IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 10, 2020, accepted August 29, 2020, date of publication September 1, 2020, date of current version September 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020924

Extracting Dense and Connected Communities in Dual Networks: An Alignment Based Algorithm

PIETRO HIRAM GUZZI[1], (Member, IEEE), EMANUEL SALERNO[1], GIUSEPPE TRADIGO[1,2], AND PIERANGELO VELTRI[1], (Member, IEEE)

[1]Department of Surgical and Medical Sciences, Magna Græcia University, 88100 Catanzaro, Italy
[2]Università degli Studi eCampus, 22060 Novedrate, Italy

Corresponding author: Pietro Hiram Guzzi (hguzzi@unicz.it)

This work was supported in part by the Healthing Project, a Programma Operativo Regione (POR) Calabria Region Project, and in part by the POR SISTABENE Project.

ABSTRACT Networks-based models have been used to represent and analyse datasets in many fields such as computational biology, medical informatics and social networks. Nevertheless, it has been recently shown that, in their standard form, they are unable to capture some aspects of the investigated scenarios. Thus, more complex and enriched models, such as heterogeneous networks or dual networks, have been proposed. We focus on the latter model, which consists of a pair of networks having the same nodes but different edges. In dual networks, one network, called physical, has unweighted edges representing binary associations among nodes. The other is an edge-weighted one where weights represent the strength of the associations among nodes. Dual networks capture in a single model some aspects that cannot be described by using a standard model. Dual networks can be used, for instance, to capture a co-authorships network, where physical network represents co-authors. In contrast, the conceptual network is used to model topics sharing among a couple of authors by means of edge connections. This allows capturing similar interests among authors even though they are not co-authors. We propose an innovative algorithm to find the Densest Connected Subgraph (DCS) in dual networks. DCS is the largest density subgraph in the conceptual network, which is also connected in the physical network. A DCS represents a set of highly similar nodes. Moreover, since DCS is a computationally hard problem, we propose novel heuristics to solve it. We tested the proposed algorithm on social, biological, and co-authorship networks. Results demonstrate that our approach is efficient and is able to extract meaningful information from dual networks.

INDEX TERMS DCS, dual networks, graph alignment, social networks.

I. INTRODUCTION

The use of network-based models to analyse data is currently growing in many research fields. For instance, in biology and medicine, many approaches use graphs both to model and analyse data [1], [2]. Similarly, social networks data can be modelled using graphs and analysed to extract relevant information regarding connections among people [3]. Most of the problems are modelled by using a *s*ingle network, i.e., the same structure is used to model data and, then, to extract information by studying network properties, as well as to identify community-related structures [4]–[7]. For instance, considering networks of genes and proteins, communities represent groups of related genes or proteins in biology. The network instance is then used to study relations among proteins (or genes) with molecular mechanisms. In social networks, a single network can be used to identify the existence of community-based structures which usually indicates the presence of related users [8].

Recently, more complex models have also been introduced. The use of a pair of graphs representing two different views of the same scenario has been introduced to detect hidden properties which are not detected by single network-based models [9]. For instance, there are scenarios where given a set of nodes, we need to model weighted connections as well as (minimal) set of simple connections. In these cases, two graphs are required.

We focus on problems requiring two graphs, which are modelled with the so-called *dual networks model* (or simply dual networks). This model is based on the use of two graphs with the same vertices set and two different edges sets. One graph is unweighted and referred to as physical

The associate editor coordinating the review of this manuscript and approving it for publication was Ali Salehzadeh-Yazdi.

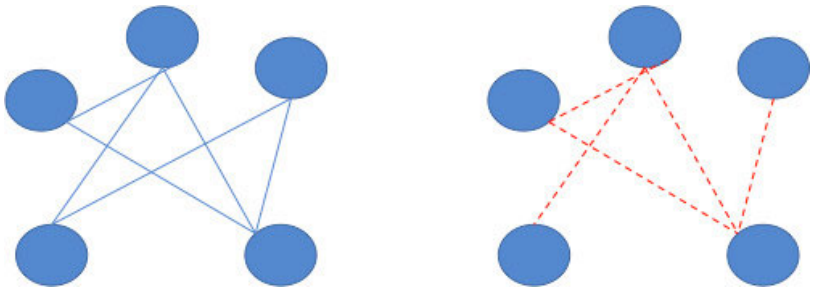


FIGURE 1. An example of dual network. Graph on the left (with solid edges) represents the conceptual network, while the other one (with red dashed edges) represents the physical network (for sack of simplicity we omitted the weight of edges on conceptual network).

graph. The other graph, called conceptual graph, is edge-weighted. Figure 1 reports an example of dual network. The use of dual networks finds natural applications whenever it is needed to model two kinds of relations among the same set of nodes. The two networks represent physical and conceptual interactions.

Phillips *et al.*, used dual networks to analyse interactions among genetic variants [10], while Tornow *et al.*, use the dual network to analyse expression data and their functional relations [11]. In such a scenario, networks representing the co-expression of genes (functional networks) may be jointly analysed with other one presenting known interactions among proteins. The integration of data may help to find relations among genes co-expression and known interactions. Ulitsky *et al.*, use a graph representing genetic interactions, i.e. a graph whose nodes are genes, and edges represent the association of two genetic perturbations affecting the phenotype (*genetic conceptual network*), and a graph representing physical interactions among genes (*physical network*) [12].

In this paper we focus on the problem of finding Densest Connected Subgraphs (DCS) in dual networks. Formally, let $G_p = (V, E_p)$, and $G_c = (V, E_c)$ be two undirected graphs defined on the same set of vertices V, where G_p is unweighted and G_c is weighted. Such two graphs represent a dual network that can also be formally indicated as $G(V, E_c, E_p)$. Finding DCS in the dual network G, consists in selecting nodes $I \in V$, edges $E_c^I \in E_c$ and $E_p^I \in E_p$ such that: (*i*) the subgraph $G_p^I = (I, E_p^I)$ is connected and (*ii*) the subgraph $G_c^I = (I, E_c^I)$ is the densest, i.e. where the density is the maximum possible value. The density of a graph is the ratio between the sum of edge's weights versus the number of nodes. We formally introduce such definitions in Section III.

Finding DCS in dual networks allows to capture real cases which cannot be captured with single networks. For instance, in co-authorship networks, the physical networks model the existence of a co-authorship relation, while the conceptual network models the similarity of research among authors, independently of the property of being co-authors. The DCS in this case represents a community of authors having similar research interests and also mutual co-authorship relations (e.g. 'A' is co-author of 'B', 'B' is co-author of 'C' but 'A' may not be co-author of 'C'). The co-authorship DCS may be used for instance by a recommendation system to optimise conference proposals. Similarly, in the social network case, the conceptual network of the DCS can be used to model the geographical distance among users, while the physical network could model friendship relations. Therefore a DCS is a community of geographically related users having common friends, therefore DCS may suggest novel friendship relations or some targeted information. DCS evaluation in social networks allows to identify set of users that share common interests and are geographically related.

