Technical Report

A list of FSM Algorithms and available Implementations in Centralized Graph Transaction Databases

Rihab AYED, Mohand-Saïd HACID, Rafiqul HAQUE and Abderrazek JEMAI

Author's Address

Université Claude Bernard, Lyon 1 (UCBL)
Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS)
43, boulevard du 11 Novembre 1918
69622 Villeurbanne cedex
France
Email: rihab.ayed@liris.cnrs.fr

Abstract

In this report, we list the algorithms proposed in the literature of Frequent Subgraph Mining (FSM) in Centralized Graph Transaction Databases. We categorize FSM algorithms in four categories of their search and matching strategy which is affecting their returned output (*i.e.,* frequent subgraphs). For each category, we list the algorithms. We filter algorithms usable in specific cases of graphs (*i.e.,* not applicable to general graphs cases). We enumerate the available software of the algorithms. This report could be helpful for having a list of FSM algorithms in Centralized Graph Transaction Databases.

Keywords

Graph Mining, Graph Transaction Databases, Centralized Environment, Frequent Subgraph Mining(FSM), FSM Algorithms

Table of Contents

1 Introduction

Graph mining is one of the most important approaches in data mining that transforms graph data into knowledge [3]. Frequent Subgraph Mining (FSM) is a subcategory of Graph Mining [34]. The objective of traditional FSM [43] is to extract subgraphs, in a given dataset, whose occurrence counts (*aka* frequency) are above a specified threshold (Minimum Support Threshold). The extracted subgraphs, called Frequent Subgraphs, are (directly) useful for analysis in areas like, biology, co-citations, chemistry, semantic web, social science and finance trade networks [48,80]. They could also be used for other relevant purposes such as classification, clustering, graph indexing and similarity search [51,80].

In the scope of the CAIR[1] project, we are investigating efficient and effective approaches for aggregated search [57] over distributed repositories. One of the building blocks of our approach is graph indexing. The index data can be provided by resorting to an FSM algorithm [99].

Given that there are many algorithms proposed for FSM in the literature, the first task was to categorize the algorithms according to some relevant and desired features.

A lot of studies [4,23,23,28,31,45,48,53,56,66,74,77,79,91] in the literature enumerated some of the existing algorithms, optionally defined their categories and/or performed experimental comparison of them. To our knowledge, there is no study that list all FSM algorithms in Centralized Graph Transaction databases. In this report, we try to list the maximum number of FSM algorithms in the literature. Please contact authors if an algorithm is not mentioned in the list.

This report is organized as follows: Section 2 present some classifications of FSM algorithms. Section 3.4, contains the list of FSM algorithms classified by four categories of algorithms. Section 4 proposes a selection of available software usable for general graphs for the four categories. Algorithms known to be not efficient are mentioned also. We conclude in Section 5 and present some perspectives.

2 Preliminaries

FSM algorithms can be classified according to three different aspects [?,53]: (i) database setting which depends on the applications (*e.g.,* Chemical graphs, Social Networks), (ii) nature of input and output graphs and (iii) the strategy of subgraph search and matching. In what follows, we briefly explain these aspects.

2.1 Database Setting

There are two distinct problem formulations for frequent subgraph mining in graph datasets: (i) graph-transaction setting and (ii) single-graph setting.

2.1.1 Graph-Transaction Setting In this case, the input is a collection of moderate sized graphs (transactions), and a subgraph is considered frequent if it appears in a large number of graphs. A subgraph occurrence is counted only once per transaction, independently of the possible multiple occurrences in the same transaction [43]. Graph Transaction mining is applied in biochemical structure analysis, program control flow analysis, XML structure analysis, image processing and analysis [3,48].

2.1.2 Single-Graph Setting This setting involves mining frequent subgraphs in different regions of one large sized graph. The frequency of a subgraph is based on the number of its occurrences (*i.e.,* embeddings) in the large graph. Special support metrics are used, by considering, for example, the overlapping of two subgraphs [55]. Single Graph mining is dedicated to applications such as social networks, citation graphs, or protein-protein interactions in bioinformatics [25].

In this paper, we are interested in algorithms that mine a collection of graphs instead of a single large garph.

2.2 Nature of Input and Output Graphs

In centralized graph transaction mining, the input graphs which are used in most of the FSM algorithms are assumed to be *labeled (vertices and edges) simple*[1] *connected undirected graphs* and the output are *connected subgraphs*. There are algorithms developed for specific graphs (*e.g.,* complex graphs [65], unconnected subgraphs [81], vertex labeled graphs [103], see Table 1) rather than the general ones.

Table 1: FSM Algorithms with specific graphs

Input Cases	**Algorithms**
Complex graphs	MgVEAM [65]
Directed graphs	mSpan [60]
Directed Acyclic graphs	DIGDAG [83]
Unlabeled graphs	The smoothing-clustering framework [12]
Vertex-labeled graphs	Cocain [103], TSMiner [49]
Relational graphs	CODENSE [39], CLOSECUT & SPLAT [100], Fp-GraphMiner [87]
Geometric graphs	gFSG [55], MaxGeo [9], FREQGEO [76]
Uncertain graphs	Monkey [105], RAM [104], MUSE [108]
Output Cases	**Algorithms**
Cliques and quasi-cliques from dense graphs	CLAN [89], Cocain [103]
Unconnected subgraphs	UGM [81]

2.3 Subgraph Search and Matching Strategy

FSM algorithms can be classified according to search strategy : *complete* and *incomplete* (or heuristic) search. Also, they can be classified according to the type of isomorphism test (matching) performed between the mined subgraphs : *exact* and *inexact* matching. We explain these categories in what follows.

2.3.1 Complete Search The complete search[2] algorithms perform a complete mining *i.e.,* it guarantees to find all frequent subgraphs from the input data, above a minimum frequency threshold [44,54].

