An entropy heuristic to optimize decision diagrams for index-driven search in biological graph databases

Nicola Licheri
University of Turin
10149, Turin, Italy
nicola.licheri@unito.it

Elvio Amparore
University of Turin
10149, Turin Italy
elvio.amparore@unito.it

Vincenzo Bonnici
University of Verona
37134, Verona, Italy
vincenzo.bonnici@univr.it

Rosalba Giugno
University of Verona
37134, Verona, Italy
rosalba.giugno@univr.it

Marco Beccuti
University of Turin
10149, Turin Italy
marco.beccuti@unito.it

Abstract

Graphs are a widely used structure for knowledge representation. Their uses range from biochemical to biomedical applications and are recently involved in multi-omics analyses. A key computational task regarding graphs is the search of specific topologies contained in them. The task is known to be NP-complete, thus indexing techniques are applied for dealing with its complexity. In particular, techniques exploiting paths extracted from graphs have shown good performances in terms of time requirements, but they still suffer because of the relatively large size of the produced index. We applied decision diagrams (DDs) as index data structure showing a good reduction in the indexing size with respect to other approaches. Nevertheless, the size of a DD is dependent on its variable order. Because the search of an optimal order is an NP-complete task, variable order heuristics on DDs are applied by exploiting domain-specific information. Here, we propose a heuristic based on the information content of the labeled paths. Tests on well-studied biological benchmarks, which are an essential part of multi-omics graphs, show that the resultant size correlates with the information measure related to the paths and that the chosen order allows to effectively reduce the index size.

In: A. Editor, B. Coeditor (eds.): Proceedings of the XYZ Workshop, Location, Country, DD-MMM-YYYY, published at http://ceur-ws.org

1 Introduction

Graphs are mathematical objects used to represent items, also called vertices, and relations between them. In the bioinformatics context, they are exploited to express relationships at any biochemical, biological, and medical level. For example, graphs can represent physical molecule structure by expressing chemical bonds among atoms [Tri18]. At the cellular system level, graphs are instead applied to represent biological actors, such as genes, proteins, or RNAs, and their relations, such as physical interactions or causal inference [HPRL08, DL05, BDCC+18]. Differently, in medical applications, graphs are exploited in decision support systems to connect patient data with disease states and treatments [XWJF19]. For what concerns integration and analysis of multi-omics data, graphs are becoming popular for integrating biomedical information with data regarding multiple omics. In such a model, items compose a heterogeneous set of biological and meta-biological objects. Graphs of genetic interactions are enriched by embedding their relationship with diseases, drugs, anatomic phenotypes, biological functions, or cellular localization. Then, multi-omics data linked to the genetic actors are integrated. The result is a knowledge base that can be exploited for drug repurposing, for prioritizing disease-associated genes, or for patient classification and biomarker identification [HB15, FWY+21, WSH+21].

Among the computational tasks that can be performed on top of such structures, the search of specific topologies within biological graphs is one of the most challenging problems. In particular, the subgraph isomorphism problem is known to be NP-complete [Coo71]. In this context, indexing of labeled graphs is a widely used technique for dealing with such complexity. In fact, it provides a good compromise between precision in filtering unmatch-

ing parts of the graphs and time to compute such an operation [LVCF21, GBB+13]. Practically, indexing approaches store topological features, ranging from paths to frequent substructures, in order to provide a fast pruning of the labeled graphs or parts of them that do not contain the queried topology. However, this kind of approaches may lead to relatively large indexes, that can compromise performance.

Recently, Decision Diagrams (DDs) have successfully been applied for reducing the indexing size [LBBG21]. In fact, DDs are particularly efficient for detecting common portions among the paths and storing efficiently them. Their efficiency is known to be strongly affected by the ordering of variables describing a path: a good ordering can substantially reduce the memory consumption and the execution time to generate and encode the indexing. Unfortunately, discovering the optimal variable ordering is known to be NP-complete [BW96a]. Thus, various heuristics depending on the specific application field for the selection of (sub)optimal orderings were proposed in the literature [FFM93].

Starting from this observation, in this paper, we extend the results presented in [LBBG21] by investigating how the variable ordering may affect the performance of such an approach in terms of memory consumption. This task was carried out by first proposing a new metric called *Sum of Entropies* (*SOE*), which experimentally highlighted a medium-to-strong anti-correlation value with respect to the final size of the DD encoding the indexing. Then, the metric was exploited as starting point to derive a sub-optimal heuristic EntropyHeu that finds a variable order greedily by optimizing the *SOE* metric.