Finding DCS is an NP-hard problem [9] in its general formulation. Indeed, it can be reduced from the set cover problem [13]. Thus, the need for introducing novel heuristics able to solve it arises. Nevertheless, while finding the densest graph in a single network has been solved by many approaches employing different heuristics, finding a DCS in a dual network is still a challenging problem. E.g., in [9] authors propose two heuristics based on pruning for solving DCS problem. The here proposed contribution consists in modelling the problem of finding the DCS as a local network alignment problem, for which we propose a novel algorithm. Such an algorithm, implemented in a tool named DN-Aligner, uses a merge-and-mine approach as in [14]–[16], and receives as input a pair of networks. It merges these networks in a single *Weighted Alignment Graph* (WAG). Each node of the built WAG belongs to both conceptual and physical networks. Each weighted edge is added to the WAG using a scoring function. This way, any sub-graph of the obtained alignment network represents a connected sub-graph of the input one. The weights of the edges are derived from the input conceptual network. Finally, we extract the densest sub-graph by using a variation of the Charikar algorithm [17]. Such densest sub-graph represents a connected graph in the physical network; therefore, it is a solution to the problem. Figure 2 depicts the main steps of our algorithm. We implemented such an algorithm, and we show the effectiveness of our approach presenting three case studies: (i) the first one based on social networks data, (ii) the second one based on biological networks and (iii) the third one based on a co-authorship network. Results confirm the effectiveness of our approach.

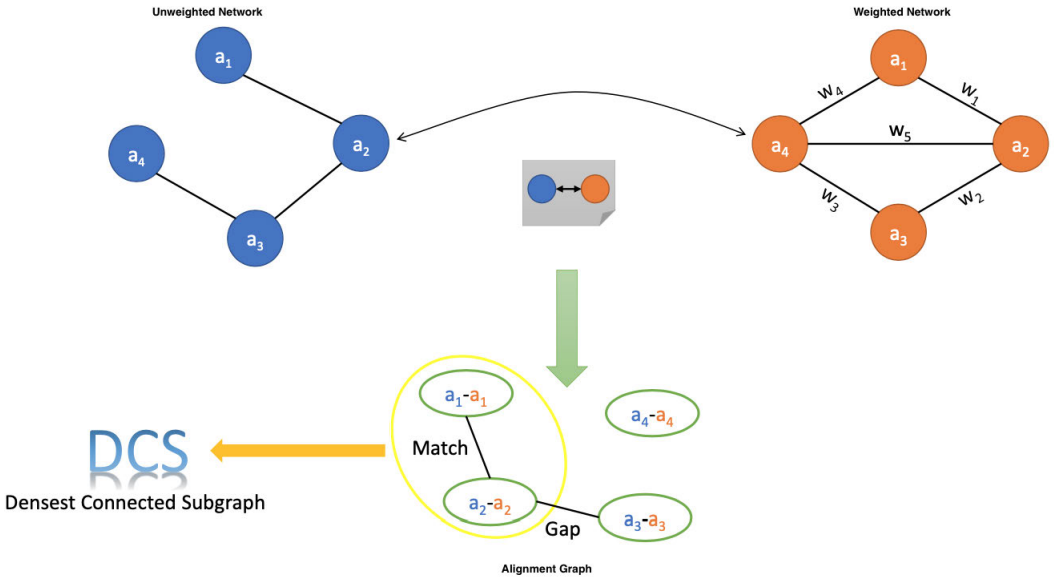


FIGURE 2. Workflow of the DN-Algorithm. The algorithm receives as input the two networks representing a dual network. In the first step the two networks are merged together into a single Alignment Graph. Each node of the alignment graph represents a pair of nodes of the input network. Edges are inserted considering the two input networks. Then the Charikar algorithm is used for extracting the densest sub-graph of the alignment graph. Each sub-graph of the alignment graph represents a connected sub-graph of the unweighted networks. Therefore it is a densest connected sub-graph for the dual network.

The paper is structured as follows: Section II discusses main related works. Section III reports a formal formulation of DCS and describe the proposed algorithm. Section IV discusses the case studies and results of applying the DN-Aligner tool, and Section V concludes the paper.

II. RELATED WORK

We focus on finding the Densest Connected Subgraph (DCS) in a dual network. Detecting dense components of a graph is one of the most challenging problems in graph analysis [18], [19]. Recently, it found applications in many important fields such as social network analysis [20]–[22]. The problem is based on the definition of *density* for a graph, and literature contains many definitions that have been applied in different contexts. One of the first definitions of dense sub-graph is a fully connected sub-graph, i.e. a clique. However the identification of a maximal clique, also referred to as the *maximum clique problem*, is NP-hard [23], and it is difficult to approximate [24].

Wu *et al.* proposed an algorithm for finding densest connected sub-graph in a dual network. The approach is based on a two-step strategy [9]. In the first step, the algorithm prunes the dual network without eliminating the optimal solution. In the second step two greedy approaches are developed to build a search strategy for finding the DCS. Briefly, the first approach finds the densest sub-graph in the conceptual network first, and then it is refined to guarantee that it is connected in the physical network. The second approach maintains the sub-graph connected in the physical network while deleting low-degree nodes in the conceptual network. Authors also propose a possible solution for finding the DCS with fixed number of nodes and for maintaining a set of input seed nodes in the identified sub-graph.

The DN-Aligner algorithm proposed here is more stable than the one presented by Wu *et al.*, since it allows to define the input correspondence among nodes, while the approach of Wu *et al.*, is based on the correspondence of nodes with the same name. Therefore our approach may be easily extended when the set of nodes are not the same and may be also used to find other kinds of communities (e.g., by using a different algorithm for mining the alignment graph, as we show in the following).

The problem of finding the *densest sub-graph* may be also treated by using heuristic in polynomial time, for instance the algorithm developed by Goldberg based on maximum-flow approach [25]. Asashiro *et al.* proposed a greedy algorithm based on the strategy of deleting the node with minimum degree [26]. Our method includes also an heuristic by implementing a similar approach but we improve it by extending the method to weighted graphs.

The here proposed algorithm uses network alignment methods to build the initial alignment graph. Network alignment algorithms aim to find a mapping among two (or more) input networks and are categorised as *local* or *global*. The global alignment algorithms (GNAs, Global Network Alignment) search the best superimposition of the whole compared networks by exploiting one-to-one node mapping. Moreover, algorithms may be designed for homogeneous networks or

TABLE 1. The main notations used for graphs.

