Mining Interesting Outlier Subgraphs in Attributed Graphs

Ahmad Mel[1], Tijl De Bie[1]

[1]AIDA, IDLab, Ghent University, Belgium

Abstract

A common task in local pattern mining on attributed graphs revolves around detecting subgraphs with predefined similarity measures on the attributes of the nodes. In this work we focus on mining cohesive subgraphs with attribute values that stand out when compared to the rest of the graph. We tackle a more specific problem: given a vertex-attributed graph, we aim to detect a cohesive set of connected vertices , such that for a maximal set of the attributes, at least one of the vertices in the subgraph shows an abnormal value. Such patterns are very relevant in the biological field, where anomaly detection is used to identify parts of a corrupt gene network. We develop a pattern syntax and an interestingness score to mine such subgraphs, and implement it using a branch-and-bound algorithm. Furthermore, we design an experimental setup to qualitatively assess the results, discuss our findings, limitations and future improvements.

Keywords

Subgraph Mining, Subjective Interestingness, Attributed Graphs

1. Introduction

The graph mining field saw a rise in interest in recent years, benefiting from the surge in the amount of network data available from the ever-growing data revolution, from social media to biological protein-protein networks.

For example, in the field of cancer biology, a cancer driver gene is a gene whose mutations cause or facilitate cellular cancer initiation and progression [1], and thus exploratory pathway and network analysis is becoming an integral tool for cancer driver gene discovery[2], and associating infrequently mutated genes as cancer genes based on their interaction with recurrently mutated genes [3, 4].

As more fields are employing graph data, the use of rich graphs for exploratory data mining became more popular. While classical graphs solely rely on node topology, rich graphs add another layer of data in the form of node and/or edge characteristics[5]. Thus, data mining on node-attributed graphs (referred to as attributed graphs throughout the rest of the paper) combines additional information and offers more insights for enhanced graph mining tasks[6].

Proceedings of the Second International Workshop on Explainable and Interpretable Machine Learning (XI-ML 2022) co-located with the 45rd German Conference on Artificial Intelligence (KI 2022), September 20, 2022, Trier (Virtual), Germany

ahmad.mel@ugent.be (A. Mel); tijl.debie@ugent.be (T. D. Bie)
https://aida.ugent.be/ (A. Mel); http://www.tijldebie.net/ (T. D. Bie)
0000-0003-4249-4005 (A. Mel); 0000-0002-2692-7504 (T. D. Bie)

One of those tasks is graph clustering, or community detection, where the goal is to detect a subgraph (typically a densely connected subset of nodes and the edges between them) of an attributed graph [7, 8], where the nodes have similar (according to some metric) attribute values [9].

Similarly, subgraph anomaly detection on attributed graphs aims to discover outlier substructures within such graphs that are rare with respect to their node attribute values, deviating significantly from the majority of the other nodes in the graph[10].

Both approaches fall under community detection, while the former seeks subgraphs sharing similarities with their attribute values, and the latter finds subgraphs with differing attribute values with respect to the rest of the graph.

In this paper, we introduce a new intuitive interpretable method for exploratory subgraph mining in node attributed graphs, that finds easily describable cohesive set of nodes with attribute values that stand out compared to the rest of graph. We propose and employ a principled interestingness measure to rank the discovered patterns, by maximizing the amount of information contained in a pattern while reducing its size. Recent work [11] on this topic focused on discovering interesting cohesive patterns where for a subset of the attributes, *every* node in the subgraph has exceptional attribute value compared to the rest of the graph. Our work generalizes this work by defining a pattern syntax such that for a subset of the attributes *at least one* node in the subgraph has exceptional value. Our novel method aligns with the application of pathway analysis for gene discovery, where the aim is to find oncogenic pathways (subgraphs) of interacting genes (nodes) and encoded patient mutation data (attributes), such that the size of the pathway is minimal while maximizing the patient mutation on the genes.

This approach overcomes many limitations including the need to define exceptions for the detected patterns, where some nodes in the subgraph do not match the described pattern. We detail the formulation in section 2. An implementation of the search and rank approach is detailed in section 3. In section 4 we design an experimental study using real world datasets from attributed gene interaction networks to validate our method qualitatively. Finally, we conclude our paper by discussing the results and limitations of our work while setting up future improvement in section 5.