In detail, Section 2 introduces the concepts of indexing of labeled graphs and DDs. Moreover, the section recalls how DD can be efficiently exploited for encoding graph indexing and we discuss how its efficiency is strongly affected by the choice of a "reasonably good" *variable order*.

In Section 3, the new metric *SOE* and the derived new heuristic are formally introduced.

Then, the effectiveness of the new heuristic is assessed in Section 4 reporting its performance on a set of well-known biological benchmarks. Finally, Section 5 concludes the paper.

2 Background

In this section, we firstly introduce the concepts of graphs, labeled paths and path-based graph indexing. Then, the Multi-Terminal Multi-way Decision Diagrams (MTMDDs) are introduced as an efficient data structure to encode and manipulate a set of paths with their occurrences. Finally, we describe GRAPES-DD, a tool using MTMDDs for effective searching in graphs.

2.1 Graphs and paths as indexing features

Formally, a graph is a pair $G = (V, E)$ where V is the set of vertices and $E : V \times V$ is the set of relations, also called edges. Given a set of labels Σ, labeled graphs are enriched with a function $f_\sigma : \Sigma \mapsto V$ which maps each vertex to a label in Σ. The same label can be associated with different vertices. A path p of length l is a vector $(v_1^p, v_2^p, \ldots, v_l^p)$ such that $v_i^p \in V$, for $1 \leq i \leq l$, and $(v_i^p, v_{i+1}^p) \in E$, for $1 \leq i < l$. A labeled path $\hat{p}$ is obtained by mapping the vertices of a path to their corresponding labels via the f_σ function, thus $\hat{p} = (f_\sigma(v_1^p), f_\sigma(v_2^p), \ldots, f_\sigma(v_l^p)) = (\sigma_1^p, \sigma_2^p, \ldots, \sigma_l^p)$.

Given a query graph $G_Q = (V_Q, E_Q)$ and a target graph $G_T(V_T, E_T)$, the subgraph isomorphism problem consists in finding the occurrences of G_Q within G_T. An occurrence is a mapping $m : V_Q \mapsto V_T$, thus between the vertices of V_Q and the vertices in V_T, which preserves label compatibility and graph topology. Label compatibility ensures that, for each $v \in V_Q$, $f_\sigma(m(v)) = m(v)$. Topology compatibility is ensured by asserting that, for each $(u, v) \in E_Q$, $(m(u), m(v)) \in E_T$. The search space of the problem is at most $O({V_Q}^{V_T})$, because each possible combination of assignment of a target vertex to a query vertex must be explored and verified. However, several techniques can be used to reduce the search space by avoiding visiting the unfeasible parts of it. One of these techniques consists in extracting features of graph vertices for computing compatibility between target and query vertices. Features of a target graph are thus extracted and stored in an index with the aim of reusing it for multiple queries. A key property of indexes is the costs for building and querying them, as well as the size they require in memory [Din17].

In particular path-based indexing uses labeled paths as features that describe the topological neighborhood of a vertex. According to this, labeled paths are stored together with the identifier of their starting vertex [GS02].

In this way, a set of target vertices to be candidates to match to a given query vertex v can efficiently be retrieved. All labeled paths starting from v are extracted, then the set of target vertices that are starting points of the same labeled paths in the index are retrieved. After such a filtering phase,all the exact occurrences are retrieved by a subgraph isomorphism solver, such as VF2 [CFSV01] or RI [BGP+13]. Because the number of paths can exponentially grow on increasing their length, a maximum length is usually set for the paths stored.

2.2 Decision diagrams in a nutshell.

Decision diagrams (DDs) are a family of data structures proposed to encode and manipulate a set of values efficiently. Multi-Terminal Multi-way Decision Diagram (MTMDD) is a type of DD that can be effectively exploited to encode the function counting the occurrences of an el-

ement into a multiset[1], where elements are tuples with format $\langle v_1, \ldots, v_n \rangle$ with $v_i \in \mathcal{N}$. Let $O : \mathcal{N} \to [1 \ldots n]$ be a bijective *variable ordering* function that assigns a unique DD level in $[1 \ldots n]$ to each encoded variable. Formally, an MTMDD is a rooted directed acyclic graph ordered by O, where the first n levels represent the variables $\langle O(v_1), \ldots, O(v_n) \rangle$ of the encoded tuple, and the terminal level the number of occurrences of each tuple in the multiset. Let us count the levels of an MTMDD in a bottom-up fashion, so that the first level is above the terminal one and the root node is at n-th level.