---

[1] A simple graph is "an un-weighted and un-directed graph with no loops and no multiple links between any two distinct nodes" [32]

[2] Complete search is also called "exact search" [48,80], we will use, in this paper, only the designation "complete search"

a) Exact Matching: It consists in finding all possible frequent subgraphs as they appear in the input data [44, 54]. The complete search must return a frequent subgraph (*e.g.,* subgraph (1) shown in Figure 1) and all of its possible subgraphs that are necessarily frequent as well (*e.g.,* subgraphs (2), (4), (5), (6) and (7) shown in Figure 1).

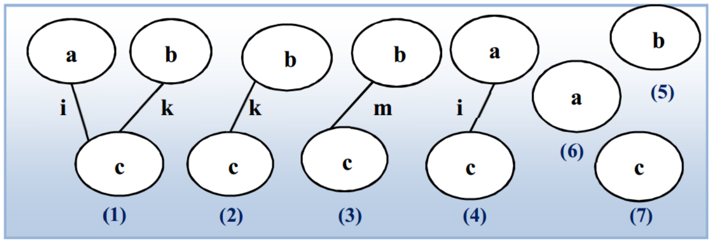

Fig. 1: Example of all Frequent Subgraphs (Exact Matching)

b) Approximate Matching: It consists in finding all frequent subgraphs, with an assumption that subgraphs having the same structure and different labels, will all be returned as the same subgraph [60]. This is considered as a complete search because all possible frequent subgraphs could be verified in the output set with the abstraction of labels (edges or vertices). Figure 2 illustrates the approximate matching where graphs with different edge labels are considered the same. For example, the subgraphs (2) and (3) in Figure 1 could be represented with the approximate matching by one subgraph (2') in Figure 2.

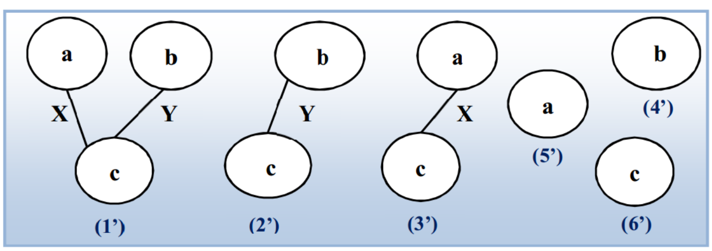

Fig. 2: Example of all Frequent Patterns (Approximate Matching)

2.3.2 Incomplete or Heuristic Search

The incomplete and heuristic search algorithms discover a set of frequent subgraphs whose cardinality is greater or lower than the one returned by the complete search. This category of FSM search is used to : (i) reduce the set of frequent subgraphs (use of exact [98] or approximate matching [17]), or (ii) add more frequent subgraphs than the complete search in order to consider the innacuracy or uncertainty of the input data (use of approximate matching) [109].

c) Exact Matching: It consists in returning a subset of frequent subgraphs [90] by setting a supplementary calculable parameter (*e.g.,* maximum size of frequent subgraphs, closed subgraphs, maximal subgraphs, maximum support threshold) [7, 41, 98], besides the minimum support threshold.

Figure 3 shows an example returning a subset of frequent subgraphs (see all frequent subgraphs, Figure 1) where the set parameter is the maximum size of frequent subgraphs (set to 2 edges).

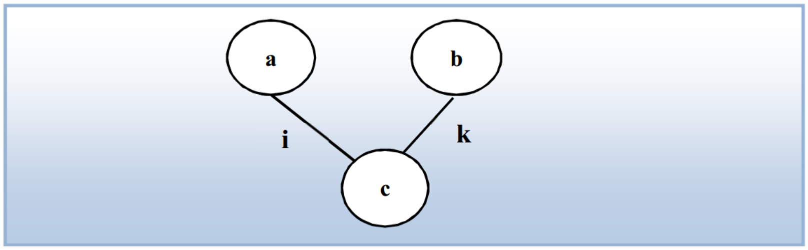

Fig. 3: Example of a subset of Frequent Subgraphs (Exact Matching)

d) Approximate Matching: It consists in either (i) reducing the output by returning a set of representative frequent patterns or (ii) enriching the frequent subgraphs output by considering the inaccuracy or uncertainty of data [47, 109]. For the first case, a representative frequent pattern is a frequent subgraph similar to a set of other frequent subgraphs. In other words, frequent subgraphs that have some differences regarding edges, vertices and labels are represented by one pattern in the output [5]. For the second case, it consists in adding infrequent subgraphs that are similar to frequent subgraphs with respect to the structure or labels [1]. For the four subcategories a, b, c and d, there

Table 2: A list of Complete Search & Exact Matching FSM Centralized Algorithms

Algorithm	**Author**	**Algorithm**	**Author**
WARMR	[20]	ADI-Mine & GraphMiner	[90, 95]
AGM	[43]	TSMiner	[49]
FARMER	[71]	FSP	[35]
MOLFEA	[52]	DMTL	[6]
AcGM	[46]	gRed	[28]
B-AGM	[44, 45]	FSMA	[92]
FSG	[54]	FREQGEO	[76]
MoFa/MoSS	[11]	SyGMA	[21]
DPMine	[33]	CGM & UGM	[81]
gSpan	[96, 97]	gdFil	[27]
Topology	[38]	grCAM	[29]
FFSM	[40]	ADI-Minebio	[19]
DSPM	[15]	Fp-GraphMiner	[87]
AGM-H	[68]	FSMA	[30]
GASTON	[72, 73]	LC-Mine framework : FGMAC & AC-miner	[24]
IDFP-tree	[67]		

are respectively 31, 1, 22 and 15 Algorithms. In the following section, we provide the list of a these FSM algorithms.

3 A list of FSM Algorithms

For each of the four selected categories, we list the FSM algorithms of the literature. The list of complete search & exact or inexact matching categories is exhaustive. The list of incomplete search & exact or inexact matching categories contains the maximum number of found algorithms, some algorithms may be missing.

Table 3: A list of Complete Search & Inexact Matching FSM Centralized Algorithms

Algorithm	**Author**
mSpan	[60]

3.1 Complete Search & Exact Matching

We identified thirty-one algorithms in the literature designed to extract all possible frequent subgraphs above a minimum support threshold (see Table 2).

