Photo(Graph) Gallery: An " exhibition " of graph classification

Frédéric Rayar, Zeina Abu-Aisheh

Photo(Graph) Gallery: An "exhibition" of graph classification

Frédéric Rayar*, Zeina Abu-Aisheh*
*Université François Rabelais, LI EA-6300, Tours, France
frederic.rayar@univ-tours.fr, zeina.abu-aisheh@univ-tours.fr

***Abstract*—This paper is an exhibition of graph matching results, in a classification context. We present Photo(Graph) Gallery, a platform that allows one to visually interpret graph matchings. We aim at understanding the computed matchings in order to improve the rates of graph classification. Preliminary results of the study performed on two data sets are also illustrated. Furthermore, a demonstrator of our proof-of-concept platform is available online at http://rfai.li.univ-tours.fr/PublicData/phogg/.**

I. Why stopping by this exhibition?

Graphs have been extensively used to represent objects in the Pattern Recognition field [1]. Once a graph representation has been adopted to describe objects, the question of comparing graphs arises. This issue gave birth to the Graph Matching (GM) area and has been addressed by many researchers in the past half century. Among GM paradigms, Graph Edit Distance (GED) [2] gains more and more importance from researchers. Its main idea is to find the *best* sequence of edit operations that allows to transform graph G_1 into graph G_2. The edit operations can be either the insertion, the deletion or the substitution of vertices and their corresponding edges.

Most of the studies that take benefit of GED in the context of graph classification evaluate their contributions using the classification rate obtained on given data sets. However, as mentioned in [3]: "*the matching quality is evaluated indirectly through the recognition rate, which highly depends on the classifier and does not allow a clear analysis of the matching algorithms*". The authors of the latter study have proposed a first attempt to address this issue by providing graph low level information for evaluating GED algorithms. However, it was not dedicated to the analysis of graph classification.

Can we go even further in the observation and evaluation of GED algorithms, in order to improve the classification rate? How to put into question the choice of the cost of edit operations and their parameters? One path, that is currently being studied, is to rely on existing optimization methods to automatically learn these values [4]. In this paper, we propose to investigate another path through an exploratory study that combines Graph Drawing and Information Visualisation techniques to put into perspective the results of GED methods. More specifically, we allow one to visualise the results of graph classification: namely the misclassified graphs and the effective matching that is found by a given GED algorithm. To do so, we present "*Photo(Graph) Gallery*" (aka. Phogg), a proof-of-concept light web platform that allows one to study the aforementioned results. A demonstrator is available online at http://rfai.li.univ-tours.fr/PublicData/phogg/.

No need for tickets: this is a free exhibition where we will run into several topics. Some related works are presented in Hall II and the Phogg platform is described in Hall III. Initial results of this early-stage work are presented in Hall IV. Once you arrive to the exit, do not forget to stop at the gift shop for free souvenirs. Note that the main idea of this exhibition is have a first demonstration of our platform, share fruitful discussions and get useful feedback from visitors.

II. History Hall

A. Graph Edit Distance for Graph Classification

Let G_1 and G_2 be attributed graphs where attributes can be numeric and/or symbolic and can exist on vertices and/or edges. The basic idea of GED is to assign a penalty cost to each matching operation according to the amount of distortion that it introduces in the transformation of G_1 into G_2 [2]. When (sub)graphs differ in their attributes or structures, a high penalty cost is added during the matching process. Such a cost prevents dissimilar (sub)graphs from being matched since they are different. Likewise, when (sub)graphs are similar, a small penalty cost is added to the overall cost. This cost includes substituting, inserting and/or deleting a vertex/edge.

Formally saying, GED is based on a set of edit operations ed_i where $i = 1 \dots k$ and k is the number of edit operations. The GED between two graphs is defined as follows.

Definition 1. *Graph Edit Distance*
Let G_1 and G_2 be two attributed graphs, the graph edit distance between G_1 and G_2 is defined as:

$$d_{\lambda_{min}}(G_1, G_2) = \min_{\lambda \in \gamma(G_1,G_2)} \sum_{ed_i \in \lambda} c(ed_i) \quad (1)$$

where $c(ed_i)$ denotes the cost function measuring the strength of an edit operation ed_i and $\gamma(G_1, G_2)$ denotes the set of all edit paths transforming G_1 into G_2. The best correspondence, λ_{min}, is one of the correspondences that obtains the minimum cost (i.e. $d_{\lambda_{min}}(G_1, G_2)$).

