Significant Subgraph Mining with Multiple Testing Correction[*][†]

Mahito Sugiyama[‡][§][¶] Felipe Llinares López[‖] Niklas Kasenburg[**] Karsten M. Borgwardt[‖]

Abstract

The problem of finding itemsets that are statistically significantly enriched in a class of transactions is complicated by the need to correct for multiple hypothesis testing. Pruning *untestable hypotheses* was recently proposed as a strategy for this task of significant itemset mining. It was shown to lead to greater statistical power, the discovery of more truly significant itemsets, than the standard Bonferroni correction on real-world datasets. An open question, however, is whether this strategy of excluding untestable hypotheses also leads to greater statistical power in *subgraph* mining, in which the number of hypotheses is much larger than in itemset mining. Here we answer this question by an empirical investigation on eight popular graph benchmark datasets. We propose a new efficient search strategy, which always returns the same solution as the state-of-the-art approach and is approximately two orders of magnitude faster. Moreover, we exploit the dependence between subgraphs by considering the *effective number of tests* and thereby further increase the statistical power.

Keywords: Statistical significance, Multiple hypothesis testing, Frequent subgraph mining, Bonferroni correction, Testability

1 Introduction

A *graph* is one of the most general data types to represent structured objects, and massive amounts of structured data are now available as graphs across a wide range of domains, such as chemical compounds in PubChem [5], biological pathways in KEGG [13], protein structures in PDB [3], and social networks on the web. Analyzing such databases, that is, *graph mining*, has evolved into an important branch of data mining and knowledge discovery. Graph databases often include two or more distinct classes of graphs and, in many application domains, the ultimate purpose is to discover *significant subgraphs* that are statistically significantly enriched in one particular class of graphs. In drug discovery, for instance, chemists try to identify a key substructure of chemical compounds which is significantly associated with a particular activity, e.g., anticancer activity [25]. In a similar fashion, biologists seek substructures of proteins that are required for particular docking events [31].

Finding such significant subgraphs is an open problem, as the large number of candidate subgraphs causes both a computational and a statistical problem: the computational problem is that it is often extremely expensive to check all subgraphs for enrichment, given that their number scales exponentially in the number of nodes of the largest graph in the database. The statistical problem is the multiple hypothesis testing problem caused by the fact that a huge number—often billions—of subgraphs are being tested for significant enrichment, each of which represents a hypothesis. If one ignores this multiple testing problem, one may find an enormous number of *false positives*, subgraphs that are deemed to be significant by mistake. In particular in the natural sciences, where significant subgraphs typically undergo further experimental investigation, a large number of false positives leads to a severe waste of time and resources. Thus *multiple testing correction*, calibration of the significance level in each test, is needed to control the total error rate of false positives.

Our goal in this paper is to overcome these two problems: we present *efficient* strategies to detect significantly enriched subgraphs *while correcting for multiple testing.*

A common approach to multiple testing correction is Bonferroni correction [6]. It tends to be highly conservative, that is, it will miss many significant observations if the number of tests performed is massive, as in graph mining or pattern mining in general. Tarone [26] proposed an improved, less conservative Bonferroni correction on categorical data. Key to this strategy is that on categorical data, only a subset of tests, called

[*] Supplementary information is available at `http://mahito.info/files/sdm15supp.pdf`

[†] This work was funded in part by a Grant-in-Aid for Scientific Research (Research Activity Start-up) 26880013 (MS), the SNSF Starting Grant “Significant Pattern Mining” (KMB), the Alfried Krupp von Bohlen und Halbach-Stiftung (KMB), and the Marie Curie Initial Training Network MLPM2012, Grant No. 316861. (FLL, KMB).

[‡] ISIR, Osaka University

[§] JST, PRESTO

[¶] Contact: mahito@ar.sanken.osaka-u.ac.jp

[‖] D-BSSE, ETH Zürich

[**] Department of Computer Science, University of Copenhagen

testable hypotheses, can reach significance, thereby hypotheses that are not testable can be safely removed without affecting the probability of reporting false positives. Terada et al. [27] recently made it possible to enumerate testable hypotheses using a frequent itemset mining algorithm and successfully applied Tarone's insight for discovering significant combinations of transcription factors in gene regulatory network analysis.

A relevant question for graph mining is whether Tarone's strategy of only correcting for testable hypotheses can be successfully transferred to significant subgraph mining as well. This is not a trivial question as the search space in graph mining is often exponentially larger than that in itemset mining due to combinations of vertices and edges. In this paper, we give a positive answer to this question by (1) extending the approach by Terada et al. [27] to solve the important open problem of significant subgraph mining with multiple testing correction via frequent subgraph mining [7, 12, 19, 34], (2) proposing efficient search strategies for detecting testable subgraphs, one of which is empirically orders of magnitude faster than their method, and (3) further improving over naïve Bonferroni correction by considering the dependence between subgraph occurrences [18, 20].

This paper is organized as follows: we present our approach to significant subgraph mining in Section 2. First we provide the necessary statistical concepts and problem statements (Sections 2.1, 2.2, and 2.3), then we propose search algorithms for significant subgraph detection in Section 2.4, followed by introducing the improved multiple testing correction via the effective number of tests in Section 2.5. We discuss related work in Section 3 and evaluate our algorithms on real-world datasets in Section 4. Finally, we summarize our contributions in Section 5.

2 Method

Let G be a *graph*, which is mathematically defined as an ordered pair of vertices $V(G)$ and edges $E(G) \subseteq V(G) \times V(G)$. A graph H is a *subgraph* of G, denoted by $H \sqsubseteq G$, if its vertex set $V(H)$ is a subset of $V(G)$ and its edge set $E(H)$ is a subset of $E(G)$ and is restricted to its vertices, i.e., $V(H) \subseteq V(G)$ and $E(H) \subseteq (V(H) \times V(H)) \cap E(G)$.