3.2 Complete Search & Inexact Matching

We identified one algorithm in the literature designed to extract all possible frequent subgraphs above a minimum support threshold with an abstraction of edges/vertices (see Table 3).

3.3 Incomplete Search & Exact Matching

To our knowledge, there are twenty-two Algorithms performing an Incomplete Search and Exact Matching (see table 4).

Table 4: A list of Incomplete Search & Exact Matching FSM Centralized Algorithms

Algorithm	**Author**	**Algorithm**	**Author**
SUBDUE	[37]	CloseGraph	[98]
LCGMiner	[93]	SPIN	[41]
CLOSECUT & SPLAT	[100]	CLAN	[89]
Cocain	[103]	MARGIN	[86]
DIGDAG	[84]	gFSG	[55]
MaxGeo	[9]	LEAP	[95]
RP-FP & RP-GD	[61] [58]	JPMiner	[62]
FOGGER	[102]	MUSK	[7]
Al Hasan algorithm[3]	[8]	gdClosed	[27]
TGP	[59]	Takigawa algorithm[4]	[82]
TD-MG	[14]	FS^3	[80]

[3] Output Space Sampling
[4] It proposes to return δ-tolerance closed frequent subgraphs

3.4 Incomplete Search & Inexact Matching

There are fifteen algorithms performing Incomplete Search and Inexact Matching (see table 5).

Table 5: A list of Incomplete Search & Inexact Matching FSM Centralized Algorithms

Algorithm	**Author**	**Algorithm**	**Author**
SUBDUE	[37]	MgVEAM	[65]
ORIGAMI	[5]	Monkey	[105]
RAM	[104]	Chen algorithm[5]	[12]
ISG	[85]	RING	[106]
APGM	[47]	MUSE	[108]
VEAMwP	[2]	VEAM	[1]
Summarization Framework	[63]	TRS	[22]
		MUG	[18]

In the following section, we propose a usability filtering of these algorithms in order to help in the choice of the most useful FSM algorithms and their respective implementations.

4 Selection of available FSM Algorithms' Implementations

The list of FSM algorithms is large and hence selecting suitable candidates is a non-trivial task. The choice is generally done in the literature implicitly according to the usedness of an algorithm (*e.g.,* the recent algorithm LC-Mine [24] (2014) is compared with the known gSpan (2002) and FSG (2001). But there is no experimental justification of the non comparison with more recent algorithms). In order to justify the selection process, we defined a set of criteria : *performance reported in the literature*, *miscellaneous weaknesses* and *availability of software*. We propose a filtering for each of the four categories of algorithms (see Section 2) using the three criteria. When a criterion is ommitted in a category, this means that we did not found algorithms to filter for this criterion.

4.1 Complete Search & Exact Matching

4.1.1 Performance of algorithms We found in comparative studies (reported in literature) that the performance of four algorithms namely WARMR [20], FARMER[6] [71], UGM & CGM[7] [81] and MOLFEA [52] is commonly agreed to be the worst. The algorithm FSMA [30] has a very modest experimentation and has not been compared with any FSM algorithm. Thus, we removed these five algorithms from the list of potential candidates.

4.1.2 Weaknesses We eliminated four algorithms for limitations regarding input graphs (see Table 6). In fact, we intend to compare algorithms that propose general usage. The weakness analysis of the algorithms produced a list of twenty-three remaining algorithms.

[5] The smoothing-clustering framework
[6] WARMR and FARMER were both used for itemsets and complex relations
[7] MoFa is competitive with UGM&CGM. MoFa has a bad performance compared to Gaston, gSpan, FFSM [70, 91]

Table 6: Complete Search & Exact Matching FSM Algorithms with specific uses

Algorithm	**Input Graphs Case**
FREQGEO [76]	Geometric Graphs (2D or 3D)
TSMiner [49]	Graphs with unlabeled edges
SyGMA [21]	The number of labels has to be small
ADI-MineBio [19]	- The input data is relational tables - Dedicated for specific biomedical data

4.1.3 Availability of Software

We tried to collect the codes/software of the remaining twenty-three algorithms from their authors or from available links. Only one-third implementations (7) out of them are public. The others are unavailable (see Table 7) due to : (i) legal constraint (intellectual property right); (ii) their code is lost or (iii) no answer is given by authors.[8]

There are different implementations of the seven remaining algorithms. We list them in Table 8. We had to eliminate AcGM implementation and four implementations of gSpan, FFSM and Gaston due to the technical shortcomings (see Table 9). We could have tried to debug the algorithms but the goal is to list usable as-is implementations.

Table 7: Unavailable Complete Search & Exact Matching FSM algorithms software

Algorithms	**Unavailability**
AGM [43], Topology [38], AGM-H [68], B-AGM [44], ADI-Mine [88], FSP [35], FSMA [92], LC-Mine framework [24], IDFP-tree [67]	No answer from authors
gRed [28], gdFil [27], grCAM [29]	Under intellectual propoerties
DPMine [33], DSPM [15], Fp-GraphMiner [87]	The code is lost

The final list of candidate algorithms in Complete Search & Exact Matching category contains six algorithms with their respective thirteen implementations. An experimental study is proposed for these implementations in another work.[9]

4.2 Complete Search & Exact Matching

There is one algorithm in this category. We inspected its availability in the following section.

4.2.1 Availability of Software

The algorithm mSpan [60] is unavailable due to no answer given from authors.

[8] 1 request and 2 reminders have been sent to authors
[9] see https://liris.cnrs.fr/rihab.ayed/ESFSM.pdf for more details
[10] ParMol framework could be provided by authors [64, 91]
[11] No theoretical work is related to this software

Table 8: Complete Search & Exact Matching FSM Implementations with Technical Drawbacks

Implementation	**Technical Drawbacks**
gSpan ParSeMis	- Quality of Frequent Subgraphs (redundancy) - Error during the execution
gSpan (Kudo, 2004)	- Requiring an additional software (MATLAB)
FFSM Original	- Error with Input Files (No answer from authors about this error)
AcGM Original	- No information about Memory Consumption or Runtime (binary code and no response from authors) - The output is only the DFS code of frequent subgraphs
Gaston ParSeMis	- Error during the execution

4.3 Incomplete Search & Exact Matching

Some algorithms in FSM propose to return a subset of frequent subgraphs in order to reduce the output. We filter in the following these algorithms.