Our main contributions can be summarized as follows:

- We introduce a novel interpretable pattern syntax for attributed subgraph mining
- We implement an interestingness measure to rank the described patterns using a heuristic branch-and-bound search algorithm
- We design an experimental study to validate our method qualitatively on a real world dataset

2. Pattern Syntax

In this section, we formalize the problem statement and the proposed pattern syntax, and define the interestingness measure used to rank the patterns.

2.1. Preliminaries

Let $G = (V, E, \hat{A})$ be a vertex-attributed undirected graph. V is a set of n vertices. $E \subseteq V \times V$, a set of m edges, With $\hat{A}$ we denote a set of p numerical attributes. They are formalized as functions $\hat{a}(v) \in \text{Dom}_a$, denoting the value of attribute $\hat{a} \in \hat{A}$ for node $v \in V$. Here, Dom_a denotes the set of possible values to which the attribute values $\hat{a}(v)$ belong. We use hats over $\hat{A}$ and $\hat{a}$ to emphasize that we are talking about their empirically observed values here. Whenever we consider them to be random variables, the hats will be omitted.

We define a cohesive subgraph with anomalous attributes (CSAA) as a tuple (U, S) for which the following holds:

- $U \subseteq V$ is a set of vertices in the graph.
- $S \subseteq \{(a, k_a) \mid a \in A,\ k_a \in \text{Dom}_a\}$ is a set of so-called *characteristics*.
- $\forall (a, k_a) \in S,\ \exists u \in U: \quad a(u) \geq k_a$

We refer to the set of vertices U as the CSAA *cover* of the CSAA, while S represents the (exceptional) *characteristics* of those vertices, restricting the attribute domains of the vertices within the cover.

We also define $N_d(v)$ as the neighborhood of range d of a vertex v, as the set of vertices whose shortest path to v is at most d:

$$N_d(v) = \{u \in V \mid dist(v, u) \leq d\}.$$

While the CSAA pattern syntax has to the best of our knowledge not been proposed before, it is relevant in concrete applications. E.g., in the biological pathway network analysis problem mentioned in the introduction, where genes represent the nodes, and mutation data the characteristics, a CSAA would define a pathway (subgraph) of genes (nodes) such that at for a maximal subset of mutation characteristics (attributes), there exists at least one gene in the pathway that fits those mutation criteria.

Problem Statement Following the notations defined above, given a vertex-attributed graph G, our goal is to find a tuple (U, S), such that for every attribute in S, at least one vertex u in U, should exceed a specified threshold k_a.

2.2. Interestingness Measure

Now that we have described the pattern syntax, the next step is to define the ranking mechanism used to quantify the interestingness of the patterns. This is important, as generally speaking there will be many CSAA's in any given attributed network. Thus, in this section we cover our interestingness measure and discuss its formulation in detail, relying on the FORSIED framework, that leverages the background knowledge of the user about the data and the complexity of the pattern to quantify its interestingness using information theory. The framework aims provide as much knowledge to the user about the data, with as little cost as possible, while relying on the user's prior belief. More specifically, our subjective interestingness measure SI is defined as the ratio of the information content IC (background knowledge) over the description length DL

(complexity) of the pattern. Such formulation intuitively leads to a pattern that maximizes the information content while remaining easily describable.

Our approach takes into account both the topology of the graph and its embedded data which makes it more versatile than other anomaly detection techniques.

$$\mathrm{SI}(U, S) = \frac{\mathrm{IC}(U, S)}{\mathrm{DL}(U, S)}. \tag{1}$$

2.2.1. Information Content

Subjective measures account for both the data and its user, thus relying on the background knowledge that the user has about the data. Such prior beliefs are modeled as equality constraints assuming that the user knows general statistics about the data.

In this context, we define information content IC of the pattern as the self-information or surprisal, a measure of unexpectedness of the pattern. Meaning, the least expected patterns are the most informative, and have the highest information content.