In the following we assume that our datasets of graphs comprises two classes of graphs, but our results also transfer to more than two classes when considering one-versus-rest classification, that is enrichment of a subgraph in one class versus all others.

2.1 Statistically significant subgraphs.

Suppose we are given two collections of graphs $\mathcal{G}$ and $\mathcal{G}'$, where the numbers of graphs in these sets are $|\mathcal{G}| = n$ and $|\mathcal{G}'| = n'$ with $n \leq n'$ without loss of generality. For each subgraph $H \sqsubseteq G$ with $G \in \mathcal{G} \cup \mathcal{G}'$, we formulate a *null hypothesis* that the occurrence of the subgraph H is independent from the class membership of G. Our task is to find for which subgraphs H the data provide enough evidence to *reject* the null hypothesis and to deem H as a significant subgraph associated with the class membership.

From given data, we measure the statistical association between two binary random variables: the indicator vector of the class membership and the occurrence/absence of the subgraph H within each graph G in the database.

Let x and x' be the frequencies of H in $\mathcal{G}$ and $\mathcal{G}'$, respectively. That is, $x = |\{\, G \in \mathcal{G} \mid H \sqsubseteq G \,\}|$ and $x' = |\{\, G \in \mathcal{G}' \mid H \sqsubseteq G \,\}|$, as represented in the following 2×2 contingency table.

	Occurrences	Non-occurrences	Total
$\mathcal{G}$	x	$n - x$	n
$\mathcal{G}'$	x'	$n' - x'$	n'
Total	$x + x'$	$(n - x) + (n' - x')$	$n + n'$

The strength of the association between binary random variables is quantified as a *p-value*, defined as the probability of observing an association at least as strong as the one present in the data under the assumption that the null hypothesis of independence holds true. To compute the p-value, *Fisher's exact test* is commonly used. It relies on the fact that, when the margins $x + x'$, n, and $n + n'$ are fixed, the probability $q(x)$ of obtaining these counts x and x' is given by the hypergeometric distribution:

$$q(x) = \binom{n}{x}\binom{n'}{x'} \bigg/ \binom{n+n'}{x+x'}.$$

Formally, define P_{L} and P_{R} as the left-tail and the right-tail of the hypergeometric distribution, respectively. That is, given an observed count x, P_{L} is the probability of observing a smaller count and P_{R} the probability of observing a larger one:

$$P_{\mathrm{L}} = \sum_{X=\max\{0,x+x'-n'\}}^{x} q(X), \quad P_{\mathrm{R}} = \sum_{X=x}^{\min\{x+x',n\}} q(X),$$

They are used as one-tailed p-values, and a two-tailed p-value P_{D} is defined as[1] $P_{\mathrm{D}} = 2\min\{P_{\mathrm{L}}, P_{\mathrm{R}}\}$ [4].

[1] We can choose other definitions for a two-tailed test, e.g., summing up all probabilities that are smaller than $q(x)$. The analysis in this paper still holds with minor modifications.

We say that a subgraph H is *statistically significant* if its p-value is smaller than a predetermined significance level α. Note that, by construction, α equals to the *Type I error probability*; the probability of falsely deeming a subgraph significant.

2.2 Multiple hypothesis testing. In our setup, one must test *all* subgraphs in a database. The procedure described above guarantees that, for a single subgraph, the probability of being a false positive is upper bounded by α. However, when many hypotheses are tested in parallel, the probability that at least one subgraph is a false positive, called the *Family-Wise Error Rate* (FWER), approaches one. This is the well-known *multiple hypothesis testing* problem.

To deal with this issue, one needs to *correct* the significance level α in each test to guarantee that FWER $\leq$ α. The most common method is the *Bonferroni correction* [6], which simply divides α by the number m of tests. The resulting FWER can be readily shown to be smaller than α. The number of tests m is called the *Bonferroni factor*, which in our case is the same as the number of subgraphs. Despite its popularity, the Bonferroni correction is known to be too conservative in many cases, that is, the statistical power, the probability to detect truly significant subgraphs, becomes too small. The problem is even more extreme in our application: as m is the huge number of subgraphs tested, the Bonferroni corrected significance level α/m is so small that hardly any subgraph can ever reach the significance level.

2.3 Testable subgraphs. Tarone [26] showed that when testing the association of discrete random variables, as in our setup, one can improve the Bonferroni correction. The key idea is that the discreteness of the problem implies the existence of a minimum achievable p-value for each subgraph H. Let $f(H) = |\{ G \in \mathcal{G} \cup \mathcal{G}' \mid H \sqsubseteq G \}| = x + x'$ be the frequency of H in the whole set of graphs $\mathcal{G} \cup \mathcal{G}'$, and assume that $f(H) \leq n$.

If the marginals $f(H)$, n, and n' are fixed, the minimum p-value, denoted by $\psi(f(H)) = \psi \circ f(H)$, is achieved for the most biased case when $x = 0$ or $x = f(H)$. Since P_{L} and P_{R} are minimized at $x = \max\{0, f(H) - n'\}$ and $x = \min\{f(H), n\}$, their minimum values are $q(0)$ and $q(f(H))$, respectively. From $n \leq n'$, $q(f(H)) \leq q(0)$ holds. Thus we have

$$\psi \circ f(H) = q(f(H)) = \binom{n}{f(H)} \bigg/ \binom{n+n'}{f(H)}$$

for a one-tailed test, and this value is doubled for a two-tailed test. If $f(H) > n$ and hence $f(H) = x + x' > (n + n')/2$, we follow the definition in [27, Supporting Text 4], that is, we simply define $\psi \circ f(H) = 1/\binom{n+n'}{n}$. Then ψ is always monotonically decreasing, which is required for our algorithms.

