangledown c_2$, if one of the following holds:*

1. $c = c_1 = c_2$;
2. $c_1 = \circ$, $c = c_2 \neq \circ$;
3. $c = c_1 \neq \circ$, $c_2 = \circ$.

Observe that if $c_1, c_2 \in \Sigma$ and $c_1 \neq c_2$, $c_1 \bigtriangledown c_2$ is not defined.

The above notion of fusion generalizes to patterns as follows.

Definition 5.6. *Given three patterns x, y, z and an integer d, we say that z is the* d-fusion of x and y, *and write $z = x \bigtriangledown_d y$, if z can be obtained by removing the leading and trailing don't care characters from the pattern m defined as $m[i] = x[i+d] \bigtriangledown y[i]$, for all indices i.*

Note that if $d > |x|$ we have $x \bigtriangledown_d y = x \circ^{d'} y$ for $d' = d - |x|$, while if $d < -|y|$ we have $x \bigtriangledown_d y = y \circ^{d''} x$ for $d'' = -d - |y|$.

The breadth-first strategy adopted by our algorithm crucially relies on the following theorem, which highlights the structure of dense motifs:

Theorem 5.7. *Let x be a maximal dense motif with $dc(x) > 0$. Then:*

(a) there exists a maximal solid block b in x such that $\mathcal{M}(x) = \mathcal{M}(b)$, or

(b) there exist two maximal dense motifs y_1, y_2 such that:

- $\mathcal{M}(x) = \mathcal{M}(y_1 \bigtriangledown_d y_2)$, *for some d;*
- *there are two maximal solid blocks b_1, b_2 in x and an integer $\hat{d} > 0$ such that b_1 is a maximal solid block in y_1, b_2 is a maximal solid block in y_2, and $b_1 \circ^{\hat{d}} b_2$ is contained in $y_1 \bigtriangledown_d y_2$;*
- $f(x) < \min\{f(y_1), f(y_2)\}$;

For the proof of Theorem 5.7 we need to define another type of pattern combination, namely the operation of *merge* between two patterns, which is similar to the one introduced in [PCGS05]. Given two characters c_1, c_2, we define the operator $\oplus$ between them such that $c_1 \oplus c_2 = \circ$, if $c_1 \neq c_2$, and $c_1 \oplus c_2 = c_1 = c_2$, otherwise.

Definition 5.8. *Given two patterns x, y and an integer d, the d-merge of x and y is the pattern $z = x \oplus_d y$ which can be obtained by removing all leading and trailing don't cares from the pattern m defined as $m[i] = x[i+d] \oplus y[i]$ for all i.*

We want to stress the difference between the notions of merging and fusion: the merge of two patterns x, y is always well defined and more general than x, y, while the fusion of x, y may not exist and, if it does, is more specific than x, y.

For the proof of Theorem 5.7 we also need the property established by the following lemma.

Lemma 5.9. *Let x and y be maximal patterns, and d be an integer such that $z = x \oplus_d y \neq \epsilon$. Then z is a maximal pattern. Moreover, if $z \neq x$ (resp., $z \neq y$) then $f(z) > f(x)$ (resp., $f(z) > f(y)$).*

Proof. First we prove that z is maximal. By contradiction, suppose that this is not the case. Then, there exists a position i such that $z[i] = \circ$ and we can replace the $\circ$ with a solid character c without decreasing the frequency of the pattern. (Note that the position of the substitution can be to the left of the first solid character in z or to the right of the last character in z.) Since x and y are more specific than z, to every occurrence of x and y in the string corresponds an occurrence of z. Hence, every occurrence of x (resp., y) in the string, contains c in its $i+d$th (resp., ith) position. Therefore, by maximality of x and y, it must be $z[i] = x[i+d] = y[i] = c$, which is a contradiction. The relations between the frequencies of x, y and z follow trivially by their maximality. □

We are now ready to prove the theorem.

Proof.[Theorem 5.7] Given a pattern x and two nonnegative integers $i \leq j$, we let $x^*[i \dots j]$ denote the pattern obtained by removing all the leading and trailing don't care characters from $x[i \dots j]$. Since x is a maximal dense pattern and $dc(x) > 0$, it is easy to see that there exist two dense patterns x_1, x_2 and an integer $d > 0$ such that $x = x_1 \circ^d x_2$, hence there exists an index $s_1 > 0$ such that $x^*[0 \dots s_1 - 1]$ and $x^*[s_1 + 1 \dots |x| - 1]$ are dense. We call these two patterns the *level-1 decomposition* of x (observe that many such decompositions may exist). Also, we let $\ell_1 = 0$ and $r_1 = |x| - 1$. Now, consider the following iterative process:

1. If in the level-i decomposition of x both $x^*[\ell_i \dots s_i - 1]$ and $x^*[s_i + 1 \dots r_i]$ have frequency strictly greater than $f(x)$, *or* at least one of $x^*[\ell_i \dots s_i - 1]$ and $x^*[s_i + 1 \dots r_i]$ is a solid block with frequency equal to $f(x)$, then terminate;

2. Otherwise, let $y = x^*[\ell_{i+1} \dots r_{i+1}]$ be (an arbitrary) one of $x^*[\ell_i \dots s_i - 1]$ or $x^*[s_i + 1 \dots r_i]$ which is not a solid block and has frequency equal to $f(x)$. Since y is dense, there exists an index s_{i+1}, $\ell_{i+1} < s_{i+1} < r_{i+1}$ such that $x^*[\ell_{i+1} \dots s_{i+1} - 1]$ and $x^*[s_{i+1} + 1 \dots r_{i+1}]$ are both dense. Call these two patterns the level-$(i+1)$ decomposition of x. Set $i = i + 1$ and go to Step 1.

Assume that the decomposition process ends by finding a solid block b that is a solid block in x and has $f(b) = f(x)$. Then, $\mathcal{M}(b) = \mathcal{M}(x)$ and the theorem follows. Otherwise, at the last level j of the decomposition, we have that $f(x) < \min\{f(x^*[\ell_j \dots s_j - 1]), f(x^*[s_j + 1 \dots r_j])\}$. In this latter case, as explained in Section 5.1 (after Definition 5.2), we can determine two maximal dense patterns y_1, y_2 such that y_1 contains $x^*[\ell_j \dots s_j - 1]$, y_2 contains $x^*[s_j + 1 \dots r_j]$, and with $\mathcal{M}(y_1) = \mathcal{M}(x^*[\ell_j \dots s_j - 1])$ and $\mathcal{M}(y_2) = \mathcal{M}(x^*[s_j + 1 \dots r_j])$. Since $f(y_1) = f(x^*[\ell_j \dots s_j - 1])$ and $f(y_2) = f(x^*[s_j + 1 \dots r_j])$, we have that $f(x) < \min\{f(y_1), f(y_2)\}$. Observe that by construction there must exist two solid blocks b_1, b_2 in x and an integer $\hat{d}$ such that b_1 is a solid block in y_1, b_2 is a solid block in y_2, and $b_1 \circ^{\hat{d}} b_2$ is a sequence of two solid blocks in x. In fact, b_1 (resp., b_2) is the last (resp., the first) solid block of $x^*[\ell_j \dots s_j - 1]$ (resp., $x^*[s_j + 1 \dots r_j]$).

Next, we show that there exists a d such that the d-fusion $y_1 \bigtriangledown_d y_2$ is well defined, contains $b_1 \circ^{\hat{d}} b_2$, and $\mathcal{M}(y_1 \bigtriangledown_d y_2) = \mathcal{M}(x)$. We proceed as follows. Let us “align” $\mathcal{M}(x)$ and y_1 so to match the occurrences of b_1 in both patterns. Then, for a certain integer p, $\mathcal{M}(x)[i+p]$ corresponds to $y_1[i]$. Assume, for the sake of contradiction, that there exists an index j such that $\mathcal{M}(x)[j+p]$ is not more specific than $y_1[j]$. Then, Lemma 5.9 implies that $z = \mathcal{M}(x) \oplus_p \mathcal{M}(y_1) \neq \mathcal{M}(y_1)$, which contains $x^*[\ell_j \dots s_j - 1]$, is maximal and has frequency strictly greater than $f(y_1)$, which is impossible because we have chosen y_1 such that $\mathcal{M}(x^*[\ell_j \dots s_j - 1]) = \mathcal{M}(y_1)$ and therefore $f(x^*[\ell_j \dots s_j - 1]) = f(y_1)$. Therefore, $\mathcal{M}(x)$ contains y_1. A similar argument shows that $\mathcal{M}(x)$ contains y_2.

Since y_1 and y_2 are contained in $\mathcal{M}(x)$, there must exist a d such that $y_1 \bigtriangledown_d y_2$ is well defined and can be aligned with $\mathcal{M}(x)$ in such a way to match the blocks b_1 and b_2 of y_1 and y_2 with the corresponding blocks in $\mathcal{M}(x)$. Moreover, $\mathcal{M}(x)$ contains $y_1 \bigtriangledown_d y_2$, hence $f(y_1 \bigtriangledown_d y_2) \geq f(\mathcal{M}(x)) = f(x)$. However, since $y_1 \bigtriangledown_d y_2$ contains both $x^*[\ell_j \dots s_j - 1]$ and $x^*[s_j + 1 \dots r_j]$, it contains also $x^*[\ell_j \dots r_j]$, which, by the decomposition process, has frequency equal to $f(x)$. Therefore, $f(y_1 \bigtriangledown_d y_2) \leq f(x)$, and the theorem follows since $f(y_1 \bigtriangledown_d y_2) = f(x)$. □

In essence, Theorem 5.7 guarantees that we can find any maximal dense motif x either within $\mathcal{M}(b)$, for some maximal solid block b, or by d-fusing two higher-frequency maximal dense motifs y_1, y_2, for some d, finding $z = \mathcal{M}(y_1 \bigtriangledown_d y_2)$ and then possibly "trimming" z on both sides to obtain x. Also, the theorem shows that in the latter case the trimmed sequence must contain at least one maximal solid block b_1 of y_1 and one maximal solid block b_2 of y_2. Moreover, we can disregard those d-fusions $y_1 \bigtriangledown_d y_2$ for which no pair of dense subsequences b_1 of y_1 and b_2 of y_2 exists such that $b_1 \circ^{\hat{d}} b_2$ contained in $y_1 \bigtriangledown_d y_2$ for some $\hat{d} > 0$.

Algorithm 5.1: MADMX

```
   Input: String $s$, frequency threshold $\sigma$, density threshold $\rho$
   Output: Maximal dense motifs
1  previous ← $\emptyset$, current ← $\emptyset$, next ← $\emptyset$ ;
2  blocks ← maximal solid blocks of $s$ with frequency $\geq \sigma$;
3  for each $b \in$ blocks do
4     find $\mathcal{M}(b)$ ;
5     current ← current$\cup$ extractMaximalDense($\mathcal{M}(b)$);
6  while current $\neq \emptyset$ do
7     for each $x_1 \in$ current do
8        for each $x_2 \in$ previous $\cup$ current do
9           for each $d$ s.t. $z = x_1 \bigtriangledown_d x_2$ is a valid fusion do
10             find $\mathcal{M}(z)$;
11             $\mathcal{DM}$ ← extractMaximalDense($\mathcal{M}(z)$);
12             for each $x \in \mathcal{DM}$ do
13                if $f(x) \geq \sigma$ and $x \notin$ previous $\cup$ current then next ← next $\cup \{x\}$;
14    previous ← previous $\cup$ current;
15    current ← next; next ← $\emptyset$;
16 return previous;
```

MADMX implements the strategy inspired by Theorem 5.7 and pseudocode is given below as Algorithm 5.1. It employs three (initially empty) sets *previous*, *current*, and *next*. In Line 2, the algorithm first stores the maximal solid blocks b in s for the given frequency in the set *blocks* (see Section 5.1). Then, it extracts all of the appropriate maximal dense motifs from $\mathcal{M}(b)$ in lines 3–5, using the function extractMaximalDense, as implied by Theorem 5.7(a). Finally, lines 6–15 implement the strategy as implied by Theorem 5.7(b). (In Line 9 a d-fusion $y_1 \bigtriangledown_d y_2$ is considered *valid* if it satisfies the second property of Theorem 5.7(b).) Given a maximal motif x, extractMaximalDense returns all the maximal dense motifs in x which satisfy the second condition of Theorem 5.7. In practice, when called on Line 5, it returns all the maximal dense substrings of $\mathcal{M}(b)$ that contains b. When called on

Line 11, the maximal motif passed in input will be $x = \mathcal{M}(y_1 \bigtriangledown_d y_2)$. In this case extractMaximalDense returns all the maximal dense substrings of x that satisfy the second property of Theorem 5.7(b), and thus contain at least one block b_1 of y_1 and at least one block b_2 of y_2.