Few optimal approaches have been proposed to solve GED due to its exponential complexity. On this basis, researchers shed light on suboptimal GED methods. These methods often have a polynomial running time in the size of the input graphs and are much faster than the optimal ones. Thus, they fit for classification. For an extensive study of GED methods and GM methods in general, we refer the interested reader to [5], [1]. Recently, a new GM category, called "anytime GM", has been put forward in the literature [6]. Anytime GM methods allow a trade-off between valuable properties of the both previously existing types of GM methods: speed for suboptimal methods and quality of the provided solution for optimal ones. For this exhibition, we have used *ADF*, the first anytime GM algorithm proposed in [6].

Most of the studies that evaluated graph classification use publicly available graph repositories [7]. However, they did not provide any clear analysis of the GM results. Recently, a new GM repository [3] has been proposed to evaluate the quality and scalability of the GM algorithms. Yet, this latter work was not dedicated to the classification analysis. In this study, we aim at evaluating the results of a given GM algorithm in a classification context.

B. Graph Drawing

One can wonder about the relevance of using visualisation to extract knowledge from data. To what extent visualisation is useful for getting insight from data? As mentioned in Card's book [8]: *The purpose of visualisation is insight, not pictures. The main goal of this insight are discovery, decision making and explanation.* Thus, by definition, Information Visualisation is a useful support in the process of having insight from data. Other studies, in the psychological field, back up this statement with the advantages of the visual perception and the cognitive benefits.

Graph Drawing (GD) is a field that addresses the issue of visual depiction of graphs in two (or three) dimensional surfaces. To do so, it takes benefit of Graph Theory and Information Visualisation fields. A common way to represent graphs is node-link diagrams. In such depictions, vertices of the graph are represented as disks, boxes, or textual labels. The edges are represented as segments or curves in the plane. In this study, we do not thoroughly present the state of the art of GD algorithms. Excellent surveys on this topic can be found in Di Battista et al.'s book [9] and Tamassia's one [10]. Nonetheless, we briefly describe below the main ideas of force-directed algorithms since one of them will be used in our work.

Force-directed graph drawing algorithms consider physical metaphors: vertices are considered as physical entities, and forces are assigned to pairs of vertices. These forces are either attractive in the case of pairs of vertices linked by an edge, or repulsive for pairs that are not linked by an edge in the graph. The main idea is then to place the vertices at a random position, and let the system find an equilibrium state. Figure 1 illustrates such a process. Several metaphors can been considered, in particular, mechanical, electrical and magnetic systems and laws. The main advantages of such algorithms is that, as mentioned in [11], they *"tend to be aesthetically pleasing, exhibit symmetries and tend to produce crossing-free layout for planar graphs"*. In our work, we have used the Fruchterman-Reingold algorithm [12].

III. PHOGG PLATFORM HALL

Welcome to the second hall in which "*Photo(Graph) Gallery*" is demonstrated. This platform allows users to visualise and interact with the results of graph classification. More specifically, it allows users to visualise the effective matching that has been done between a graph G_1 to be classified and its nearest graph G_2, in term of GED. Note that Phogg has been developed as a proof-of-concept platform, and not an end product. The latest version of the platform is available online at http://rfai.li.univ-tours.fr/PublicData/phogg/.

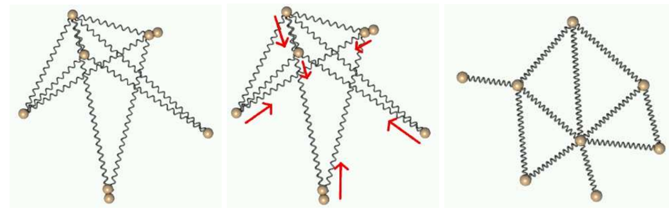

Fig. 1: Force-directed algorithms. Illustration from [13] of a generic spring metaphor. Starting from randomly placed vertices, the graph is treated as a spring system and the objective is to look for a stable configuration.