If the minimum p-value $\psi \circ f(H)$ is larger than the significance threshold, the subgraph H can never be significant regardless of the class membership of the graphs in which it occurs. Tarone's insight is that such *untestable* subgraphs do not increase the FWER, and hence we can exclude them from candidate subgraphs and reduce the Bonferroni factor. Formally, let $\mathcal{H} = \{ H \sqsubseteq G \mid G \in \mathcal{G} \cup \mathcal{G}' \}$ be the set of all subgraphs in the database and define for each natural number k

$$m(k) = |\{ H \in \mathcal{H} \mid \psi \circ f(H) \leq \alpha/k \}|,$$

as the number of subgraphs whose minimum achievable p-value is smaller than α/k. Let k_{rt} satisfy

$$m(k_{\mathrm{rt}} - 1) > k_{\mathrm{rt}} - 1 \text{ and } m(k_{\mathrm{rt}}) \leq k_{\mathrm{rt}},$$

that is, k_{rt} is the rounded *root* of $m(k) - k$. Since $m(k)$ monotonically decreases as k increases, we have $m(k) - k > 0$ for all $k < k_{\mathrm{rt}}$ and $m(k) - k \leq 0$ for all $k \geq k_{\mathrm{rt}}$. Then we can see that FWER $\leq \alpha$ even if we reduce the Bonferroni factor from $|\mathcal{H}|$ to $m(k_{\mathrm{rt}})$, since we have

$$\begin{aligned} \text{FWER} &\leq \sum \{ \psi \circ f(H) \mid \psi \circ f(H) \leq \alpha/k_{\mathrm{rt}},\ H \in \mathcal{H} \} \\ &\leq m(k_{\mathrm{rt}}) \frac{\alpha}{k_{\mathrm{rt}}} \leq \alpha. \end{aligned}$$

As a result, we have the set of *testable subgraphs* $\tau(\mathcal{H})$, which is given by

$$\tau(\mathcal{H}) = \{ H \in \mathcal{H} \mid \psi \circ f(H) \leq \alpha/k_{\mathrm{rt}} \},$$

and our task of detecting all significant subgraphs is achieved by finding the root k_{rt} and enumerating the set $\tau(\mathcal{H})$ of testable subgraphs.

Terada et al. [27] used Tarone's method in the context of itemset mining for discovering gene regulatory motifs, where efficient enumeration of testable itemsets was achieved by applying a frequent itemset mining algorithm. Next we show how to apply Tarone's method to significant subgraph mining.

2.4 Enumeration of testable subgraphs. To use Tarone's results for our purpose, the challenge is now to efficiently compute all testable hypotheses, that is all testable subgraphs. Here we show how to use *frequent subgraph mining* to enumerate all testable subgraphs. Frequent subgraph mining algorithms find all subgraphs whose frequencies are higher than the user specified threshold σ (or its ratio $\theta = \sigma/(n + n')$). Since the minimum p-value ψ is a monotonically decreasing function (the proof is provided in [27, Supporting Text 4]), we have $\psi \circ f(H) \leq \psi(\sigma)$ for every frequent subgraph H.

PROPOSITION 2.1. *The set of testable subgraphs $\tau(\mathcal{H})$ coincides with the set of frequent subgraphs for the threshold σ_{rt} such that*

$$|\{ H \in \mathcal{H} \mid f(H) \geq (\sigma_{\text{rt}} - 1) \}| > \alpha/\psi(\sigma_{\text{rt}} - 1),$$
$$|\{ H \in \mathcal{H} \mid f(H) \geq \sigma_{\text{rt}} \}| \leq \alpha/\psi(\sigma_{\text{rt}})$$

Proof. We have for $k_{\text{rt}} = \alpha/\psi(\sigma_{\text{rt}})$, $m(k_{\text{rt}}) = m(\alpha/\psi(\sigma_{\text{rt}})) = |\{ H \in \mathcal{H} \mid \psi \circ f(H) \leq \alpha/k_{\text{rt}} \}| = |\{ H \in \mathcal{H} \mid \psi \circ f(H) \leq \psi(\sigma_{\text{rt}}) \}| = |\{ H \in \mathcal{H} \mid f(H) \geq \sigma_{\text{rt}} \}|$. ∎

In the following, we present four variants to efficiently find this (rounded) *root frequency* σ_{rt} and enumerate all testable subgraphs (their pseudocodes are in Supplementary Notes). Note that every single method gives exactly the same root frequency and testable subgraphs, resulting in the same significant subgraphs. Our search procedures can be combined with any of the many algorithms for frequent subgraph mining (an FSM algorithm for short), e.g., with AGM [12], gSpan [34], Mofa [7], or Gaston [19], as long as they report actual frequencies of detected frequent subgraphs.

An important property of our search algorithms is that they require a significance level α as an input but do not require the frequency threshold to be prespecified, which is attractive as it is often difficult to find an appropriate frequency threshold for a particular problem in practice.