The correctness of Algorithm MADMX is proved by the following.

Theorem 5.10. *Given a string s, frequency threshold σ and density threshold ρ, Algorithm* MADMX *produces in output all the maximal dense motifs in s.*

Proof. Let assume that there exists a maximal dense motif x that is not returned by MADMXṠince MADMX produces all the maximal dense motifs that can be generated from $\mathcal{M}(b)$, where b is a maximal solid block (lines 3–6), if x is not produced in output then there exists a pair of maximal dense motifs y_1, z_1 such that x can be found from $\mathcal{M}(y_1 \bigtriangledown_d z_1)$, where y_1, z_1 satisfy the properties of Theorem 5.7(b), such that one of y_1, z_1 is not produced by MADMXL̇et assume that y_1 is the maximal dense motif not produced by MADMXẆe can apply the same reasoning to y_1, thus we can find another maximal dense motif y_2 not produced by MADMXİterating this reasoning, we can find a sequence $y_1, y_2, \ldots, y_i, \ldots$ of dense motifs such that (i) $\forall i, y_i$ are maximal dense motifs (ii) $f(y_{i+1}) > f(y_i)$, and (iii) y_i is derived from the fusion of y_{i+1} with another maximal dense motif Theorem 5.7 implies that this sequence must be finite, and that the last element of this sequence, $\tilde{y}$, is either a solid block or can be found in the maximal extension of a solid block. Therefore $\tilde{y}$ has been generated by the algorithm (lines 3–5), that is a contradiction. □

An important issue for the efficiency of MADMX is that it needs to compute the exact frequency of each generated pattern. For what concerns the fusion operation of two patterns x_1, x_2 in Line 10, observe that a simple computation on the pairs $(\ell_1, \ell_2) \in \mathcal{L}_{x_1} \times \mathcal{L}_{x_2}$ is sufficient to yield the frequencies of all the valid fusions of two patterns. However, given $z = x_1 \bigtriangledown_d x_2$, for a maximal dense pattern w which does not contain z in its entirety, we can only conclude that $f(w) \geq f(z)$.

Therefore, in the course of the algorithm we generate two classes of maximal dense motifs: those whose exact frequencies are known (*final* motifs), and those for which only a lower bound to their frequencies is known (*tentative* motifs). Algorithm 5.1 is modified accordingly, requiring that x_1 and x_2 in lines 8 and 9 of the pseudocode be final. Whenever the set *current* contains no final motifs,we can label as final the motif in *current* with the highest lower bound to its frequency, and continue with the generation. The correctness of this assumption is proved by the following.

Theorem 5.11. *Let x be the tentative motif x with the highest lower bound $lb(x)$ on its frequency $f(x)$ when current does not contain any final motif. Then $f(x) = lb(x)$.*

Proof. For the sake of contradiction, assume that $f(x) \neq lb(x)$. In particular, it must be $f(x) > lb(x)$. From Theorem 5.7 we know that there must be two dense motifs x_1, y_1 with $\min\{f(x_1), f(y_1)\} > f(x)$ and an integer d such that x can be obtained, with its exact frequency, from $\mathcal{M}(x_1 \bigtriangledown_d y_1)$. If both x_1 and y_1 have already been moved to the *previous* list from Algorithm 5.1, we have $f(x) = lb(x)$. The only possibility is then that at least one of x_1 and y_1 has not been moved to *previous*. Let x_1 be this dense motif. Then x_1 is either a *tentative* motif or has not been generated by any fusion yet. Applying the same reasoning to x_1, we have that there exists two dense motifs x_2, y_2 such that at least one of them (let say x_2) has not been put in *previous*, $\min\{f(x_2), f(y_2)\} > f(x_1)$ and x_1 can be obtained, with its real frequency, from a valid fusion of x_2, y_2. Iterating this reasoning, we can find a sequence $x_1, x_2, \ldots, x_i, \ldots$ of dense motifs such that (i) $\forall i, x_i$ has not been put in *previous*, (ii) $f(x_{i+1}) > f(x_i)$, and (iii) x_i is derived from the fusion of x_{i+1} with another pattern. Theorem 5.7 implies that this sequence must be finite, and that the last element of this sequence, $\tilde{x}$, is either a solid block or can be found in the maximal extension of a solid block. Therefore $\tilde{x}$ has been generated by the algorithm (lines 3–5) with its correct frequency, thus it is in *previous*, that is a contradiction. □

A crude upper bound on the running time of MADMX can be derived by observing that, for each pair of dense maximal motifs in output, the time spent during all the operations concerning that pair is (naively) $O(n^3)$, where n is the length of the input string. If P patterns are produced in output, the overall time complexity is $O(n^3 P^2)$.

5.3 Experimental validation of MADMX

We developed a first, non-optimized, implementation of MADMX in `C++` also including an additional feature which eliminates, from the set of initial maximal solid blocks, those shorter than a given threshold min_ℓ. The purpose of this latter heuristics is to speed up motif generation driving it towards the discovery of (possibly) more significant motifs, with the exclusion of spurious, low-complexity ones. (The code is available for download at `http://www.dei.unipd.it/wdyn/?IDsezione=4534`.)

We performed two classes of experiments to evaluate how significant is the set of motifs found using our approach. The first class of experiments, described in Section 5.3.1, compares our motifs with the known biological repetitions available in `RepBase` [JKP+05], a very popular genomic database. The second class of experiments, described in Section 5.3.2, aims at comparing the motifs extracted by MADMX with those extracted by VARUN using the same z-score metric employed in [ACP09] for assessing their relative statistical significance.

5.3.1 Evaluating significance by known biological repetitions

RepBase [JKP$^+$05] is one of the largest repositories of prototypic sequences representing repetitive DNA from different eukaryotic species, collected in several different ways. RepBase is used as a reference collection for masking and annotation of repetitive DNA through popular tools such as REPEATMASKER [SHG04]. REPEATMASKER screens an input DNA sequence s for simple repeats and low complexity portions, and it uses RepBase to screen for interspersed repeats. Sequence comparisons are performed through Smith-Waterman scoring. REPEATMASKER returns a detailed annotation of the repeats occurring in s, and a modified version of s in which all of the annotated repeats are masked by a special symbol (N or X). With the current version of RepBase, on average, almost 50% of a human genomic DNA sequence will be masked by the program [SHG04].

Most of the interspersed repeats found by REPEATMASKER belong to the families called SINE/ALU and LINE/L1: the former are *Short INterspersed Elements* that are repetitive in the DNA of eukaryotic genomes (the Alu family in the human genome); the latter are *Long Interspersed Nucleotide Elements*, which are typically highly repeated sequences of 6K–8K bps, containing RNA polymerase II promoters. The LINE/L1 family forms about 15% of the human genome.

We have conducted an experimental study using MADMX and REPEATMASKER on *Human Glutamate Metabotropic Receptors* HGMR 1 (410277 bps) and HGMR 5 (91243 bps) as input sequences. We have downloaded the sequences from the March 2006 release of the UCSC Genome database (http://genome.ucsc.edu). REPEATMASKER version was open-3.2.7, sensitive mode, with the query species assumed to be homologous; it ran using blastp version 2.0a19MP-WashU, and RepBase update 20090120.

The experiments to assess the biological significance of the maximal dense motifs extracted by MADMX involved three separate stages. In the first stage, we ran REPEATMASKER on the input sequences HGMR 1 and HGMR 5, focusing the attention only on interspersed repeats using RepBase. One of the output files (.out) of REPEATMASKER contains the list of found repeats, and provides, for each occurrence, the substring $s[i \dots j]$ of the input sequence s which is locally aligned with (a substring of) the repeat.

In the second stage, we ran MADMX on the same DNA sequences, with density threshold $\rho = 0.8$, frequency threshold $\sigma = 4$, and $\min_\ell = 15$. In order to filter out simple repeats and low complexity portions, which are dealt with by REPEATMASKER without resorting to `RepBase`, we modified MADMX eliminating periodic maximal solid blocks (with short periods), which are the seeds of simple repeats. Then, we identified the occurrences of the motifs returned by MADMX in the input sequences, using REPEATMASKER as a pattern matching tool (i.e., replacing `RepBase` with the set of motifs returned by MADMX as the database of known repeats). The underlying idea behind this use of REPEATMASKER was to employ the same local alignment algorithms, so to make the comparison fairer.

In the third stage, we cross-checked the intervals associated with the occurrences of the `RepBase` repeats against those associated with the occurrences of our motifs. Surprisingly, MADMX was able to identify and characterize *all* of the intervals of the known SINE/ALU repeats in HGMR 1 and HGMR 5 (respectively, 56 repeats plus an extra unclassified for HGMR 1, and 20 plus an extra unclassified for HGMR 5). The remaining occurrences of the motifs permitted to identify 29 repeats out of 78 of the LINE/L1 family in HGMR 1.

The choice of the parameters ρ, σ, and $\min_l$ was done using values that seemed reasonable to us, and the results obtained seem to confirm our definition. However, a more in depth study of the effectivenes

5.3.2 Evaluating significance by statistical z-score ranking

The z-score is the measure of the distance in standard deviations of the outcome of a random variable from its expectation. Consider a DNA sequence s of length n as if it was generated by a stationary, i.i.d. source with equiprobable symbols; an approximation to the z-score for a motif of length m that contains c solid characters and appears f times in s is given by $Z = \frac{f-(n-m+1)\times p}{\sqrt{(n-m+1)\times p\times(1-p)}}$, where $p = (1/4)^c$. This metric was used in [ACP09] to assess the significance of the motifs extracted by VARUN and to rank them in the output. VARUN is designed to extract extensible motifs from one or more input sequences, and works by converting the input into a sequence of possibly overlapping cells, built during an initialization phase, so that a maximal extensible pattern corresponds to a sequence of cells. All the sequences of cells corresponding to maximal extensible patterns are fund during an iteration phase.

We employed the code for VARUN provided by the authors to extract the rigid motifs from the DNA sequences analyzed in [ACP09]. We then ran MADMX on the same sequences using the same frequency threshold σ, and setting the minimum density threshold ρ in such a way to obtain a comparable yet smaller output size. In this fashion, we tested the ability of MADMX to produce a succinct yet significant set of motifs, by virtue of its more flexible notion of density.

The results are shown in Table 5.1 and Table 5.2. For VARUN we used $D = 1$, thus allowing at most one don't care between two solid characters, and ran MADMX with $min_\ell = 1$, so to obtain the *complete* family of maximal dense motifs. In the table, there is a row of the table for each sequence (identified in the first column). Each sequence, whose total length is reported in the second column, is obtained as the concatenation of a number of smaller subsequences, reported in the third column. We used the concatenation of input sequences since MADMX is designed to run on one input sequence. On each sequence, both tools were run with the same frequency threshold σ, and the table reports for both the output size in terms of the number of motifs returned and the execution time in seconds. Also, for MADMX, the table reports the density threshold ρ used in each experiment.

				VARUN		MADMX		
name	length	#	σ	|output|	time	ρ	|output|	time
`ace2`	500	1	2	1866	3s	0.7	1762	18s
`ap1`	500	1	2	1555	1s	0.7	1304	5s
`gal4`	3000	6	4	9764	12s	0.67	7606	67s
`gal4`$^{(*)}$	3000	6	4	9764	12s	0.65	11733	191s
`uasgaba`	1000	2	2	4586	30s	0.70	4194	90s

Table 5.1: Results of the comparison with VARUN: output size and running time.

				best top-m z-scores				
name	length	#	σ	m=10	m=50	m=100	m^*	$\hat{m}$
`ace2`	500	1	2	10	50	100	1571	1067
`ap1`	500	1	2	10	50	100	392	13
`gal4`	3000	6	4	10	49	99	16	16
`gal4`$^{(*)}$	3000	6	4	10	50	100	9764	301
`uasgaba`	1000	2	2	10	50	100	175	175

Table 5.2: Results of the comparison with VARUN: z-scores.

