Performance Comparison of Five Exact Graph Matching Algorithms on Biological Databases

Vincenzo Carletti, Pasquale Foggia, and Mario Vento

DIEM - University of Salerno
{vcarletti,pfoggia,mvento}@unisa.it
http://mivia.unisa.it

Abstract. Graphs are a powerful data structure that can be applied to several problems in bioinformatics. Graph matching, in its diverse forms, is an important operation on graphs, involved when there is the need to compare two graphs or to find substructures into larger structures. Many graph matching algorithms exist, and their relative efficiency depends on the kinds of graphs they are applied to. In this paper we will consider some popular and freely available matching algorithms, and will experimentally compare them on graphs derived from bioinformatics applications, in order to help the researchers in this field to choose the right tool for the problem at hand.

Keywords: Graph matching, Benchmarking, Graph methods in Bioinformatics.

1 Introduction

Graphs are a powerful and flexible data structure that can be used for the representation of several kinds of data in many applicative fields [9] [5] [7]. Among these, bioinformatic applications play a very important role. For instance, molecular structures can be represented using a graph whose nodes corresponds to atoms and whose edges represent chemical bonds. The secondary structure of a protein can be represented by a graph with aminoacids as nodes, and edges used for encoding the contact s between adjacent and non adjacent nodes (e.g. hydrogen bonds); a similar representation can be used for other large biomolecules. Also, graph structures have been recently used to represent information describing networks of relations. For instance, the biologists are collecting huge quantities of data related to the interactions between the activities of different proteins in a given species. The resulting data are interaction networks, i.e. graphs whose nodes represent proteins, and whose edges represent interactions; by comparing the interaction networks of different species, it is possible to predict unknown interactions, or to discover functionally equivalent groups of proteins that perform a specific activity.

Given this wide number of cases where biological information is represented by means of (possibly large) graphs, it is important to have some graph-based methods and techniques to extract information from graphs. *Graph Matching*,

A. Petrosino, L. Maddalena, P. Pala (Eds.): ICIAP 2013 Workshops, LNCS 8158, pp. 409–417, 2013.

in its different meanings, plays a major role within such techniques. The simplest form of Graph Matching, *Graph Isomorphism*, can be used to determine if two graphs have the same structure, regardless the order of the nodes. *Graph-Subgraph Isomorphism* can be used to establish if a smaller graph is completely contained within a larger one as a subgraph, for example for searching all the graphs in a database that contain a certain desired substructure. *Monomorphism* is a weaker form of Graph-Subgraph Isomorphism, suitable when the desired substructure can be characterized by the presence of some relations, but not by their absence. Finally, the *Maximum Common Subgraph* (MCS) matching is aimed at finding common substructures within larger graphs. In the bioinformatics field, a substructure within a larger structure is often called a *motif*; *motif search* and *motif discovery* are considered problems of primary importance, since it is often a reasonable conjecture to assume that a motif common to different structures may have a perhaps undiscovered biological significance.

In the literature, hundreds of Graph Matching algorithms have been proposed in the last 40 years for the problems mentioned above. Still, none of them has proved to be the most effective in all situations. Different algorithms have different advantages and disadvantages, that can make them more or less suited for a particular kind of graphs. While in some cases a theoretical analysis is sufficient to exclude some algorithms, for the most commonly used ones there is an insufficient theoretical characterization of their computational cost. So, benchmarking is the only viable option to choose the right algorithm for a given problem; and this benchmarking has to be done using graphs with the same characteristics as the ones the will be encountered in the chosen application domain.

This paper aims at presenting the results of such a benchmarking activity using graphs from the bioinformatics research field. Of course, such an experimentation cannot reasonably aspire to be comprehensive and complete, given the large number of existing algorithms and of existing kinds of graph structures. However, it is our opinion that even within this limitations, this activity may provide useful information to the bioinformatics researchers looking for a graph matching algorithm, especially considering that we have included well known algorithms, for which stable and reliable implementations are freely available; such algorithms are very often the first choices for a researcher that views graph matching as a tool, and not as the goal of his research activity.

In the following, after a brief overview of the literature, we will present the chosen algorithms and the databases used for this experimentation. Then we will show and analyze the obtained results, especially the computation time, and draw some final conclusions.