We also consider only attributed graphs with non-negative integer values for the attributes. In other words $(a : V \ \rightarrow \mathbb{N}, \forall a \in A)$. In our experiments we will even consider the situation where the attribute value domains are binary (1 or 0).

We consider also that the user knows a priori general statistics about the graph, more specifically the averages over the attributes and vertices. This leads to prior beliefs formalized as equality constraints on the attribute values A on all the vertices, more specifically, one constraint modeled after the total sum of the attributes for each vertex, and the other after the total sum of all the vertices for each attribute in the graph, this is denoted below:

$$\sum_{A} \Pr(A) \left(\sum_{a \in A} a(v) \right) = \sum_{\hat{a} \in \hat{A}} \hat{a}(v), \quad \forall v \in V, \tag{2}$$

$$\sum_{A} \Pr(A) \left(\sum_{v \in V} a(v) \right) = \sum_{v \in V} \hat{a}(v), \quad \forall a \in A. \tag{3}$$

Additionally, we model the prior beliefs of the user as a Maximum Entropy distribution (MaxEnt) subject to those constraints, as such distribution maximizes the entropy (the uncertainty about the data) while minimizing the bias (using only prior information as bias), subject to the constraints in eq. (2) and eq. (3), and normalization. We refer to the MaxEnt distribution as background distribution, and we formalize the information content of a pattern $\mathrm{IC}(U, S)$ as the negative logarithm of the probability that the pattern is present under the background distribution.

$$\mathrm{IC}(U, S) = -\log(\Pr(U, S)). \tag{4}$$

And following our problem statement we can define $\Pr(U, S)$ as

$$\Pr(U, S) = \Pr(\forall (a, k_a) \in S, \exists u \in U : a(u) \geq k_a). \tag{5}$$

Furthermore, as demonstrated in [12], the maximum entropy problem is a convex optimization problem, of which the solution is a product of independent Geometric distributions, one for each vertex attribute-value $a(v)$, of the form

$$\Pr(a(v) = z) = p_{av}(1 - p_{av})^z, \text{ with } z \in \mathbb{N} \text{ and } p_{av} = 1 - e^{(\lambda_a + \lambda_v)},$$

where p_{av} is the success probability, and λ_a and λ_v the Lagrange multipliers corresponding to the two constraint types.

Thus, eq. (5) can be written as:

$$\Pr(\forall (a, k_a) \in S, \exists u \in U : a(u) \geq k_a) = \prod_{(a,k_a)\in S} \Pr(\exists u \in U : a(u) \geq k_a).$$

The factors in this product can be written as follows:

$$\begin{aligned}
\Pr(\exists u \in U : a(u) \geq k_a) &= 1 - \Pr(\forall u \in U : a(u) < k_a) \\
&= 1 - \prod_{u \in U} \Pr(a(u) < k_a) \\
&= 1 - \prod_{u \in U} (1 - \Pr(a(u) \geq k_a)) \\
&= 1 - \prod_{u \in U} (1 - (1 - p_{au})^{k_a}).
\end{aligned}$$

Furthermore,

$$\Pr(\forall (a, k_a) \in S, \exists u \in U : a(u) \geq k_a) = \prod_{(a,k_a)\in S} (1 - \prod_{u \in U} (1 - (1 - p_{au})^{k_a})).$$

Putting things together, eq. (4) would lead to the following Information Content (IC):

$$\mathrm{IC}(U, S) = \sum_{(a,k_a)\in S} (-\log(1 - \prod_{u \in U} (1 - (1 - p_{au})^{k_a}))).$$

2.2.2. Description Length

The description length DL represents the encoding length of the pattern. Ideally, we aim to find patterns that are the most informative with minimal description cost. Our approach, provides a more intuitive interpretation of the descriptional complexity of a pattern, such that, a higher complexity would negatively impact the total interestingness of a pattern, as the description length DL functions as a regulator.

