Simple Graph Comparison Inspired on Metabolic Pathway Correlation

Esteban Arias-Méndez *, Alonso Montero-Marín, Danny Chaves-Chaves, Francisco Torres-Rojas
Ingeniería en Computación, Escuela de Computación, Tecnológico de Costa Rica
[Costa Rica Institute of Technology, School of Computing, Computing Engineering]
*Member of the PAttern Recognition and Machine Learning (PARMA) Group and The Happy Few.
Cartago, Costa Rica
esteban.arias@tec.ac.cr

***Abstract*—Graph comparison has been defined as a computationally complex task [1][3]. After a work about the correlation of metabolic pathways with two proposed approaches to simplify the problem of comparing its associated representation as graphs [2], we extend this work to general graph structures as a simple way to compare them. The first algorithm consists in the transformation of a two-dimensional graph structure to a one-dimensional structure, and thus aligning the corresponding 1D structure using sequence alignment tools from bioinformatics. The second algorithm consists in performing a pair analysis of 1:1 connected nodes between graphs and thus eliminating all similarities, finally, showing these differences to the user. These algorithms were developed as a low-cost process to correlate metabolic pathways with good results. Here we review the extension of this work as an application to a more general graph data structure. Results shows evidence of a quick, simple and effective way to resolve the described problem.**

Keywords— graph comparison; metabolic pathway correlation; depth-first traversal; breadth-first traversal

I. Introduction

To represent metabolic pathways, graphs and dynamic data structures have been used to model various relationships between processes of all kinds. Generally, a route is represented in a computer as a directed graph. From there, different techniques have been developed to align and compare its related directed graphs, corresponding to the routes of interest; the associated comparison complexity cost is NP, as demonstrated in [1] and [3]. Complex algorithm solutions have then been applied generally with the use of heuristic techniques that seek to reduce the time of graph-alignment, like the works in [8] and [12]. This causes some loss of generality but makes the data easier to process.. This problem is much more complex when looking for a comparison between multiple pathways or graphs at the same time.

The difference between two homologous routes and two similar routes must be considered. Homology can be described as a high-level comparison, more intuitive; while similarity is defined as a measurable and tangible valuation. We can say that two people are homologous because their general form is similar: 1 head, 2 arms, 1 trunk, 2 legs, 2 eyes, etc. However, although 2 people are homologous they might not be similar. In the case of routes, multiple paths may possess the same amount of interactions or reactions and thus have a homologous shape, but the reacting components or nodes differ.

II. Graphs and Metabolic Pathways

Metabolomics is the set of biochemical and physiochemical reactions that happen on a cellular level. A metabolic pathway is an ordered sequence of biochemical reactions between various actors that are transformed into a product through a series of reactions catalyzed by enzymes [4], [10]. Metabolic pathways provide key information to achieve a better understanding of life and all its processes; this is useful information for the improvement of medicine, agronomy, pharmacy and other similar areas.

The main analysis tool used to study these pathways is based on the idea of pathway comparison, using graph data structures. Computer scientists have proposed several mechanisms for effective comparison. To this regard Abaka et. al [1] have done a review of the most important tools developed until now, including NP costs associated with the alignment of two routes treated as graphs, tasks that have been described as computationally complex. This is mentioned also by Ay & Kahveci [3] who make a proposal called SubMAP (Subnetwork Mappings in Alignment of Pathways) which does not provide a 1-to-1, or 1-to-many, comparison as in the first approaches, but rather focuses on finding common sub-parts between different pathways, or similar subnets. The CAMPways algorithm of Abaka et. al [1] promises to be more efficient at runtime than do so-called “state of the art” algorithms. However, this algorithm refers to two evaluations or measures that can be self-conflicting: similarity and topological similarity of the given pathway. The analysis of the information obtained from paths, as well as the analysis of the previous tools, is a complex

mechanism dependent upon interpretation and the processing of existing information.

Another approach was used by [8], in which the M-GRAAL algorithm was used. This method relies on the calculation of *edge correctness*, which represents the percentage in which a graph is topographically like another graph. The goal of the M-GRAAL algorithm is to align two different networks in such a way that *edge correctness* is maximized. This task has great computational complexity, but M-GRAAL is a very good approximation. Pinter et. al [12], proposed a bottom-up dynamic programming method to align pathways of different graphs; however, this implementation requires a transformation from the original graph to a multi-source tree (which is a directed acyclic graph)

On a recent work, two different low-cost approaches were developed as mechanisms for the correlation of two metabolic pathways that can be used as a previous step to a deeper and more time-consuming analysis to be applied for the graph comparison associated to the pathways; the full details of the general step-by-step description of these algorithms can be observed in the work by [2]. Below we present a short description of the algorithms as a guide to the reader.