4.3.1 Weaknesses Seven algorithms out of twenty-two are useful for specific cases only. These algorithms propose to process (as an input) or to return (as an output) specific graphs. For example, CLAN [89] proposes to return frequent cliques, this case is specific to the application of cliques not

Table 9: Available Implementations of Complete Search & Exact Matching FSM algorithms

Algorithm	**Available software**	**Last Release**
FSG [54]	FSG Original v1.37 (PAFI v1.0.1) [50]	2003
gSpan [97]	gSpan Original v.6 [94]	2009
	gSpan Original 64-bit v.6 [94]	2009
	gSpan ParSeMis [36, 78]	2011
	gSpan (Kudo) [75]	2004
	gSpan ParMol [10]	2013
	gSpan (Zhou)[11] [107]	2015
	gSpan (Chen) v0.2.2 [13]	2018
MoFa/MoSS [11]	MoFa ParMol [64, 91]	2013
	MoSS ParMol [64, 91]	2013
	MoFa/Moss Original (Miner v6.13) [10]	2015
AcGM [46]	AcGM Original [42]	-
FFSM [40]	FFSM Original v3.0 [26]	2010
	FFSM ParMol [64, 91]	2013
Gaston [72]	Gaston Original v1.1 [69]	2005
	Gaston Original RE v1.1 [69]	2005
	Gaston ParMol [64, 91]	2013
	Gaston ParSeMis [36, 78]	2011
	Gaston (Conduff) [16]	2017
DMTL [6]	DMTL Original v1.0 (g++ 4.8 compiler) [101]	2006

usable for general cases. Another example is TGP [59] that proposes to return frequent subgraphs of a specific size with no possibility of use of minimum support threshold (see Table 10).

Table 10: Incomplete Search & Exact Matching FSM Algorithms with specific uses

Algorithm	**I/O Graphs**	**Specific Case**
CLOSECUT& SPLAT [100]	Input	Relational[12] graph set
CLAN [89]	Output	Frequent cliques[13]
Cocain [103]	Output	Closed Quasi-cliques
gFSG [55] MaxGeo [9]	Input	Geometric Graphs
DIGDAG [84]	Input	Directed Acyclic Graphs with distinct labels
TGP [59]	Input	No minimum support threshold A minimum size of FS is set For large graphs, TGP does not work well [59]

4.3.2 Availability of Software

We collected eight algorithms software out of fifteen from authors and available links (see Table 11).

Table 11: Available Implementations of Incomplete Search & Exact Matching FSM algorithms

Algorithm	**Available software**	**Last Release**
SUBDUE [37]	subdue 5.2.2	2011
CloseGraph [98]	gSpan ParMol[14]	2013
SPIN [41]	SPIN	2007
LEAP [95]	LeapMine	2007
Al Hasan algorithm[15] [8]	OSS	2011
MUSK [7]	Uniform Maximal	2015
Takigawa algorithm [82]	deltol gspan	2015
FS^3 [80]	randomminer	2014

The rest of algorithms are unavailable for specific reasons (see Table 12). The eight selected algorithms have each one implementation.

---

[12] Vertex labels are distinct in each graph
[13] fully connected subgraphs
[14] CloseGraph is gSpan with the Closed Subgraphs Only option. ParMol framework could be provided by authors [64, 91]
[15] Output Space Sampling

Table 12: Unavailable Incomplete Search & Exact Matching FSM algorithms software

Algorithms	**Unavailability**
LCGMiner [93], MARGIN [86], RP-FP & RP-GD [61] [58], JPMiner [62], FOGGER [102], TD-MG [14]	No answer from authors
gdClosed [29]	Under intellectual propoerties

4.4 Incomplete Search & Inexact Matching

Some algorithms in FSM propose to return a subset of frequent subgraphs in order to reduce the output or to return a richer set than the complete one in order to take into consideration uncertainty and innaccuracy of data. We filter in the following these algorithms.

4.4.1 Performance of FSM algorithms We detected one algorithm that had limits of performance ORIGAMI [5].Its authors [7] cites that "The sampling of maximal frequent graphs in ORIGAMI is far from uniform, which may result in poor quality representatives".

4.4.2 Weaknesses Two algorithms are useful in specific cases (see Table 13). For example, ISG with perform on graphs with unique edge labels.

Table 13: Incomplete Search & Inexact Matching FSM Algorithms with specific uses

Algorithm	**Input Graph Case**
ISG [85]	Graphs with unique edge labels
MUSE [108]	Uncertain labeled undirected graphs

4.4.3 Availability of Software We found two available algorithms' software out of the remaining 12 ones (see Table 14). The rest are unavailable for specific reasons (see Table 15).

Table 14: Available Incomplete Search & Inexact Matching Implementations of FSM algorithms

Algorithm	**Available software**	**Last Release**
SUBDUE [37]	subdue 5.2.2	2011
APGM [47]	APGM	2010

5 Conclusion

A selection is made in this report of FSM algorithms implementations available to be used for general graphs in Graph Transaction Databases. As a total, fifteen algorithms are freely available as an implementation out of sixty eight FSM algorithms. Respectively, 6 and 9[16] of the Complete and Incomplete Search category. The choice of one algorithm instead of another is justified by an experimental study between the software. For complete search algorithms, we provide an experimental comparison of all of the 13 implementations of the 6 algorithms[17]. No study proposes to compare all the software at once for Incomplete search implementations. A further work should consider this.