The high storage efficiency of DDs is strongly conditioned by the choice of a "reasonably good" *variable order*, i.e. the assignment of the problem variables to the DD levels. It is known [BW96b] that finding the optimal order is a NP-complete problem. Some heuristics exist to help searching at least sub-optimal orders [FFM93], but these algorithms typically use problem-specific information. However, to the best of our knowledge, no such heuristic is currently available for reordering the variables of DDs encoding biological graph databases.

2.3 GRAPES-DD: a tool using Decision Diagrams for searching in graphs.

In [LBBG21] we proposed a new version of GRAPES, a path-based graph indexing tool [GBB+13], which exploits the decision diagram (i.e MTMDD) to achieve a substantial reduction of the memory footprint of the index graphs. The goal was reached thanks to DD ability to efficiently handle the presence of similar patterns in the indexed graph paths.

Roughly speaking the GRAPES-DD workflow is composed of three main phases: (1) the index building phase in which MTMDD indexing the set of target graphs is created by extracting all the labeled paths up to length l_p, (2) the filtering phase in which, given a query graph, the set of target graphs is restricted to those subgraphs potentially containing the query, and (3) the verification phase in which subgraph isomorphism algorithm (i.e. VF2 algorithm [CFSV01] or RI [BGP+13]) is applied only on the subset of candidate targets.

3 Methods

In this section, after introducing the GRAPES-DD strategy, we focus on the formal definition of a new sub-optimal heuristic inspired to decision tree learning in machine learning [Qui86].

GRAPES-DD exploits an MTMDD with n variables: $n-1$ variables v_i, $i = 1, \ldots, n-1$ encoding the paths' i-th label; and v_n encoding the identifier of the starting node of the labeled path. Note that v_n and the v_i, $1 \leq i < n$, are different because they belongs to different domain spaces

[1] Multiset extends the concept of a set allowing for multiple instances for each of its elements.

```
1:  function EntropyHeu
2:      O ← EntropyHeuLabels()
3:      minSize ← ∞
4:      for i ← [1 ... n] do
5:          O' ← InsertAt(O, {v_n ↦ i})
6:          DD' ← BuildDD(O')
7:          if SizeDD(DD') < minSize then
8:              minSize ← SizeDD(DD')
9:              O* ← O'
10:     return O*
11: function EntropyHeuLabels
12:     U ← {v_1 ... v_{n-1}}
13:     O ← {}
14:     for i from 1 to n − 1 do
15:         v_sel ← null
16:         H_sel ← −∞
17:         for v' ∈ U do:
18:             U' ← U \ {v'}
19:             H' ← Entropy(U')
20:             if H' > H_sel then
21:                 v_sel ← v'
22:                 H_sel ← H'
23:         U ← U \ {v_sel}
24:         O ← Append(O, {v_sel ↦ i})
25:     return O
```

Algorithm 1: Variable ordering selection heuristic

(graph vertices and labels, resp.) with largely different domain cardinalities. Given a fixed position $v_n \mapsto k$ for the v_n variable, we define the stratum k as the subset of variable orders $\{O\}_k$ sharing the fixed position for v_n. We shall see that the strata show significant clustering of the results in Section 4.

Each tuple $x = \langle v_1 \ldots v_n \rangle$ has an associated integer multiplicity $mult(x)$. Let X be the multiset of all the encoded tuples x. Given a multiset X, let $H(X)$ be the *entropy* of X, defined according to the standard definition [Sha01]

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x), \quad \text{with: } p(x) = \frac{mult(x)}{\sum_{x' \in X} mult(x')} \tag{1}$$

Let $\mathcal{U} \subseteq \mathcal{N}$ be a subset of the problem's variables. Let $x' = x/\mathcal{U}$ be a new tuple x' obtained from a tuple x by removing all the variables not in $\mathcal{U}$. Let $X/\mathcal{U}$ be the projection of the multiset X over the sole variables $\mathcal{U}$, with

$$mult(x') = \sum_{x \in X,\ x' = x/\mathcal{U}} mult(x) \tag{2}$$

the multiplicity of each tuple x'.