As mentioned earlier in section 2.1, we describe the vertex set U in the pattern as the intersection of a set of neighborhoods $N_d(v)$, with $v \in V$. More formally, let us define the set of all neighborhoods $\mathcal{N} = \{N_d(v) \mid v \in V \wedge d \in \mathbb{N} \wedge d \leq D\}$ (with D the maximum radius d considered), and let $\mathcal{N}(U) = \{N_d(v) \in \mathcal{N} \mid U \subseteq N_d(v)\}$ be the subset of neighborhoods that

contain U. The length of a description of the set U as the intersection of all neighborhoods in a subset $X \subseteq \mathcal{N}(U)$

$$\mathrm{DL}(X, U) = \log(|\mathcal{N}|) + |X| \cdot \log(|\mathcal{N}|).$$

The first term accounts for encoding of the number of neighborhoods $\log(|\mathcal{N}|)$, and the second term for the description of which neighborhoods are involved ($|X| \log(|\mathcal{N}|)$).

3. Interesting Outlier Subgraph Detection

Now that we have defined a CSAA pattern, we will introduce our method in this section, and subsequently our algorithm[1] that searches for patterns and rank them according to the subjective interesting score defined in section 2.

The main idea is that the ranking is based on the subjective interestingness score SI moderated by the background information and the size of the intersection. The bigger the size of the pattern, the higher the description length DL the lower the total SI. On the other hand, the higher the information content IC, the higher the total SI. So our algorithm tries to optimize over those two variables, to maximize SI. Again, our aim is to find patterns such that for a maximal subset of the attributes, there exists at least one vertex in the pattern that contains anomalous value compared to the rest of the graph. Applying a brute force search strategy is inefficient, thus we rely on a beam search approach for the breadth-first-search branch-and-bound algorithm.

3.1. Beam search

Traversing a search tree, branch-and-bound methods keep track of the best solution found so far and its associated value, and then it computes an upper bound on the highest objective value that can possibly be achieved when exploring new parts of the search space. If such upper bound is worse than the current best solution then this portion of the search space cannot contain the optimal solution, and is pruned. Beam search [13] operates similarly, in each level of the search tree, beam search expands towards the k most promising solutions, and discards the rest, where the k is an integer called the beam width. By varying k we can control the search strategy, from a greedy search for $k = 1$, mirroring the classic branch-and-bound approach, to a larger limit to perform a complete search approach, which in most cases is intractable. One of the main challenge is to define a suitable upper bound for the objective function to effectively prune irrelevant branches.

3.2. Interesting Outlier Subgraph Miner: MIOS

Our aim is to explore CSAA patterns in attributed graphs and rank them efficiently. To do so, we employ a beam search branch-and-bound algorithm. Given an attributed graph G, maximum radius d for the ego graph of a given vertex, and k representing the width of the beam search, the initial step is to enumerate all the neighborhoods $N_d(v)$ (ie. ego graph of vertex v of radius d) for every vertex v in the graph G, using the function `G.AllNeighborhoods(d)`. This would

[1] Our algorithm code is available on the following github link: https://github.com/aida-ugent/mios

constitute our initial search space `candidate-patterns`. We initialize `BestScore`$= \emptyset$ to keep track of the SI values of the most promising k solutions so far. Then for every enumerated neighborhood $N_d(v)$, we generate a pattern object P_0. The function `G.getNeighborhoods` generate candidate neighborhoods X_i that intersects with P_0 to produce a new pattern P with the interestingness score $P.si$, and an upper bound on the interestingness score that we consider as an optimistic bound $P.upperbound = SI^*$.

The upper bound on the interestingness score is computed such that the intersection of the set of neighborhoods $P.X$ of this pattern P with an additional neighborhood would produce a pattern P^* with its cover $P^*.U$ containing a single vertex with maximal information content. Thus $|P^*.X| = |P.X| + 1$, and $P^*.DL$ is the same in this case for all the patterns formed by the different vertices in $P.U$. Consequently, the upper bound on $P^*.si$ would depend solely on the vertex with the maximal IC $P^*.ic$. Thus the upper bound on SI would be:

$$SI^* = \frac{\max_{\forall v \in U}\{IC(\{v\}, S^*)\}}{(|X| + 2) \cdot \log(|\mathcal{N}|)}.$$

This pattern is the best we can hope for as its description length is minimal, and infomation content is maximal for all the childrens node formed with the additional intersection. New candidate patterns are generated for next iteration and the algorithm is repeated until the candidates list is empty. To summarize, the algorithm uses the neighborhoods as search space, and iteratively adds more neighborhoods to the intersection forming new patterns, thus reducing the number of vertices in the newly formed patterns, to maximize the interestingness of the patterns resulting from the intersection of these neighborhoods. The pseudocode for our algorithm is provided in algorithm 1.