It is not the intention of this work to give a definitive answer to the result of comparing two graphs, rather we seek to provide an additional point of view as support, to be considered by an expert in the matter at the time of making their observations, evaluations and conclusions about the process they are studying represented as a graph It is not sought to give a "correct" answer on which is the best, only to provide reference information for the interested party.

III. Methodology

First, we describe the two algorithms proposed for the problem of correlating metabolic pathways to be applied as a general graph comparison.

A. *Algorithm 1: Transformation of 2D graph to a 1D or linear structure for later alignment and evaluation.*

In the case of metabolic pathways, it is common to observe in the description of the raw data obtained from the various databases that, although they are modeled as a graph with different relations between them and even containing internal cycles, it is characteristic that every route has two key elements: a starting point substrate and a final product as output. If the path is then viewed as a graph, this graph will have at least a root and an important target product or leaf node.

Concerning graphs which represents a metabolic pathway, when applying a traversal algorithm to the graph (which visits all the nodes) it becomes trivial to obtain the list of elements that conform said graph using a known root and desired target node. This would be a 2D to 1D transformation of the graph. If we take the starting point of the graph as the root of the graph, then all the nodes must be visited until arriving at the node of interest that would be the final product of the pathway as such. There are two common graph traversal strategies, by depth [14] or by breadth [5], (see also [6], [7], [9]).

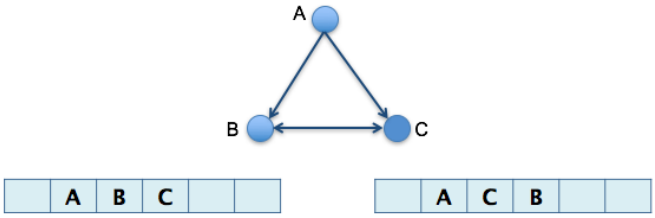

Fig. 1. Loss of information due to 2D to 1D transformation.

It should be noted that there will be a loss of information in such transformation. Figure 1 shows this fact, mainly on the order of the elements and their original relationships. As demonstrated in the previous work, such loss of information during the process is tolerable and acceptable for a correct comparison result.

On the other hand, in bioinformatics, alignment techniques are valid for a step-by-step comparison of each stage of the metabolic pathway, but, an efficient comparison mechanism at the computational level, which can then be used with different sources of information for the proper study of the metabolic pathways of interest and their subsequent analysis, is still required. So, After the transformation of the graphs to be analyzed, the linear information obtained is used to apply conventional alignment algorithms: global (GA) [11], semiglobal (SGA), and local (LA) [13]; with which we can obtain comparison values of said linear sequences. Once the route data is raised to obtain the routes in a 1D format we proceed to apply the traditional alignment algorithms. For the examples shown we used standard values: +1 for match, -1 for mismatch and -2 for gaps.

Below, we propose 5 different ways to implement a 2D to 1D graph transformation, to give the user different options to personalize the analysis required. Since, there are five versions of the algorithm, every one of them will be explained now. It is important to note that the result of any of these algorithms is a sequence of nodes from the original graph. This sequence is the 1D representation given of each of the graphs to be compared to by aligned later.

1) Algorithm 1.1: Traversal of the graph (starting at any node, finishing at any node)

This first version consists of an exploration of the graph using breadth-first traversal (BFT) and depth-first traversal (DFT). However, as the original implementation explained on [2], BFT has shown to be more useful when analyzing metabolic pathways. However, DFT could be more useful in some other cases. In this case, neither the starting nor the finishing nodes are selected by the user. As in most implementations of graph traversal, the starting point is selected randomly, and the traversal only stops when every node has been explored. To create a 1D sequence of nodes, this algorithm simply selects nodes when they appear during the traversal.

2) Algorithm 1.2: Traversal of the graph (starting at a given node, finishing at any node)

When looking for similarities between two graphs, it may occur that the first node in each of them has relevance or special

interest for the analysis, for this case, this version of the algorithm might be useful. This is because this algorithm works just as algorithm 1.1, with the property that the starting node is not selected randomly but manually by the user. Once again, the nodes are explored one by one using BFT and DFT. The final node is not selected by the user. Instead, the search ends when all the nodes have been found.