For each experiment, we compared the best top-m z-scores, with $m = 10, 50,$ and 100, as follows. Note that, in general, the top-m motifs found by MADMX and VARUN differ. Thus, we let z^i_M (resp., z^i_V) be the z-score of the ith motif in decreasing z-score order obtained by MADMX (resp., VARUN). For each m, the table reports how many times it was $z^i_M \geq z^i_V$, for $1 \leq i \leq m$. Also, column m^* (resp., column $\hat{m}$) gives the maximum m such that $z^i_M \geq z^i_V$ (resp., $z^i_M > z^i_V$) for every $1 \leq i \leq m$.

The results of the experiment show that even when MADMX is calibrated to yield a slightly smaller output, the quality of the motifs extracted, as measured by the z-score, is higher than those output by VARUN. Indeed, for sequences ace2 and uasgaba a very large prefix of the top-ranked motifs extracted by MADMX features strictly greater z-scores of the corresponding top-ranked ones extracted by VARUN. In fact, for all of the four sequences, at least the thirteen top-ranked motifs enjoy this property. To shed light on the slightly worse performance of MADMX on gal4, we re-ran MADMX with a different density threshold, so to obtain a slightly larger output (see row gal4$^{(*)}$). In this case, the top-301 motifs extracted by MADMX have z-score strictly greater than the corresponding motifs extracted by VARUN, while the execution time remains still acceptable.

For all runs, the top z-score of a motif discovered by MADMX is considerably higher than the one returned by VARUN. Specifically, on ace2 our best z-score is 387 763 vs. 12 027 of VARUN; on ap1, we have 12 027 vs. 1 490; on gal4 it is 75 vs. 28; on gal4$^{(*)}$ it is 150 vs. 28; on uasgaba we have 134 532 vs. 67 059. This reflects the high selectivity of MADMX, which is to be attributed mostly to adoption of a more flexible density constraint.

We must remark that MADMX (in its current nonoptimized version) is slower than VARUN, but it still runs in time acceptable from the point of view of a user. To further investigate the tradeoff between execution time and significance of the discovered motifs, we repeated the experiments running MADMX with $\min_\ell = 2$ and $\rho = 0.65$, for all sequences. The running time of MADMX was almost halved, while the small output produced still featured high quality. Notably, for sequences ace2, ap1, and uasgaba the top-100 motifs extracted by MADMX have z-score greater or equal than the corresponding ones returned by VARUN.

We also have attempted a comparison between VARUN and MADMX on longer sequences (such as HGMR 1) at higher frequencies (since, unfortunately, VARUN does not seem to be able to handle low frequencies on very long sequences). Even allowing a higher number of don't cares between solid characters ($D = 2$) for the motifs of VARUN, all of the top-m z-scores featured by the motifs extracted by MADMX are greater than or equal to the corresponding scores in the ranking of VARUN, with m reaching the size of VARUN's output. The small values of D considered ($D = 1, 2$) are consistent with the experiments reported in [ACP09] for the input DNA sequences we considered. In [ACP09] those values have been shown to produce biological significant motifs. In fairness, we remark that VARUN was designed to work at its best on protein sequences, while MADMX's main target are DNA sequences. Hence, these two tools should be regarded as complementary. Moreover, VARUN has the advantage of retrieving flexible motifs, while MADMX focuses only on rigid ones.

Chapter 6

Significantly Mutated Pathways in Biological Networks

In this chapter we propose a rigorous framework for *de novo* identification of significantly mutated subnetworks. The naïve approach is to examine mutations on all subnetworks, or all subnetworks of a fixed size and to apply statistical standard multi-hypothesis testing. This approach is problematic. First, the enumeration of all such subnetworks is prohibitive even for subnetworks of reasonable size. Second, the extremely large number of hypotheses that are tested makes it difficult to achieve statistical significance. Finally, biological interaction networks typically have small diameter due to the presence of *hubs*, genes of high degree. There are reports that cancer-associated genes have more interaction partners than non-cancer genes [L$^+$07a, JB06], and indeed highly mutated cancer genes like TP53 have high degree in most interaction networks (e.g. the degree of TP53 in HPRD is 238). Such correlations might lead to a large number of "uninteresting" subnetworks being deemed significant, since any subnetwork containing an highly mutated hub will be returned as significant.

Our framework employs two strategies to overcome the difficulties described above. First, we formulate an *influence* measure between pairs of genes in the network using a diffusion process defined on the graph. This quantity considers a gene to influence another gene if they are both close in distance on the graph *and* the number of paths between them is relatively high compared to all paths starting from one of the two genes. We use this measure to build a smaller *influence graph* that includes only the mutated genes but encodes the neighborhood information from the larger network. We then identify significant subnetworks using two techniques. In the combinatorial model we consider a graph in which each mutated gene is represented by a node, and two genes are connected if the influence between them is larger then some threshold. We formulate on this graph the *connected maximum coverage* problem of finding the connected subgraph that is altered in the highest number of patients. We show that this problem is NP-hard and describe an efficient approximation algorithm. We then derive an alternative approach, the *enhanced influence model*, in which the influence between pairs of genes is enhanced by the number of mutations observed on these genes. Again we consider a graph on the set of mutated genes with edges connecting pairs of genes with enhanced influence above a given threshold. Since the mutation information is already encoded in the edge weights, the computational problem is reduced to just finding connected components in the graph. Finally, we derive a *two-stage multiple hypothesis test* that mitigates the testing of a large number of hypotheses by focusing on the number of discovered subnetworks of a given size rather than on individual subnetworks. We also show how to estimate the false discovery rate (FDR) incurred by this test.

We tested our approach on the HPRD human interaction network using somatic mutation data from two recently published studies: (i) 601 genes in 91 glioblastoma multiforme patients from The Cancer Genome Atlas (TCGA) project; (ii) 623 genes in 188 lung adenocarcinoma patients sequenced during the Tumor Sequencing Project (TSP). In both datasets, we identify statistically significant mutated subnetworks that are enriched for genes on pathways known to be important in these cancers, including the p53 and RTK/RAS/PI(3)K pathways. We also identify the Notch signaling pathway as significantly mutated in the lung samples. Notch signaling is known to be deregulated in a number of cancers, but was not reported as mutated in the TSP publication. Our work is the first, to our knowledge, to propose a computationally efficient strategy for *de novo* identification of *statistically significant* mutated subnetworks. We anticipate that our approach will find increasing use as cancer genome studies increase in size and scope.

The rest of the chapter is organized as follows: in Section 6.2 the influence graph is defined, while Section 6.3 presents the two methods we design to find significantly mutated pathways. Section 6.4 presents the statistical method we design to address the significance of our findings, and Section 6.5 illustrates the results we obtained with our method.

The results presented in this chapter were published in a preliminary version in [VUR09, VUR10].

6.1 Mathematical model

We model the interaction network by a graph $G = (V, E)$, where the vertices in V represent individual proteins (and their associated genes), and the edges in E represent (pairwise) protein-protein or protein-DNA interactions. Let $\mathcal{T} \subseteq V$ be the subset of genes that have been tested, or assayed, for mutations in a set $\mathcal{S}$ of samples (patients). The size of $\mathcal{T}$ will vary by study; e.g. some recent works resequenced hundreds of genes [Net08, D$^+$08] while others examine nearly all known protein-coding genes in the human genome [W$^+$07, J$^+$08, P$^+$08]. We assume that each gene g is assigned one of two labels, *mutated* or *normal*, in each sample. Let M_i denote the subset of genes in $\mathcal{T}$ that are mutated in the ith sample, for $i = 1, \ldots |\mathcal{S}|$. Let $\mathcal{S}_j$ be the samples in which gene $g_j \in \mathcal{T}$ is mutated, for $j = 1, \ldots, |\mathcal{T}|$, let $m = \sum_i |M_i|$ be the total number of occurrences of altered genes observed in all samples.

We define a *pathway* or *subnetwork* to be a connected subgraph of G. Note that this definition matches the common biological usage of the term where pathways may have arbitrary topology in the graph, and are not restricted to be linear chains of vertices. We generally do not know whether more than one gene must be mutated to perturb a pathway in a sample, and thus will assume that a pathway is mutated in a sample if *any* of the genes in the pathway are mutated. For a subset $T \subseteq \mathcal{T}$, let $S(T)$ denote the set of samples in which *at least one* gene in T is mutated.

6.2 Influence graph

Given the protein interaction network and the mutation data observed for tested genes in the samples $\mathcal{S}$, we want to identify subnetworks of genes that are significantly mutated. The genes in a subnetwork should correspond to a pathway, where the mutation of a gene corresponds to the alteration of the pathway. The mutation of a gene g in a subnetwork should then have a significant effect on at least one other gene g' in the same subnetwork. Using the original interaction network we can observe only effects on the neighbours of a gene, but the mutation of g can in general alter the functionality of gene g' even if g is not directly interacting with g'. Consider for example the linear chain of Figure 6.1. The mutation of the gene at the bottom of the chain can have the effect of altering the functionality of the gene at the top of the chain, even if the two nodes are not directly interacting. We thus need a procedure to identify the genes whose functionality can be altered by the mutation of gene g. A first possibility is to use the distance between two genes g, g' in the protein interaction network as measure for this functional influence. However the distance is not an accurate measure, since it does not take into account the topology of the network containing g and g', that must be considered when relating the functionality of g and g'.

We can quantify the alteration that mutation of g induces in g' taking into account the whole network topology using a diffusion process. The significance of a subnetwork is derived from: (i) the number of samples that have mutations in the genes of the subnetwork, and (ii) the interactions between genes in the subnetwork in the context of the whole network topology. For example, consider the two scenarios of mutated nodes of Figure 6.1. In the first scenario, the two mutated nodes are part of a linear chain in the interaction network. In the second scenario, the two mutated nodes are connected through a high-degree node. In the first case, there is a single path joining the two mutated nodes, thus we expect the functionality of the two nodes to be more related than in the second case, where the two nodes are connected by a node that is active in a large number of possible pathways. If the number of samples in which the two genes are altered is the same in both scenarios, we would assign greater significance to the linear chain. Most human interaction networks have a number of nodes of high-degree, or *hubs*, and these produce many paths between mutated nodes. A simple correction for this problem is to remove high-degree nodes. However, a number of genes that are commonly mutated in cancer have high-degree in interaction networks,, and thus removal of high-degree nodes results in loss of information.

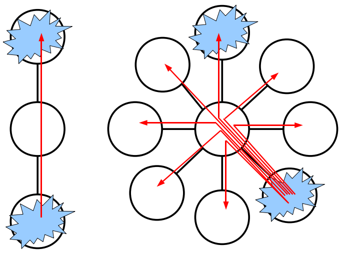


Figure 6.1: Mutation on chain vs. star graph.

We use a diffusion process on the interaction network to define a rigorous measure of *influence* between all pairs of nodes. To measure the influence of node s on all the other nodes in the graph, consider the following process, described by [QSL+08]. Fluid is pumped into the source node s at a constant rate, and fluid diffuses through the graph along the edges. Fluid is lost from each node at a constant first-order rate γ. Let $f_v^s(t)$ denote the amount of fluid at node v at time t, and let $\mathbf{f}^s(t) = [f_1^s(t), \dots, f_n^s(t)]^T$ be the column vector of fluid at all nodes. Let L be the Laplacian matrix of the graph[1], and let $L_\gamma = L + \gamma I$. Then the dynamics of this continuous-time process are governed by the vector equation

$$\frac{d\mathbf{f}^s(t)}{dt} = -L_\gamma \mathbf{f}^s(t) + \mathbf{b}^s u(t), \tag{6.1}$$

[1] $L = -A + D$, where A is the adjacency matrix of the graph and D is a diagonal matrix with $D_{i,i} = degree(v_i)$.

where $\mathbf{b}^s$ is the elementary unit vector with 1 at the s^{th} place and 0 otherwise, and $u(t)$ is the unit step function. As $t \to \infty$, the system reaches the steady state. The equilibrium distribution of fluid density on the graph is $\mathbf{f}^s = L_\gamma^{-1}\mathbf{b}^s$ (See [QSL+08]). Note that this diffusion process is related to the diffusion kernel [KL02], or heat kernel [Chu07], which models the diffusion of heat on a graph, and these diffusion processes are also related to certain random walks on graphs [DS84, Lov93]. Diffusion processes and their related flow problems have been used in protein function prediction on interaction networks [TN04, NJA+05] and to define associations between gene expression and phenotype [MLWS07].