Symbol	Definition
$G = (V, E)$	Graph G with node set V and edge set E
$G = (V, E_p, E_c)$	a dual network made by a conceptual network $G_c(V, E_c)$ and a physical network $G_p(V, E_p)$
$I \subset V$	a subset of nodes
$vol(v), v \in V$	the sum of the weights of the edges incident to the node v
$\rho(V) = \frac{\sum_{\forall v \in V}(vol(v))}{\lvert V \rvert}$	density of a graph G defined as the average vol of the nodes

heterogeneous ones [4]. Many implementation have been proposed, such as: (*i*) MAGNA [27]; (*ii*) MAGNA++ [28] and (*iii*) IGLOO [29]. Traditional GNAs employ a two-stage procedure. During the first stage, they apply a cost function to estimate pairwise similarities among nodes. Then, they use an alignment method to quickly determinate, among all probable alignments, the one with a high score with the overall similarity on all aligned node [16].

Local Network Alignment algorithms (LNAs) find multiple (relatively small) regions of similarity among input networks. Each region is usually mapped independently of other regions. Each subgraph represents a conserved motif or pattern of activities. Prominent examples of LNAs are NetworkBLAST [30], MaWish algorithm [31], Graemlin [32], NetAligner [33], and AlignNemo [34].

We used local network alignement starting from methods proposed in [14] and [35]. We built the alignment graph similarly to L-HetNetAligner by employing the same algorithmic approach.

III. FINDING THE DENSEST CONNECTED SUBGRAPH (DCS)

In this section, we formalise the problem of finding a Densest Connected Subgraph (DCS) in a dual network. We describe the here proposed DN-Aligner algorithm to find DCS from a dual network.

A. GRAPH FORMULATION

A dual network comprises two networks sharing the same node-set. One network, called physical network, has unweighted edges. A second network, called conceptual network, has weighted edges. Edge sets are in general different in the two networks. Using the standard notation reported in Table 1, we formulate DCS problem [9] by using the following definitions.

Definition 3.1: Dual Network.

A dual network $G = (V, E_p, E_c)$ consists of two networks: a conceptual weighted network $G_c(V, E_c)$ and a physical unweighted one $G_p(V, E_p)$.

Definition 3.2: Density of unweighted graph.

Given an unweighted graph $G(V, E)$ the density ρ is defined as the ratio of number of edges to the number of nodes, i.e. $\rho = \frac{|E|}{|V|}$

The definition may be extended to weighted graphs as described in literature [9] by considering the sum of the weights of the edges for each node, known as the *vol* of the node. Then, the density is calculated using the average weight of each node.

Definition 3.3: Density of a weighted graph.

Given an weighted graph $G(V, E)$, let $v \in V$ a node of G, and $vol(v) = \sum_{(v,w)\in E} weight(v, w)$ be the sum of the weights of the edges for the node v. The density of weighted graph is defined as $\rho(G) = \frac{\sum_{v\in V}(vol(v))}{|V|}$

Given a dual network we may consider the subgraphs G_p^I and G_c^I induced in the two networks by the same node set $I \subset V$. A densest common subgraph *DCS* is the subgraph defined on a subset of nodes I such that the density of the induced conceptual network is maximised and the induced physical network is connected. Thus the problem consists in identifying the set of node I.

Definition 3.4: Densest Common Subgraph.

Given a dual network $G(V, E_c, E_p)$, the densest connected subgraph is a subset of nodes $I \subset V$ such that G_p^I is connected and the density of G_c^I is maximised.

Table 1 summarizes the above reported formulation.

B. DN-ALIGNER ALGORITHM DESCRIPTION

To identify the DCS in dual network, we use an alignment graph algorithm and then aligned graph is used to extract the DCS. Given a dual network, the Algorithm 1 reported briefly in the sequel is based on the following two steps:

- (step 1) Merge the input networks into a single alignment graph;
- (step 2) Analyse the alignment graph using (an adapted version of) Charikar algorithm [17].

Algorithm 1 DN-Aligner Algorithm

Input: : A Conceptual Network $G_c = (W, E)$, and a Physical Network $G_p = (V, E)$,
Input: : A Correspondence File F indicating the nodes to be merged
Input: : A distance threshold δ (optional)
Output: : DCS
Begin
1: : WAG ← BuildAlignmentGraph(G_c,G_p,δ,F)
2: : DCS ← Analyse(WAG)
3: **return** DCS
End

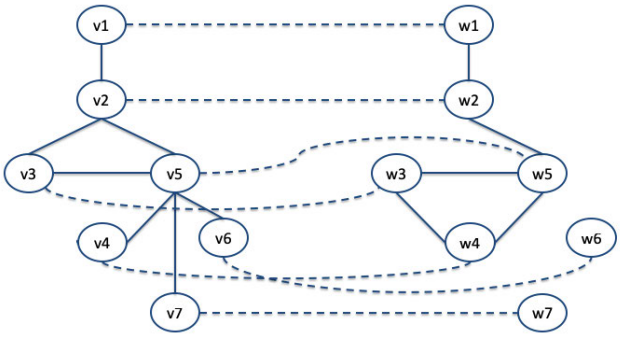


FIGURE 3. Alignment example: the algorithm receives as input two networks and a set of similarity relationship among nodes of the networks (dashed lines).

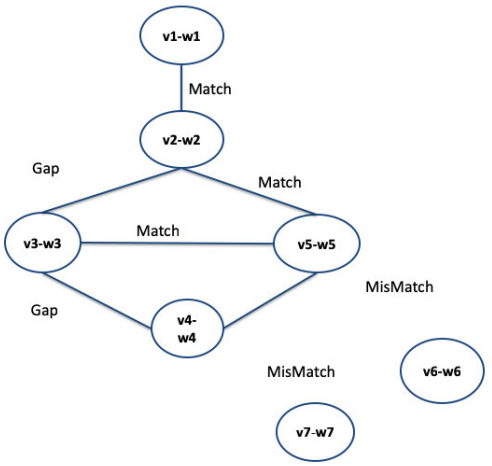


FIGURE 4. First, the algorithm builds the nodes of the heterogeneous alignment graph. The edges are then added according to the analysis of input networks.

In the first step the two networks are merged together in a single alignment graph. The algorithm has two additional parameters: (i) a file F that stores the correspondence among nodes, i.e. which nodes belonging respectively to conceptual and physical networks have to be merged; (ii) a distance threshold δ that represents the maximum distance threshold that two nodes should have in the physical network (the parameter is optional and it is used to prune the possible solutions).

In the first step the algorithm merges the input network in a single weighted **alignment graph**. Each node of the input graph represents a pair of corresponding nodes of the input ones. Each weighted edge of this graph is added using a *match-mismatch-gap* model. In this way, each connected sub-graph of this graph represents a pair of connected sub-graphs of the input ones. Weight of the edges are derived from the input conceptual networks without modifying them. Finally, the algorithm extracts the densest sub-graph by using a modified version (see below) of the Charikar algorithm [17]. Such densest sub-graph represents a connected graph in the physical network, and it is a solution for the initial problem.