A. Description

Figure 2 shows the platform interface. It consists of two parts: *(i)* the list of matched graphs pair, and *(ii)* the matching visualisation for a given pair of graphs.

The proposed visual analysis system has been realised using web technologies. This choice is explained as follows:

1) the platform targets experts from various fields such as computer science, health or digital humanities. Nowadays, a majority of users that are not experts in computer science can still manage well web navigation. Thus, we think that presenting the system as a light web platform would allow users to be more disposed to use and take advantage of it.
2) the choice of discarding a server can be justified by two arguments: *(i)* there is no upload of the images to a server since such an operation may take time and *(ii)* as the images are not sent nor stored in an external server, we respect the potential confidentiality or license issues that are related to data. Hence, we allow one to perform an offline study.

Hence, we have used HTML5, CSS3 and JavaScript web technologies to implement our platform. Regarding the JavaScript part, we have taken advantage of the AngularJS [1] framework. Along with this framework, we have used the Linkurious [2] library to visualise graphs in web browsers. All these libraries are open-source and have either a GPL or MIT license.

Regarding the interactions, a few functionalities have been implemented in Phogg. First, common interactions such as zoom and pan are available to allow users to manipulate the graph and select a region of interest. Second, labels on both the vertices and edges can be displayed when hovering the elements on the graph canvas. One can also select and highlight a specific pair of matched vertices. Third, if users do not want to have a visual notification of the matching edges, they can hide or show them on demand.

We also advocate for the fact that, when available, the images represented by the graphs should be used and displayed to give more visual support to users. To this end, in the left panel, misclassified graph images are displayed.