We interpret f_i^s as the influence $i(g_s, g_i)$ of gene g_s on gene g_i. Computing the diffusion process for all tested genes gives us, for each pair of genes $g_j, g_k \in \mathcal{T}$, the influence $i(g_j, g_k)$ that gene g_j has on gene g_k. Note that in general the influence is not symmetric; i.e. $i(g_j, g_k) \neq i(g_j, g_k)$. We define an *influence graph* $IG = (\mathcal{T}, IE)$ with the set of nodes corresponding to the set of tested genes, the weight of an edge (g_j, g_k) is given by

$$w(g_j, g_k) = \min[i(g_k, g_j), i(g_j, g_k)] = \min[f_j^k, f_k^j].$$

If n is the number of nodes in the interaction network, then the cost of computing IG is dominated by the complexity of inverting an $n \times n$ matrix.

6.3 Discovering significant subnetworks

6.3.1 Combinatorial model

Given an influence measure between genes, the obvious first approach for discovering significant subnetworks is to identify sets of nodes in the influence graph IG that are (1) connected through edges with high influence measure; and (2) correspond to mutated genes in a significant number of samples. We fix a threshold δ and compute a *reduced influence graph* $IG(\delta)$ of IG by removing all edges with $w(g_i, g_j) < \delta$, and all nodes corresponding to genes with no mutations in the sample data. The computational problem is reduced to identifying connected subgraphs of $IG(\delta)$ such that the corresponding set of genes is altered in a significant number of patients.

The size of the connected subgraphs we discover is controlled by the threshold δ. We choose sufficiently small δ such that in the null hypothesis, in which the mutations are randomly placed in nodes corresponding to tested genes, it is unlikely that our procedure finds connected subgraphs with similar properties. Note that value of δ depends only on the null hypothesis and not on the observed sample data (see Section 6.4 for details of the statistical analysis).

Computational problem

Finding the connected subgraph of k genes that is mutated in the highest number of samples requires to solve the following problem, that we define as *connected maximum coverage* problem: given a graph G defined on a set of m vertices V, a set of elements I, a family of subsets $\mathcal{P} = \{P_1, \ldots, P_m\}$, with $P_i \in 2^I$ associated with $v_i \in V$, and a value k, find the connected subgraph $\mathcal{C}^* = \{v_{i_1}, \ldots, v_{i_k}\}$ with k nodes in G that maximize $| \cup_{j=1}^{k} P_{i_j}|$. In our case we have $G = IG(\delta)$, V is the subset of genes in $\mathcal{T}$ mutated in at least one sample, and for each $g_i \in V$ the associated set is $\mathcal{S}_i$. The connected maximum coverage problem is related to the maximum coverage problem (see e.g. [Hoc97] for a survey) where given a set I of elements, a family of subsets $F \subset 2^I$, and a value k, one needs to find a collection of k sets in F that covers the maximum number of elements in I. This problem is NP-hard as set cover is reducible to it.

If the graph G is a complete graph, the connected maximum coverage problem is the same as the maximum coverage problem. Thus the connected maximum coverage problem is NP-hard for a general graph. Moreover we prove that the problem is still hard even on simple graphs such as the star graph (similar result was shown in [SH06] for the connected set cover problem).

Theorem 6.1. *The connected maximum coverage problem on star graphs is NP-hard.*

Proof. The proof is by reduction from the maximum coverage problem. Given an instance of the maximum coverage problem, consisting of I, F, and k, we build an instance of the connected maximum coverage problem. We define $I' = I \cup \{v_0\}$, with $v_0 \notin I$; and $F' = F \cup \{v_0\}$. Moreover, we build the graph $G = (V, E)$ where $V = F'$ and $E = \{(v_0, s) | s \in F\}$. It is easy to verify that G is a star graph, and then each non-trivial (i.e., with more than 1 vertex) subgraph of G will contain the vertex v_0. The solution X to the connected maximum coverage problem on the graph G is then of the form $X = Y \cup \{v_0\}$, where $Y \subseteq F$. It is easy to verify that X is a connected maximum coverage of size $k + 1 > 1$ if and only if Y is maximum coverage of size $k > 0$. □

Since the connected maximum coverage problem is NP-hard even for simple graphs we turn to approximate solutions. It is not hard to construct a polynomial time $1 - \frac{1}{e}$ approximation algorithm for spider graphs (analogously to the result in [SH06] for the connected set cover problem). Since the biological network of interest are not spider graphs, we construct an alternative polynomial time algorithm that gives $O(1/r)$ approximation when the radius of the optimal solution $\mathcal{C}^*$ is r.

Our algorithm obtains a solution $\mathcal{C}_v$ (thus, a connected subgraph) starting from each node $v \in V$, and then returns the best solution found. To obtain $\mathcal{C}_v$, our algorithm executes an *exploration phase*, i.e. for each node $u \in G$ it finds a shortest path $p_v(u)$ from v to u. Let $\ell_v(u)$ be the set of nodes in $p_v(u)$, and $P_v(u)$ the elements of I they cover. After this *exploration phase*, the algorithm builds a connected subgraph $\mathcal{C}_v$ starting from v. At the beginning we have $\mathcal{C}_v = \{v\}$. $P_{\mathcal{C}_v}$ is the set of elements covered by the current connected subgraph $\mathcal{C}_v$. Then, while $|\mathcal{C}_v| < k$, the algorithm chooses the node $u \notin \mathcal{C}_v$ such that: $u = \arg\max_{u \in V} \left\{ \frac{|P_v(u) \setminus P_{\mathcal{C}_v}|}{|\ell_v(u) \setminus \mathcal{C}_v|} \right\}$ and $|\ell_v(u) \cup \mathcal{C}_v| \leq K$; the new solution is then $\ell_v(u) \cup \mathcal{C}_v$. The main computational cost of our algorithm is due to the exploration phase, that can be performed in polynomial time. We have the following:

Theorem 6.2. *The algorithm above gives a $\frac{1}{cr}$-approximation for the connected maximum coverage problem on G, where $c = \frac{2e-1}{e-1}$ and r is the radius of optimal solution in G.*

Proof. We first analyze the solution obtained assuming the nodes in the solution are inserted one at the time (i.e., $|\ell_v(u) \setminus \mathcal{C}_v| = 1$ for each node u inserted in the solution). We will then show that when the nodes are not inserted in the solution one at the time, the solution obtained cannot have a worse solution.

Let $z^*(v)$ be the value of the best solution $OPT(v)$ that can be found starting at node v. Define

$$OPT_i(v) = \left| \left\{ \bigcup_{\substack{g_j \in OPT(v): \\ d(g_j,v) = r-i+1}} S_j - \bigcup_{\substack{g_j \in OPT(v): \\ r-i+1 < d(g_j,v) \leq r}} S_j \right\} \right|,$$

thus $OPT(v) = \sum_{i=1}^{r_v} OPT_i(v)$, where r_v is the radius of $OPT(v)$, and $z^*(v) = \sum_{i=1}^{r_v} |OPT_i(v)|$. We divide the execution of our algorithm in r_v phases: in phase i our algorithm inserts $|OPT_i(v)|$ new nodes in the solution. Note that in phase i, our algorithm always has the possibility to reach each node in $OPT_i(v)$. Thus, in phase i, the algorithm above is equivalent to the greedy algorithm for the maximum coverage problem where the sets that can be chosen are all the sets at distance at most $r-i+1$, and then all the sets in $OPT_i(v)$ can be chosen by the greedy algorithm. Let $A_i(v)$ be the increment in the value of the solution found by our algorithm between the end of phase i and the end of phase $i-1$. Since the approximation factor for the maximum coverage is $1-1/e$ and each element in $OPT_i(v)$ is seen with weight reduced of a factor $1/(r-i+1)$ (since it is at distance $r-i+1$), in phase i our algorithm improve the current solution of a factor

$$A_i \geq \frac{1}{r}\left(1-\frac{1}{e}\right)\left(OPT_i(v) - \sum_{j=1}^{i-1} A_{i-1}(v)\right).$$

Let A denote the value of the solution returned by our algorithm. Summing the terms above for all i we obtain:

$$\begin{aligned}
A(v) &\geq \frac{1}{r}\left(1-\frac{1}{e}\right)\left(\sum_{i=1}^{r_v} OPT_i(v) - \sum_{j=1}^{r_v-1}(r_v-j)A_j(v)\right)\\
&\geq \frac{1}{r}\left(1-\frac{1}{e}\right)\sum_{i=1}^{r_v} OPT_i(v) - \frac{1}{r}\left(1-\frac{1}{e}\right)\sum_{j=1}^{r_v-1}(r_v-j)A_j(v)\\
&\geq \frac{1}{r}\left(1-\frac{1}{e}\right)OPT(v) - \frac{1}{r}\left(1-\frac{1}{e}\right)rA(v)\\
&\geq \frac{1}{r}\left(1-\frac{1}{e}\right)OPT(v) - \left(1-\frac{1}{e}\right)A(v).
\end{aligned}$$

We then obtain

$$\frac{2e-1}{e}A(v) \geq \frac{1}{r}\left(\frac{e-1}{e}\right)OPT(v)$$

that is

$$A(v) \geq \frac{1}{r}\left(\frac{e-1}{2e-1}\right)OPT(v).$$

Now consider the case $|\ell_v(u) \setminus \mathcal{C}_v| > 1$: this means that that we insert a path whose weight, divided by $|\ell_v(u) \setminus \mathcal{C}_v|$, is higher than the weight of any other possible reachable node (from v). Then we have that the value of the solution found by our algorithm can only improve, since we are inserting $|\ell_v(u) \setminus \mathcal{C}_v|$ nodes such that the average value of the inserted nodes is greater than the maximum value of $|\ell_v(u) \setminus \mathcal{C}_v|$ reachable nodes in the best solution including v divided by its distance (that is at most r_v). □

For our experiments we implemented a variation of this algorithm, that for each pair of nodes (u, v) considers all the shortest paths between u and v, and then keeps the one that maximizes $\frac{|P_v(u)|}{|\ell_v(u)|}$ to build the solution $\mathcal{C}_v$. With this modification the algorithm is not guaranteed to run in polynomial time in the worst-case, but ran efficiently for all our experiments.

6.3.2 The Enhanced Influence model

We developed an alternative, computationally efficient, approach for identifying subnetworks that are significant with respect to the gene mutation data. The *Enhanced Influence Model* is based on the idea of enhancing the influence measure between genes by a function of the number of mutations observed in each of these genes, as explained below, and then decomposing an associated *enhanced influence graph* into connected components.

We define the *enhanced influence* graph H. It has a node for each gene g_i with at least one mutation in the data. The weight of edge (g_j, g_k) in H is given by

$$hw(g_j, g_k) = \min\{i(g_j, g_k), i(g_k, g_j)\} \times \max\{|\mathcal{S}_j|, ||\mathcal{S}_k|\}.$$

Thus, the strength of connection between two nodes in the enhanced influence graph is a function of both the interaction between the nodes in the interaction network and the number of mutations observed in their corresponding genes. Next we remove all edges with weight smaller than a threshold δ to obtain a graph $H(\delta)$. We return the connected components in $H(\delta)$ as the significant subnetworks with respect to the mutation data and the threshold δ. The computational cost is the complexity of computing all connected components in a graph with $|S|$ nodes (number of mutated genes), which is linear in the size of the graph. The significance of the discovered subnetworks depends on the choice of δ. We choose sufficiently small δ such that in the null hypothesis, in which the mutations are randomly placed in nodes corresponding to tested genes, it is unlikely that our procedure finds connected components of similar size (see Section 6.4 for details of the statistical analysis).