Given a variable order $O = \{k_1 \ldots k_n\}$, we define the i-th variable subset $\mathcal{U}_{O,i}$ as the set of the first i variable indices

of O. We define the *SOE* metric for a variable order O as

$$SOE(O) = \sum_{i=1}^{n} H(X/\mathcal{U}_{O,i}) \quad (3)$$

Research question R1: The size of the MTMDD (i.e. the sum of its nodes and edges) correlates with the *SOE* function. To test this hypothesis, we construct the MTMDD for all the variable orders (which is factorial in the number n of variables), and compute a correlation score between the value (3) and the final MTMDD size.

Unfortunately, finding the optimal MTMDD by constructing all the permutations is not feasible in practice, except for a limited number of encoded variables. Therefore, to make the technique broadly applicable in a real world context, we define a sub-optimal heuristic EntropyHeu that searches a variable order O^* by applying a greedy optimization the local entropy sum at every projection step i.

The pseudo-code of EntropyHeu is shown in Algorithm 1. The function EntropyHeu first computes the ordering for the $\{v_1 \dots v_{n-1}\}$ label variables. It then tries to insert the identifier variable v_n in all the positions, returning the order O^* that minimizes the final DD size. The function EntropyHeuLabels is the core heuristic algorithm, performing the greedy search. It starts by defining an empty variable order O and by taking into account the full set of label variables $\mathcal{U}$. At each outer iteration (lines 14-23), a variable $v_{\text{sel}} \in \mathcal{U}$ is removed and assigned to position i in the order O. The variable v_{sel} is chosen to be the one that maximizes the entropy given by the remaining set of variables $\mathcal{U} \setminus \{v_{\text{sel}}\}$, namely:

$$v_{sel} = \arg\max_{v \in U} H(U \setminus v) \quad (4)$$

We assume that BuildDD(O) generates the MTMDD for the projected variables subset with order O, and Entropy($\mathcal{U}$) computes (1) on the projected multiset $X/\mathcal{U}$.
Research question R2: The function EntropyHeu selects reasonably good variable orders, comparable with the theoretical-optimal order derived by the *SOE* metric.

In the next section, results are analysed to find answers to the research questions **R1** and **R2**.

4 Results

We empirically tested our two research questions **R1** and **R2** using a set of well-known biological benchmarks, described hereafter. The first 5 benchmarks are protein-protein interaction (PPI) networks of 5 different species: *Caenorhabditis elegants* (CE), *Drosophila melanogaster* (DROSOFILA), *Homo sapiens* (HOMO), *Mus musculus* (MUS) and *Saccaromyces cerevisae* (YEAST) [SFK+10]. Vertices are proteins and edges are predicted physical interactions between them. For each species, different thresholds on the accurateness of the prediction were applied, ranging from 0.4, 0.5, 0.6 to 0.7. The retrieved graphs have from 2k to 10k vertices, and from 2k to 89k edges, with average degrees ranging from 1.3 to 15. PPIs belonging to the same species were then merged into a single benchmark to be indexed. The obtained benchmarks have from 13k to 21k vertices, and from 39k to 260k edges.

We also included in the benchmark the standard database for *Antiviral Screen*(AIDS) [ci]. It consists of 40k chemical structures representing small molecules. Vertices are atoms and edge are the chemical bounds linking them. Vertex labels represent atomic elements, and there are a total of 62 distinct elements. The average number of vertices per graph is 44.98, and the average degree is 4.17.

We conducted a set of experiments over the graph databases described above. We indexed each database using labeled paths up to length 4, so that each index MTMDD is defined over 5 variables. Then, for each collection, we obtain the size of the index MTMDD for all possible variable orders.

Figure 1 reports the results for the **R1** question on the 6 benchmarks. Each dot represents one of the 120 possible variable orders. Dots are colored by their respective stratification induced by the level of the identifier variable. In each stratum, a dashed line represents the trend of the relation between the *SOE* metric and the final DD size. The number indicates the value Spearman's correlation coefficient. We can observe that the metric has medium-to-strong anticorrelation values in all stratum except for the one where the identifier is positioned at the bottom, whose sizes are almost insensible to the reordering of the label variables. The figure shows a very positive result, because it shows that a heuristic that maximizes the *SOE* metric has an high chance of selecting a good order that minimizes the DD size. Moreover, the cross on each stratum identifies the ordering that would be selected by the proposed heuristic EntropyHeu when fixing the position of the identifier variable.