Table 15: Unavailable Incomplete Search & Inexact Matching FSM algorithms software

Algorithms	**Unavailability**
Monkey [105], Chen algorithm [12], RAM [104], RING [106], TRS [22], Summarization Framework [63], MUG [18]	No answer from authors
VEAM [1], VEAMwP [2], MgVEAM [65]	Under intellectual propoerties

6 Acknowledgments

This work has been elaborated as a part of the CAIR[18] project. Special thanks are addressed to FSM authors especially to Xifeng Yan, Thorsten Meinl, Andrés Gago-Alonso, Christian Borgelt, Mohammad Al Hasan and Sabeur Aridhi for sending us software, datasets, also for providing clarifications and for their availability.

References

1. N. Acosta-Mendoza, A. Gago-Alonso, and J. E. Medina-Pagola. Frequent approximate subgraphs as features for graph-based image classification. *Knowledge-Based Systems*, 27:381–392, Mar. 2012.
2. N. Acosta-Mendoza, A. Gago-Alonso, and J. E. Medina-Pagola. On Speeding up Frequent Approximate Subgraph Mining. In L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, number 7441 in Lecture Notes in Computer Science, pages 316–323. Springer Berlin Heidelberg, Sept. 2012. DOI: 10.1007/978-3-642-33275-3_39.
3. C. C. Aggarwal, H. Wang, and others. *Managing and mining graph data*, volume 40. Springer, 2010.
4. M. Al Hasan. Mining Interesting Subgraphs by Output Space Sampling. *SIGKDD Explor. Newsl.*, 12(1):73–74, Nov. 2010.
5. M. Al Hasan, V. Chaoji, S. Salem, J. Besson, and M. J. Zaki. ORIGAMI: Mining Representative Orthogonal Graph Patterns. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 153–162, Oct. 2007.
6. M. Al Hasan, V. Chaoji, S. Salem, N. Parimi, and M. J. Zaki. DMTL: A Generic Data Mining Template Library.
7. M. Al Hasan and M. Zaki. Musk: Uniform Sampling of k Maximal Patterns. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, Proceedings, pages 650–661. Society for Industrial and Applied Mathematics, 2009.

[16] It is worth noting that the SUBDUE algorithm performs both exact and inexact search, it is counted once
[17] see https://liris.cnrs.fr/rihab.ayed/ESFSM.pdf for more details
[18] https://www.irit.fr/CAIR/fr/

8. M. Al Hasan and M. J. Zaki. Output Space Sampling for Graph Patterns. *Proc. VLDB Endow.*, 2(1):730–741, Aug. 2009.
9. H. Arimura, T. Uno, and S. Shimozono. Time and Space Efficient Discovery of Maximal Geometric Graphs. In V. Corruble, M. Takeda, and E. Suzuki, editors, *Discovery Science*, number 4755 in Lecture Notes in Computer Science, pages 42–55. Springer Berlin Heidelberg, Oct. 2007. DOI: 10.1007/978-3-540-75488-6_6.
10. C. Borgelt. Moss - molecular substructure miner. `http://www.borgelt.net/moss.html`. [Online; accessed 2016-05-30].
11. C. Borgelt and M. R. Berthold. Mining molecular fragments: finding relevant substructures of molecules. In *2002 IEEE International Conference on Data Mining, 2002. ICDM 2003. Proceedings*, pages 51–58, 2002.
12. C. Chen, C. X. Lin, X. Yan, and J. Han. On Effective Presentation of Graph Patterns: A Structural Representative Approach. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 299–308, New York, NY, USA, 2008. ACM.
13. Q. Chen. gspan with python. `https://github.com/betterenvi/gSpan`. [Online; accessed 2018-08-25].
14. X. Chen, C. Zhang, F. Liu, and J. Guo. Algorithm Research of Top-Down Mining Maximal Frequent SubGraph Based on Tree Structure. In P. Sénac, M. Ott, and A. Seneviratne, editors, *Wireless Communications and Applications*, number 72 in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pages 401–411. Springer Berlin Heidelberg, Aug. 2011. DOI: 10.1007/978-3-642-29157-9_38.
15. M. Cohen and E. Gudes. Diagonally Subgraphs Pattern Mining. In *Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, DMKD '04, pages 51–58, New York, NY, USA, 2004. ACM.
16. C. Conduff. Gaston graph mining with python. `https://github.com/ColinConduff/FrequentSubgraphMining`. [Online; accessed 2018-08-25].
17. D. J. Cook and L. B. Holder. Substructure Discovery Using Minimum Description Length and Background Knowledge. *J. Artif. Int. Res.*, 1(1):231–255, 1994.
18. B. Cule, B. Goethals, and T. Hendrickx. Mining Interesting Itemsets in Graph Datasets. In J. Pei, V. S. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining*, number 7818 in Lecture Notes in Computer Science, pages 237–248. Springer Berlin Heidelberg, Apr. 2013. DOI: 10.1007/978-3-642-37453-1_20.
19. R. de Sousa Gomide, C. D. de Aguiar Ciferri, R. R. Ciferri, and M. T. P. Vieira. ADI-Minebio: A Graph Mining Algorithm for Biomedical Data. *Journal of Information and Data Management*, 2(3):433, Sept. 2011.
20. L. Dehaspe, H. Toivonen, and R. D. King. Finding Frequent Substructures in Chemical Compounds. pages 30–36. AAAI Press, 1998.
21. C. Desrosiers, P. Galinier, P. Hansen, and A. Hertz. Sygma: Reducing symmetry in graph mining. Technical report, Les Cahiers du GERAD, 2007.
22. W. Dhifli, M. Moussaoui, R. Saidi, and E. M. Nguifo. Towards an Efficient Discovery of Topological Representative Subgraphs. *ResearchGate*, July 2013.
23. H. Dinari and H. Naderi. A Survey of Frequent Subgraphs and Subtree Mining Methods. *ResearchGate*, 14(1):1694–2108, May 2014.
24. B. Douar, M. Liquiere, C. Latiri, and Y. Slimani. LC-mine: a framework for frequent subgraph mining with local consistency techniques. *Knowledge and Information Systems*, 44(1):1–25, July 2014.
25. M. Elseidy, E. Abdelhamid, S. Skiadopoulos, and P. Kalnis. GraMi: Frequent Subgraph and Pattern Mining in a Single Large Graph. *Proc. VLDB Endow.*, 7(7):517–528, Mar. 2014.
26. H. Fei. Fast frequent subgraph mining (ffsm). `https://sourceforge.net/projects/ffsm/`. [Online; accessed 2016-05-30].
27. A. Gago-Alonso and J. A. Carrasco-Ochoa. Full Duplicate Candidate Pruning for Frequent Connected Subgraph Mining. *Integrated Computer-Aided Engineering*, 17(3):211–225, 2010.