C. ALIGNMENT GRAPH DESCRIPTION

The first step of the Algorithm 1 is based on a previous work on graph alignment [14], which has been improved to fulfil the DCS evaluation problem.

We explain the building of the alignment graph through an example. Let us consider two input graphs: a weighted graph $G_1 = (W, E)$, and an unweighted graph $G_2 = (V, E)$, as depicted in Figure 3. We build the alignment graph by considering input graphs and a set of similarity relations among nodes used as starting seeds.

Consider, for the sake of simplicity, two networks (conceptual and physical) with an equal number of nodes. Figure 3 shows these relationships as dashed lines connecting the nodes of the two graphs. The modified version of the alignment algorithm [14] works as follows. First, the algorithm builds a new node, defined as **composite node**, for each pair of nodes that are in a relationship. Each node of the alignment graph represents a pair of correspondent nodes. After this step, the algorithm adds the edges among nodes by examining the two input graphs. Given two nodes, a connection node is inserted whenever the corresponding nodes are connected respectively in conceptual and in physical networks. For instance, in Figure 3 both $(v1, v2)$ and $(w1, w2)$ are connected in input networks, hence the alignment graph will contain $(v1-w1)$ and $(v2-w2)$ nodes, where $w1, w2 \in W, v1, v2 \in V$ and $(v1 - w1)$ and $(v2 - w2)$ are the corresponding nodes in the alignment graph. This condition represents a *Match*. Therefore an edge is inserted between $(v1-w1)$ and $(v2-w2)$ and the weight of the edge between them will be equal to the weight of the edge between $w1$ and $w2$. Let us now consider nodes $(v3-w3)$ and $(v4-w4)$ of the alignment graph. Nodes $w3$ and $v4$ are adjacent while $v3$ and $v4$ are connected but not adjacent and let us suppose that the distance is below a given a threshold of distance $\delta = 4$, (e.g. distance less than 4). In this case, an edge will be inserted between nodes $(v3 - w3)$ and $(v4-w4)$ of the adjacent graph, while the weight of the edge is the average of the weights of the edges of the path linking $w3$ to $w4$. After the analysis of all node pairs, the final alignment graph is built, as represented in Figure 4. The steps described above are used to implement the line 1 of the Algorithm 1, returning AL graph. Such a result is obtained by running the algorithm *B*uilding the Weighted Alignment Graph, reported formally below also in pseudo code in the Algorithm 2.

The procedure `BuildAlignmentGraph` receives two networks, a set of relations among their nodes stored in the similarity file F and a threshold δ. It generates a weighted alignment graph WAG $= (G_{al}, E_{al})$. In the following, the Algorithm 2 in pseudo code is reported.

The similarity file F contains one-to-one relations among nodes belonging respectively to conceptual and physical networks of the dual network. The Algorithm 2, which implements the first line of Algorithm 1, scans the similarity file and for each pair of nodes, it builds a node of the alignment graph. Then it considers all pairs of nodes of AL. Given two nodes of the alignment graph $v_{al,1} = (v_1, w_1)$ and

Algorithm 2 BuildAlignmentGraph(G_c,G_p,δ,F)

Input: (G_c,G_p,δ,F)
Output: $WAG = G_{al}, E_{al}$ (weighted Alignment Graph)
1: BEGIN
Initialisation:
2: // Building Nodes of the Alignment Graph
LOOP Process: Scan F file
3: **for** each pair contained in F **do**
4: add Node to G_{al}
5: **end for**
6: // Building of Edges
7: **for** Each Edge $(n_1, n_2) \in G_{al}$ **do**
8: $E_{al} \leftarrow$ Analyse(G_c,G_p)
9: **end for**
10: **return** *WAG*
11: END

$v_{al,2} = (v_2, w_2)$, it adds a corresponding edge between them when the input nodes are adjacent in the two input networks. In this case, the weight of the edge in AL is the weight of the corresponding edge in the conceptual network.

Given two nodes $v_{al,1} = (v_1, w_1)$ and $v_{al,2} = (v_2, w_2)$ of the alignment graph, we say that there is a **gap** when the input nodes are adjacent only in the conceptual network and they are at distance lower than δ in the physical network. In this case, an edge will be inserted into AL, and the weight will be the average weight of the edges of the shortest path connecting them. Conversely, when they are at a distance greater than δ, no edge is inserted into the alignment graph. When δ is set to ∞, an edge will be inserted whenever the nodes are connected in the physical network (see function *Analyse*(), Algorithm 2, line 8).

D. DENSEST CONNECTED GRAPH EXTRACTION

We now describe how to obtain the DCS from the aligned graph, i.e. we explain line 2 of Algorithm 1. We improved the Charikar algorithm [17]. The latter produces a densest sub-graph S of given graph G by using a greedy approximation. The algorithm originally has been developed for unweighted graphs. The idea behind the algorithm is that the elimination of low degree vertices in an unweighted graph may produce a subgraph S having the desired properties. The algorithm starts by considering the whole graph G. For each iteration it identifies the minimum degree vertex $v_{min} \in G$ and it removes v_{min} from G. The algorithm stops when all the vertices have been removed from G. The sub-graph with maximum density is built and returned as output during the iterations. The algorithm can be modified to be used with a weighted graph by considering the weighted sum of the degree and weights [17]. We extend the Charikar algorithm for weighted graph, by using the definition of density reported below (also introduced in Table 1).

Let $G = (V, E)$ be an undirected graph with weighted edges and $S \subset V$ a sub-graph. Each node ($v \in V$) has a set of incident edges ($E(v)$) and each edge has an associated weight w. We define as $vol(v)$, $v \in V$ the sum of the weights of the edges incident to the node v, $vol(v) = \sum(E(v))$. We define as density of G $\rho(V)$ the ratio among the $vol(v)$ and the number of nodes of G:

$$\rho(V) = \frac{\sum_{\forall v \in V}(E(v))}{|V|}, \tag{1}$$

We use the above reported extension to the algorithm and include the two reported algorihms to find the DCS in a Python based program called DN-Aligner. Code and tests are available for download at https://github.com/hguzzi/DualNetworkAligner.

IV. CASE STUDIES

A. PROOF OF CONCEPT AND VALIDATION

In the next section, we show as a proof of principle results for the application of our algorithm to some synthetic networks that contain known DCS. We demonstrate that our findings have superior quality over other classical approaches. The quality of the results is evaluated in various ways: we first show the ability of our approach to recover known DCS by means of the measures of precision and recall, then we show that our solutions are better than other methods.

We build 200 test dual networks each one containing DCS ($DCS_{kn,i}, i = 1..200$). Each physical networks has 1000 nodes, and 5000 edges and the conceptual network has 1000 nodes and 5500 edges.

The quality of a result was evaluated by comparing each extracted DCS ($DCS_{ex,j}$) with each known $DCS_{kn,i}$. The DCS sensitivity (Sn_{DCS}) represents the coverage of a known DCS by its best-matching extracted DCS (the maximal fraction of nodes in the DCS found in a common extracted DCS). Reciprocally, the DCS-wise Positive Predictive Value (PPV_{DCS}) measures how well a given extracted DCS predicts its best-matching known DCS.