4. Experimental Results

In this section we define our experimental setup, by describing our dataset, presenting our qualitative findings of the application of our method on oncological pathway discovery and an assessment of the quality of the patterns.

4.1. Dataset

To test our framework, we utilized a dataset consisting of gene interaction network [14], and patient mutation sample data for each gene in the network. The network consists of 185 genes, and 463 patient samples. We additionally use the PAM50 [15] dataset for validation of the results. PAM50 is a 50-gene signature that classifies breast cancer into five molecular intrinsic subtypes: Luminal A, Luminal B, human epidermal growth factor receptor 2 (HER2)-enriched, Basal-like and Normal-like. Each of the five molecular subtypes vary by their biological properties and prognoses.

The aim is to find a pathway (subgraph) of genes (vertices) such that for a maximal subset of samples (attributes), there exist at least one gene in the pathway that is mutated (anomalous). To assess the abnormality of the gene, we combine data from 463 patients, representing mutation, copy number variation and gene expression data, such that the attribute on the nodes are binary: 1 corresponding to an abnormal gene, and 0 to normal gene.The dataset is illustrated in fig. 1.

Algorithm 1: MIOS(G, d, k)

Input:
G: Vertex-Attributed Graph Object,
d: Maximum Radius of Neighborhoods,
k: Width of the Beam Search

Output:
top-k-patterns: Top-k anomalous pattern objects

```
BestScore = ∅
candidate-patterns ← G.AllNeighborhoods(d)
while candidate-patterns ≠ ∅ do
    next-candidate-patterns= ∅
    for P_0 in candidate-patterns do
        N ← G.getNeighborhoods(P_0.X, P_0.U)
        for X_i in N do
            if P_0.X ∪ X_i ≠ P_0.X and P_0.U ∩ X_i ≠ ∅ then
                P ← Pattern(P_0.X ∪ X_i, P_0.U ∩ X_i)
                if P.upperbound > BestScore then
                    BestScore ← P.si
                    next-candidate-patterns ← addPattern(P)
                    top-k-patterns ← addPattern(P)
    candidate-patterns ← FilterPatterns(k, next-candidate-patterns)
    top-k-patterns ← FilterPatterns(k, top-k-patterns)
return top-k-patterns
```

4.2. Results and discussion

Using MIOS we were able to detect the most relevant genes in the top-50 patterns. We also perform post-processing of the results to remove redundant patterns that are produced by different descriptions (intersection of different genes ego networks). In fig. 2 we visualize the top ranking pattern, a subgraph of two vertices {TP53, FOXA1} formed by the intersection of two neighborhoods $\texttt{N(MYB)}_1$, the ego graph of radius 1 around vertex MYB and $\texttt{N(MDM2)}_1$ the ego graph of radius 1 centered around node MDM2. TP53 is commonly associated with many forms of cancer. While interpreting the pattern quality is challenging and out of the scope of this work, we can nonetheless validate the results by assessing the classification of the resulted patterns and their association with a specific molecular subtype: (LumA, LumB, HER2, Basal, Normal). The goal is to evaluate the detected patterns with gene set enrichment analysis using Fisher Exact Test.

Enrichment analysis of biological pathways is a statistical method that is used to identify pathways that are enriched in a gene list more than would be expected by chance. It is a commonly used method to identify classes of genes that are over-represented in a large set of genes, and may have an association with disease phenotypes. In table 1 we list an example of a contingency table of a pattern for a specific subtype. $\overline{\text{Subtype}}$ represents the counts of samples not associated with the subtype. $\overline{\text{Characteristics}}$ represents the counts of samples not contained in the characteristics of the patterns.