One-pass search. The first method is to apply an FSM algorithm only once to get the full spectrum of subgraphs. Since the root frequency should satisfy $\psi(\sigma_{\text{rt}}) < \alpha$, we can compute the minimum possible frequency $\sigma_{\min}$ satisfying $\psi(\sigma_{\min}) < \alpha$ from n and n' in advance. Then we run an FSM algorithm with this frequency $\sigma_{\min}$. The mining process might be expensive since this $\sigma_{\min}$ is usually small, resulting in an exponentially large number of frequent subgraphs that may include many untestable subgraphs. But once we finish mining and obtain the actual frequency $f(H)$ for all detected frequent subgraphs H, we can easily obtain the root frequency, for example, by sorting the subgraphs according to their frequencies and checking them one by one, starting with the smallest frequency.

Decremental search (LAMP search). The second approach is to decrease the frequency from the maximum possible value until reaching the root frequency, proposed in LAMP by Terada et al. [27] to find testable itemsets. We start from the maximum possible frequency $\sigma_{\max} = n$ and repeatedly run an FSM algorithm while decreasing the threshold σ one by one as long as the condition $|\{ H \in \mathcal{H} \mid f(H) \geq \sigma \}| \leq \alpha/\psi(\sigma)$ is satisfied. Otherwise if we have $|\{ H \in \mathcal{H} \mid f(H) \geq \sigma \}| > \alpha/\psi(\sigma)$ at some frequency σ, the root $\sigma_{\text{rt}} = \sigma + 1$. This search is expected to be more efficient then the above one-pass search since mining with high frequency is usually much cheaper than that with low frequency and we do not need to run the FSM algorithm with a frequency threshold lower than $\sigma_{\text{rt}} - 1$.

Incremental search. Instead of decreasing the frequency, here we newly propose the opposite strategy, that is, increasing the frequency one by one. We use an additional trick, *early termination* of an FSM algorithm for frequencies $\sigma < \sigma_{\text{rt}}$. For such a frequency σ, we know in advance that the number of admissible subgraphs at this frequency is at most $\alpha/\psi(\sigma)$ — if it is larger, σ cannot be the root frequency. Thus during the process of subgraph mining, we are able to terminate it as soon as the number of subgraphs exceeds this value. The whole process is as follows: we start from the minimum possible frequency $\sigma_{\min}$ and repeatedly apply an FSM algorithm while increasing the threshold σ one by one, as long as the search process terminates early. Otherwise if mining is finished at some frequency, this frequency is the root. This approach is also expected to work efficiently, as the number of admissible subgraphs is quite small if the frequency σ is small. Therefore we can quickly increase the frequency and, moreover, we have to finish the full mining process only once (i.e., without early termination), for the frequency σ_{rt}. Thus the complexity is the same as an FSM algorithm itself.

In parallel to our work, a sped up version of LAMP was published by Minato et al. [17] for significant itemset mining, which also uses incremental search. Unlike our approach, in which pattern mining and incremental search can be combined in an arbitrary, modular fashion, they change the mining process itself to prune untestable hypotheses as early as possible.

Bisection search (LEAP search). Since our task can be viewed as a root-finding problem, we can apply the well-known *bisection method* as our fourth approach. This strategy is used in LEAP by Yan et al. [33] to obtain top-k subgraphs in terms of a user-specified objective function in which a statistical test can be used, yet without multiple testing correction. Thereby we exploit only its search strategy to find the root frequency. It repeatedly bisects an interval of possible frequencies and selects a subinterval in which the root frequency lies. First we set the interval $[a, b]$ from the minimum possible frequency $a = \sigma_{\min}$ to the maximum possible frequency $b = \sigma_{\max} = n$. We run an FSM algorithm with the frequency $\sigma = (a+b)/2$ and set $a = \sigma$ if the mining process terminates earlier, and $b = \sigma$ otherwise, and repeat the process until $a - b = 1$. We can also use the early termination with the number of admissible subgraphs proposed in the incremental search above, which enables us to gain more efficiency and to determine whether the current frequency σ is larger than the root. This method could potentially reduce the number of frequencies to be examined.

2.5 Effective number of tests. Many subgraphs are expected to be highly correlated with each other due to combinatorial constraints on graphs such as subgraph-supergraph relationships [29]. To exploit the dependence between subgraphs and further increase the power, we use the *effective number of tests*. In the Šidák correction [24], the significance level α' for each test is given as $1 - (1-\alpha)^{1/m}$ for m *independent* tests. This means that if we have m tests and some of them are correlated, only $m_{\text{eff}} < m$ tests, defined by

$$m_{\text{eff}} := \log(1-\alpha)/\log(1-\alpha'),$$

are *effective* for controlling the FWER [18], hence m_{eff} can be used as a reduced Bonferroni factor. This m_{eff} is called the effective number of tests and estimation methods, such as the Cheverud-Nyholt estimate [20], have been proposed in particular in statistical genetics.

We directly estimate the significance level α' for each test by random permutations of class labels, which gives the null distribution of independent subgraphs. Although this method gives the optimal estimation of m_{eff} in theory, its drawback is the high computational cost $O(mh)$ ($m = |\mathcal{H}|$ in our case), where h is the number of iterations. Here we overcome this drawback by considering only testable subgraphs. Since we can ignore untestable hypotheses (subgraphs) for controlling the FWER, we apply the above permutation-based estimation to only testable subgraphs. The complexity reduces to $O(|\tau(\mathcal{H})|h)$, which is expected to be much cheaper than $O(|\mathcal{H}|h)$ if we can eliminate many untestable subgraphs. We set the number of permutations to be 1,000 throughout the paper, which is recommended for $\alpha = 0.05$ [8] and commonly used [18].