28. A. Gago-Alonso, J. E. M. Pagola, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad. Mining Frequent Connected Subgraphs Reducing the Number of Candidates. In W. Daelemans, B. Goethals, and K. Morik, editors, *Machine Learning and Knowledge Discovery in Databases*, number 5211 in Lecture Notes in Computer Science, pages 365–376. Springer Berlin Heidelberg, Sept. 2008. DOI: 10.1007/978-3-540-87479-9_42.
29. A. Gago-Alonso, A. Puentes-Luberta, J. A. Carrasco-Ochoa, J. E. Medina-Pagola, and J. F. Martínez-Trinidad. A New Algorithm for Mining Frequent Connected Subgraphs based on Adjacency Matrices. *Intelligent Data Analysis*, 14(3):385–403, Aug. 2010.
30. Z. Gao, L. Shang, and Y. Jian. Frequent subgraph mining based on the automorphism mapping. In *2012 2nd International Conference on Computer Science and Network Technology (ICCSNT)*, pages 1518–1522, 2012.
31. M. Gholami and A. Salajegheh. A survey on algorithms of mining frequent subgraphs. *International Journal of Engineering Inventions*, 1(5):60–63, 2012.
32. A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, June 1985.
33. E. Gudes, S. E. Shimony, and N. Vanetik. Discovering Frequent Graph Patterns Using Disjoint Paths. *IEEE Transactions on Knowledge and Data Engineering*, 18(11):1441–1456, Nov. 2006.
34. J. Han, J. Pei, and M. Kamber. *Data Mining: Concepts and Techniques*. Elsevier, June 2011.
35. S. Han, W. K. Ng, and Y. Yu. FSP: Frequent Substructure Pattern mining. In *2007 6th International Conference on Information, Communications Signal Processing*, pages 1–5, 2007.
36. T. Henderson. Parsemis. `https://github.com/timtadh/parsemis`. [Online; accessed 2016-05-30].
37. L. B. Holder, D. J. Cook, S. Djoko, et al. Substucture discovery in the subdue system. In *KDD workshop*, pages 169–180, 1994.
38. M. Hong, H. Zhou, W. Wang, and B. Shi. An Efficient Algorithm of Frequent Connected Subgraph Extraction. In K.-Y. Whang, J. Jeon, K. Shim, and J. Srivastava, editors, *Advances in Knowledge Discovery and Data Mining*, number 2637 in Lecture Notes in Computer Science, pages 40–51. Springer Berlin Heidelberg, Apr. 2003. DOI: 10.1007/3-540-36175-8_5.
39. H. Hu, X. Yan, Y. Huang, J. Han, and X. J. Zhou. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21(suppl 1):i213–i221, June 2005.
40. J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 549–552. IEEE, 2003.
41. J. Huan, W. Wang, J. Prins, and J. Yang. SPIN: Mining Maximal Frequent Subgraphs from Graph Databases. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 581–586, New York, NY, USA, 2004. ACM.
42. A. Inokuchi. Acgm. kwansei gakuin university. `http://ist.ksc.kwansei.ac.jp/~inokuchi/acgm.zip`. [Online; accessed 2016-05-30].
43. A. Inokuchi, T. Washio, and H. Motoda. An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. pages 13–23, 2000.
44. A. Inokuchi, T. Washio, and H. Motoda. Complete Mining of Frequent Patterns from Graphs: Mining Graph Data. *Machine Learning*, 50(3):321–354, Mar. 2003.
45. A. Inokuchi, T. Washio, and H. Motoda. A general framework for mining frequent subgraphs from labeled graphs. *Fundamenta Informaticae*, 66(1-2):53–82, 2005.
46. A. Inokuchi, T. Washio, K. Nishimura, and H. Motoda. A fast algorithm for mining frequent connected subgraphs. Technical report, IBM, 2002.
47. Y. Jia, J. Zhang, and J. Huan. An efficient graph-mining method for complicated and noisy data with real-world applications. *Knowledge and Information Systems*, 28(2):423–447, Feb. 2011.
48. C. Jiang, F. Coenen, and M. Zito. A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review*, 28(01):75–105, Mar. 2013.
49. R. Jin, C. Wang, D. Polshakov, S. Parthasarathy, and G. Agrawal. Discovering Frequent Topological Structures from Graph Datasets. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 606–611, New York, NY, USA, 2005. ACM.