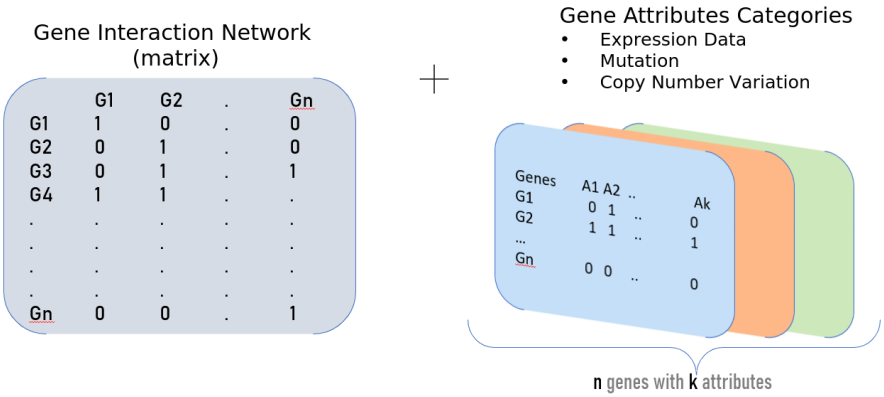

Figure 1: Data sketch for the experimental setup. Combining gene interaction network with the attributes represented by k patients sample binary data (mutation, copy number variation and gene expression), such that an attribute value of 1 corresponds to an abnormal gene, and 0 otherwise.

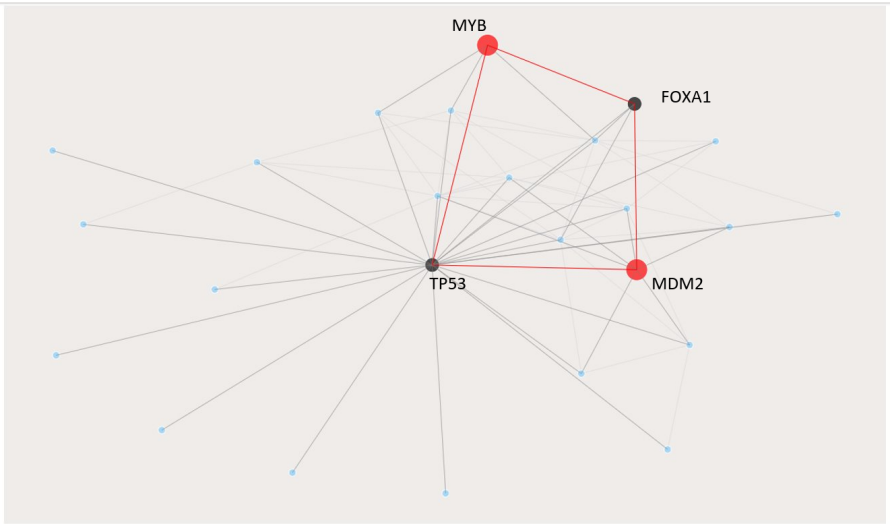

Figure 2: Top pattern ($U=\{TP53,\ FOXA1\},\ X=\{N(MYB)_1, N(MDM2)_1\}$) detected using MIOS. TP53 is a common oncological driver gene.

Fisher exact-test is used to determine the probabilities of observing the various joint values within a contingency table under the null hypothesis assumption that the marginal values are fixed and that there exist no association between the categorical values. We take the a priori stance that the categories are independent. Consequently, we calculate the probability that this contingency table with joint values would occur under the null hypothesis. A small probability is interpreted as a discrepancy between the data and the null hypothesis of no association between variables.

Thus, we can couple the subgroup information to recover certain type of molecular subtype groups.Performing enrichment analysis (fisher exact test, 2-sided and 1-sided) on the top 50 resulting patterns, resulting in fig. 3, we found that the majority of the p-values are below

Table 1
Example of a contingency table for a discovered pattern by MIOS for a specific subtype.

	Subtype	$\overline{\text{Subtype}}$
Characteristics	5	20
$\overline{\text{Characteristics}}$	66	74

0.05 for the fisher tests for HER2 and Basal (both being overrepresented), and LumA (being underrepresented),

This implies that the majority of our patterns exhibit statistically significant imbalance for HER2, Basal and LumA, rendering the classification significant.