1.1 Related Works

Several existing works [12] [14] [16] [1] point out the possibility of using graph based representations for bioinformatics data and problems; some of them specifically refer to particular kinds of graph matching, while others advocate a

plurality of techniques without focusing onto one in particular. In the following we will present some important papers, without any claim of comprehensiveness. Kuhl et al. [13] present one of the earliest applications of graph matching to bioinformatics, namely the MCS problem is used for the prediction of the ligaind-protein binding. Gifford et al. [10] apply graph matching to the prediction of biological activity of a molecule. The paper by Milo et al. [15] discusses the analysis of network motifs (i.e. subgraphs) within complex networks from genomics, proteomics and other fields of bioinformatics, besides other application domains. Tian et al. [19] propose the use of subgraph isomorphism for searching for protein complexes in protein-protein interaction networks. The paper by Aittokallio and Schwikowski [1] presents several graph-based techniques, including graph matching, that can be applied for the analysis of cell networks in cellular biology. In the literature several papers have already appeared comparing and contrasting several graph matching algorithms, mostly from a theoretical point of view, and without a reference to a particular application. In Conte et al. [4], a comprehensive survey of graph matching techniques used in Pattern Recognition is provided, presenting the different kinds of exact and inexact graph matching problems and discussing for each problem the most important approaches. While this paper does not cover methods published in the last few years, it remain an essential reference to understand the different approaches to the problem, and most of the well known and proven algorithms. The paper [16] by Raymond and Willet presents a more focused survey for MCS algorithms, with specific reference to their application to 2D and 3D molecular data. Bonnici et al. in a recent paper [3], besides presenting a subgraph isomorphism algorithm called RI, describe an experimental evaluation of several algorithms for the same problem using different databases of graphs obtained from chemical and biological applications.

This latter paper is the closest in its conception to the present article. While both the papers share the focus on the importance of a quantitative evaluation of graph matching on graphs directly derived from bioinformatics applications, the present article considers a broader range of matching problems, including isomorphism and MCS, and as a consequence different algorithms have been used for the experimental comparison.

2 Benchmarking

2.1 The Compared Algorithms

In this work we compare five graph matching algorithms on four different biological databases, focusing our attention on three kinds of matching problems [4]: Graph Isomorphism, Subgraph Isomorphism and MCS. The five considered algorithms are not applicable to all the three problems; usually each algorithm is developed for a single problem and may be extended to a second similar problem with few changes. So, in the experiments we will not present all the 15

combinations between the five algorithms and the three problems; however we have ensured that for each problem there are at least two different algorithms to be compared.

Thus, in our experimentation we have used VF2 [6], Ullmann [20], RI [3] and LAD [18] for Graph Isomorphism; VF2, Ullmann, RI, LAD for Subgraph Isomorphism; and VF2 and DPC2 [8] for MCS. Of these algorithms, Ulmann, DPC2 and VF2 are widely known and widely used algorithms, for which stable implementations are available, while RI and LAD are newer entries, that should be representative of the most recent ideas on exact graph matching. In the following a short overview of each algorithm is provided.

VF2. The VF2 algorithm by Cordella et al. [6] uses a state space representation and is based on a depth-first strategy with a set of rules to efficiently prune the search tree. The algorithm can be used both for graph isomorphism, subgraph isomorphism and MCS selecting a suiteble set of rules for the cosidered problem. VF2 is an extension of the previous VF matching algorithm, and is characterized by the use of suitably designed data structures for reducing the amount of information that has to be replicated when passing from a state to its descendants, so reducing both the memory occupation and the computation time.

Ullmann. The Ullmann algorithm [20] is among the most commonly used algorithms for exact graph matching [4]. It is based on a depth-first search like VF2, and use a complex, iterative heuristic for pruning the search space, that significantly reduces the number of visited states, at the expense of the memory and time spent for each state. This algorithm can be used both for Graph Isomorphism and for Subgraph Isomorphism.

DPC2. The DPC2 algorithm by Durand and Pasari [8] uses a well known theoretical result that connects the MCS problem to the search of the Maximum Clique (i.e. complete subgraph) of a suitably defined *association graph* [2]. DPC2 solves the Maximum Clicque problem with a state-space representation, adopting a depth-first search algorithm with a heuristic criterion to prune some unfruitful search paths. This algorithm, by its definition, can only be used for the MCS problem.

LAD. The LAD algorithm [18] by C. Solnon is also based on a state space representation with depth-first search, and uses a formulation of the matching as a Constraint Satisfaction Problem (CSP) to derive a criterion for pruning the search space. Namely, LAD uses the constraint that the mapping between the nodes of the two graphs must be injective and edge-preserving, and after each node assignment it propagates iteratively this constraint to unmatched nodes until convergence. Thus the LAD algorithm can be used both for Graph Isomorphism and for Subgraph Isomorphism.