[1] https://angularjs.org/
[2] http://linkurio.us/

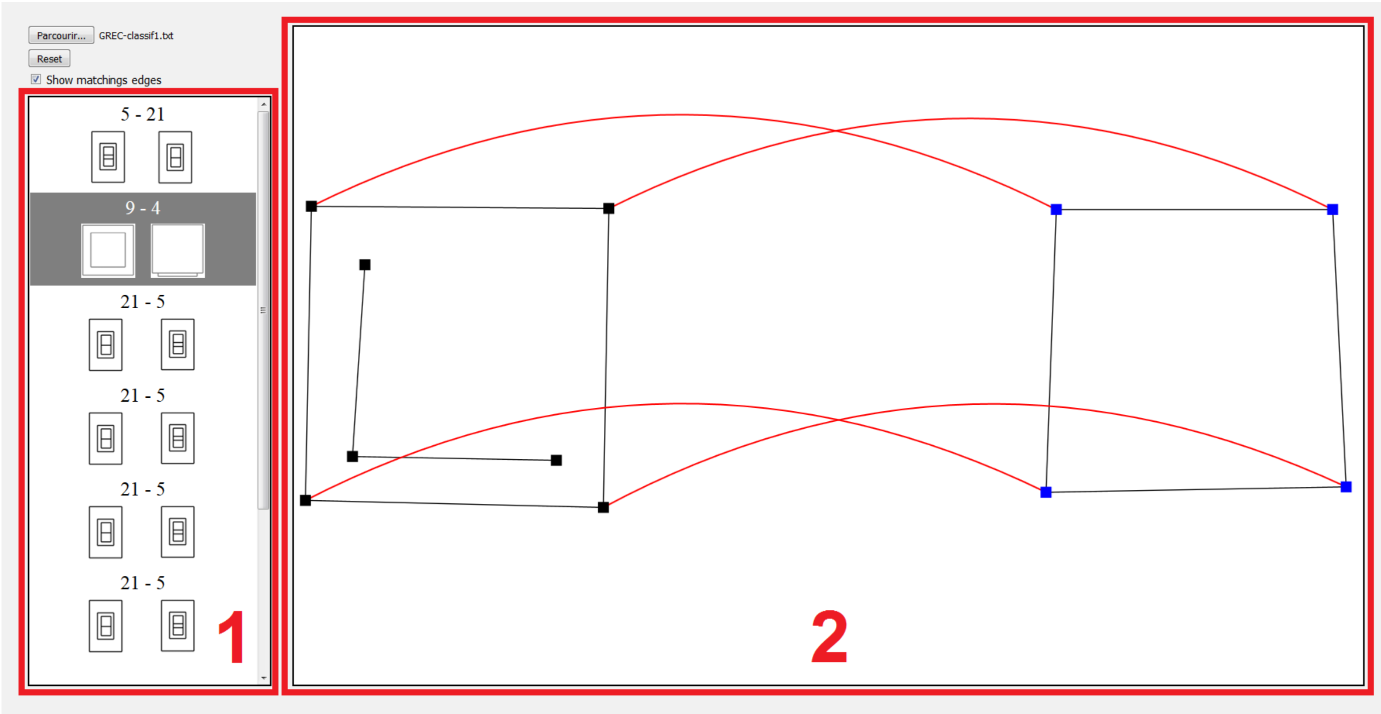

Fig. 2: Screenshot of the proposed Phogg interface. (1) corresponds to the list of misclassified graphs. (2) corresponds to the graph visualisation canvas, where the matchings between two graphs are displayed. The image (respectively graph) on the right is the one to be classified and the one the left is the nearest neighbour in the training database.

B. Discussion

The proposed platform has several advantages: First, it can be used for various data that can be represented by a planar graph, such as images, protein, *etc*. Second, one can actually visualise the graphs under study. This argument might seem naive, but reveals itself quite useful for making observations and having insights. Third, the matchings that are computed by a GM algorithm can be visualised and one can put into perspective and question the obtained results. Last, the fact that users can visualise and interact with the classification results and the graph matchings put themselves "*into the loop*" and make them pro-active in his study.

However, as aforementioned, the Phogg platform has been implemented as a proof-of-concept platform, therefore we can mention a few limitations. First, regarding the visualisation: force-directed algorithms produce aesthetically drawings of graphs, but this stands for relatively small graphs and such algorithms perform poorly for graphs with several hundreds of vertices. Using such algorithms, one can also expect that edges will tend to have a uniform length, and that vertices that are not connected by an edge tend to be placed further apart (because of the repulsion forces). Moreover, these algorithms are not perfect and some issues such as node overlap or edge crossing can be found in the computed layouts. Second, regarding the interactions: at the moment, only a limited set is proposed. A brainstorming on relevant tasks and the interactions that can be used to perform these tasks, must be conducted.

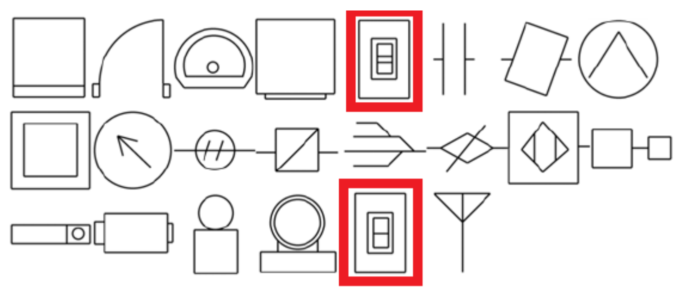

Fig. 3: Representative images of the 22 GREC classes. Classes 5 and 21 are highlighted with a surrounded box.

IV. Use Cases Hall

Wondering about the relevance of Phogg? This is normal since it is not easy to understand it without making Phogg face up to some data sets. In this hall, we present the results of *ADF* (see Section II-A) on several graph data sets. *ADF* is used to compare graphs under a time constraint of 400 milliseconds. Moreover, in our experiments, we have used the 1-nearest neighbour classifier.