5. Conclusion and future work

In this paper we presented MIOS, a novel method for interpretable attributed graph mining for interesting outlier subgraph detection. The goal is to search and rank cohesive sets of nodes with an intuitive explainable description, and anomalous attribute values compared to the rest of the graph. We formulated a pattern syntax and implemented an algorithm to mine such patterns, we then validated our method with real world applications in pathway mining with preliminary findings showing promising qualitative results. Furthermore, we performed additional analysis on the results to assess the ranked patterns using fisher exact tests. The main limitation of our work is the incompletness of the search algorithm, which would be tackled in future work by implementing better heuristic approaches, using constraint satisfaction programming to mitigate the pattern explosion problem. Finally, applications with additional datasets and frameworks with a focus on pathway mining will be another main focus for improvement.

Acknowledgments

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) (ERC Grant Agreement no. 615517), and under the European Union's Horizon 2020 research and innovation programme (ERC Grant Agreement no. 963924), from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, and from the FWO (project no. G0F9816N, 3G042220).

References

[1] A. Torkamani, G. Verkhivker, N. J. Schork, Cancer driver mutations in protein kinase genes, Cancer letters 281 (2009) 117–127.
[2] H. Horn, M. S. Lawrence, C. R. Chouinard, Y. Shrestha, J. X. Hu, E. Worstell, E. Shea,