RI. Like VF2, also the RI algorithm [3] by Bonnici et al. during the matching the algorithm performs heuristic checks that are very fast to execute, thus requiring a short time for each visited search state. However, a preprocessing is performed, resulting in a reordering of the nodes of each graph; this reordering is aimed at ensuring that the nodes that involve more constraints are matched first,

so as to reduce as early as possible the search space by means of those contraints. RI can be used for both Graph Isomorphism and Subgraph Isomorphism.

2.2 Test Databases

The benchmarks have been performed using graphs obtained by several databases from bioinformatics applications, namely molecular and protein databases. The choice of the databases is based on the paper by Bonnici et al. [3], whose authors provide six databases already converted into a common graph format.

The databases have been structured for the monomorphism problem, so each one is composed of very large graphs, called target, from which have been extracted a set of not induced subgraphs, called pattern, and so grouped by three densities: 1, 0.5 and 0.25. The aim of the authors is to use pattern graphs as queries on the target graph. We do not use all the six databases, but the following tree:

AIDS. This is molecular database containing 40000 graphs, from 4 to 256 nodes, representing the topological structure of a chemical compound tested for evidence of anti-HIV activity.

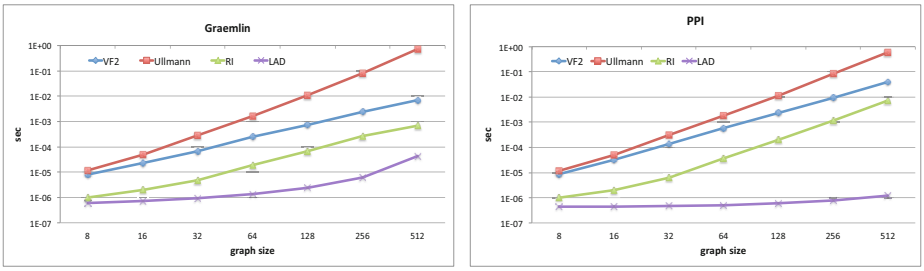
Graemlin. The Graemlin database contains 10 target graphs having up to 6726 nodes, extracted from 10 microbial networks.

PPI. The Protein-Protein interaction networks contains 10 network graphs, from 5720 to 12575 nodes, describing known protein interactions of 10 organisms. This is composed of more dense and big graphs than those in Graemlin.

The Graemlin and PPI databases have nodes with no labels. Since graph matching algorithms usually employ label information to reduce the computational costs, the matching times on unlabeled graphs are very long. For this reason, following [3], we have attached labels to the nodes of the graphs. The labels have been generated using random integer values. We have generated 5 labeled version of each database, varying the number of possible label values (128, 256, 512, 1024 and 2048 values have been used). For graph isomorphism we have generated two new databases, extracting random node-induced subgraphs from the target graphs in Graemlin and PPI; for each graph, 10 random permutations of the nodes have been computed to obtain isomorphic pairs of graphs. In order to test isomorphism algorithms on dense graphs, we have also added extra edges at random. So finally we have two isomorphism databases having 700 pairs, varying from 8 to 512 nodes.

3 Experimental Results

Experiments have been run employing of the implementations provided into VFlib for VF2, Ullmann and DPC2 algorithms, while for RI and LAD we used the code distributed by the respective authors [11] [17]. The experimental results



Medium Density Graphs High Density Graphs

Fig. 1. The matching time, in logarithmic scale, of VF2, Ullmann, RI, LAD, algorithms on Isomorfism Databases, as a function of graph size

have been obtained on an Intel i3-2100 3.1 Ghz, with 2 cores and 3 Mb cache, equipped with 4 GB of RAM and running a Linux OS with kernel 3.2.0. We have not used multithreading to avoid caching problems. Moreover, for isomorphism and MCS we set a 30 minutes time out.

Graph Isomorphism. In this experimentation we have used the VF2, Ullmann, RI and LAD algorithms on two databases, with distinct densities, generated as described in 2.2. Fig. 1 shows the results on this problem. As it can be seen, RI perfoms better than VF2 and Ullman, altough the difference with VF2 seems to be a constant, and so these two algorithms have a very similar asymptotic behavior. However, on isomorphism the best algorithms seems to be LAD, especially on very dense graphs. The reason may be due the CSP approach used by LAD, that makes a better pruning of unfruitful matches when nodes have several constraints.