A. GREC

The GREC data set consists of a subset of the symbol database underlying the GREC 2005 competition [3]. It is composed of undirected graphs of rather small size (i.e. up

[3] http://symbcontestgrec05.loria.fr/index.php

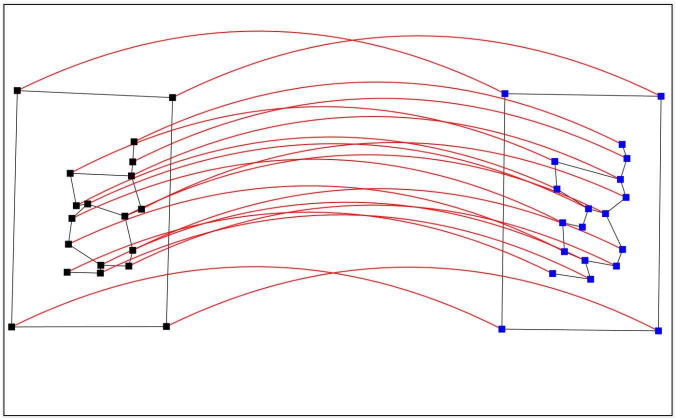

Fig. 4: Illustration of the matching for one misclassified (21,5) pair in GREC data set.

to 24 vertices), that belong to one of the 22 classes. Figure 3 illustrates the integrated classes. In this experiment, 286 graphs have been used in the training database to predict 528 graphs. The obtained classification rate is 98.5%, which corresponds to 8 misclassified graphs *(real_class-predicted_class)*: one (5-21), one (9-4) and 6 (21-5). Figure 4 illustrates the matching obtained from a misclassified (21-5) pair. In this example, one can see that the misclassification is not due to GED computation, but rather to graph representation. This could be explained by the fact that in this competition, some images were distorted: the computed graph representation does not perfectly match the prototype's graph. Hence, for future work that aims at increasing the classification rate, we can advise no to tune the graph matching algorithm (either its choice or its parameter), but rather to improve the graph extraction step.

B. Protein

The Protein [14] data set contains graphs representing proteins. They have small and medium size graphs (up to 126 vertices), that belong to 6 classes. We have used 200 graphs for training and 200 for the prediction. Using *ADF*, we have achieved a classification rate of 52%, which corresponds to 96 misclassified graphs. Using Phogg, we have visualised the misclassified graphs. Figure 5 illustrates these results. One immediate observation is that a lot of graphs (68 out of 96) have been wrongly assigned to class 1. More precisely, their nearest neighbour, in the sense of the GED implementation, was the "*enzyme_11*". This specific protein corresponds to a 4-vertex clique. In terms of topology, these results are quite strange: it seems that often a "*reverse-subgraph matching*" was performed. Hence, one can wonder: was the deletion operation often selected over the substitutions one?

Following this observation, we modified the cost function associated to the Protein data set to avoid this phenomenon, and ran the classification test again. This time, we obtained a slightly lower classification rate of 50.5% and 99 misclassified graphs. Thanks to this modification, almost 20% of the previously misclassified graphs are now correctly classified. Figure 6 shows the nearest neighbour of a given protein graph before and after the modification.

However, previously correctly classified graphs are now misclassified, hence the slightly lower classification rate. Figure 7 illustrates these new results. One can see that, even if these graphs are wrongly classified, their nearest neighbour is a graph with the same topology, at least in terms of number of vertices. Obviously, this modification on the cost function was "naive": we only focused on the topology and pushed into the background the attributes of the vertices. However, this basic scenario is promising. Indeed, using our visual and interactive platform, one can make observations, put into perspective the parameters choice and hopefully improve the graph classification rate. Since we did not have much time to "prepare this exhibition", in a foreseeable future, we plan to study the protein data set more using Phogg in order to improve the classification rate.

V. Souvenir and Gift Shop

Let us have a look back at our visit. We saw an exploratory study, at the crossroad of Information Visualisation, Graph Matching and Machine Learning fields. We came across Photo(Graph) Gallery, a web platform to visualise, interact and evaluate GM results in the context of graph classification. Early-stage results, presented in the paper, allowed us to foresee promising directions, such as the understanding of classification rates in graph classification.

In the visitor's book, we are welcoming any comments, suggestions and critics regarding the proposed work and its directions.

As for the gift shop, one can be tempted by *(i)* the online demonstrator of Phogg and its open-source code, *(ii)* the code of the Fruchterman-Reingold graph drawing algorithm or *(iii)* the laid out graphs (both in **.gxl* and **.png* format) of the data sets that have been studied in this work. In the near future, we also plan to share a simple web platform and its code, that will allow users to load, visualise and save their own graphs.