6.4 Statistical analysis

We assess the statistical significance of our discoveries with respect to null hypothesis distributions in which the mutated genes are randomly allocated in the network, that is when the occurrence of mutations are independent of the network topology. We consider two null hypothesis distributions: in H_0^{sample} a total of $m = \sum_i |M_i|$ mutations are placed uniformly at random in the nodes corresponding to the $|\mathcal{T}|$ tested genes, hence preserving the number of mutated genes in each sample. While easier to analyze, this model does not account for the fact that in the observed data a large number of mutations are concentrated in a few genes(e.g. TP53).

An alternative null hypothesis distribution we consider, H_0^{gene}, is generated by uniformly at randomly permuting the tested genes among the locations of the tested genes in the network. That is we select a random permutation σ of the set $\{1, \ldots, |\mathcal{T}|\}$, and we the set of samples $\mathcal{S}_j \subseteq \mathcal{S}$, associated to g_j in the real data, to the location of gene $g_{\sigma(j)}$ in the original network.

6.4.1 A two stage multi-hypothesis test

A major difficulty in assessing the statistical significance of the discovered subnetworks is that we test simultaneously for a large number of hypotheses; each connected subnetwork in the interaction graph with at least one tested gene is a possible significant subnetwork and thus an hypothesis. Using the standard approach of [BH95] to control the FDR would result in a reduced ability of identifying significantly mutated pathways. Instead, we adapt the ideas introduced in Section 4.3.2 to develop a two stage test for our problem that allows us to flag a number of subnetworks in our data as statistically significant while controlling the FDR of the set of flagged subnetworks.

We demonstrate our method through the analysis of the Enhanced Influence model. A similar technique was applied to the Combinatorial model. Let $C_1, \ldots, C_\ell$ be the set of connected components found in the enhanced influence graph $H(\delta)$. Testing for the significance of these discoveries is equivalent to simultaneously testing for $2^{|\mathcal{T}|}$ hypothesis. To reduce the number of hypothesis we focus on an alternative statistic (outcome) which is the number of discoveries of a given size. Let $\tilde{r}_s$ be the number of connected components of size $\geq s$ found in the graph $H(\delta)$, and let r_s be the corresponding random variable in the null hypothesis (H_0^{sample} or H_0^{gene}). We are testing now for just $\mathcal{K} = |\mathcal{T}|$ simple hypotheses, for $s = 1, \ldots, \mathcal{K}$: $E_s \equiv$ "$\tilde{r}_s$ conforms with the distribution of r_s". Testing each hypothesis with confidence level $\alpha/\mathcal{K}$, the first stage of our test identifies the smallest size s such that with confidence level α we can reject the null hypothesis that $\tilde{r}_s$ conforms with the distribution of r_s.

The fact that the number of connected components of size at least s is statistically significant does not imply necessarily that each of the connected components is significant. We now add a second condition to the test that guarantees an upper bound on the FDR:

Theorem 6.3. *Fix $\beta_1, \beta_2, \ldots, \beta_{\mathcal{K}}$ such that $\sum_{i=1}^{\mathcal{K}} \beta_i = \beta$. Let s^* be the first s such that $\tilde{r}_s \geq \frac{\mathbf{E}[r_s]}{\beta_s}$. If we return as significant all connected components of size $\geq s^*$, then the FDR of the test is bounded by β.*

Proof. Let V_i be the number of erroneous rejections of connected components of size i, i.e. the number of connected components of size i that were flagged erroneously as significant. Note that $E[V_i] \leq E[r_i]$, since if these hypothesis were erroneously rejected they were generated by the null distribution.

$$\begin{aligned}
FDR &= \sum_{i=0}^{|\mathcal{K}|} E\left[\frac{V_i}{\tilde{r}_i}\right] \Pr(E_i, \bar{E}_{i-1}, \ldots, \bar{E}_0) \\
&\leq \sum_{i=0}^{|\mathcal{K}|} \frac{\beta_i E[X_i \mid E_i \bar{E}_{i-1}, \ldots, \bar{E}_0]}{E[r_i]} \Pr(E_i, \bar{E}_{i-1}, \ldots, \bar{E}_0) \\
&= \sum_{i=0}^{|\mathcal{K}|} \frac{\beta_i \sum_j j \Pr(X_i = j, E_i, \bar{E}_{i-1}, \ldots, \bar{E}_0)}{E[r_i]} \\
&\leq \sum_{i=0}^{|\mathcal{K}|} \frac{\beta_i E[r_i]}{E[r_i]} \leq \beta.
\end{aligned}$$

□

Notice that the test above does not require to test all value $s = 1, \ldots, \mathcal{K}$. In fact, in our tests we considered only two thresholds, $s = 6$, and $s = 10$. For each hypothesis we can then compute what is the minimum threshold α for which that hypothesis would be rejected. We can moreover compute what is the FDR associated with the set of connected components returned using s^* defined in Theorem 6.3. In our tests we have used $\beta_i = \frac{\beta}{2^i}$ for the i^{th} largest s tested (with $\beta_s = \beta - \sum_i \beta_i$ for the smallest s), since we are more interested in finding large connected components.

6.4.2 Estimating the distribution of the null hypothesis

The null hypothesis distributions can be estimated by either a Monte-Carlo simulation (known as “permutation test” in the computational biology community) or through analytical bounds.

Using Monte-Carlo simulation, two features of our method significantly reduce the cost of the estimates. First, the Influence Graph IG is created *without* observing the sample data. The mutation data and IG are then combined to create the sample dependent graphs $IG(\delta)$ and $H(\delta)$. Thus, the Monte Carlo simulation needs to run on the graph IG which is significantly smaller than the original interaction network (in our data the original interaction network had 18796 nodes while the influence graph had only about 600 nodes), since the vertices of IG are the tested genes and both null distributions requires to work only on tested genes . Second, our statistical test does not use the p-values of individual connected subgraphs but the p-value of the number of connected subgraphs of a given size. Thus, since the number of hypotheses is smaller, we need p-values an order of magnitude larger than the ones that would be required if we test for single subgraphs. We then need to estimate p-values to a precision that is an order of magnitude larger, which require significantly fewer rounds of simulations. These features allowed us to compute the null distributions through Monte-Carlo simulations for the size of our data with no significant computational cost.

For larger number of tested genes we can estimate the null hypothesis through analytical bounds. Consider for example the Enhanced Influence model, and assume that the $|\mathcal{T}|$ tested genes are randomly permuted among the $|\mathcal{T}|$ nodes of the graph IG to generate a random instance graph $\bar{H}(\delta)$. Let m be the number of genes with observed mutations, and let s_{max} be the maximum number of mutations of any gene. Since we are interested in δ that partitions the graph to a number of connected components we can choose the maximum δ such that for any node g_i in IG no more than $\alpha m/|\mathcal{T}|$ of the adjacent edges have weights that satisfy $s_{max}w(g_i, g_j) \geq t$, for some fixed $\alpha < 1$. For the choice of δ above, the expected number of connected components of size k in $\bar{H}(\delta)$ is bounded by

$$\binom{|\mathcal{T}|}{k} k^{k-2}\alpha^{k-1} \leq \frac{m}{k^2}\alpha^{k-1}.$$

Since connected components are disjoint, their occurrences are negatively correlated, and we can stochastically bound the distribution of r_s with a binomial distribution with the above expectation. A similar bound can be computed for the other models and null hypothesis distributions, and for (somewhat) less restrictive conditions on δ.

6.5 Experimental results

We applied our approach to analyze somatic mutation data from two recent studies. The first dataset is a collection of 453 somatic mutations identified in 601 tested genes from 91 glioblastoma multiforme (GBM) samples from The Cancer Genome Atlas [Net08]. In total, 223 genes were reported mutated in at least one sample. The second dataset is a collection of 1013 somatic mutations identified in 623 tested genes from 188 lung adenocarcinoma samples from the Tumor Sequencing Project [D$^+$08]. In total, 316 genes were reported such that each of them was mutated in at least one sample. We use the protein interaction network from the Human Protein Reference Database (June 2008 version) [P$^+$09] which consists of 18796 vertices and 37107 edges. We derive the influence graph for each dataset by directly computing the inverse[2] of L_γ. For all our experiments we fixed the parameter $\gamma = 8$, which is approximately the average degree of a node in HPRD (after the removal of disconnected nodes). We also conducted a preliminary study of the impact of the choice of γ on the distribution of the weights in the influence graph. This preliminary study shows that the choice of γ does not have a huge impact for our random models. However, the development of a rigorous method to choose γ is an open problem. The influence graphs obtained from the inversion of L_γ have weights $i(g_j, g_k) \neq 0$ for almost all pairs (g_j, g_k) of tested genes: less than 2% of the weights are zero in the GBM graph, while all weights in the lung adenocarcinoma graph are positive. We now describe the results of the applying the combinatorial model (Section 6.5.1) and enhanced influence model (Section 6.5.2) to both datasets. Section 6.5.3 compares these results against those obtained with the naïve algorithm.

6.5.1 Combinatorial model

We used the combinatorial model to extract a subnetwork, of k mutated genes, that is mutated in the highest number of samples from GBM and lung adenocarcinoma with $k = 10$ and $k = 20$. For both data we used the procedure described in Section 6.3.1 to derive the threshold $\delta = 0.0001$ for the reduced influence graph $IG(\delta)$. Table 6.1 shows that we find statistically significant subnetworks under both the H_0^{gene} and H_0^{sample} null hypotheses (p-values for H_0^{sample} are computed without Monte-Carlo simulation). The genes in each subnetwork are reported in Table 6.2. To assess the biological significance of our findings in GBM, we compared the genes in each subnetwork to the genes in pathways that were previously implicated in GBM and used as a benchmark in the TCGA publication [Net08] (See also Figure 6.2 (a) below). We find that our subnetworks are enriched for (i.e., contains a statistically significant number of) genes in the RTK/RAS/PI(3)K pathway and to a lesser extent, the p53 pathway. For the lung adenocarcinoma samples, we find that the subnetworks share significant overlap with the pathways reported in the original publication [D$^+$08]. These results demonstrate that the combinatorial model is effective in recovering genes known to be important in each of these cancers.

6.5.2 Enhanced Influence model

We applied the enhanced influence model to the same two datasets. Following the procedure described in Section 6.3.2, we first computed the enhanced influence net-

[2] In contrast [QSL$^+$08] derive a power series approximation to L_γ^{-1} whose convergence depends on the choice of γ.

			p-val		pathway enrichment p-val		
dataset	k	samples	H_0^{sample}	H_0^{gene}	all	RTK/RAS/PI(3)K	p53
GBM	10	67	$< 10^{-10}$	4×10^{-3}	3×10^{-4}	8×10^{-4}	0.19
	20	78	$< 10^{-10}$	$< 10^{-3}$	10^{-5}	8×10^{-5}	0.05
Lung	10	140	$< 10^{-10}$	0.02	8×10^{-6}	/	
	20	151	$< 10^{-10}$	0.03	3×10^{-3}	/	

Table 6.1: Results of the combinatorial model. k is the number of genes in the subnetwork. *samples* is the number of samples in which the subnetwork is mutated. *p-val* is the probability of observing a connected subgraph of size k under the random model H_0^{sample} or H_0^{gene}. *enrichment p-val* is the p-value of the hypergeometric test for overlap between genes in the identified subgraph and genes reported significant pathways in [Net08] or [D$^+$08]. For GBM, *enrichment p-val* is the p-value of the hypergeometric test for RTK/RAS/PI(3)K and p53 pathways.

dataset	k	samples	genes
GBM	10	67	INSR BCR TP53 PTEN EGFR
			ERBB2 DST PIK3R1 PIK3CA SERPINA3
	20	78	MDM2 FGFR1 BRCA2 CHEK1 COL1A2
			ITGB3 TNK2 INSR BCR TP53
			PTEN EGFR ERBB2 DST PIK3R1
			PIK3CA NF1 SPARC PDGFRA SERPINA3
Lung	10	140	CDC25A CHEK1 TP53 STK11 HRAS
			KRAS ERBB4 EGFR NF1 PTEN
	20	150	MAPK8 PRKDC TP53 STK11 HRAS
			KRAS EGFR PRKD1 NF1 ABL1
			ERBB4 PTEN HD PRKCE SMAD2
			TGFBR1 BAX RAPGEF1 PIK3CG ACVR1B

Table 6.2: Genes in the connected component of size k that covers the maximum number of samples as reported by our algorithm for GBM and lung adenocarcinoma.

work, using a threshold of $t = 0.003$ for the GBM data and $t = 0.01$ for the lung adenocarcinoma data. Table 6.3 shows the number and sizes of the connected components identified in the GBM data, and the associated p-values, the latter obtained using the method described in Section 6.4. Table 6.4 reports the genes in the connected components of size > 3.