To estimate the overall correspondence between a result (a set of extracted DCS) and the collection of known DCS, we computed the weighted means of all PPV values (averaged over all extracted DCS) and Sn_{DCS} values (averaged over all known DCS). The resulting statistics, clustering-wise PPV and clustering-wise Sn, provide information about the quality. To integrate the two measures, we computed a geometrical accuracy (Acc_{DCS}), defined as the geometrical mean of the averaged Sn and PPV values.

Since classical clustering and community discover algorithms do not run in dual network, we applied them over conceptual networks; then we derived the induced sub-graph into the physical network. Finally, we reduced the cluster on the conceptual network to find a connected sub-graph into the physical one.

Then we calculated the same measures described before for each algorithm running on 200 synthetic networks. We used the MCL, MCODE [36], and Louvain algorithm on the conceptual network. Table 2 summarises the performance of the algorithms (i.e., DN-Aligner, MCODE, MCL and

TABLE 2. Performances on synthetic networks: average values are reported with their standard deviation.

Algorithm	PPV	SSN	ACC
DN-Aligner	0.75. ± 0.01	0.81 ± 0.02	0.78 ± 0.02
MCODE	0.62 ± 0.02	0.62 ± 0.02	0.62 ± 0.02
MCL	0.72 ± 0.03	0.65 ± 0.03	0.68 ± 0.03
LOUVAIN	0.82 ± 0.02	0.70 ± 0.01	0.75 ± 0.02

LOUVAIN), measured by using the average value evaluated on the runs over each of the 200 networks, respectively for PPV, SSN and ACC.

As evidence, DNAligner averaged over 200 networks outperforms the remaining 3 algorithms for the discovering of DCS.

We also performed additional tests, by varying the input network. We randomly added edges (as a matter of noise) and we generated 5 altered networks by randomly adding noise defined by 5%, 10%, 15%, 20%, 25% of new edges. Then we calculated same statistics as before. Results are summarised into Tables 3. We also evaluate the statistical differences among accuracy values and the values obtained by our algorithm are higher than the other algorithms through a non-parametric test.

This section presents some case studies on a social network, on a co-authorship network and on a biological network. As proof of concept we present three case studies on three different networks: (i) a social network, (ii) a biological network and (iii) a co-authorship network. In each study, we extract the densest connected graph. All experiments have been performed on a server with 16Gb Memory, Ubuntu OS and Intel Core i5 CPU.

B. EXPERIMENTS ON SOCIAL NETWORKS: THE GOWALLA DATASET

GoWalla is a social network where users share their locations (expressed as GPS coordinates) by checking-in into the web-site [37]. We downloaded data contained in SNAP datasets collection [38]. The whole network is undirected and it consists of 196,591 nodes and 950,327 edges. Each node represents a user and each edge link two friends into the network. In order to obtain a dual network we considered two possible networks starting from these data. The physical network represents the friendship network. Figure 7 depicts an extract of the dual network.

Therefore each user of GoWalla is represented by a node, while an edge represents a friendship relation derived from data. Since each user is associated with information about the position, we calculated the distances among the users expressed as distance among check-ins. In case of multiple check-ins we considered the average of all the check-ins. Then we normalised all the distances by considering the maximum distance among all the users. Therefore nodes representing users that may be considered close will be connected by edges having a weight close to one, while a weight close to zero will represent user whose positions are not close.

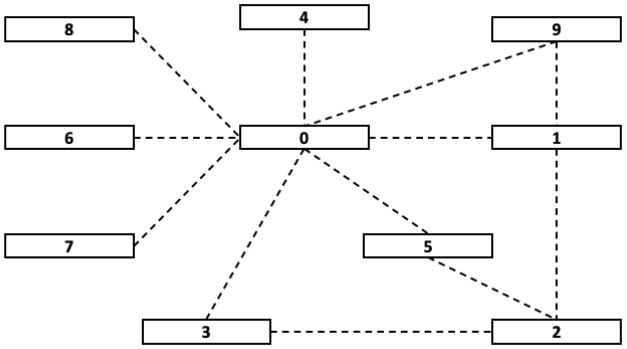


FIGURE 5. Physical instance.

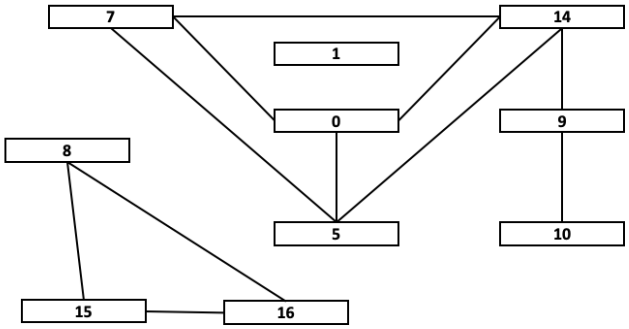


FIGURE 6. Conceptual instance.

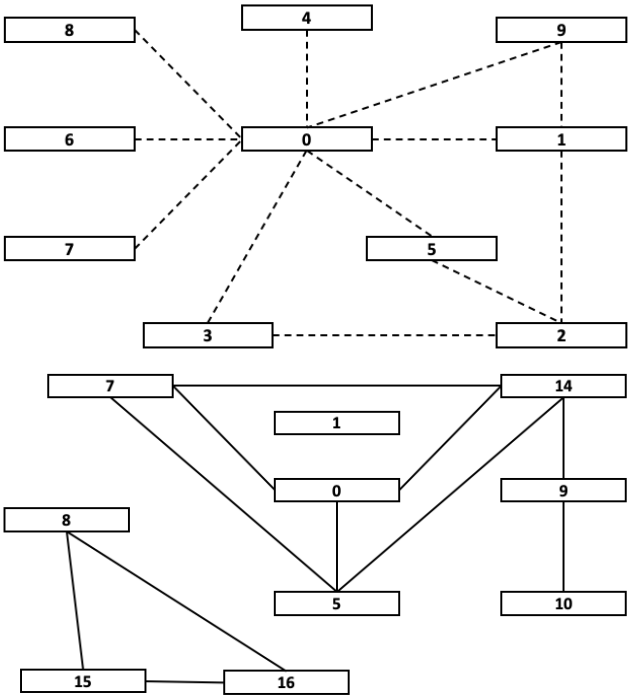


FIGURE 7. An extract of Go Walla network.

It should be noted that two users that are geographically near, they might be not friends and that two friends may be far geographically. A *Densest Common Sub-graph* in this case represents a set of users that are very close geographically and that are connected among them in a friendship network.