3) Algorithm 1.3: Traversal of the graph (starting at a given node, finishing at a given node)

In this third version of the algorithm, an initial node and final node can now be selected by the user. The algorithm will then proceed to explore the graph using BFT and DFT, starting with the initial node proposed by the user. However, instead of stopping when every node has been found, the search will stop when the final node appears. It is likely that the sequence of nodes created by this algorithm will not contain every node in the graph, which might be useful for the analysis of disjoint graphs or subgraphs.

4) Algorithm 1.4: Traversal of the graph (starting at any node, finishing at a given node)

This algorithm explores the graph in a similar way as the original algorithm does, with a single exception: when the final node is found, the search will stop. As it happens with the third version, it may occur that the entire graph is not represented in the sequence generated by this algorithm. This specific version can be of interest when analyzing graphs that represent a procedure. Since this research was inspired by the analysis of metabolic pathways represented with graphs, we decided that it is relevant to provide the user with an option in which the final node is more important than the origin for a certain path (as it happened with former algorithms 1.2 and 1.3), here is more important the ways to get to a node than the starting point.

5) Algorithm 1.5: Evaluation of all possible paths (starting at a given node, finishing at a given node)

In some cases, it is more useful to evaluate a single part of the graph instead of trying to obtain too much information at the same time (which might be difficult to represent as a sequence). This version of the algorithm might be better for some instances of interest, and it is very different from the previous versions. We observed that for some biotechnology analysis regarding metabolic pathways it was more interesting to study a subsection of the pathway than the entire graph.

This algorithm asks for an initial node and a final node, and it will recursively search for all possible paths from one to another. In some instances, there might be cycles on paths that are important for analysis (they might be irrelevant on some cases). Because of this, an additional parameter specifies how much repetitions of the same node are allowed on a path. A bigger number will result in more paths (when there are cycles in the graph).

The output of this algorithm is a list of paths from the initial node to the final node. This means that, when comparing two different graphs, it will compare all paths obtain between given initial and final nodes from first graph with all paths obtain between given initial and final nodes from the second graph. This would be a little more expensive, due to the number of comparisons, but it might yield better results, since then we select the best two alignments and provide the corresponding paths to the user for its analysis.

B. Algorithm 2: Differentiation by pairs

Many times, when we desire to compare two objects, the common elements are evident, it is then in such situations that it becomes more relevant to concentrate on looking for the differences. Based on this idea, this second algorithm seeks to eliminate from the graphs the common pairs of objects, that is, equal edges between the two routes or graphs, to find the differences between the two of them.

This would be a different approach to traditional numeric measured alignment that performs a comparison to highlight a value instead the divergent points between a given pair of routes. Our proposal is to look for the pairs of common edges between both graphs and to eliminate them. The process is shown step by step in detail in [2].

IV. Discussion: Tests and results

First, we provide a sample case of graphs to show the outputs of the algorithms and its implementations. On Figures 2 and 3 we show graphs G1 and G2 which, on an intuitive manner, look very similar.

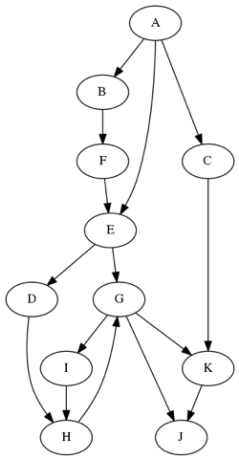

Fig. 2. Graphical visualization of sample graph G1.

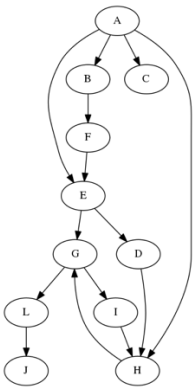

Fig. 3. Graphical visualization of sample graph G2.

In this case we show the outputs of the different implementations for Algoritm 1 for G1 and G2.