Figure 2 shows the results for the **R2** question on the effectiveness of the EntropyHeu heuristic on the 6 benchmarks. Relative DD sizes are shown on the y-axis, while the x-axis has no meaning (it is only used for visualization purposes to separate the dots). The green cross identifies the relative DD size of the selected order, while the blue bar identifies the average size that would be obtained by taking an order randomly among the possible 120 orders. We can observe that the greedy heuristic that follows the metric *SOE* is actually capable of selecting almost-optimal orders in all the tested cases, showing the effectiveness of the proposed information-based strategy.

5 Discussion and conclusions

In this paper, we extended the approach proposed in [LBBG21] investigating how the MTMDD variable order may affect the performance of such an approach in

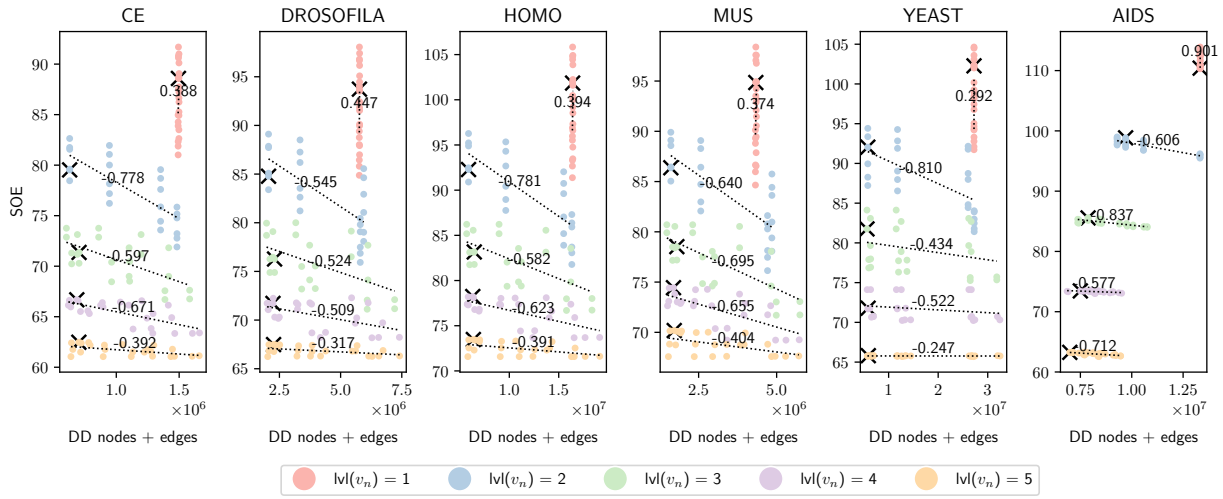

Figure 1: Spearman's correlation and trend lines of the *SOE* metric value w.r.t. the DD sizes, divided by sample strata. The black cross on each strata identifies the order that is selected by EntropyHeu.

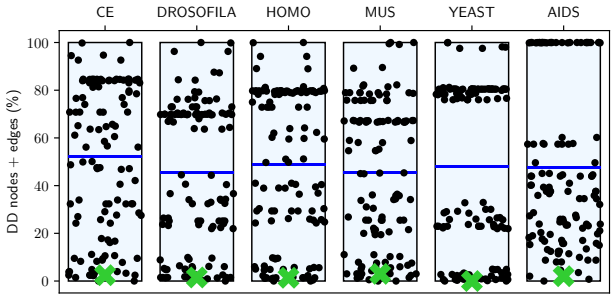

Figure 2: Relative position of the variable order selected by EntropyHeu among all the other possible orders.

terms of memory consumption. To achieve this task we first proposed the new metric *SOE* based on the Shannon entropy which experimentally showed a medium-to-strong anticorrelation with respect to the DD size encoding the graph indexing. Then we developed the sub-optimal heuristic EntropyHeu inspired to the information gain which is able to derive a variable order comparable with the theoretical-optimal order derived by the *SOE* metric. As a future extension, we will apply the EntropyHeu heuristic on a bigger set of benchmarks coming from different research fields and we will evaluate its performance by increasing the length of labeled paths.

Acknowledgements

This work is partially supported by "Creation of a computational framework to model and study West Nile Disease" project supported by "Fondazione CRT".

References

[BDCC+18] Vincenzo Bonnici, Giorgio De Caro, Giorgio Constantino, Sabino Liuni, Domenica D'Elia, Nicola Bombieri, Flavio Licciulli, and Rosalba Giugno. Arena-idb: a platform to build human non-coding rna interaction networks. *BMC bioinformatics*, 19(10):25–38, 2018.