TABLE 3. Performances of DN-aligner versus MCODE, MCL and LOUVAIN measured on the PPV, SSN and ACC values, with the standard deviation evaluated by introducing randomly edges in the graph with introducing 5%, 10%, 15% 20% and 25% of variation noise.

% of added edges	Algorithm	PPV (± SD)	SSN (± SD)	ACC (± SD)
5%	DN-Aligner	0.71 ± 0.02	0.79 ± 0.01	0.74 ± 0.01
	MCODE	0.61 ± 0.02	0.58 ± 0.01	0.60 ± 0.0
	MCL	0.70 ± 0.01	0.61 ± 0.02	0.65 ± 0.02
	LOUVAIN	0.72 ± 0.01	0.69 ± 0.01	0.7 ± 0.01
10%	DN-Aligner	0.71 ± 0.02	0.75 ± 0.03	0.73 ± 0.03
	MCODE	0.60 ± 0.02	0.52 ± 0.02	0.56 ± 0.02
	MCL	0.68 ± 0.03	0.60 ± 0.03	0.64 ± 0.03
	LOUVAIN	0.71 ± 0.01	0.69 ± 0.01	0.70 ± 0.01
15%	DN-Aligner	0.70 ± 0.01	0.73 ± 0.01	0.73 ± 0.03
	MCODE	0.55 ± 0.02	0.5 ± 0.02	0.56 ± 0.02
	MCL	0.67 ± 0.03	0.58 ± 0.03	0.64 ± 0.03
	LOUVAIN	0.71 ± 0.01	0.69 ± 0.01	0.70 ± 0.01
20%	DN-Aligner	0.61 ± 0.01	0.65 ± 0.01	0.62 ± 0.03
	MCODE	0.51 ± 0.01	0.55 ± 0.01	0.45 ± 0.02
	MCL	0.61 ± 0.01	0.45 ± 0.01	0.51 ± 0.03
	LOUVAIN	0.61 ± 0.01	0.62 ± 0.01	0.57 ± 0.01
25%	DN-Aligner	0.61 ± 0.02	0.62 ± 0.02	0.61 ± 0.02
	MCODE	0.52 ± 0.02	0.45 ± 0.03	0.48 ± 0.03
	MCL	0.51 ± 0.02	0.51 ± 0.02	0.51 ± 0.02
	LOUVAIN	0.54 ± 0.02	0.57 ± 0.02	0.55 ± 0.01

The analysis of the conceptual network alone may miss all the information about friendships. The extracted DCS contains 2442 nodes and 149530 edges. This community represents a set of users that are friends and that are close from a geographical point of view.

C. EXPERIMENTS ON CO-AUTHORSHIP NETWORK

We evaluate our approach in a dual network representing authors and the similarity of the activity of their research. We use the DBLP dataset[1]. We considered published papers in five bioinformatics conferences: BCB, BIBM ISMB, RECOMB and EMBC. For each conference we extracted all the information about papers and authors. The dataset contains 20,563 authors.

The physical network represents co-authorship relations. Therefore each node represents an author and edge links two authors that have co-authored a paper. The conceptual network models the research interest similarity among authors, and it is constructed by analysing the similarity of the paper titles. We considered the Jaccard Index to compute the research interest similarity. We obtained two graphs having 20,563 nodes, and the physical network has 58536 edges while the conceptual network has 200530 nodes.

It should be evidenced that a dense sub-graph in the conceptual network represents a set of authors that have large similarity research interests that may be not collaborators considering the co-authorship networks. Therefore the analysis of the only conceptual network may miss information about the chain of collaborations evidencing the need for the use of dual networks. DN-Aligner tool found a DCS with 573 nodes and 95823 edges, Figure 8 depicts an extract of the found DCS. The DCS contains co-authors that share common research interests.

We also extracted a dense subgraph only in the conceptual network and we derived the induced subgraph in the co-author network. We obtained a graph with 1073 nodes and 198746 edges. We found that this graph is not connected in the physical network thus the analysis of only a network is missing many important information.

D. EXPERIMENTS ON BIOLOGICAL DATA: ANALYSIS OF PROTEIN INTERACTIONS

We considered data from the STRING database [39]. This database contains data about proteins and their interactions. Each node represents a protein, and each edge takes into account the reliability of the interaction between two proteins with a value in the interval $(0 - 1)$. Therefore, we obtained two networks:

- a conceptual network, which represents the strength of associations among proteins;
- a physical network, which stores the binary interactions among proteins.

We obtained two networks having 19.354 nodes and 5.879.727 edges. We ran DN-Aligner algorithm, and it

[1]The dblp team: dblp computer science bibliography. Monthly snapshot release of November 2019. https://dblp.org/xml/release/dblp-2019-11-01.xml.gz

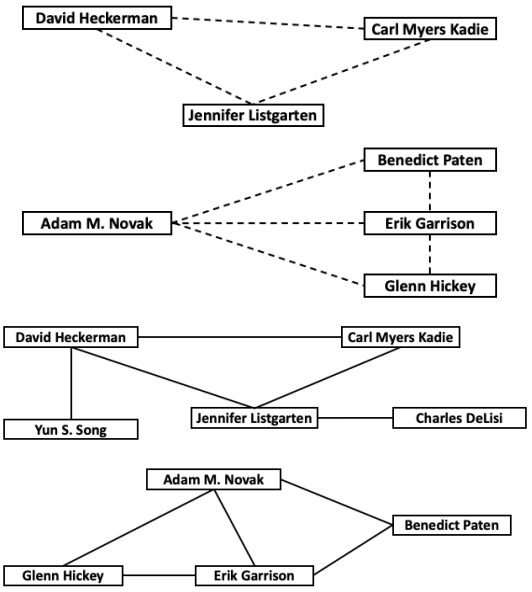


FIGURE 8. Dual networks (physical and conceptual) representing co-authorship and similar interests.

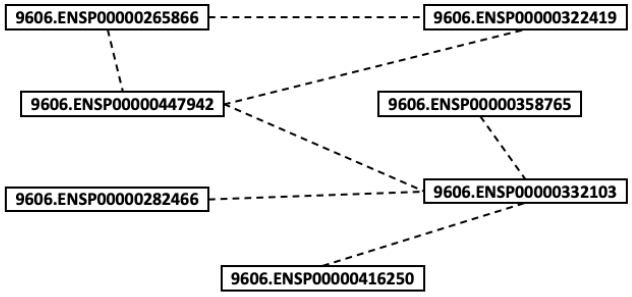


FIGURE 9. Physical network.