3 Related Work

The statistical significance of subgraph occurrence in networks has been investigated before, first in specific application domains, such as social networks [30] and gene regulatory networks [22], and the formulation was later extended to general graphs [1, 10, 16, 21, 33]. In all of these studies, however, the significance is defined using a random database, that is, the p-value of a subgraph is the probability of its frequency being larger than the user-specified threshold under a certain distribution of graphs (or labels on graphs) and, to the best of our knowledge, no study directly detects subgraphs that are significantly associated with class memberships of graphs. Moreover, our method overcomes the following three drawbacks of previous approaches: (1) their p-values depend on the frequency threshold, which is often difficult to determine in practice, while our method requires only the significance level α; (2) their p-value computation requires a distribution of graphs, which is not trivial to estimate, while our method does not need to consider such a distribution and can still calculate the exact p-values; (3) to the best of our knowledge, all previous studies did not consider the multiple testing problem, which leads to many false positives, while our method strictly controls the FWER.

Subgraph detection has also been intensively studied in *graph classification*, where subgraphs are used as *features* to describe graphs. This means that each graph G is represented as a feature vector in which each feature corresponds to another graph H and the value is one if $H \sqsubseteq G$ and zero otherwise. The general objective is to find informative subgraphs for discrimination to improve the accuracy of the subsequent classification, which can also be viewed as a supervised feature selection problem. A number of methods have been proposed, for example, gBoost [14] and a Lasso-based method [28]. Note that, however, in classification we do not need to control the FWER (false positives) as long as we can build a good classifier, while our ultimate goal in this paper is to detect key substructures for a better understanding of the target phenomenon and the FWER must be controlled to avoid false positives for further investigation in application domains.

Multiple (hypothesis) testing is a classical problem in statistics, with Bonferroni correction [6] being the most prominent correction technique. Since Bonferroni correction is known to be too conservative, other correction methods have been proposed, for instance, Holm's correction [11]. However, these methods also require the exact number of tests (subgraphs) for correction, which is highly expensive to compute in graph mining. Another approach is to use random subsampling to estimate the correction factor [9], but this also needs high computational cost if the number of tests is massive. Controlling the false discovery rate (FDR) [2] is recently becoming popular as an alternative to the FWER, which leads to more power in multiple testing. However, it also requires the exact number of tests and hence is also extremely expensive to compute.

Table 1: Statistics of datasets, where $|L(V)|$ and $|L(E)|$ denote the number of node and edge labels.

Dataset	Size	#positive	avg.$\|V\|$	avg.$\|E\|$	max$\|V\|$	max$\|E\|$	min$\|V\|$	min$\|E\|$	avg.deg	$\|L(V)\|$	$\|L(E)\|$
PTC (MR)	584	181	31.96	32.71	181	181	2	1	2.01	7	4
MUTAG	188	125	17.93	39.59	28	66	10	20	4.38	7	11
ENZYMES	600	300	32.63	62.14	126	149	2	1	3.86	3	1
D&D	1178	691	284.32	715.66	5748	14267	30	63	4.98	82	1
NCI1	4208	2104	60.12	62.72	462	468	4	3	2.08	8	4
NCI41	27965	1623	47.97	50.15	462	468	3	2	2.09	8	4
NCI167	80581	9615	39.70	41.05	482	478	2	1	2.06	8	4
NCI220	900	290	46.87	48.52	239	255	2	1	2.05	7	3

4 Experiments

We examined our methods on real-world graph data and compared them to the brute-force approach (BF for short) and two state-of-the-art approaches (LAMP and LEAP) in our framework. BF naïvely enumerates subgraphs occurring more than once to set the Bonferroni correction factor. Notice that, with respect to assessing the quality of results, that is, the number of significant subgraphs, BF can be our only comparison partner, since there exists no method for finding significant subgraphs while controlling the FWER by multiple testing correction. On the efficiency side, we compare BF and our four search strategies, in which two of them (decremental LAMP search and bisection LEAP search) are the state-of-the-art.

As an FSM algorithm, we employ Gaston [19] since it is reported to be one of the fastest FSM algorithms [32]. We integrated our search strategies into Gaston, which are written in C++ and compiled with gcc 4.6.3. The significance level α was always set to 0.05 and a two-tailed test was used. We repeated $1,000$ permutations to obtain the effective number of tests. We used Ubuntu version 12.04.3 with a single 2.6 GHz AMD Opteron CPU and 512 GB of memory. All experiments were performed in R 3.0.1.

We used eight real-world graph datasets: PTC(MR), MUTAG, ENZYMES, D&D, and four NCI datasets, where ENZYMES and D&D are proteins and others are chemical compounds. Statistics for these datasets are summarized in Table 1. These datasets have been frequently used as benchmarks in previous studies [15, 23, 36]. They are labeled undirected graphs: Graph nodes are labeled in all datasets and edges are also labeled except for ENZYMES and D&D.

Effectiveness. First we compare the Bonferroni correction factors and our reduced correction factors, that is, the number of testable subgraphs $|\tau(\mathcal{H})|$ and that of effective subgraphs m_{eff}, and evaluate the improvement of our method in terms of the power for detecting significant subgraphs and the empirical FWERs obtained from 10,000 permutations of class labels. In each dataset, we varied the upper bound of the subgraph size from 4 to 16 and without size bound ("Limitless").

The resulting correction factors are plotted in Figure 1 and the numbers of significant subgraphs we detected and the empirical FWERs are shown in Figures 2 and 3, respectively. There are some missing values in the plots, in particular results of the Bonferroni factor (red cross marks), due to a huge amount of computation time. These plots clearly show that, in all datasets, our correction factor is much smaller than the Bonferroni factor and the difference between them becomes larger as the maximum subgraph size increases. In particular in PTC(MR) and D&D, our factors (blue circles and green triangles) become stable in large maximum subgraph sizes while the Bonferroni factors increase exponentially. The reason might be that most of large subgraphs become untestable because they tend to have small frequencies in general. Moreover, we can confirm that in all datasets correction factors are further reduced using the effective number of tests. This is because many subgraphs are highly correlated with each other due to combinatorial constraints of graphs [29].