Please note in your agenda that in the near future, another exhibition will take place, where improvements will be done on the Phogg platform (e.g. top k-nearest neighbours display and new interactions). Issues such as the embedding of user relevance feedback will also be addressed. Hence, one will find results on research direction towards "*Visual Analytics*", where automated analysis could be done between and along with the user's interactions.

Acknowledgment

The authors would like to give credits to Prof. Mario Vento and his trip in the charming world of graphs for Pattern Recognition [1], which inspired us the exhibition metaphor of this paper.

References

[1] M. Vento, "A long trip in the charming world of graphs for pattern recognition," *Pattern Recognition*, vol. 48, no. 2, pp. 291–301, 2015.

[2] H. Bunke and G. Allermann., "Inexact graph matching for structural pattern recognition," *Pattern Recognition Letters*, vol. 1, no. 4, pp. 245–253, 1983.

[3] Z. Abu-Aisheh, R. Raveaux, and J. Ramel, "A graph database repository and performance evaluation metrics for graph edit distance," in *GBR'2015*, 2015, pp. 138–147.

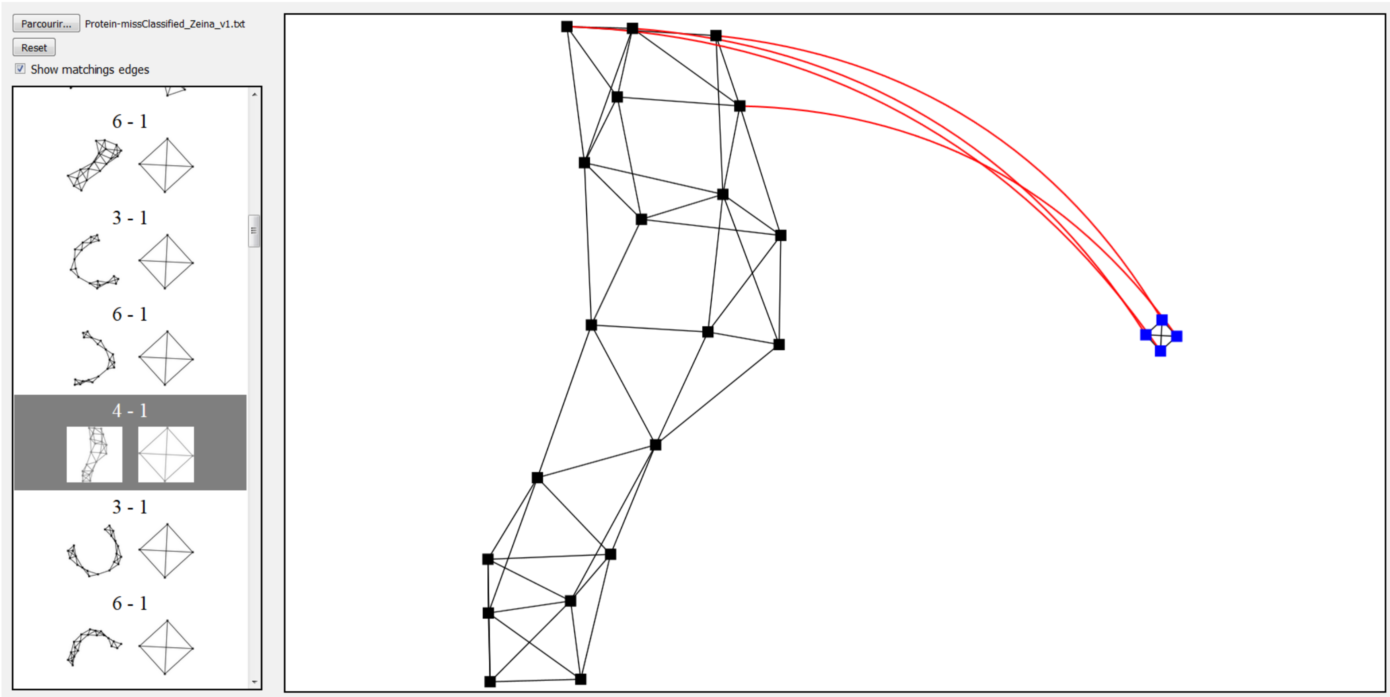

Fig. 5: Illustration of the misclassified graphs for the protein data set. One can see in the left panel, that several misclassification occurrences involve the 4-vertex clique.