50. G. Karypis. Pafi software package for finding frequent patterns in diverse datasets. Karypis Lab. `http://glaros.dtc.umn.edu/gkhome/project/dm/software?q=pafi/overview`. [Online; accessed 2016-05-30].
51. Y. Ke and J. Cheng. Efficient Correlation Search from Graph Databases. *IEEE Trans. Knowl. Data Eng.*, 20(12):1601–1615, 2008.
52. S. Kramer, L. De Raedt, and C. Helma. Molecular Feature Mining in HIV Data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 136–143, New York, NY, USA, 2001. ACM.
53. V. Krishna, N. R. Suri, and G. Athithan. A comparative survey of algorithms for frequent subgraph discovery. *CURRENT SCIENCE*, 100(2):190, 2011.
54. M. Kuramochi and G. Karypis. Frequent Subgraph Discovery. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages 313–320, Washington, DC, USA, 2001. IEEE Computer Society.
55. M. Kuramochi and G. Karypis. Finding Frequent Patterns in a Large Sparse Graph. *Data Min. Knowl. Discov.*, 11(3):243–271, Nov. 2005.
56. K. Lakshmi and T. Meyyappan. Frequent subgraph mining algorithms - a survey and framework for classification. 2012.
57. M. Lalmas. Aggregated Search. In M. Melucci and R. Baeza-Yates, editors, *Advanced Topics in Information Retrieval*, number 33 in The Information Retrieval Series, pages 109–123. Springer Berlin Heidelberg, 2011. DOI: 10.1007/978-3-642-20946-8_5.
58. J. Li, Y. Liu, and H. Gao. Efficient Algorithms for Summarizing Graph Patterns. *IEEE Transactions on Knowledge and Data Engineering*, 23(9):1388–1405, Sept. 2011.
59. Y. Li, Q. Lin, R. Li, and D. Duan. TGP: Mining Top-K Frequent Closed Graph Pattern without Minimum Support. In L. Cao, Y. Feng, and J. Zhong, editors, *Advanced Data Mining and Applications*, number 6440 in Lecture Notes in Computer Science, pages 537–548. Springer Berlin Heidelberg, Nov. 2010. DOI: 10.1007/978-3-642-17316-5_51.
60. Y. Li, Q. Lin, G. Zhong, D. Duan, Y. Jin, and W. Bi. A Directed Labeled Graph Frequent Pattern Mining Algorithm Based on Minimum Code. In *Third International Conference on Multimedia and Ubiquitous Engineering, 2009. MUE '09*, pages 353–359, June 2009.
61. Y. Liu, J. Li, and H. Gao. Summarizing Graph Patterns. In *2008 IEEE 24th International Conference on Data Engineering*, pages 903–912, Apr. 2008.
62. Y. Liu, J. Li, and H. Gao. JPMiner: Mining Frequent Jump Patterns from Graph Databases. In *Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09*, volume 5, pages 114–118, Aug. 2009.
63. Z. Liu, R. Jin, H. Cheng, and J. X. Yu. Frequent Subgraph Summarization with Error Control. In J. Wang, H. Xiong, Y. Ishikawa, J. Xu, and J. Zhou, editors, *Web-Age Information Management*, number 7923 in Lecture Notes in Computer Science, pages 1–12. Springer Berlin Heidelberg, June 2013. DOI: 10.1007/978-3-642-38562-9_1.
64. T. Meinl, M. WÃ¶rlein, O. Urzova, I. Fischer, and M. Philippsen. The ParMol Package for Frequent Subgraph Mining. *Electronic Communications of the EASST*, 1(0), July 2007.
65. N. A. Mendoza, J. A. Carrasco-Ochoa, A. Gago-Alonso, J. F. Martínez-Trinidad, and J. E. Medina-Pagola. Representative Frequent Approximate Subgraph Mining in Multi-Graph Collections. 2015.
66. K. Mohammad Reza and A. Fereshteh. Classification and analysis of frequent subgraphs mining algorithms. *JSW*, 7(1):220–227, 2012.
67. M. H. Nadimi-Shahraki, M. Taki, and M. Naderi. IDFP-TREE: An Efficient Tree for interactive mining of frequent subgraph patterns. *Journal of Theoretical and Applied Information Technology*, 74(3), 2015.
68. P. C. Nguyen, T. Washio, K. Ohara, and H. Motoda. Using a Hash-Based Method for Apriori-Based Graph Mining. In J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, editors, *Knowledge Discovery in Databases: PKDD 2004*, number 3202 in Lecture Notes in Computer Science, pages 349–361. Springer Berlin Heidelberg, Sept. 2004. DOI: 10.1007/978-3-540-30116-5_33.
69. S. Nijssen. Gaston - download. `http://liacs.leidenuniv.nl/~nijssensgr/gaston/download.html`. [Online; accessed 2016-05-30].

70. S. Nijssen. Performance comparison of graph mining algorithms on pte. `http://liacs.leidenuniv.nl/~nijssensgr/farmer/results.html`, 2003. [Online; accessed 2016-05-30].
71. S. Nijssen and J. Kok. Faster association rules for multiple relations. In *In International Joint Conference on Artificial Intelligence*, pages 891–896. Morgan Kaufmann, 2001.
72. S. Nijssen and J. N. Kok. A Quickstart in Frequent Structure Mining Can Make a Difference. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 647–652, New York, NY, USA, 2004. ACM.
73. S. Nijssen and J. N. Kok. The Gaston Tool for Frequent Subgraph Mining. *Electronic Notes in Theoretical Computer Science*, 127(1):77–87, Mar. 2005.
74. S. Nijssen and J. N. Kok. Frequent subgraph miners: Runtime dont say everything. In *Proceedings of the International Workshop on Mining and Learning with Graphs (MLG 2006*, pages 173–180, 2006.
75. S. Nowozin. A fork of sebastian nowozin's and koji tsuda's gboost code. `https://github.com/rkwitt/gboost`. [Online; accessed 2019-07-07].
76. S. Nowozin and K. Tsuda. Frequent Subgraph Retrieval in Geometric Graph Databases. In *2008 Eighth IEEE International Conference on Data Mining*, pages 953–958, Dec. 2008.
77. H. J. Patel, R. Prajapati, M. Panchal, and M. Patel. A Survey of Graph Pattern Mining Algorithm and Techniques. *International Journal of Application or Innovation in Engineering & Management*, 2(1):125–129, 2013.
78. M. Philippsen. Parsemis - the parallel and sequential mining suite. `https://www2.cs.fau.de/EN/research/zold/ParSeMiS/index.html`. [Online; accessed 2016-05-30].
79. T. Ramraj and R. Prabhakar. Frequent Subgraph Mining Algorithms – A Survey. *Procedia Computer Science*, 47:197–204, Jan. 2015.
80. T. K. Saha and M. A. Hasan. FS^3: A Sampling based method for top-k Frequent Subgraph Mining. *arXiv:1409.1152 [cs]*, Sept. 2014. arXiv: 1409.1152.
81. Ł. Skonieczny. Mining for Unconnected Frequent Graphs with Direct Subgraph Isomorphism Tests. In K. A. Cyran, S. Kozielski, J. F. Peters, U. Stańczyk, and A. Wakulicz-Deja, editors, *Man-Machine Interactions*, number 59 in Advances in Intelligent and Soft Computing, pages 523–531. Springer Berlin Heidelberg, 2009. DOI: 10.1007/978-3-642-00563-3_55.
82. I. Takigawa and H. Mamitsuka. Efficiently mining δ-tolerance closed frequent subgraphs. *Machine Learning*, 82(2):95–121, Sept. 2010.
83. A. Termier, Y. Tamada, K. Numata, S. Imoto, T. Washio, and T. Higuchi. Digdag, a first algorithm to mine closed frequent embedded sub-dags. In *Mining and Learning with Graphs, MLG 2007, Firence, Italy, August 1-3, 2007, Proceedings*, 2007.
84. A. Termier, Y. Tamada, K. Numata, S. Imoto, T. Washio, and T. Higuchi. DIGDAG, a First Algorithm to Mine Closed Frequent Embedded Sub-DAGs. In *ResearchGate*, Jan. 2007.
85. L. Thomas, S. Valluri, and K. Karlapalem. Isg: Itemset based subgraph mining. Technical report, Technical Report, IIIT, Hyderabad, December2009, 2009.
86. L. T. Thomas, S. R. Valluri, and K. Karlapalem. MARGIN: Maximal Frequent Subgraph Mining. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 1097–1101, Dec. 2006.
87. N. Vijayalakshmi. FP-GraphMiner - A Fast Frequent Pattern Mining Algorithm for Network Graphs. *Journal of Graph Algorithms and Applications*, 15(6):753–776, 2011.
88. C. Wang, W. Wang, J. Pei, Y. Zhu, and B. Shi. Scalable Mining of Large Disk-based Graph Databases. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 316–325, New York, NY, USA, 2004. ACM.
89. J. Wang, Z. Zeng, and L. Zhou. CLAN: An Algorithm for Mining Closed Cliques from Large Dense Graph Databases. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 73–73, Apr. 2006.
90. W. Wang, C. Wang, Y. Zhu, B. Shi, J. Pei, X. Yan, and J. Han. GraphMiner: A Structural Pattern-mining System for Large Disk-based Graph Databases and Its Applications. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 879–881, New York, NY, USA, 2005. ACM.