In terms of the number of significant subgraphs (Figure 2), we can find more subgraphs because of the reduced correction factor across our datasets. On several datasets the effect is dramatic, such as MUTAG, ENZYMES or D&D, where our methods find thousands of significant subgraphs missed by the standard Bonferroni correction (see Supplementary Notes for examples). In PTC(MR), one cannot find any significant subgraphs by the Bonferroni correction when the maximum subgraph size is larger than 6, but one can detect 2 to 4 (testable) or 3 to 8 (effective) significant subgraphs using our factors. Moreover, the number of significant subgraphs in the Bonferroni factor rapidly decreases in the D&D dataset as the maximum subgraph size increases, while numbers are stable in our methods even if the maximum subgraph size is unlimited. Since it is of-

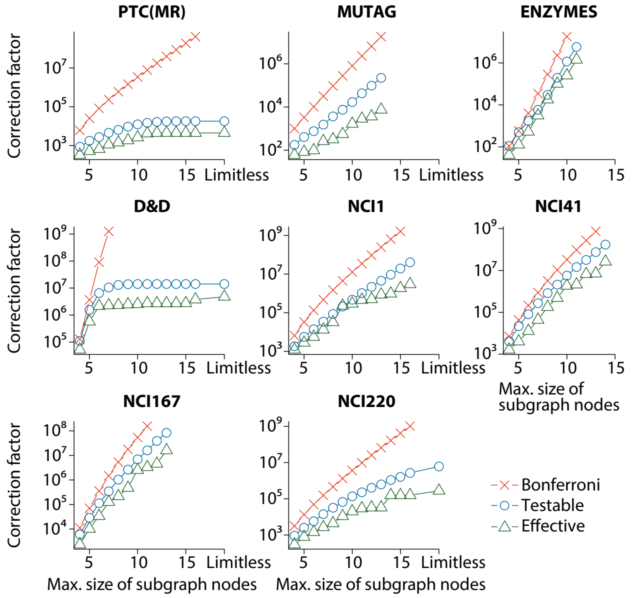

Figure 1: The Bonferroni correction factor $|\mathcal{H}|$ (red cross marks), the number of testable subgraphs $|\tau(\mathcal{H})|$ (blue circles), and the effective number of tests m_{eff} (green triangles). Note that the y-axis has a logarithmic scale.

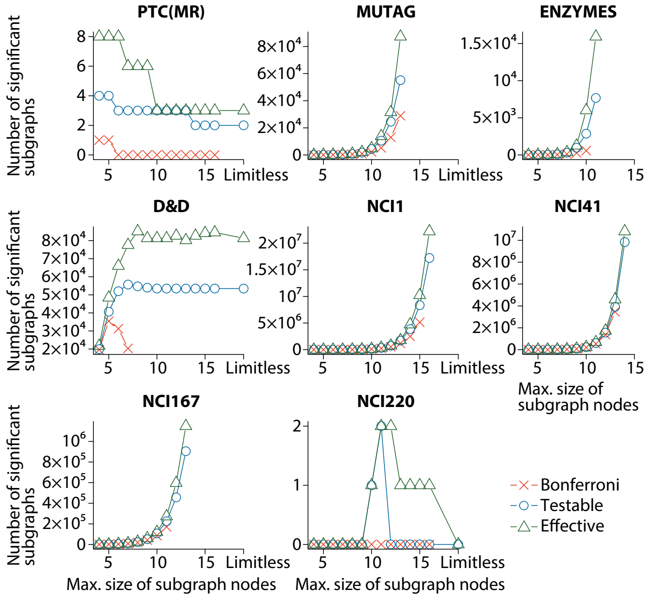

Figure 2: The number of significant subgraphs discovered with the Bonferroni correction (red cross marks) and our method with the testability criterion (blue circles) and the effective number of tests (green triangles).

ten difficult to appropriately upper bound the subgraph size beforehand in practice, this is another advantage in practical applications. In NCI220 the number of significant subgraphs exhibits an interesting behavior, that is, significant subgraphs are detected only if the maximum subgraph size is 10 or 11 (testable) and from 10 to 16 (effective). The reason is that the size of these significant subgraphs is 10 or 11 and we cannot detect them if the maximum subgraph size is smaller than that. Furthermore, these subgraphs are no longer significant if the maximum subgraph size becomes larger due to the increase of the correction factor.

We can also confirm the higher statistical power from the empirical FWERs (Figure 3). Note that the FWER should be $\alpha = 0.05$ in the best case, and the correction factor is too large if the FWER is smaller than α. By reducing the correction factor with the testability criterion and the effective number of tests, the FWERs get closer to $\alpha = 0.05$.

Efficiency. Next we analyze the efficiency of our strategies compared to BF and the state-of-the-art (LAMP and LEAP). The resulting running times are plotted in Figure 4 and are summarized in Table 2 as RMSD (root mean square deviation) to the best (fastest) running time on each dataset and for each maximum subgraph size. In addition, we also plot the running time of computing the effective number of testable subgraphs by using 1,000 permutations in Figure 4.