We identify two significant connected components with more than 19 genes (FDR ≤ 0.14). We find significant overlap ($P < 10^{-2}$ by hypergeometric test) between the 68 genes in our connected components and the set of all mutated genes in the same RTK/RAS/PI(3)K, p53, and RB pathways examined in the TCGA study [Net08] (see Table 6.5). The second largest connected component with 19 genes has significant overlap to the p53 pathway, while the largest connected component with 22 genes has significant overlap with the RTK/RAS/PI(3)K signaling pathway. In contrast to the combinatorial model, the enhanced influence model separates these two pathways into different connected components. Figure 6.2 (a) illustrates the overlap between the mutated genes in connected components returned by our method and genes in the pathways reported in [Net08].

		H_0^{sample}		H_0^{gene}	
s	# c.c. $\geq s$	μ	p-val	μ	p-val
2	15	22.18	0.97	13.63	0.38
3	3	6.37	0.98	4.38	0.6
19	2	$< 10^{-3}$	$< 10^{-3}$	0.07	$< 10^{-3}$
22	1	$< 10^{-3}$	$< 10^{-3}$	0.05	0.05

Table 6.3: Results of the enhanced influence model on GBM samples. s is the size of connected components (c.c.) found with our method. *# c.c.* $\geq s$ is the number of c.c. with *at least* s nodes. μ is the expected number of c.c. with $\geq s$ nodes under random models H_0^{gene}, H_0^{sample}. *p-val* is the probability of observing *at least # c.c.* $\geq s$ with at least s nodes in a random dataset.

size	genes
22	MSH2 ATM MSH6 PRKDC ATR BCR KLF6 GLI3 KLF4 PML MAPK9 CHEK1 BRCA2 ING4 MDM2 MDM4 TP53 TOP1 PTEN KPNA2 STK36 GLI1
19	ANXA1 TNK2 ERBB3 SERPINA3 SOCS1 TNC PIK3C2B PDGFRB ERBB2 NRAS VAV2 EGFR EPHA2 MET ADAM12 PIK3R1 PIK3CA CENTG1 AXL

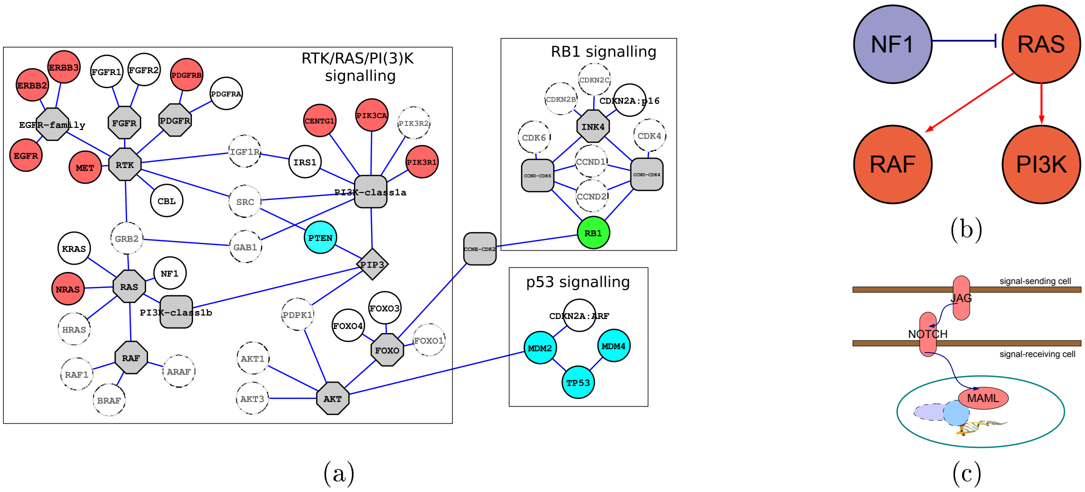
Table 6.4: Genes in connected components obtained for GBM the diffusion model with $\gamma = 8, t = 0.003$.

	enrichment p-val	
s	RTK/RAS/PI(3)K	p53
19	0.9	4×10^{-3}
22	4×10^{-6}	–

Table 6.5: Result of the hypergeometric test for enrichment for RTK/RAS/PI(3)K, and p53 pathways respectively. s is the size of connected components (c.c.) found with our method.

For the lung data, Table 6.6 shows the sizes of connected components returned by the enhanced influence model and the p-values associated with each. Table 6.7 lists the genes in each connected component of size > 5. The 88 genes in the union of the connected components derived by our method overlap significantly ($P < 7 \times 10^{-9}$

by the hypergeometric test) with the mutated pathways reported in the network of Figure 6 in the TSP publication [D$^+$08]. We identify 4 connected components of size ≥ 7 (FDR ≤ 0.28). The first connected component of size 10 contains genes in the p53 pathway, and the second one is enriched ($P < 10^{-2}$) for the MAPK pathway (Figure 6.2 (b)). The third component is the ephrin receptor gene family, a large family of membrane-bound receptor tyrosine kinases, that were reported as mutated in breast and colorectal cancers [S$^+$06]. Notably, only one of the genes in this component, EPHA3, is mentioned as significantly mutated in [D$^+$08]. Finally, the connected component of size 7 consists exclusively of members of the Notch signaling pathway (Figure 6.2 (c)). The mutated genes include: the Notch receptor (NOTCH2/3/4); Jagged (JAG1/2), the ligand of Notch; and Mastermind (MAML1/2), a transcriptional co-activator of Notch target genes. The Notch signaling pathway is a major developmental pathway that has been implicated in a variety of cancers [Axe04] including lung cancer [CKB04]. Mutations in this pathway were not noted in the original TSP publication [D$^+$08], probably because no single gene in this pathway is mutated in more than 3 samples. Because our method exploits both mutation frequency and network topology, we are able to identify these more subtle mutated pathways, and in this case identify an entire "signaling circuit".



(a)

(c)

Figure 6.2: **(a)** Overlap between subnetworks found by the enhanced influence model and significant pathways reported in [Net08]. The genes in the network shown have been reported as involved in significant pathways in [Net08]. Each circle is a gene, gray nodes represents protein families or complexes, or small molecules. For each protein family and complex, tested genes are shown. "Dashed" nodes are tested genes that were not mutated in GBM, and thus cannot be returned as significant. Red nodes are found in the c.c. of size 22, blue nodes in the c.c. of size 18, and the green node in a c.c. of size 2. **(b)** Pathway corresponding to one of the connected components extracted with enhanced influence model in lung. **(c)** Notch signaling pathway identified in the lung dataset.

		H_0^{sample}		H_0^{gene}		
s	# c.c. $\geq s$	μ	p-val	μ	p-val	enrichment p-val
2	24	23.4	0.7	17.67	0.4	/
3	11	6.51	0.13	7.27	0.2	/
4	7	3.21	0.07	4.98	0.13	/
5	5	2.09	0.01	2.18	0.01	/
7	4	0.54	0.01	0.56	0.01	–
10	3	$< 10^{-3}$	$< 10^{-3}$	0.4	0.02	0.34 10^{-5} 9×10^{-8}

Table 6.6: Results of the enhanced influence model on lung adenocarcinoma samples. Columns are as described in Table 6.3. Last column shows, for c.c. with $s \geq 7$, the result of the hypergeometric test for enrichment all genes reported in significant pathways in [D+08] (the 3 values shown refers to c.c. of size 10).

size	genes
10	WT1 CDKN2A TP53 CCNG1 KLF6 ATR CDKN2C TP73L TFDP1 CHEK1
10	RAP2B PIK3CA HRAS RASSF2 NRAS MRAS PIK3CG BRAF NF1 RHOB
10	EPHB1 EPHB6 EPHA7 EPHA6 EPHA5 EPHA4 EPHA3 EPHA2 EPHA1 FGFR4
7	MAML2 MAML1 NOTCH4 NOTCH2 NOTCH3 JAG2 JAG1

Table 6.7: Connected components of size $\geq$ 7 for lung adenocarcinoma using the diffusion model with $\gamma = 8, t = 0.01$.

6.5.3 Naïve approach

To demonstrate the impact of the influence graph on the results, we implemented a naïve approach that examines all paths in the original HPRD network that connect two tested genes and contain at most 3 nodes. We extracted all paths that were altered in a significant number of samples with FDR ≤ 0.01 using the standard Benjamini-Yekutieli method [BY01], considering each path as an hypothesis. More than 1700 paths in GBM and > 2200 in lung adenocarcinoma are marked as significant with this method. A major reason for this large number of paths is the presence of highly mutated genes that are also high-degree nodes in the HPRD network (e.g. TP53). *Each* path through these high degree nodes is marked as significant, thus a large number of "uninteresting" subnetworks are deemed significant. One possible solution is to remove any path that contains a subpath that is significant. However, these filtered paths include *none* through highly-mutated and high degree genes that are biologically important for cancer (like TP53). Our influence graph uses both mutation frequency and local topology of the network, allowing us to recover subnetworks containing these genes. Finally, we note that finding larger, statistically significant subnetworks (e.g. those with 10 or 20 nodes) with the naïve approach is impossible in the GBM and lung datasets because of the severe multiple hypotheses correction for the large number of subnetworks tested; e.g., the number of connected components with 10 tested nodes in the HPRD network is $> 10^{10}$. For the same reason the enumeration of all the paths or connected components of reasonable size is impossible.

Table 6.8 shows the significant paths containing at most 3 analyzed genes that have been found significant using the random model H_0^{sample} and the Benjamini-Yekutieli method to correct for multiple hypothesis test using GBM somatic mutations. In the table only paths that do not contain any subpath that is significant are shown (e.g., all the paths with > 1 gene that are significant and contain TP53 are not reported). Table 6.9 shows the analogous table for Lung adenocarcinoma

genes	# mutated samples	p-value
TP53	31	1.11022×10^{-16}
PTEN	28	1.11022×10^{-16}
EGFR	15	2.55351×10^{-15}
NF1	13	1.00975×10^{-12}
PIK3R1	9	6.87229×10^{-08}
RB1	9	6.87229×10^{-08}
DST	8	8.75524×10^{-07}
ERBB2	7	9.93594×10^{-06}
PDGFRB , PIK3CA	8	0.00010412
PIK3CA, PRKCD, EP300	10	5.71599×10^{-05}
PIK3CA, IRS4, PRKCZ	8	0.00010412

Table 6.8: Statistically significant mutated paths (FDR = 0.01) using the HPRD network [P+09] and the glioblastoma mutations dataset [Net08]. For each significant path, the genes in the path, the number of samples with at least one mutation in the path, and the (non-corrected) p-value are shown.

genes	# mutated samples	p-value
TP53	64	$< 10^{-16}$
KRAS	60	$< 10^{-16}$
STK11	34	$< 10^{-16}$
EGFR	30	$< 10^{-16}$
LRP1B	16	1.97591×10^{-11}
ATM	13	1.65488×10^{-08}
NF1	13	1.65488×10^{-08}
APC	11	1.02906×10^{-06}
CDKN2A, E4F1, RB1	15	1.28117×10^{-06}
CDKN2A, WRN, PRKDC	15	1.28117×10^{-06}
EPHA7, EFNA1, EPHA3	15	1.28117×10^{-06}
PRKDC, HSP90AA1 , KDR	15	1.28117×10^{-06}
EPHA3 , EFNA2, EPHA5	15	1.28117×10^{-06}
NTRK3, DYNLL1, NTRK1	14	6.16984×10^{-06}
NTRK1, CAV1 , KDR	14	6.16984×10^{-06}
KDR, ITGB3, PDGFRA	14	6.16984×10^{-06}

Table 6.9: Statistically significant mutated paths (FDR = 0.001) using the HPRD network [P$^+$09] and the lung adenocarcinoma mutations dataset [D$^+$08]. For each significant path, the genes in the path, the number of samples with at least one mutation in the path, and the (non-corrected) p-value are shown.