Algorithm 1.1

```
BFT  Pathway G1: A B E C F G D K J I H
     Pathway G2: A B E C H F G D L I J

DFT  Pathway G1: A C K J E D H G I B F
     Pathway G2: A H G I L J C E D B F

   GA-BFT:  +2, GA-DFT:  -5
   LA-BFT:  +5, LA-DFT:  +5
   SGA-BFT: +3, SGA-DFT: +3
```

Algorithm 1.2, G1 - starting node: A, G2 - starting node: B

```
BFT   Pathway G1: A B E C F G D K J I H
      Pathway G2: B F E G D L I H J

DFT   Pathway G1: A C K J E D H G I B F
      Pathway G2: B F E D H G I L J

   GA-BFT:  -6,  GA-DFT:  -3
   LA-BFT:  +2,  LA-DFT:  +2
   SGA-BFT: -18, SGA-DFT: -18
```

Algorithm 1.4, G1 - starting node: A, final node: H; G2 - starting node: A, final node: J

```
BFT   Pathway G1: A B E C F G D K J I H
      Pathway G2: A B E C H F G D L I J

DFT   Pathway G1: A C K J E D H G I B F
      Pathway G2: A H G I L J C E D B F

   GA-BFT:  +2, GA-DFT:  -5
   LA-BFT:  +5, LA-DFT:  +5
   SGA-BFT: +3, SGA-DFT: +3
```

Algorithm 1.5, G1 - starting node: A, final node: J; G2 - starting node: A, final node: J

Paths found for Graph G1:

```
   1.0 - A->B->F->E->G->K->J
   1.1 - A->B->F->E->G->J
   1.2 - A->B->F->E->G->I->H->G->K->J
   1.3 - A->B->F->E->G->I->H->G->J
   1.4 - A->B->F->E->D->H->G->K->J
   1.5 - A->B->F->E->D->H->G->J
   1.6  -  A->B->F->E->D->H->G->I->H->G->K->J
   1.7 - A->B->F->E->D->H->G->I->H->G->J
   1.8 - A->E->G->K->J
   1.9 - A->E->G->J
   1.10 - A->E->G->I->H->G->K->J
   1.11 - A->E->G->I->H->G->J
   1.12 - A->E->D->H->G->K->J
   1.13 - A->E->D->H->G->J
   1.14 - A->E->D->H->G->I->H->G->K->J
   1.15 - A->E->D->H->G->I->H->G->J
   1.16 - A->C->K->J
```

Paths found for Graph G2:

```
   2.0 - A->B->F->E->G->L->J
   2.1 - A->B->F->E->G->I->H->G->L->J
   2.2 - A->B->F->E->D->H->G->L->J
   2.3  -  A->B->F->E->D->H->G->I->H->G->L->J
   2.4 - A->E->G->L->J
   2.5 - A->E->G->I->H->G->L->J
   2.6 - A->E->D->H->G->L->J
   2.7 - A->E->D->H->G->I->H->G->L->J
   2.8 - A->H->G->L->J
   2.9 - A->H->G->I->H->G->L->J

Most valuable pathways
   Pathway G1 1.4: A->B->F->E->D->H->G->K->J
   Pathway G2 2.2: A->B->F->E->D->H->G->L->J

   GA pathways  1.4 and 2.2: +7
   LA pathways  1.4 and 2.2: +7
   SGA pathways 1.4 and 2.2: +7
```

Algorithm 2, output for G1 and G2 comparison.

```
Differences Identified from G1 to G2
   C -> K
   K -> J
   G -> K
   G -> J
Differences Identified from G2 to G1
   A -> H
   G -> L
   L -> J
```

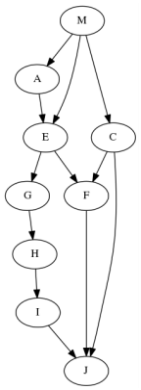

Fig. 4. Graphical visualization of sample graph G3..

If we now compare the graph G1 on Figure 2 and graph G3 on Figure 4, can be observed the structures are not very similar. The purpose of this is to test the algoritms proposed with data alike.

Algorithm 1.1

```
BFT   Pathway G1: A B E C F G D K J I H
      Pathway G3: A E F G J H I

DFT   Pathway G1: A C K J E D H G I B F
      Pathway G3: A E G H I J F

   GA-BFT:  -5,  GA-DFT:  -5
   LA-BFT:  +2,  LA-DFT:  +2
   SGA-BFT: -14, SGA-DFT: -14
```

Algorithm 1.2, G1 - starting node: A, G3 - starting node: B

```
BFT   Pathway G1: A B E C F G D K J I H
      Pathway G3: B F E G D L I H J

DFT   Pathway G1: A C K J E D H G I B F
      Pathway G3: B F E D H G I L J

   GA-BFT:  -3,  GA-DFT:  -4
   LA-BFT:  +4,  LA-DFT:  +4
   SGA-BFT:     -18,     SGA-DFT:     -18
```