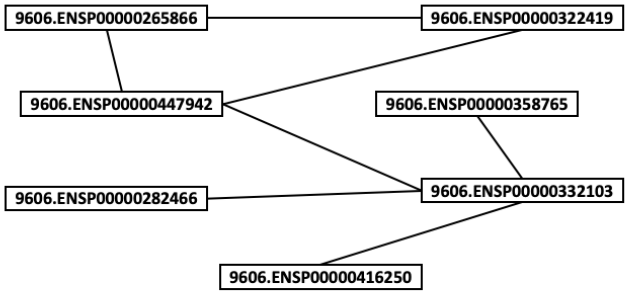


FIGURE 10. Conceptual network.

resulted in a DCS having 756 nodes and 154.142 edges. We performed a biological interpretation of the results by using a functional enrichment algorithm provided by the DAVID software [40]. Main enriched functions of the DCS are:

- GO:0006281 - DNA repair
- GO:0006302 - double-strand break repair
- GO:0070182 - DNA polymerase binding
- GO:0003676 - nucleic acid binding

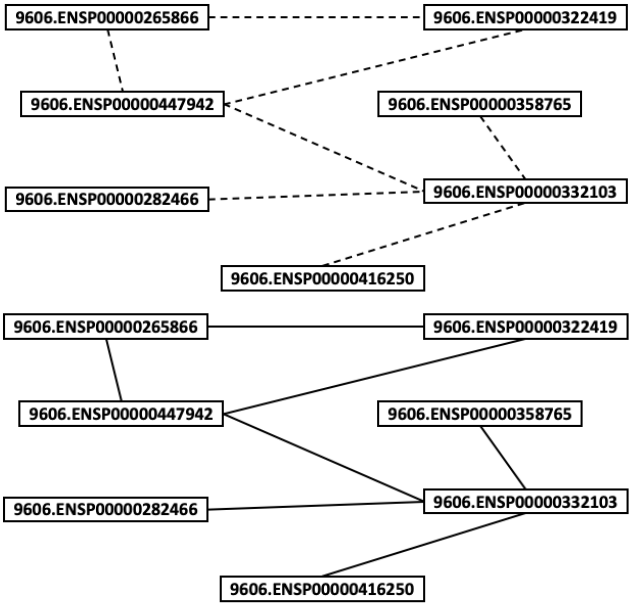


FIGURE 11. Dual networks representing biological data.

Similarly to social networks, we extracted the densest graph in the conceptual network, and we verified that the induced graph in the physical one is not connected.

V. CONCLUSION

In a dual network model a pair of graphs is used to model complex scenarios in which one of the two graph is unweighted (physical network) while the other is edge-weighted (conceptual network). In the present paper we presented an heuristic algorithm for obtaining the densest connected sub-graph (DCS) having the largest density in the conceptual network and being also connected in the physical network. We formalised the problem and we then mapped the DCS problem into a graph alignment problem. Finally, we proposed a possible solution and presented a set of experiments, which demonstrate the effectiveness of our approach.

REFERENCES

[1] M. Cannataro, P. H. Guzzi, and P. Veltri, "Protein-to-protein interactions," *ACM Comput. Surv.*, vol. 43, no. 1, pp. 1–36, Nov. 2010.

[2] M. T. Di Martino, P. H. Guzzi, D. Caracciolo, L. Agnelli, A. Neri, B. A. Walker, G. J. Morgan, M. Cannataro, P. Tassone, and P. Tagliaferri, "Integrated analysis of micrornas, transcription factors and target genes expression discloses a specific molecular architecture of hyperdiploid multiple myeloma," *Oncotarget*, vol. 6, no. 22, p. 19132, 2015.

[3] A. Sapountzi and K. E. Psannis, "Social networking data analysis tools & challenges," *Future Gener. Comput. Syst.*, vol. 86, pp. 893–913, Sep. 2018.

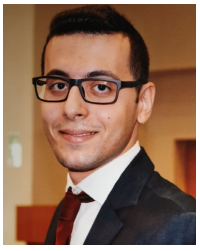
[4] C. Clark and J. Kalita, "A comparison of algorithms for the pairwise alignment of biological networks," *Bioinformatics*, vol. 30, no. 16, pp. 2351–2359, Aug. 2014.

[5] F. E. Faisal, L. Meng, J. Crawford, and T. Milenković, "The post-genomic era of biological network alignment," *EURASIP J. Bioinf. Syst. Biol.*, vol. 2015, no. 1, pp. 1–19, Dec. 2015.

[6] M. Cannataro, P. H. Guzzi, and A. Sarica, "Data mining and life sciences applications on the grid," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 3, no. 3, pp. 216–238, May 2013.