Chapter 7

Conclusions

In this final chapter we summarize the main contributions of this thesis and discuss some future research directions.

7.1 Summary

In this thesis we contributed novel results on the mining of significant patterns, focusing on the problem of frequent itemsets mining, a fundamental primitive that arises in many data mining problems, on the extraction of motifs from biological sequences, and on the discovery of significantly mutated pathways in cancer.

In chapter 3 we studied the algorithmic aspects of the extraction of top-K frequent closed itemsets and the use of sampling to extract the top-K frequent items/itemsets. For the first primitive we provide the first analytical evidence of its effectiveness, proving a tight upper bound on the ratio between the number of closed itemsets returned in output and the input parameter K. We then developed a new algorithm for mining top-K frequent closed itemsets in order of decreasing support, TopKMiner, which attains substantial improvements w.r.t. the best previously know algorithm. A peculiar feature of our algorithm is that it allows the user to dynamically raise the value K, without requiring the computation to restart from scratch. For the extraction of top-K frequent items/itemsets through sampling we proved a tight bound on the sufficient sample size to obtain an approximation the top-K frequent items/itemsets with probabilistic guarantees on the quality of the output. Moreover, we develop an algorithm based on progressive sampling to extract the top-K frequent items/itemsets.

In Chapter 4 we proposed a novel methodology to identify a meaningful support threshold σ^* for a dataset such that the itemsets with support at least σ^* can be flagged as statistically significant with a small False Discovery Rate (FDR), which is the expected ratio of false discoveries among all discoveries. Our methodology hinges on a Poisson approximation to the distribution of the number of itemsets in a random dataset with support at least s, for any s greater than or equal to a minimum threshold $s_{\min}$. We obtained this result through a novel application of the Chen-Stein approximation method, which is of independent interest. A crucial feature of our approach is that, unlike most previous work, it takes into account the entire dataset rather than individual discoveries. It is therefore better able to distinguish between significant observations and random fluctuations. The results of our comparison to a standard procedure for multi-hypothesis testing provide experimental evidence of the higher power of our approach.

In Chapter 5 we studied the discovery of *motifs*, possibly including don't care characters, in biological sequences. This problem is highly relevant to computational biology. We introduced the *density*, defined as the ratio of solid characters to the total length of the motif, as a simple and flexible measure for bounding the number of don't cares in a motif,. We define a natural notion of *maximality* for *dense motifs* and devise an efficient algorithm, called MADMX which performs complete MAximal Dense MOtif eXtraction from an input sequence, with respect to user-specified frequency and density thresholds. We provided experimental evidence of the efficiency and the quality of the motifs returned by MADMX, comparing them with the known biological repetitions, and with the motifs extracted by the recently developed tool VARUN [ACP09] using the same statistical metric employed in [ACP09] for assessing their relative significance.

Finally, in Chapter 6 we addressed the problem of identifying *significantly mutated pathways* in large scale gene and protein interaction networks. We proposed a new framework based on an *influence* measure between pairs of genes obtained using a diffusion process defined on the interaction network. We then proposed two algorithms to identify significantly mutated pathways, both using the influence measure between pairs of genes. Moreover, we derived a statistical test that identifies significantly mutated pathways and estimates the FDR of the identified subnetworks. This test is built on the technique we developed in Chapter 4 in the context of frequent itemset mining. We tested the algorithms on a large human protein-protein interaction network using mutation data from recent studies on two different type of cancers. The tests showed that our methods successfully recover pathways that are known to be involved in the considered cancers, and moreover identify additional pathways that have been implicated in cancer but not previously reported as mutated in the samples we considered.

7.2 Further research

There are a number of interesting avenues to improve the results presented in this thesis and to develop new methods to mine significant patterns.

A first set of possible directions regards the mining primitives we have studied in Chapter 3. For the extraction of top-K frequent closed itemsets, a natural direction is the development and testing of an external memory algorithm for the problem. Since many datasets of interest for this problem are huge, they will probably not fit in main memory, and new algorithms explicitly designed to work on external memory are needed. For the use of sampling to extract top-K frequent items/itemsets it would be interesting to study, both analytically and experimentally, the performance of our algorithm on datasets with different items/itemsets distributions, trying to characterize what are the distributions for which our algorithm gives the best performance. Another direction for future work is the experimental assessment of the algorithm based on min-count Bloom filter we proposed.

For what concern instead the mining of statistically significant itemsets, the framework we have introduced offers several interesting directions for further work. Naturally, one goal is to adapt our test to different random models, for example the one introduced in [GMMT07]. Another interesting direction is the design of a method that extract statistically significant itemsets with low supports. Moreover, the statistical test we have proposed can be adapted to the extraction of other patterns, as we have done in Chapter 6 for the extraction of significantly mutated pathway. We think that the mining of graphs, for example, would provide an interesting application of our method.

The extraction of significant motifs provides many interesting directions for future work. Our definition of density provides a way to constrain the structure of the motifs so to enforce significance more general. than the ones previously employed, but the choice of the density and frequency thresholds are left to the user. An important problem is then to understand what is the relation between those parameters and the biological significance of the corresponding motifs. Another interesting direction is the design of an algorithm that extracts the maximal dense motifs from a set of sequences, where the frequency of a pattern is the number of sequences in which it appears. MADMX can be used to solve this problem (by concatenating the input sequences), but novel algorithmic solutions could result in better performance.

For the identification of significant pathways in cancer much work remains to be done. For example, we model the protein interaction network as an undirected graph, while information on the directionality of some interactions is already available, and more will be produced in the next few years. Adapting our models and methods to directed graphs requires new solutions. Moreover, somatic mutations are not the only causes that lead to cancer. Other genomic alterations, like copy number modifications or epigenetic alterations, have been related to cancer. How to analyze different type of alterations, and how to combine them, to identify the pathways specific to cancers is one of the most interesting problems that our method does not currently tackle.

Bibliography

[A$^+$00] M. Ashburner et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics*, 25:25–29, 2000.

[ACP09] A. Apostolico, M. Comin, and L. Parida. Varun: Discovering extensible motifs under saturation constraints. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 99(1), 2009.

[AGG90] R. Arratia, L. Goldstein, and L. Gordon. Poisson approximation and the Chen-Stein method. *Statistical Science*, 5(4):403–434, 1990.

[AIS93] R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[AP04] A. Apostolico and L. Parida. Incremental paradigms of motif discovery. *Journal of Computational Biology*, 11(1):15–25, 2004.

[AS94] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the International Conference on Very Large Data Bases*, pages 487–499, 1994.

[AT07] A. Apostolico and C. Tagliacollo. Optimal offline extraction of irredundant motif bases. In *Proceedings of the International Computing and Combinatorics Conference*, pages 360–371, 2007.

[AT08] A. Apostolico and C. Tagliacollo. Incremental discovery of the irredundant motif bases for all suffixes of a string in n time. *Theoretical Computer Science*, 408(2-3):106–115, 2008.

[AU07] H. Arimura and T. Uno. Mining maximal flexible patterns in a sequence. In *Proceedings of the Annual Conference of The Japanese Society for Artificial Intelligence*, pages 307–317, 2007.

[Axe04] H. Axelson. Notch signaling and cancer: emerging complexity. *Seminars in Cancer Biology*, 14:317–319, 2004.

[AY98] C. C. Aggarwal and P. S. Yu. A new framework for itemset generation. In *Proceedings of the ACM Symposium on Principles of Database Systems*, pages 18–24, 1998.

[B$^+$01] G. D. Bader et al. BIND–The Biomolecular Interaction Network Database. *Nucleic Acids Research*, 29:242–245, Jan 2001.

[Bay98] R. J. Bayardo Jr. Efficiently mining long patterns from databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 85–93, 1998.

[BGKM03] E. Boros, V. Gurvich, L. Khachiyan, and K. Makino. On maximal frequent and minimal infrequent sets in binary matrices. *Annals of Mathematcs and Artificial Intelligence*, 39(3):211–221, 2003.

[BGZ04] R. J. Bayardo Jr., B. Goethals, and M. J. Zaki, editors. *FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, volume 126 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2004.

[BH95] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[BHA02] R. J. Bolton, D. J. Hand, and N. M. Adams. Determining hit rate in pattern search. In *Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery*, pages 36–48, 2002.

[Bod04] F. Bodon. Surprising results of trie-based fim algorithms. In *FIMI, Proceedings of the ICDM 2004 Workshop on Frequent Itemset Mining Implementations*, 2004.

[BY01] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of Statistics*, 4(29):1165–1188, 2001.

[CCFC04] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3–15, 2004.

[CF04] Y.-L. Cheung and A. W.-C. Fu. Mining frequent itemsets without support threshold: With and without item constraints. *IEEE Transactions on Knowledge Data Engineering*, 16(9):1052–1069, 2004.

[CGK08] E. Cohen, N. Grossaug, and H. Kaplan. Processing top-k queries from samples. *Computer Networks*, 52(14):2605–2622, 2008.

[CHS02] B. Chen, P. Haas, and P. Scheuermann. A new two-phase sampling based algorithm for discovering association rules. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*, pages 462–468, 2002.

[Chu07] F. Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735, 2007.

[CKB04] B. J. Collins, W. Kleeberger, and D. W. Ball. Notch in lung development and lung cancer. *Seminars in Cancer Biology*, 14:357–364, 2004.

[CLL$^+$07] H. Y. Chuang, E. Lee, Y. T. Liu, D. Lee, and T. Ideker. Network-based classification of breast cancer metastasis. *Molecular Systems Biology*, 3:140, 2007.

[D$^+$08] L. Ding et al. Somatic mutations affect key pathways in lung adenocarcinoma. *Nature*, 455(7216):1069–75, 2008.

[DP01] W. DuMouchel and D. Pregibon. Empirical bayes screening for multi-item associations. In *Proceedings of the ACM International conference on Knowledge Discovery and Data Mining*, pages 67–76, 2001.

[DS84] P.G. Doyle and J.L. Snell. *Random Walks and Electric Networks.* The Mathematical Association of America, 1984.

[DSB03] S. Dudoit, J. P. Schaffer, and J. C. Boldrick. Multiple hypothesis testing in microarray experiments. *Statistical Science*, 18(1):71–103, 2003.

[DuM99] W. DuMouchel. Bayesian data mining in large frequency tables, with an application to the FDA spontaneous reporting system. *The American Statistician*, 53(3):177–190, August 1999.

[FA07] P. G. Ferreira and P. J. Azevedo. Evaluating deterministic motif significance measures in protein databases. *Algorithms for Molecular Biology*, 2, 2007.

[FKT00] A. W.-C. Fu, R. W.-w. Kwong, and J. Tang. Mining n-most interesting itemsets. In *Proceedings of the International Symposium on Foundations of Intelligent Systems*, pages 59–67, 2000.

[G$^+$07] C. Greenman et al. Patterns of somatic mutation in human cancer genomes. *Nature*, 446:153–158, 2007.

[GMMT07] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas. Assessing data mining results via swap randomization. *ACM Transactions on Knowledge Discovery from Data*, 1(3):14, 2007.

[GPP$^+$09] R. Grossi, A. Pietracaprina, N. Pisanti, G. Pucci, E. Upfal, and F. Vandin. MADMX: A novel strategy for maximal dense motif extraction. In *Proceedings of Workshop on Algorithms in Bioinformatics*, pages 362–374, 2009.

[GZ03] B. Goethals and M. J. Zaki, editors. *FIMI '03, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations*, volume 90 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.

[HCXY07] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

[HK01] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Mateo, CA, 2001.

[HLCS09] B. J. Hescott, M. D. M. Leiserson, L. Cowen, and D. K. Slonim. Evaluating between-pathway models with expression data. In *Proceedings of the International Conference on Research in Computational Molecular Biology*, pages 372–385, 2009.

[HN08] W. Hämäläinen and M. Nykänen. Efficient discovery of statistically significant association rules. In *Proceedings of the IEEE International Conference on Data Mining*, pages 203–212, 2008.

[Hoc97] D. S. Hochbaum, editor. *Approximation algorithms for NP-hard problems*. PWS Publishing Co., Boston, MA, USA, 1997.