Algorithm 1.4, G1 - starting node: A, final node: H; G3 - starting node: M, final node: I

```
BFT   Pathway G1: A B E C F G F K J I H
      Pathway G3: M A E C F G J H I

DFT   Pathway G1: A C K J E D H G I B F
      Pathway G3: M C J F E G H I A

   GA-BFT:  -3,  GA-DFT:  -5
   LA-BFT:  +4,  LA-DFT:  +4
   SGA-BFT: -18, SGA-DFT: -13
```

Algorithm 1.5, G1 - starting node: A, final node: J; G3 - starting node: M, final node: J

Paths found for Graph G1:

```
   1.0 - A->B->F->E->G->K->J
   1.1 - A->B->F->E->G->J
   1.2 - A->B->F->E->G->I->H->G->K->J
   1.3 - A->B->F->E->G->I->H->G->J
   1.4 - A->B->F->E->D->H->G->K->J
   1.5 - A->B->F->E->D->H->G->J
   1.6  -  A->B->F->E->D->H->G->I->H->G->K-
>J
   1.7 - A->B->F->E->D->H->G->I->H->G->J
   1.8 - A->E->G->K->J
   1.9 - A->E->G->J
   1.10 - A->E->G->I->H->G->K->J
   1.11 - A->E->G->I->H->G->J
   1.12 - A->E->D->H->G->K->J
   1.13 - A->E->D->H->G->J
   1.14 - A->E->D->H->G->I->H->G->K->J
   1.15 - A->E->D->H->G->I->H->G->J
   1.16 - A->C->K->J
```

Paths found for Graph G3:

```
   3.0 - M->A->E->F->J
   3.1 - M->A->E->G->H->I->J
   3.2 - M->E->F->J
   3.3 - M->E->G->H->I->J
   3.4 - M->C->F->J
   3.5 - M->C->J

Most valuable pathways
   Pathway G1 1.9: A->E->G->J
   Pathway G1 1.8: A->E->G->J
   Pathway G3 3.0: M->A->E->F->J
   Pathway G3 3.2: M->E->F->J

   GA pathways  1.9 and 3.0: +2
   LA pathways  1.8 and 3.2: +3
   SGA pathways 1.8 and 3.2: +2
```

Algorithm 2, output for G1 and G3 comparison.

Differences Identified from G1 to G3

```
A -> B
A -> C
A -> H
B -> F
E -> D
H -> G
F -> E
G -> L
G -> I
D -> H
L -> J
I -> H
```

Differences Identified from G3 to G1

```
E -> F
C -> F
C -> J
F -> J
G -> H
H -> I
I -> J
M -> A
M -> E
M -> C
```

Shown the results, it is the moment to analyze the results of the test. It should be noted that when we specify that two graphs are "similar" or "different", we are talking in an intuitive way about how we evaluate them. More concrete results about the similarities between these graphs can be inferred from the data obtained.

A. *Testing Graph 1 vs Graph 2 (Similar graphs)*

The original algorithm (1.1) performed very well. It rated both graphs as "similar", with a similarity value of +5 in the best cases (local alignment combined with BFT, local alignment with DFT). In the average case, the similarity values were also good (+3 or +2). In the worst case (global alignment with DFT), the graphs were rated as "different" with a similarity value of –5. However, this is only one of six cases, and it depended on a depth-first traversal, which, as suggested by [2] is not very useful for paths alignments.

The similarity values obtained by the second algorithm (1.2) were not as high as those obtained with the first algorithm. However, it is likely because the selected initial node was different in both graphs (it was A on Graph 1 and B on Graph 2). The mostly negative values of similarity are just a consequence of this selection and is actually right.

With the algorithm 1.3, the results are very similar to the ones obtained with 1.1, which implies that they are very good, indeed. The same error occurs with the global alignment performed over the results provided by depth-first traversal. However, every other result is positive and indicates that the two graphs are actually similar (which is the case).

In the case of algorithm 1.4, many paths were found and the result of comparing all of the paths from the first graph with all of the paths from the second graph was that the most similar alignments were very positive (+7). This experiment was considered to be a success because it showed that there are paths on the both graphs which are almost the same. This would not happen if the graphs were "mostly different".