[BGP+13] Vincenzo Bonnici, Rosalba Giugno, Alfredo Pulvirenti, Dennis Shasha, and Alfredo Ferro. A subgraph isomorphism algorithm and its application to biochemical data. *BMC bioinformatics*, 14(7):1–13, 2013.

[BW96a] Beate Bollig and Ingo Wegener. Improving the variable ordering of obdds is np-complete. *IEEE Trans. Computers*, 45(9):993–1002, 1996.

[BW96b] Beate Bollig and Ingo Wegener. Improving the variable ordering of OBDDs is NP-complete. *IEEE Trans. Comp.*, 45(9):993–1002, September 1996.

[CFSV01] Luigi Pietro Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. An improved algorithm for matching large graphs. In *3rd IAPR-TC15 workshop on graph-based representations in pattern recognition*, pages 149–159, 2001.

[ci] National cancer institute. National cancer institute. Accessed: 2021 september 21.

[Coo71] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the*

third annual ACM symposium on Theory of computing, pages 151–158, 1971.

[Din17] Hamed Dinari. A survey on graph queries processing: techniques and methods. *International Journal of Computer Network and Information Security*, 9(4):48, 2017.

[DL05] Eric Davidson and Michael Levin. Gene regulatory networks. *Proceedings of the National Academy of Sciences*, 102(14):4935–4935, 2005.

[FFM93] Masahiro Fujita, Hisanori Fujisawa, and Yusuke Matsunaga. Variable ordering algorithms for ordered binary decision diagrams and their evaluation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(1):6–12, 1993.

[FWY+21] Jiansong Fang, Qihui Wu, Fei Ye, Chuipu Cai, Lvjie Xu, Yong Gu, Qi Wang, Ai-lin Liu, Wenjie Tan, and Guan-hua Du. Network-based identification and experimental validation of drug candidates toward sars-cov-2 via targeting virus–host interactome. *Frontiers in Genetics*, 12:1590, 2021.

[GBB+13] Rosalba Giugno, Vincenzo Bonnici, Nicola Bombieri, Alfredo Pulvirenti, Alfredo Ferro, and Dennis Shasha. Grapes: A software for parallel searching on biological graphs targeting multi-core architectures. *PloS one*, 8(10), 2013.

[GS02] Rosalba Giugno and Dennis Shasha. Graphgrep: A fast and universal method for querying graphs. In *Object recognition supported by user interaction for service robots*, volume 2, pages 112–115. IEEE, 2002.

[HB15] Daniel S Himmelstein and Sergio E Baranzini. Heterogeneous network edge prediction: a data integration approach to prioritize disease-associated genes. *PLoS computational biology*, 11(7):e1004259, 2015.

[HPRL08] Luke Hakes, John W Pinney, David L Robertson, and Simon C Lovell. Protein-protein interaction networks and biology-what's the connection? *Nature biotechnology*, 26(1):69–72, 2008.

[LBBG21] Nicola Licheri, Vincenzo Bonnici, Marco Beccuti, and Rosalba Giugno. GRAPES-DD: exploiting decision diagrams for index-driven search in biological graph databases. *BMC bioinformatics*, 22(1):1–24, 2021.

[LVCF21] David Luaces, José RR Viqueira, José M Cotos, and Julián C Flores. Efficient access methods for very large distributed graph databases. *Information Sciences*, 573:65–81, 2021.

[Qui86] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[SFK+10] Damian Szklarczyk, Andrea Franceschini, Michael Kuhn, Milan Simonovic, Alexander Roth, Pablo Minguez, Tobias Doerks, Manuel Stark, Jean Muller, Peer Bork, et al. The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic acids research*, 39(suppl_1):D561–D568, 2010.

[Sha01] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[Tri18] Nenad Trinajstic. *Chemical graph theory*. Routledge, 2018.

[WSH+21] Tongxin Wang, Wei Shao, Zhi Huang, Haixu Tang, Jie Zhang, Zhengming Ding, and Kun Huang. Mogonet integrates multi-omics data using graph convolutional networks allowing patient classification and biomarker identification. *Nature Communications*, 12(1):1–13, 2021.

[XWJF19] Xiayu Xiang, Zhongru Wang, Yan Jia, and Binxing Fang. Knowledge graph-based clinical decision support system reasoning: a survey. In *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, pages 373–380. IEEE, 2019.