[HW02] W. C. Hahn and R. A. Weinberg. Modelling the molecular circuitry of cancer. *Nature Reviews Cancer*, 2(5):331–41, 2002.

[IOSS02] T. Ideker, O. Ozier, B. Schwikowski, and A. F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18 Suppl 1:S233–240, 2002.

[J$^+$08] S. Jones et al. Core signaling pathways in human pancreatic cancers revealed by global genomic analyses. *Science*, 321(5897):1801–6, 2008.

[J$^+$09] L. J. Jensen et al. STRING 8–a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Research*, 37:D412–416, Jan 2009.

[JB06] P. F. Jonsson and P. A. Bates. Global topological features of cancer proteins in the human interactome. *Bioinformatics*, 22:2291–2297, 2006.

[JKP$^+$05] J. Jurka, V. V. Kapitonov, A. Pavlicek, P. Klonowski, O. Kohany, and J. Walichiewicz. Repbase update, a database of eukaryotic repetitive elements. *Cytogenetic and Genome Research*, 110(1-4):462–467, 2005.

[JL96] G. H. John and P. Langley. Static versus dynamic sampling for data mining. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*, pages 367–370, 1996.

[JS05] S. Jaroszewicz and T. Scheffer. Fast discovery of unexpected patterns in data, relative to a bayesian network. In *In Proceedings of the ACM International conference on Knowledge Discovery in Data Mining*, pages 118–127, 2005.

[KL02] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the International Conference on Machine Learning*, pages 315–322, 2002.

[KMP$^+$09a] A. Kirsch, M. Mitzenmacher, A. Pietracaprina, G. Pucci, E. Upfal, and F. Vandin. An efficient rigorous approach for identifying statistically significant frequent itemsets. In *Proceedings of the ACM Symposium on Principles of Database Systems*, pages 117–126, 2009.

[KMP$^+$09b] A. Kirsch, M. Mitzenmacher, A. Pietracaprina, G. Pucci, E. Upfal, and F. Vandin. An efficient rigorous approach for identifying statistically significant frequent itemsets. Submitted to *Journal of the ACM* (JACM), 2009.

[Knu73] D. E. Knuth. *The Art of Computer Programming, Volume III: Sorting and Searching.* Addison-Wesley, 1973.

[KSS09] S. Karni, H. Soreq, and R. Sharan. A network-based method for predicting disease-causing genes. *Journal of Computational Biology*, 16:181–189, 2009.

[L$^+$07a] J. Lin et al. A multidimensional analysis of genes mutated in breast and colorectal cancers. *Genome Research*, 17:1304–1318, 2007.

[L$^+$07b] M. Liu et al. Network-based analysis of affected biological processes in type 2 diabetes models. *PLoS Genetics*, 3:e96, 2007.

[LACB09] M. Lavallée-Adam, B. Coulombe, and M. Blanchette. Detection of locally over-represented go terms in protein-protein interaction networks. In *Proceedings of the International Conference on Research in Computational Molecular Biology*, pages 302–320, 2009.

[LG04] Y. Li and R. P. Gopalan. Effective sampling for mining association rules. In *Proceedings of Australian Conference on Artificial Intelligence*, pages 391–401, 2004.

[LOP06] C. Lucchese, S. Orlando, and R. Perego. Fast and memory efficient mining of frequent closed itemsets. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):21–36, 2006.

[LOPS04] C. Lucchese, S. Orlando, R. Perego, and F. Silvestri. Webdocs: a real-life huge transactional dataset. In *FIMI, Proceedings of the ICDM 2004 Workshop on Frequent Itemset Mining Implementations*, 2004.

[Lov93] L. Lovász. Random walks on graphs: A survey, 1993.

[LV03] P. Lyman and H. Varian. How much information? 2003. School of Information Management and Systems at the University of California at Berkeley, 2003.

[MAA05] A. Metwally, D. Agrawal, and A. El Abbadi. Efficient computation of frequent and top-k elements in data streams. In *Proceedings of International Conference on Database Theory*, pages 398–412, 2005.

[MLWS07] X. Ma, H. Lee, L. Wang, and F. Sun. CGI: a new approach for prioritizing genes by combining gene expression and protein-protein interaction data. *Bioinformatics*, 23:215–221, 2007.

[MNU08] M. Michael, F. Nicolas, and E. Ukkonen. On the complexity of finding gapped motifs. *CoRR*, abs/0802.0314, 2008.

[MS98] N. Megiddo and R. Srikant. Discovering predictive association rules. In *Proceedings of the ACM International Conference on Knowledge Discovery in Databases and Data Mining*, pages 274–278, 1998.

[MU05] M. Mitzenmacher and E. Upfal. *Probability and Computing : Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, January 2005.

[NCTLH07] S. Nacu, R. Critchley-Thorne, P. Lee, and S. Holmes. Gene expression network analysis and applications to immunology. *Bioinformatics*, 23:850–858, 2007.

[Net08] The Cancer Genoma Atlas Network. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–8, 2008.

[NJA$^+$05] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21 Suppl 1:i302–310, 2005.

[P$^+$08] D. W. Parsons et al. An integrated genomic analysis of human glioblastoma multiforme. *Science*, 321(5897):1807–12, 2008.

[P$^+$09] T. S. K. Prasad et al. Human Protein Reference Database–2009 update. *Nucleic Acids Research*, 37:D767–772, 2009.

[Par00] L. Parida. Some results on flexible-pattern discovery. In *Proceedings of the Annual Symposium on Combinatorial Pattern Matching*, pages 33–45, 2000.

[Par02] S. Parthasarathy. Efficient progressive sampling for association rules. In *Proceedings of IEEE International Conference on Data Mining*, pages 354–361, 2002.

[Par07] L. Parida. *Pattern Discovery in Bioinformatics: Theory & Algorithms.* Chapman & Hall/CRC, 2007.

[PBTL99] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Information Systems*, 24(1):25–46, 1999.

[PCGS05] N. Pisanti, M. Crochemore, R. Grossi, and M.-F. Sagot. Bases of motifs for generating repeated patterns with wild cards. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2(1):40–50, 2005.

[Pis02] N. Pisanti. Segment-based distances and similarities in genomic sequences, 2002. PhD thesis, University of Pisa, Italy.

[PRUV09] A. Pietracaprina, M. Riondato, E. Upfal, and F. Vandin. Mining top-k frequent itemsets through sampling. Manuscript, 2009.

[PV07] A. Pietracaprina and F. Vandin. Efficient incremental mining of top-k frequent closed itemsets. In *Proceedings on International Conference on Discovery Science*, pages 275–280, 2007.

[PVGG04] P. W. Purdom, D. Van Gucht, and D. P. Groth. Average case performance of the apriori algorithm. *SIAM Journal on Computing*, 33 (5):1223–1260, 2004.

[PZ03] A. Pietracaprina and D. Zandolin. Mining frequent itemsets using patricia tries. In *FIMI, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations*, 2003.

[QSL$^+$08] Y. Qi, Y. Suhail, Y. Y. Lin, J. D. Boeke, and J. S. Bader. Finding friends and enemies in an enemies-only network: a graph diffusion kernel for predicting novel genetic interactions and co-complex membership from yeast genetic interactions. *Genome Research*, 18:1991–2004, 2008.

[RF98] I. Rigoutsos and A. Floratos. Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. *Bioinformatics*, 14(1):55–67, February 1998.

[S$^+$06] T. Sjoblom et al. The consensus coding sequences of human breast and colorectal cancers. *Science*, 314(5797):268–74, 2006.

[SA96] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. *ACM SIGMOD Record*, 25(2):1–12, 1996.

[SBM98] C. Silverstein, S. Brin, and R. Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2(1):39–68, 1998.

[SH06] T.-P. Shuai and X.-D. Hu. Connected set cover problem and its applications. In *Proceedings of International Conference on Algorithmic Aspects in Information and Management*, pages 243–254, 2006.

[SHG04] A.F.A. Smit, R. Hubley, and P. Green. *RepeatMasker Open-3.0*, 1996–2004. `http://www.repeatmasker.org`.

[SM04] J. K. Seppänen and H. Mannila. Dense itemsets. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*, pages 683–688, 2004.

[SMS$^+$04] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg. The Database of Interacting Proteins: 2004 update. *Nucleic Acids Research*, 32:D449–451, Jan 2004.

[SSPT98] L. Shen, H. Shen, P. Prithard, and R. Topor. Finding the n largest itemsets. In *Proceedings of the IEEE International Conference on Data Mining*, 98.

[ST96] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6):970–974, 1996.

[SVGP05] B. Sayrafi, D. Van Gucht, and P. W. Purdom. On the effectiveness and efficiency of computing bounds on the support of item-sets in the frequent item-sets mining problem. In *Proceedings of the International Workshop on Open Source Data Mining*, pages 46–55, 2005.

[TN04] K. Tsuda and W. S. Noble. Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, 20 Suppl 1:i326–333, 2004.

[Toi96] H. Toivonen. Sampling large databases for association rules. In *Proceedings of International Conference on Very Large Data Bases*, pages 134–145, 1996.

[TSK06] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining.* Addison Wesley, 2006.

[UAUA03] T. Uno, T. Asai, Y. Uchida, and H. Arimura. LCM: An efficient algorithm for enumerating frequent closed item sets. In *FIMI, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations*, 2003.

[UAUA04] T. Uno, T. Asai, Y. Uchida, and H. Arimura. An efficient algorithm for enumerating closed patterns in transaction databases. In *Proceedings of the International Conference on Discovery Science*, pages 16–31, 2004.

[Ukk07] E. Ukkonen. Structural analysis of gapped motifs of a string. In *Proceedings of the International Symposium on Mathematical Foundations of Computer Science*, pages 681–690, 2007.

[UKS08] I. Ulitsky, R. M. Karp, and R. Shamir. Detecting disease-specific dysregulated pathways via analysis of clinical expression profiles. In *Proceedings of the International Conference on Research in Computational Molecular Biology*, pages 347–359, 2008.

[VK04] B. Vogelstein and K. W. Kinzler. Cancer genes and the pathways they control. *Nature Medice*, 10:789–799, 2004.

[VUR09] F. Vandin, E. Upfal, and B. J. Raphael. Identification of significantly mutated pathways in cancer. *Abstract, RECOMB Satellite (Systems Biology)*, 2009.

[VUR10] F. Vandin, E. Upfal, , and B. J. Raphael. Algorithms for detecting significantly mutated pathways in cancer. In *Proceedings of the International Conference on Research in Computational Molecular Biology, to appear*, 2010.

[VV09] D. Vasudevan and M. Vjnović. Ranking through Random Sampling. *Microsoft Reasearch Technical Report*, 2009.

[W$^+$07] L. D. Wood et al. The genomic landscapes of human breast and colorectal cancers. *Science*, 318(5853):1108–13, 2007.

[WF06] R. C.-W. Wong and A. W.-C. Fu. Mining top- frequent itemsets from data streams. *Data Mining and Knowledge Discovery*, 13(2):193–217, 2006.

[WHLT05] J. Wang, J. Han, Y. Lu, and P. Tzvetkov. TFP: An efficient algorithm for mining top-k frequent closed itemsets. *IEEE Transactions on Knowledge and Data Engineering*, 17(5):652–664, 2005.

[XHYC05] D. Xin, J. Han, X. Yan, and H. Cheng. Mining compressed frequent-pattern sets. In *Proceedings of the International Conference on Very Large Data Bases*, pages 709–720. VLDB Endowment, 2005.

[Yan04] G. Yang. The complexity of mining maximal frequent itemsets and maximal frequent patterns. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pages 344–353, 2004.

[ZL77] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.

[ZPLO97] M. J. Zaki, S. Parthasarathy, W. Li, and M. Ogihara. Evaluation of sampling for data mining of association rules. In *Proceedings of International Workshop on Research Issues in Data Engineering*, page 42, 1997.

[ZPT04] H. Zhang, B. Padmanabhan, and A. Tuzhilin. On the discovery of significant statistical quantitative rules. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pages 374–383, 2004.

[ZPZ$^+$09] R. Zielinski, P. F. Przytycki, J. Zheng, D. Zhang, T. M. Przytycka, and J. Capala. The crosstalk between EGF, IGF, and Insulin cell signaling pathways–computational and experimental analysis. *BMC Systems Biology*, 3:88, 2009.