B. *Testing Graph 1 vs Graph 3 (Mostly Different)*

In this case, the algorithm 1.1 rated these graphs as "very different", with the best rating being +2 and the worst being –14. On this specific case, DFT and BFT obtained the results. Since DFT normally gives bad results, this would imply that even BFT not able to find many coincidences when doing alignments (the best alignment was only of +2 which is still low).

For the second algorithm, the results were not very different from those shown by algorithm 1.1. The worst similarity value obtained was of –18, but the best was of +4. However, as it happened with the previous algorithms, only the local alignments obtained positive results. The only case in which BFT did not obtained the same rating as DFT was in the global alignment, when BFT got –3, while DFT got -4. This reinforces the idea that BFT is better that DFT when it comes to finding similarities. With algorithm 1.3, the results were very similar to those obtained with algorithm 1.2. which is good because the differences between both graph are shown.

With algorithm 1.4, the results were very good. Even when comparing all the paths from graph 1 with all the paths from graph 2, the best alignments were low (+2, +3 and +2, with the highest being local alignment). And even in those cases, the paths that yielded those results were very short (four nodes on most cases, six nodes in the best case). This shows that the similarities on both graphs are very small compared to those found between graph 1 and graph 2 (both paths had seven nodes).

C. *Testing Graph 2 vs Graph 3 (Mostly Different)*

As it happened on the second test (where both graphs were mostly different), these graphs were found to be again very different when comparing them with the algorithm 1.1. The best similarity value is +2, while the worst is -14. The average of all cases is -6. The BFT found more similarities (-5) than the DFT (-7) once again, in the global alignment. With algorithm 1.2, the results are very similar. The best value is the same (found in local alignment), while the worst is now -18 (semiglobal alignment). When it came to global alignment, BFT and DFT were further away (BFT gave a rating of -3, while DFT gave -12). Once again, the results yielded by the algorithms 1.2 and 1.3 are practically the same (they found the same path). The average rating was of -7.8.

In the case of algorithm 1.4, the results obtained were like those of the previous test: the best possible alignments were very low (+1, +3 and +1, with the highest being local alignment). Also, the paths were very short (5 or 6 nodes).

Now, we evaluate the cost of the algorithms used to show that they are less costly than the ones used so far. The second step is to demonstrate that the procedure provides an accurate and useful result of the comparison.

For the procedure of the first algorithm, we make use of graph traversal by breadth or by depth. As previously indicated, using a depth-first traversal does not provide information like the one described by a pathway and the results for different

graphs can be seemingly random. In the case of breadth-first search, a level crossing is performed, like the way a metabolic route works in nature. So, depth-first traversal is not relevant for the process proposed. The cost of these algorithms approaches in the order of O (| V | + | E |), where V: is the set of vertices or nodes of the graph and E $\subseteq$ VxV: is the set of edges or arcs.

For the second algorithm, it must be considered that for each reaction that exists in the first G1 path or graph, it must be found in the second G2 path or graph. That is, if R1 is the number of reactions counted by G1 and R2 the quantity for G2, there will be a maximum R1 x R2 comparisons, when it is common for a half-time on average to perform such comparisons. Thus, we can establish a worse case in the order of O (R1 x R2).

We can observe that the routes of figures 4 and 5 are like each other. The goal is to achieve a good value of precision in the comparison without sacrificing accuracy in the process. The result achieved is to gain time by means of a simple procedure and without losing accuracy.

After applying the proposed algorithms, in the above example, we obtained effective comparison values of +3 for the global alignment and of +5 for the local alignment. It is easy to verify the evident similarity between both routes analyzed until now and we present an evaluation mechanism that provides us a similarity score.

When testing with a different pathway (on purpose very different), in both shape and contents, shown in figure 12, and after applying the transformation using algorithm 1 and its subsequent alignment we observe that the results vary as expected.

In the case of the alignment of the routes of figures 4 and 12 (which is completely different biological process), the values reached were -10 for the overall alignment, 0 for the local alignment and a value of -20 if we applied a semi global alignment. Again, if a comparison is made between the two in this case both are quite dissimilar. This is to show that the algorithms will provide also information about differences.

For the case of the algorithm 2, we did not find an algorithm with which to compare it since it is a different strategy than the ones proposed so far about similarity. But it does provide useful information to the expert who performs an analysis of the observed differences. After applying the algorithm shown step by step in figure 11 to the same working paths for figures 4 and 5, the differences listed below were obtained. For each metabolic pathway, 4 and 5, the reactions that are present and are not present in the opposite path are listed below.