[7] M. Cannataro and P. H. Guzzi, "μ-CS: An extension of the TM4 platform to manage Affymetrix binary data," *BMC Bioinf.*, vol. 11, no. 1, p. 315, 2010.
[8] X. Liu, C. Shen, X. Guan, and Y. Zhou, "Digger: Detect similar groups in heterogeneous social networks," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 1, pp. 1–27, Jan. 2019.
[9] Y. Wu, X. Zhu, L. Li, W. Fan, R. Jin, and X. Zhang, "Mining dual networks: Models, algorithms, and applications," *ACM Trans. Knowl. Discovery Data*, vol. 10, no. 4, pp. 1–37, 2016.
[10] P. C. Phillips, "Epistasis—The essential role of gene interactions in the structure and evolution of genetic systems," *Nature Rev. Genet.*, vol. 9, no. 11, pp. 855–867, Nov. 2008.
[11] S. Tornow, "Functional modules by relating protein interaction networks and gene expression," *Nucleic Acids Res.*, vol. 31, no. 21, pp. 6283–6289, Nov. 2003.
[12] I. Ulitsky and R. Shamir, "Pathway redundancy and protein essentiality revealed in the saccharomyces cerevisiae interaction networks," *Mol. Syst. Biol.*, vol. 3, no. 1, p. 104, Apr. 2007.
[13] R. M. Karp, "Reducibility among combinatorial problems," in *50 Years of Integer Programming 1958-2008*. Berlin, Germany: Springer, Nov. 2009, pp. 219–241.
[14] M. Mina and P. H. Guzzi, "Improving the robustness of local network alignment: Design and extensive assessmentof a Markov clustering-based approach," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 11, no. 3, pp. 561–572, May 2014.
[15] P. H. Guzzi, M. Mina, C. Guerra, and M. Cannataro, "Semantic similarity analysis of protein data: Assessment with biological features and issues," *Briefings Bioinf.*, vol. 13, no. 5, pp. 569–585, Sep. 2012. [Online]. Available: http://bib.oxfordjournals.org/content/early/2011/12/02/bib.bbr066.short
[16] P. H. Guzzi and T. Milenković, "Survey of local and global biological network alignment: The need to reconcile the two sides of the same coin," *Briefings Bioinf.*, vol. 19, Jan. 2017, Art. no. bbw132.
[17] M. Charikar, "Greedy approximation algorithms for finding dense components in a graph," in *Proc. Int. Workshop Approximation Algorithms Combinat. Optim.* Cham, Switzerland: Springer, 2000, pp. 84–95.
[18] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal, "A survey of algorithms for dense subgraph discovery," in *Managing and Mining Graph Data*. Cham, Switzerland: Springer, 2010, pp. 303–336.
[19] S. Khuller and B. Saha, "On finding dense subgraphs," in *Proc. Int. Colloq. Automata, Lang., Program.* Cham, Switzerland: Springer, 2009, pp. 597–608.
[20] S. Parthasarathy, Y. Ruan, and V. Satuluri, "Community discovery in social networks: Applications, methods and emerging trends," in *Social Network Data Analytics*. Cham, Switzerland: Springer, 2011, pp. 79–113.
[21] X. Ma, G. Zhou, J. Shang, J. Wang, J. Peng, and J. Han, "Detection of complexes in biological networks through diversified dense subgraph mining," *J. Comput. Biol.*, vol. 24, no. 9, pp. 923–941, Sep. 2017.
[22] H. Hu, X. Yan, Y. Huang, J. Han, and X. J. Zhou, "Mining coherent dense subgraphs across massive biological networks for functional discovery," *Bioinformatics*, vol. 1, no. 1, pp. 1–9, 2005.
[23] J. Hastad, "Clique is hard to approximate within $n^{1-\varepsilon}$," in *Proc. 37th Conf. Found. Comput. Sci.*, 1996, pp. 627–636.
[24] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo, "The maximum clique problem," in *Handbook of Combinatorial Optimization*. Cham, Switzerland: Springer, 1999, pp. 1–74.
[25] A. Goldberg, "Finding a maximum density subgraph," Uni. California, Berkeley, CA, USA, Tech. Rep., 1984.
[26] Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama, "Greedily finding a dense subgraph," *J. Algorithms*, vol. 34, no. 2, pp. 203–221, Feb. 2000.
[27] V. Saraph and T. Milenković, "MAGNA: Maximizing accuracy in global network alignment," *Bioinformatics*, vol. 30, no. 20, pp. 2931–2940, Oct. 2014.
[28] V. Vijayan, V. Saraph, and T. Milenković, "MAGNA++: Maximizing accuracy in global network alignment via both node and edge conservation," *Bioinformatics*, vol. 31, no. 14, pp. 2409–2411, Jul. 2015.
[29] L. Meng, J. Crawford, A. Striegel, and T. Milenkovic, "IGLOO: Integrating global and local biological network alignment," 2016, *arXiv:1604.06111*. [Online]. Available: http://arxiv.org/abs/1604.06111
[30] R. Sharan and T. Ideker, "Modeling cellular machinery through biological network comparison.," *Nature Biotechnol.*, vol. 24, no. 4, pp. 427–433, Apr. 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16601728
[31] M. Koyutürk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama, "Pairwise alignment of protein interaction networks," *J. Comput. Biol.*, vol. 13, no. 2, pp. 182–199, Mar. 2006.
[32] J. Flannick, A. Novak, C. B. Do, B. S. Srinivasan, and S. Batzoglou, "Automatic parameter learning for multiple local network alignment," *J. Comput. Biol.*, vol. 16, no. 8, pp. 1001–1022, Aug. 2009.
[33] R. A. Pache and P. Aloy, "A novel framework for the comparative analysis of biological networks," *PLoS ONE*, vol. 7, no. 2, Feb. 2012, Art. no. e31220.
[34] G. Ciriello, M. Mina, P. H. Guzzi, M. Cannataro, and C. Guerra, "AlignNemo: A local network alignment method to integrate homology and topology," *PLoS ONE*, vol. 7, no. 6, Jun. 2012, Art. no. e38107.
[35] M. Milano, T. Milenković, M. Cannataro, and P. H. Guzzi, "L-HetNetAligner: A novel algorithm for local alignment of heterogeneous biological networks," *Sci. Rep.*, vol. 10, no. 1, pp. 1–20, Dec. 2020.
[36] G. Bader and C. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinf.*, vol. 4, no. 1, p. 2, 2003. [Online]. Available: http://www.biomedcentral.com/1471-2105/4/2
[37] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2011, pp. 1082–1090.
[38] J. Leskovec and A. Krevl. (Jun. 2014). *SNAP Datasets: Stanford Large Network Dataset Collection*. [Online]. Available: http://snap.stanford.edu/data
[39] D. Szklarczyk, J. H. Morris, H. Cook, M. Kuhn, S. Wyder, M. Simonovic, A. Santos, N. T. Doncheva, A. Roth, P. Bork, and L. J. Jensen, "The STRING database in 2017: Quality-controlled protein–protein association networks, made broadly accessible," *Nucleic Acids Res.*, Oct. 2016, Art. no. gkw937.
[40] D. Huang, B. T. Sherman, Q. Tan, J. R. Collins, W. G. Alvord, J. Roayaei, R. Stephens, M. W. Baseler, H. C. Lane, and R. A. Lempicki, "The DAVID gene functional classification tool: A novel biological module-centric algorithm to functionally analyze large gene lists," *Genome Biol.*, vol. 8, no. 9, p. R183, 2007.

PIETRO HIRAM GUZZI (Member, IEEE) received the Ph.D. degree in biomedical engineering from Magna Græcia University, Italy, in 2008. He has been an Associate Professor of computer engineering with Magna Græcia University since 2008. He has been a Visiting Researcher with Georgia Tech University, Atlanta. He has authored two books. His research interests include semantic-based and network-based analysis of biological and clinical data. He is a member of the ACM, BITS, ISMB, and NETBIO COSI. He is an Editor of a newsletter of the *ACM Special Interest Group on Bioinformatics, Computational Biology, and Biomedical Informatics* (SIGBio), and the IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS. He serves the scientific community as a reviewer for many conferences.



EMANUEL SALERNO received the degree from Magna Græcia University. He worked under the supervision of Prof. P. H. Guzzi. He currently works as a software developer in a private company.

GIUSEPPE TRADIGO was a Postdoctoral Fellow of the University of Calabria and a Visiting Research Fellow of the University of Florida, in 2016. He was a Visiting Researcher at the University of Dublin with Prof. G. Pollastri. He is an Associate Professor with Università degli Studi eCampus, Italy. His main research interests include bioionformatics, protein structure prediction, proteins modeling, and health informatics.

PIERANGELO VELTRI (Member, IEEE) received the Ph.D. degree in computer science from Paris XI, in 2002. He is an Associate Professor with Magna Græcia University. He worked as a Researcher at the INRIA, France, from 1998 to 2002, working on database models and query languages for semistructured data. He moved to Italy as a faculty member of medicine, in 2002, and served as assistant professor until 2010. His main interests include data modeling, protein and molecular modeling, spatial and geographic database systems, health informatics and biological modeling, agritech, and life data. He teaches database and clinical informatics systems, and he serves as an Editor of the *ACM Special Interest Group on Bioinformatics, Computational Biology, and Biomedical Informatics* (SIGBio) newsletter, and an Associate Editor of *BMC Medical Informatics and Decision Making* and the *Journal of Healthcare and Informatics Research*.