Induced Subgraph Isomorphism. The experiments on induced subgraph isomorphism involved the VF2, Ullmann and RI algorithms on the Graemlin and PPI databases. Tests have been run for five different uniform node label distributions, from 128 to 2048 values, but we only show a reduced set of results. RI always outperfoms both VF2 and Ullmann, as shown in Fig. 2. The distance, in execution time, between RI and VF2 seems to be the same for all considered cases, irrespective of label distribution or target graph size and density.

MCS. MCS experimentation only involved VF2 and DPC2 algorithms, because the others do not deal with this problem. Tests have been run using AIDS database, by choosing random pairs of graphs and computing their MCS. As shown in Fig. 3 VF2 outperforms DPC2, even though there is significant variance in the matching times. This is caused by the fact that the graphs in the database have all different number of nodes, and so the resulting times are not mediated over a large number of cases.

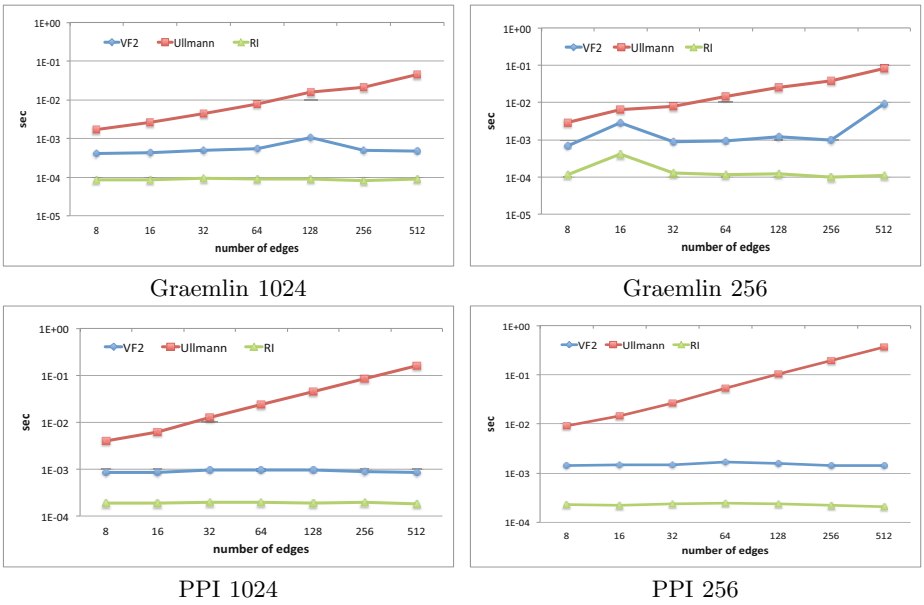


Fig. 2. The matching time, in logarithmic scale, of VF2, Ullmann, RI algorithms on Graemlin ad PPI Databases, as a function of query graph edges

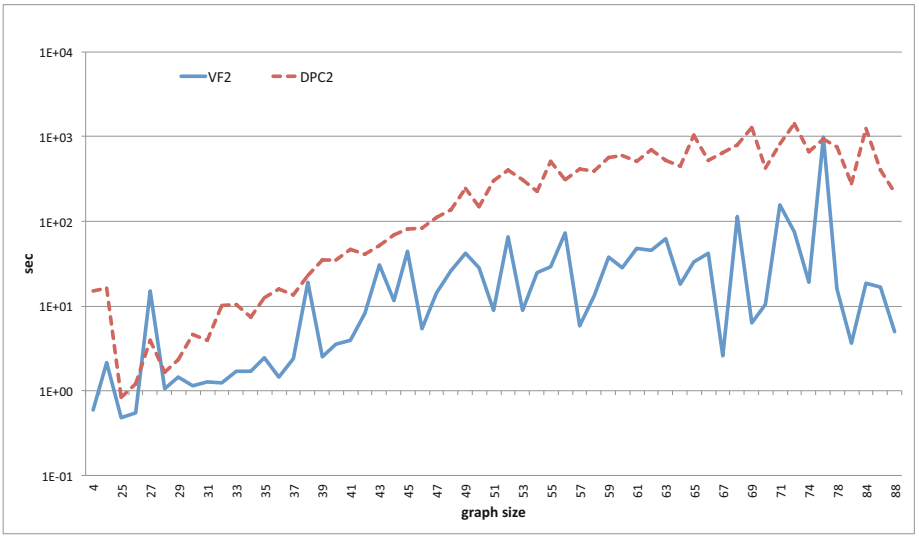


Fig. 3. The matching time, in logarithmic scale, of VF2, DPC2 algorithms on AIDS Dataset, as a function of graph size