Differences Identified:

It should be noted that the reactions are bidirectional in the original routes, which is why the description of each reaction is given in each direction.

After applying the second algorithm to the routes of figures 4 and 12, it is evident that there are many different reactions between the two routes as presented in the following obtained output. And the algorithm provides information about that fact. For each metabolic pathway, 4 and 12, the reactions that are present and are not present in the opposite path are listed below.

Differences Identified:

V. Conclusions

It can be established that the mechanism proposed in algorithm 1: "transforming a 2D structure to 1D structure for later alignment and evaluation" can be used as a prior evaluator to predict good comparisons in case a deeper analysis is desired.

In the case of the algorithm 2: "Differentiation by pairs" the proposal is to offer the expert an additional point of view for his evaluation of the pathway in question. In this case, no score is provided but the listed differences.

We demonstrated that through the use of the proposed algorithms they are a fast and of relatively low computational cost, it is possible to provide relevant information for the comparison study about metabolic routes of interest and other analyzes. This is achieved by simplifying the information, in the case of transforming data from a 2D to 1D structure, the loss of information or precision does not affect much the result, which is to give the user a similarity score between the two analyzed pathways. However, as mentioned before, the depth-first traversal of the graph is not relevant for analysis, since the product can appear in other position of the 1D sequence rather than at the end of the sequence as one can expect for a product of a metabolic process.

VI. Future Work

Having verified that the proposed algorithms can provide relevant information for the analysis and comparison of metabolic pathways, it would be useful to implement a complete software tool, capable of: directly accessing metabolic databases, extracting information from metabolic routes of interest and finally applying the proposed algorithms to the benefit of experts.

In the other way make the tool more accessible for people that doesn't know about programming languages and develop a intuitive and easy to manage, using a webpage that the experts can upload the models and compare with another or detect some similarities between different models, but generating the same reactions at given point.

Another functionality to be implemented in the website is generate the graph of the given metabolic pathways, of this mode can see if reaction occur in determined model.

Finally, just as there are multiple alignment algorithms for several genetic sequences, it is important to continue working on the problem of comparing multiple pathways to find, for example, similar factors among different species.

Acknowledgments

This work would not have been possible without the support of School of Computing and VIE at TEC Costa Rica.

References

[1] Abaka, G., Bıyıkoğlu, T., & Erten, C. (2013). CAMPways: constrained alignment framework for the comparative analysis of a pair of metabolic pathways. Bioinformatics, 29(13), i145-i153.

[2] Arias-Mendez, E., & Torres-Rojas, F. (2017, July). Alternative low-cost algorithms for metabolic pathway comparison. In Bioinspired Intelligence (IWOBI), 2017 International Conference and Workshop on (pp. 1-9). IEEE.

[3] Ay, F., Kellis, M., & Kahveci, T. (2011). SubMAP: aligning metabolic pathways with subnetwork mappings. Journal of computational biology, 18(3), 219-235.

[4] Bruce, A., Dennis. B., Julian, L., Martin, R., Keith, R., James D., W. (1994) Molecular Biology Of The Cell third edition.

[5] Bundy, A., & Wallen, L. (1984). Breadth-First Search. In Catalogue of Artificial Intelligence Tools (pp. 13-13). Springer Berlin Heidelberg.

[6] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). Introduction to algorithms second edition.

[7] Knuth, D. (1968). The Art of Computer Programming 1: Fundamental Algorithms 2: Seminumerical Algorithms 3: Sorting and Searching. MA: Addison-Wesley, 30.

[8] Kuchaiev, O., & Przulj, N. (2010). Global Network Alignment.

[9] Lee, C. Y. (1961). An algorithm for path connections and its applications. IRE transactions on electronic computers, (3), 346-365.

[10] Lee, J. M., Gianchandani, E. P., Eddy, J. A., & Papin, J. A. (2008). Dynamic analysis of integrated signaling, metabolic, and regulatory networks. PLoS Comput Biol, 4(5), e1000086.

[11] Needleman, S. B., & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology, 48(3), 443-453.

[12] Pinter, R. Y., Rokhlenko, O., Yeger-Lotem, E., & Ziv-Ukelson, M. (2005). Alignment of metabolic pathways. Bioinformatics, 21(16), 3401-3408.

[13] Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. Journal of molecular biology, 147(1), 195-197.

[14] Tarjan, R. (1972). Depth-first search and linear graph algorithms. SIAM journal on computing, 1(2), 146-160.