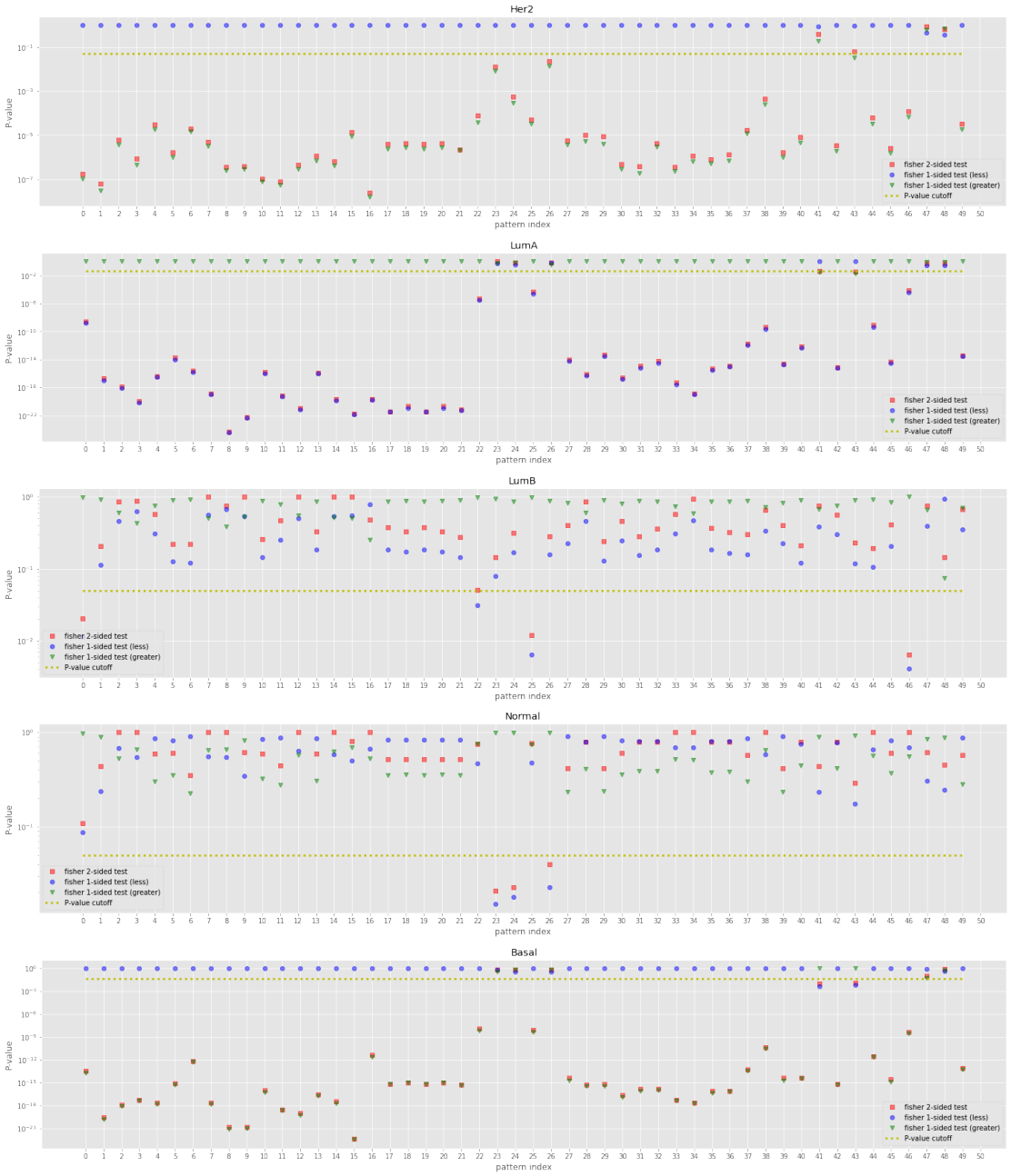

Figure 3: Fisher exact test (2-sided and 1-sided) to evaluate the quality of the detected patterns. Used to assess the statistical significance of the representation of a specific subtype in the resulting patterns. The smaller the p-value is the more significant is the (over: green or under: blue) representation.

N. Ilic, E. Kim, A. Kamburov, et al., Netsig: network-based discovery from cancer genomes, Nature methods 15 (2018) 61–66.

[3] P. Creixell, J. Reimand, S. Haider, G. Wu, T. Shibata, M. Vazquez, V. Mustonen, A. Gonzalez-

Perez, J. Pearson, C. Sander, et al., Pathway and network analysis of cancer genomes, Nature methods 12 (2015) 615.
[4] M. A. Reyna, D. Haan, M. Paczkowska, L. P. Verbeke, M. Vazquez, A. Kahraman, S. Pulido-Tamayo, J. Barenboim, L. Wadi, P. Dhingra, et al., Pathway and network analysis of more than 2500 whole cancer genomes, Nature communications 11 (2020) 1–17.
[5] J. Kim, J.-G. Lee, Community detection in multi-layer graphs: A survey, ACM SIGMOD Record 44 (2015) 37–48.
[6] C. Bothorel, J. D. Cruz, M. Magnani, B. Micenkova, Clustering attributed graphs: models, measures and methods, Network Science 3 (2015) 408–444.
[7] M. Atzmueller, S. Günnemann, A. Zimmermann, Mining communities and their descriptions on attributed graphs: a survey, Data Mining and Knowledge Discovery 35 (2021) 661–687.
[8] A. Bendimerad, Mining useful patterns in attributed graphs, Ph.D. thesis, Université de Lyon, 2019.
[9] P. Chunaev, Community detection in node-attributed social networks: a survey, Computer Science Review 37 (2020) 100286.
[10] L. Akoglu, H. Tong, D. Koutra, Graph based anomaly detection and description: a survey, Data mining and knowledge discovery 29 (2015) 626–688.
[11] A. Bendimerad, A. Mel, J. Lijffijt, M. Plantevit, C. Robardet, T. De Bie, Sias-miner: mining subjectively interesting attributed subgraphs, Data Mining and Knowledge Discovery 34 (2020) 355–393.
[12] T. De Bie, Maximum entropy models and subjective interestingness: an application to tiles in binary databases, Data Mining and Knowledge Discovery 23 (2011) 407–446.
[13] R. Bisiani, Beam search, Encyclopedia of Artficial Intelligence (1992) 1467–1468.
[14] M. Gillespie, B. Jassal, R. Stephan, M. Milacic, K. Rothfels, A. Senff-Ribeiro, J. Griss, C. Sevilla, L. Matthews, C. Gong, et al., The reactome pathway knowledgebase 2022, Nucleic acids research 50 (2022) D687–D692.
[15] J. S. Parker, M. Mullins, M. C. Cheang, S. Leung, D. Voduc, T. Vickery, S. Davies, C. Fauron, X. He, Z. Hu, et al., Supervised risk predictor of breast cancer based on intrinsic subtypes, Journal of clinical oncology 27 (2009) 1160.