[4] X. Cortès and F. Serratosa, "Learning graph-matching edit-costs based on the optimality of the oracle's node correspondences," *Pattern Recognition Letters*, vol. 56, pp. 22–29, 2015.

[5] K. Riesen, *Structural Pattern Recognition with Graph Edit Distance - Approximation Algorithms and Applications*, ser. Advances in Computer Vision and Pattern Recognition. Springer, 2015.

[6] Z. Abu-Aisheh, R. Raveaux, and J.-Y. Ramel, "Anytime graph matching," *Pattern Recognition Letters*, vol. 84, pp. 215 – 224, 2016.

[7] K. Riesen and H. Bunke, "IAM graph database repository for graph based pattern recognition and machine learning," *Pattern Recognition Letters*, vol. 5342, pp. 287–297, 2008.

[8] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in Information Visualization: Using Vision to Think*. M. K. Publishers Inc., 1999.

[9] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall PTR, 1998.

[10] R. Tamassia, *Handbook on Graph Drawing and Visualization*. Chapman and Hall/CRC, 2013.

[11] S. G. Kobourov, "Spring embedders and force directed graph drawing algorithms," *CoRR*, 2012.

[12] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Softw. Pract. Exper.*, vol. 21, no. 11, pp. 1129–1164, 1991.

[13] S. G. Kobourov, "Force-directed drawing algorithms," in *Handbook of Graph Drawing and Visualization*. CRC Press, 2013, pp. 383–408.

[14] K. M. Borgwardt, C. S. Ong, S. Schnauer, S. V. N. Vishwanathan, A. J. Smola, and H.-P. Kriegel, "Protein function prediction via graph kernels," *Bioinformatics*, vol. 21, 2005.

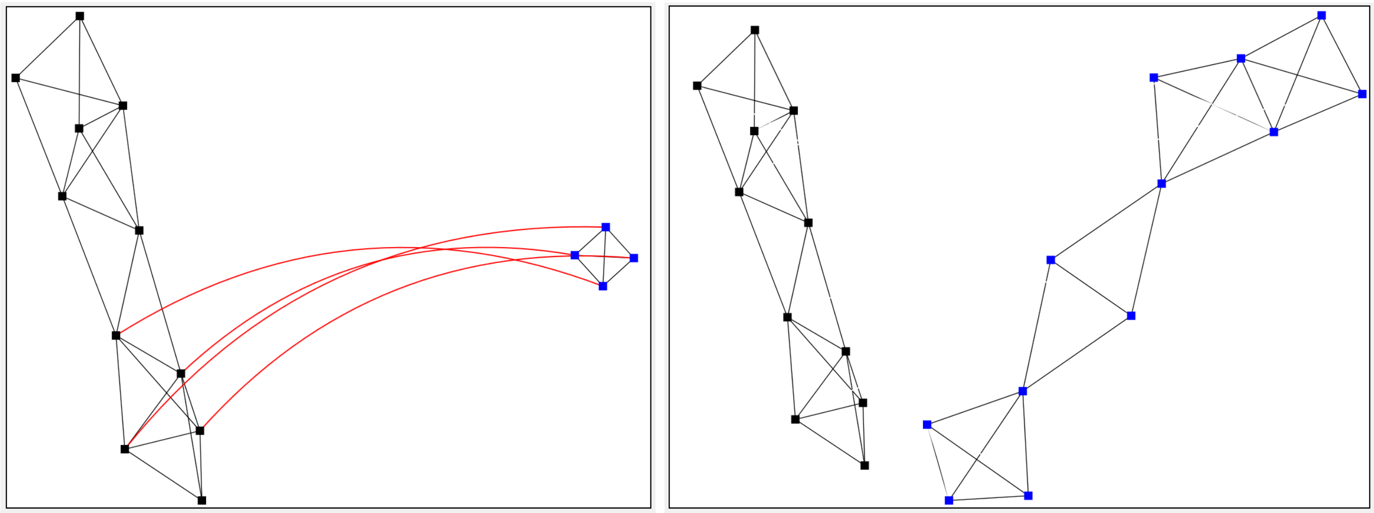

Fig. 6: Illustration of the nearest neighbour of a graph before (left) and after (right) our modification on the cost functions. Matching edges are hidden on the right figure to have a better perception of the graph structures.