91. M. Wörlein, T. Meinl, I. Fischer, and M. Philippsen. A Quantitative Comparison of the Subgraph Miners MoFa, gSpan, FFSM, and Gaston. In A. M. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, editors, *Knowledge Discovery in Databases: PKDD 2005*, number 3721 in Lecture Notes in Computer Science, pages 392–403. Springer Berlin Heidelberg, Oct. 2005. DOI: 10.1007/11564126_39.
92. J. Wu and L. Chen. A Fast Frequent Subgraph Mining Algorithm. In *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, pages 82–87, Nov. 2008.
93. A. Xu and H. Lei. LCGMiner: levelwise closed graph pattern mining from large databases. In *16th International Conference on Scientific and Statistical Database Management, 2004. Proceedings*, pages 421–422, June 2004.
94. X. Yan. Software - gspan: Frequent graph mining package. `http://www.cs.ucsb.edu/~xyan/software/gSpan.htm`. [Online; accessed 2016-05-30].
95. X. Yan, H. Cheng, J. Han, and P. S. Yu. Mining Significant Graph Patterns by Leap Search. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 433–444, New York, NY, USA, 2008. ACM.
96. X. Yan and J. Han. gSpan : Graph-Based Substructure Pattern Mining. Technical report, UIUC, UIUCDCS-R-2002-2296, 2002.
97. X. Yan and J. Han. gSpan: graph-based substructure pattern mining. In *2002 IEEE International Conference on Data Mining, 2002. ICDM 2003. Proceedings*, pages 721–724, 2002.
98. X. Yan and J. Han. CloseGraph: Mining Closed Frequent Graph Patterns. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 286–295, New York, NY, USA, 2003. ACM.
99. X. Yan, P. S. Yu, and J. Han. Graph Indexing: A Frequent Structure-based Approach. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, pages 335–346, New York, NY, USA, 2004. ACM.
100. X. Yan, X. J. Zhou, and J. Han. Mining Closed Relational Graphs with Connectivity Constraints. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 324–333, New York, NY, USA, 2005. ACM.
101. M. J. Zaki. Data mining template library (dmtl). `http://www.cs.rpi.edu/~zaki/www-new/pmwiki.php/Software/Software`. [Online; accessed 2016-05-30].
102. Z. Zeng, J. Wang, J. Zhang, and L. Zhou. FOGGER: An Algorithm for Graph Generator Discovery. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '09, pages 517–528, New York, NY, USA, 2009. ACM.
103. Z. Zeng, J. Wang, L. Zhou, and G. Karypis. Coherent Closed Quasi-clique Discovery from Large Dense Graph Databases. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 797–802, New York, NY, USA, 2006. ACM.
104. S. Zhang and J. Yang. RAM: Randomized Approximate Graph Mining. In B. Ludäscher and N. Mamoulis, editors, *Scientific and Statistical Database Management*, number 5069 in Lecture Notes in Computer Science, pages 187–203. Springer Berlin Heidelberg, July 2008. DOI: 10.1007/978-3-540-69497-7_14.
105. S. Zhang, J. Yang, and V. Cheedella. Monkey: Approximate Graph Mining Based on Spanning Trees. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 1247–1249, Apr. 2007.
106. S. Zhang, J. Yang, and S. Li. Ring: An integrated method for frequent representative subgraph mining. In *ICDM*, 2009.
107. K. Zhou. gspan algorithm in data mining. `https://github.com/Jokeren/DataMining-gSpan`. [Online; accessed 2016-05-30].
108. Z. Zou, J. Li, H. Gao, and S. Zhang. Frequent Subgraph Pattern Mining on Uncertain Graph Data. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 583–592, New York, NY, USA, 2009. ACM.
109. Z. Zou, J. Li, H. Gao, and S. Zhang. Mining Frequent Subgraph Patterns from Uncertain Graph Data. *IEEE Transactions on Knowledge and Data Engineering*, 22(9):1203–1218, Sept. 2010.