The results clearly show that all four searches using the testability criterion are faster than BF on average. This means that reducing the number of subgraph candidates using the testability of them contributes not only to the effectiveness in terms of finding significant subgraphs but also the efficiency of the whole process. Furthermore, our new incremental search is one to two orders of magnitude faster than the other state-of-the-art search strategies (decremental LAMP and bisection LEAP) and more than two orders of magnitude faster than the one-pass search and BF on average. In contrast, the decremental LAMP search is slow, with its speed being similar to the one-pass search on average, and it is often even slower than BF. The reason is that in practice the root frequency σ_{rt} is relatively small (around 20, see Table S2 in Supplementary Notes) and hence the decremental search needs to repeat an FSM algorithm many times until reaching this frequency. This is also the reason for the efficiency of the incremental search as it can quickly find the root frequency. Although the bisection LEAP search is faster than the decremental and the one-pass search on average, it is slower than the incremental search. The reason is the same as in the discussion above, that is, the root frequency is usually small and it tends to repeat subgraph mining with high frequencies.

The running time for computing the effective num-

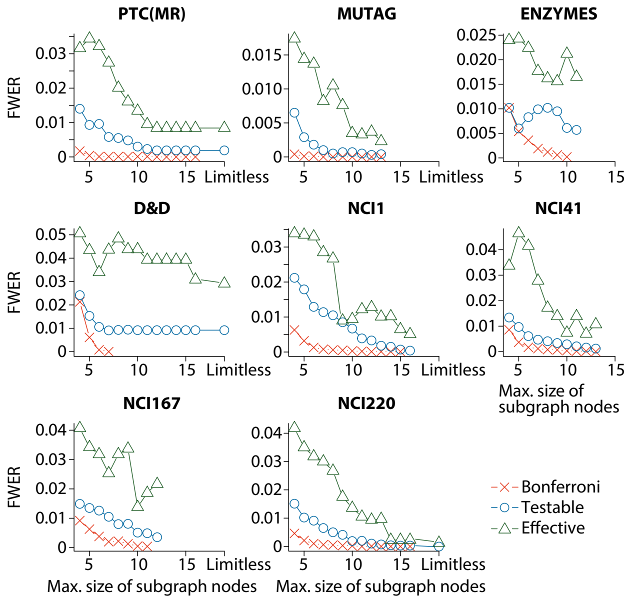

Figure 3: The empirical FWERs with 10,000 permutations of class labels with the Bonferroni correction (red cross marks), our method with the testability criterion (blue circles) and the effective number of tests (green triangles).

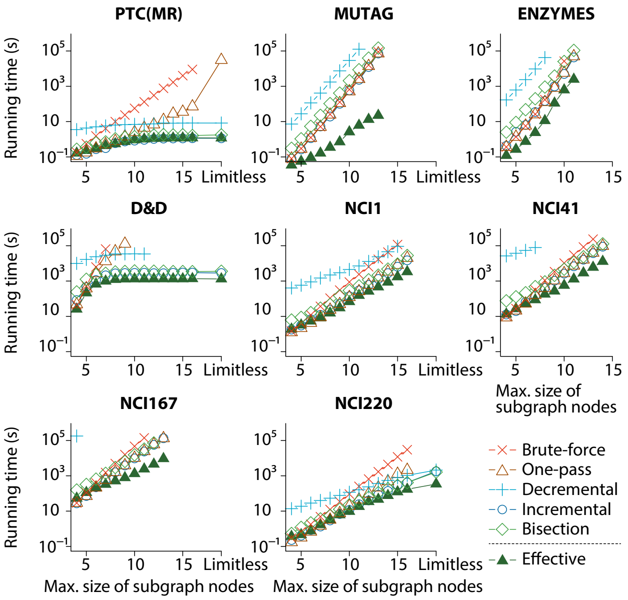

Figure 4: Running time (seconds). Note that the y-axis is in logarithmic scale.

ber of tests is faster than the above-mentioned search of testable subgraphs in most cases. This means that the testability criterion also contributes to the efficiency of computing the effective number of tests and makes it feasible within a reasonable time.

5 Conclusion

In this paper, we have presented a solution for finding subgraphs that are statistically significantly enriched in one class of graphs but not another. The difficulty of the problem stems from the two facts that (1) one has to consider an enormous search space of candidate subgraphs and that (2) one has to correct the significance level for multiple testing to control the FWER, as one tests a large number of candidate subgraphs simultaneously. The first problem leads to enormous computational runtime problems, the second one to a loss in the statistical power to detect significant subgraphs.

We have shown that the problem can be exactly and efficiently solved by considering only *testable* subgraphs, which include all significant subgraphs and dramatically reduce the number of tests performed, thereby leading to a gain in statistical power. Moreover, we can further increase the power using the effective number of tests, which reduces the correction factor according to the dependence between subgraphs. We have presented several search strategies that use frequent subgraph mining algorithms to efficiently retrieve the set of testable subgraphs. Experimental results show that our method finds significant subgraphs with higher speed and higher statistical power than any state-of-the-art approach. This result promises to open the door to many interesting applications in chemoinformatics, structural biology and personalized medicine.

We also believe that our approach lays the foundation for follow-up studies in several important directions: developing and integrating other approaches which exploit the dependence between tests [35], considering other types of structured data such as strings, and summarizing the solution set of significant subgraphs, which sometimes grows extremely large.

References

[1] Arora, A., Sachan, M., Bhattacharya, A.: Mining statistically significant connected subgraphs in vertex labeled graphs. In: SIGMOD. pp. 1003–1014 (2014)

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

[3] Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic Acids Research 28, 235–242 (2000), www.rcsb.org

[4] Bland, M.: An Introduction to Medical Statistics. Oxford University Press (2000)

[5] Bolton, E.E., Wang, Y., Thiessen, P.A., Bryant, S.H.: PubChem: Integrated platform of small molecules and biological activities. Annu. rep. comp. chem. 4, 217–241 (2008), pubchem.ncbi.nlm.nih.gov

Table 2: RMSD (root mean square deviation) of running time (seconds) in Figure 4 to the best (fastest) running time on all datasets and maximum subgraph sizes. This measure rewards methods that are always close to the fastest running time on each dataset and each maximum subgraph size.