4 Conclusions

In this paper we have proposed a preliminary banchmark aimed at presenting a first performance evaluation beetween some widely used exact graph matching algoritms, like VF2 and Ullmann, and some recently introduced ones, RI and LAD, on graphs from bioinformatics databases. Even though a more extended experimentation is needed, some conclusions can already be drawn. RI seems to be currently the best algorithm for subgraph isomorphism, but for dense graphs the CSP approach makes LAD to perform better than RI on isomorphism. VF2 is ten years older than the latter two, but the performance gap is not so wide; furthermore, VF2 is more general than LAD and RI, since its feasibilty rules can be efficiently adapted to resolve other exact matching problems, as we show in the MCS experimentation. Future work will involve an extension of this benchmarking, considering both other algorithms and other bioinformatics-related graph databases.

References

1. Aittokallio, T., Schwikowski, B.: Graph-based methods for analysing networks in cell biology. Briefings in Bioinformatics 7(3), 243–255 (2006)
2. Bomze, M., Budinich, M., Pardalos, M., Pelillo, M.: The maximum clique problem. Handbook of Combinatorial Optimization 4 (1999)
3. Bonnici, V., Giugno, R., Pulvirenti, A., Shasha, D., Ferro, A.: A subgraph isomorphism algorithm and its application to biochemical data. BMC Bioinformatics 14 (2013)
4. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in Pattern Recognition. IJPRAI 18(3), 265–298 (2004)
5. Conte, D., Foggia, P., Sansone, C., Vento, M.: How and why pattern recognition and computer vision applications use graphs. In: Kandel, A., Bunke, H., Last, M. (eds.) Applied Graph Theory in Computer Vision and Pattern Recognition. SCI, vol. 52, pp. 85–135. Springer, Heidelberg (2007)
6. Cordella, L., Foggia, P., Sansone, C., Vento, M.: A (sub)graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 1367–1372 (2004)
7. De Santo, M., Foggia, P., Percannella, G., Sansone, C., Vento, M.: An unsupervised algorithm for anchor shot detection. In: Proceedings - International Conference on Pattern Recognition, vol. 2, pp. 1238–1241 (2006)
8. Durand, P., Pasari, R., Baker, J., Tsai, C.C.: An efficient algorithm for similarity analysis of molecules. Internet Journal of Chemistry 2 (1999)
9. Foggia, P., Percannella, G., Sansone, C., Vento, M.: A graph-based algorithm for cluster detection. International Journal of Pattern Recognition and Artificial Intelligence 22, 843–860 (2008)
10. Gifford, E., Johnson, M., Smith, D., Tsai, C.C.: Structure-reactivity maps as a tool for visualizing xenobiotic structure-reactivity relationships. Network Science 2, 1–33 (1996)
11. Giugno, R.: Ri website, `http://ferrolab.dmi.unict.it/ri/ri.html`
12. Huan, J., et al.: Comparing graph representations of protein structure for mining family-specific residue-based packing motif. Journal of Computational Biology (2005)

13. Kuhl, F.S., Crippen, G.M., Friesen, D.K.: A combinatorial algorithm for calculating ligand binding. Journal of Computational Chemistry 5(1), 24–34 (1984)
14. Lacroix, V., Fernandez, C., Sagot, M.: Motif search in graphs: Application to metabolic networks. Transactions on Computational Biology and Bioinformatics (December 2006)
15. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: Simple building blocks of complex networks. Science 298(5594), 824–827 (2002)
16. Raymond, J., Willett, P.: Maximum common subgraph isomorphism algorithms for the matching of chemical structures. Journal of Computer-Aided Molecular Design 16(7), 521–533 (2002)
17. Solnon, C.: Lad website, `http://liris.cnrs.fr/csolnon/LAD.html`
18. Solnon, C.: Alldifferent-based filtering for subgraph isomorphism. Artificial Intelligence 174(12-13), 850–864 (2010)
19. Tian, Y., McEachin, R.C., Santos, C., States, D.J., Patel, J.M.: Saga: A subgraph matching tool for biological graphs. Bioinformatics 23(2), 232–239 (2007)
20. Ullman, J.R.: An algorithm for subgraph isomorphism. J. Assoc. Comput. Mach. 23, 31–42 (1976)