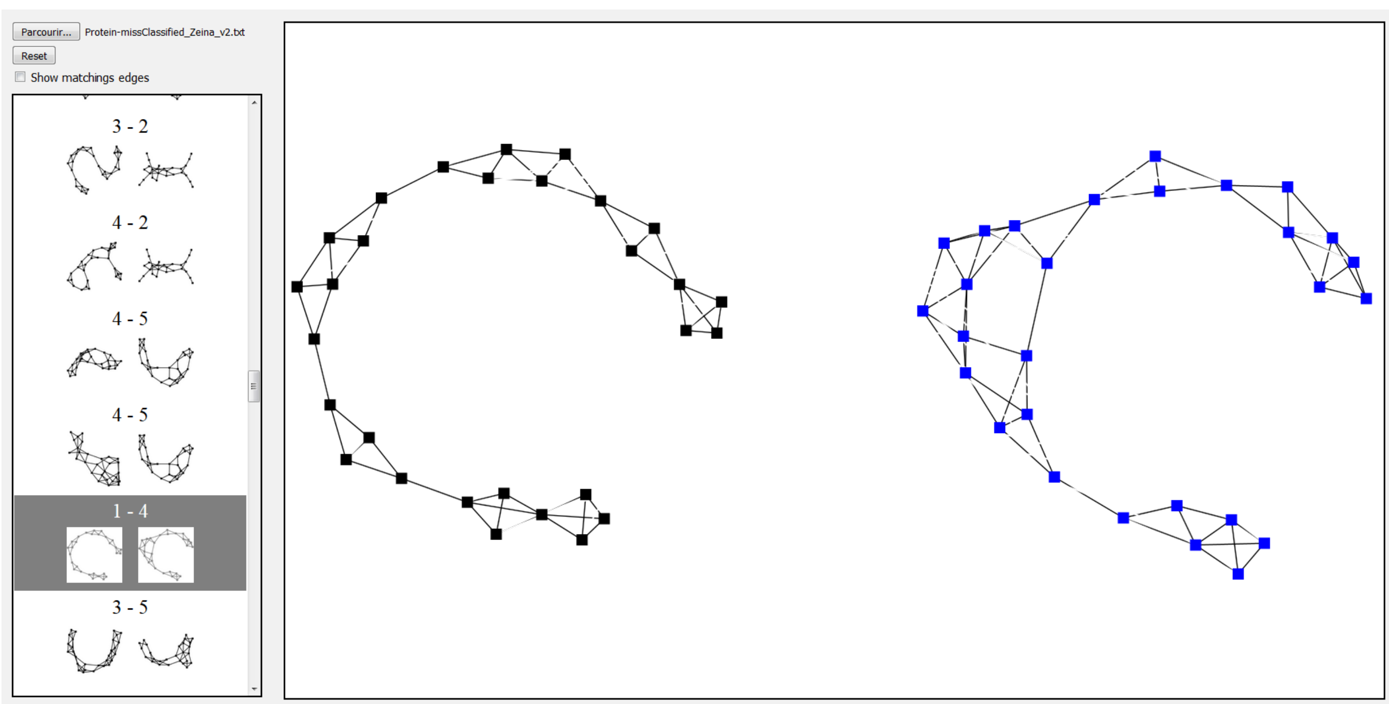

Fig. 7: Illustration of the misclassified graphs for the protein data set, after the modification on the cost functions. Matching edges are hidden to have a better perception of the graph structures. One can see on the left panel that graphs are matched with graphs with a similar topology, at least in terms of number of vertices.