Brute-force (BF)	One-pass	Decremental (LAMP)	Incremental	Bisection (LEAP)
6.994×10^4	2.635×10^4	2.410×10^4	$\mathbf{1.230 \times 10^2}$	9.554×10^3

[6] Bonferroni, C.E.: Teoria statistica delle classi e calcolo delle probabilità. Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze 8, 3–62 (1936)

[7] Borgelt, C., Berthold, M.R.: Mining molecular fragments: Finding relevant substructures of molecules. In: ICDM. pp. 51–58 (2002)

[8] Churchill, G.A., Doerge, R.W.: Empirical threshold values for quantitative trait mapping. Genetics 138(3), 963–971 (1994)

[9] Dudoit, S., Shaffer, J.P., Boldrick, J.C.: Multiple hypothesis testing in microarray experiments. Statistical Science pp. 71–103 (2003)

[10] He, H., Singh, A.K.: GraphRank: Statistical modeling and mining of significant subgraphs in the feature space. In: ICDM. pp. 885–890 (2006)

[11] Holm, S.: A simple sequentially rejective multiple test procedure. Scandinavian J. Statistics pp. 65–70 (1979)

[12] Inokuchi, A., Washio, T., Motoda, H.: An Apriori-based algorithm for mining frequent substructures from graph data. In: PKDD, pp. 13–23. LNCS 1910 (2000)

[13] Kanehisa, M., Goto, S.: KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Research 28, 27–30 (2000), www.kegg.jp

[14] Kudo, T., Maeda, E., Matsumoto, Y.: An application of boosting to graph classification. In: NIPS. pp. 729–736 (2004)

[15] Li, G., Semerci, M., Yener, B., Zaki, M.J.: Effective graph classification based on topological and label attributes. Statistical Analysis and Data Mining 5(4), 265–283 (2012)

[16] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: Simple building blocks of complex networks. Science 298(5594), 824–827 (2002)

[17] Minato, S., Uno, T., Tsuda, K., Terada, A., Sese, J.: A fast method of statistical assessment for combinatorial hypotheses based on frequent itemset enumeration. In: ECML/PKDD, pp. 422–436. LNCS 8725 (2014)

[18] Moskvina, V., Schmidt, K.M.: On multiple-testing correction in genome-wide association studies. Genetic epidemiology 32(6), 567–573 (2008)

[19] Nijssen, S., Kok, J.N.: A quickstart in frequent structure mining can make a difference. In: KDD. pp. 647–652 (2004)

[20] Nyholt, D.R.: A simple correction for multiple testing for single-nucleotide polymorphisms in linkage disequilibrium with each other. The American Journal of Human Genetics 74(4), 765–769 (2004)

[21] Ranu, S., Singh, A.K.: GraphSig: A scalable approach to mining significant subgraphs in large graph databases. In: ICDE. pp. 844–855 (2009)

[22] Shen-Orr, S., Milo, R., Mangan, S., Alon, U.: Network motifs in the transcriptional regulation network of *Escherichia coli*. Nature Genetics 31(1), 64–68 (2002)

[23] Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-Lehman graph kernels. JMLR 12, 2359–2561 (2011)

[24] Šidák, Z.: Rectangular confidence regions for the means of multivariate normal distributions. J. American Statistical Association 62(318), 626–633 (1967)

[25] Takigawa, I., Mamitsuka, H.: Graph mining: Procedure, application to drug discovery and recent advances. Drug Discovery Today 18(1–2), 50–57 (2013)

[26] Tarone, R.E.: A modified Bonferroni method for discrete data. Biometrics 46(2), 515–522 (1990)

[27] Terada, A., Okada-Hatakeyama, M., Tsuda, K., Sese, J.: Statistical significance of combinatorial regulations. PNAS 110(32), 12996–13001 (2013)

[28] Tsuda, K.: Entire regularization paths for graph data. In: ICML. pp. 919–926 (2007)

[29] Ugander, J., Backstrom, L., Kleinberg, J.: Subgraph frequencies: Mapping the empirical and extremal geography of large graph collections. In: WWW. pp. 1307–1318 (2013)

[30] Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press (1994)

[31] Weill, N., Rognan, D.: Development and validation of a novel protein-ligand fingerprint to mine chemogenomic space: application to G protein-coupled receptors and their ligands. Journal of Chemical Information and Modeling 49(4), 1049–1062 (2009)

[32] Wörlein, M., Meinl, T., Fischer, I., Philippsen, M.: A quantitative comparison of the subgraph miners MoFa, gSpan, FFSM, and Gaston. In: PKDD, pp. 392–403. LNCS 3721 (2005)

[33] Yan, X., Cheng, H., Han, J., Yu, P.S.: Mining significant graph patterns by leap search. In: SIGMOD. pp. 433–444 (2008)

[34] Yan, X., Han, J.: gSpan: Graph-based substructure pattern mining. In: ICDM. pp. 721–724 (2002)

[35] Zhang, X., Pan, F., Wang, W., Nobel, A.: Mining non-redundant high order correlations in binary data. Proc. VLDB 1(1), 1178–1188 (2008)

[36] Zhao, Y., Kong, X., Yu, P.S.: Positive and unlabeled learning for graph classification. In: ICDM. pp. 962–971 (2011)