# Improving the Robustness of Local Network Alignment: Design and Extensive Assessment of a Markov Clustering-Based Approach

Marco Mina and Pietro Hiram Guzzi

**Abstract**—The analysis of protein behavior at the network level had been applied to elucidate the mechanisms of protein interaction that are similar in different species. Published network alignment algorithms proved to be able to recapitulate known conserved modules and protein complexes, and infer new conserved interactions confirmed by wet lab experiments. In the meantime, however, a plethora of continuously evolving protein-protein interaction (PPI) data sets have been developed, each featuring different levels of completeness and reliability. For instance, algorithms performance may vary significantly when changing the data set used in their assessment. Moreover, existing papers did not deeply investigate the robustness of alignment algorithms. For instance, some algorithms performances vary significantly when changing the data set used in their assessment. In this work, we design an extensive assessment of current algorithms discussing the robustness of the results on the basis of input networks. We also present AlignMCL, a local network alignment algorithm based on an improved model of alignment graph and Markov Clustering. AlignMCL performs better than other state-of-the-art local alignment algorithms over different updated data sets. In addition, AlignMCL features high levels of robustness, producing similar results regardless the selected data set.

**Index Terms**—PPI Network, network alignment, neighborhood topology, graph matching

## 1 Introduction

Molecular biology has focused on the study of relevant molecules (such as genes and proteins) on a system scale, considering the whole set of relationships intertwining them. The rationale is that proteins rarely work alone, but they form a complex network of interactions.

This new perspective encouraged the development of techniques for the determination, and consequent analysis, of the whole set of protein-protein interactions (PPIs) within organisms, also known as interactomes. The development of novel high-throughput technologies lead to the accumulation of a large amount of data, collected in several databases publicly available [1]. Formalism from graph theory provides the best framework to represent and analyze PPI data. A set of protein-protein interactions is generally modeled as a graph $G = \{V, E\}$, referred to as protein-protein interaction network (PIN), where $V$ is the set of labeled nodes representing the proteins, and $E$ is the set of edges representing protein interactions. The analysis of protein behaviors from a network perspective adds a new dimension to the understanding of the cellular machinery, since it exposes combinatorial effects otherwise not observable when considering single proteins alone [2].

- *M. Mina is with MPBA, Fondazione Bruno Kessler (FBK), Trento, Italy. E-mail: marco.mina@fbk.eu.*
- *P.H. Guzzi is with the Department of Surgical and Medical Sciences, Magna Graecia University of Catanzaro, Italy. E-mail: hguzzi@unicz.it.*

*Manuscript received 23 Sept. 2013; revised 21 Mar. 2014; accepted 10 Apr. 2014. Date of publication 17 Apr. 2014; date of current version 5 June 2014. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TCBB.2014.2318707*

More recently, the availability of PINs for different organisms fostered the extension of comparative studies to the interactome-level. Such analysis, formally known as *network alignment*, is the counterpart of sequence and structure alignment for primary and secondary structure of genes and proteins. In the basic formulation, it consists of finding common interaction patterns within the PINs of two (or more) species [3]. It should be noted that in this context the problem of network alignment presents some differences respect to classical graph alignment problems.

There are two different instances of the alignment problem. The *local network alignment* [4] searches relatively small similar subnetworks that are likely to represent conserved functional structures. Instead, the *global network alignment* looks for the best superimposition of the whole input networks (i.e., the alignment that minimizes a cost function). From a biological perspective, global alignment answers an evolutionary question, searching for a single comprehensive mapping of the whole set of protein interactions from different species. Instead, local alignment looks for evolutionary conserved building blocks of the cellular machinery, disregarding the overall similarity between the networks.

In recent years, many works focused on these problems, leading to the development of a plethora of algorithms (synopsis available in Table S1 of supplementary materials, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TCBB.2014.2318707). Despite the existence of different algorithms, to the best of our knowledge some important aspects have not been fully addressed. For instance, the robustness of algorithms has not been exhaustively assessed, as well as their general applicability and scalability.

In this work, we critically review local network alignment algorithms, and define an extensive assessment framework. We also introduce AlignMCL, a novel local network alignment algorithm based on Markov CLustering (MCL).

The paper is structured as follows. Section 2 critically reviews network alignment approaches, common alignment issues, and proposed solutions. AlignMCL is described in Section 3. Performance of local alignment algorithms are extensively assessed on 114 alignments in Section 5, following the assessment criteria defined in Section 4. Supporting data and AlignMCL code are available in the supporting website http://sites.google.com/site/alignmcl.

## 2 Local Network Alignment

There are many types of functional structures, henceforth referred to as modules, that have been conserved across evolution. Each module is represented by subgraphs with particular topological properties. For instance, many protein complexes are represented by densely connected subgraphs [5], [6]. Signalling pathways, instead, are usually linear chains of interactions, even though redundant paths might be present [7]. Hub Proteins are proteins with a significant number of interactors that play fundamental biological roles [1].

Modules from different organisms are functionally and evolutionary related if they share evolutionary related proteins (ortholog proteins), and present similar interaction patterns [8].

The local alignment of two organisms corresponds to identify, in their interactomes, the subgraphs representing modules evolutionary conserved. Conserved modules are expected to have:

- similar topologies in the two interactomes, according to some criterion (usually based on an evolutionary model [9], [8])
- meaningful interaction patterns, according to a given module model (i.e., for protein complexes [10])

Literature contains different formalizations of conserved module (as those proposed, for instance, in [4], [8], [11], [12]) that we generalize in the following formulation. Given two input graphs, $G_a = \{V_a, E_a\}$ and $G_b = \{V_b, E_b\}$, a correspondence between two regions of $G_a$ and $G_b$ can be expressed as a set of node pairs

$$S_i = (x^a, y^b) \mid x^a \in V_a \cup \eta,\ y^b \in V_b \cup \eta, \tag{1}$$

where $\eta$ is a fictitious symbol that means the associated protein has no ortholog in the other species. Let

$$S_i^a = \{x^a \mid (x^a, y^b) \in S_i\},$$

$$S_i^b = \{y^b \mid (x^a, y^b) \in S_i\},$$

be the sets of proteins belonging to $V_a$ and $V_b$, respectively, involved in $S_i$. Let $G_i^a$ ($G_i^b$) be the subgraph induced by $S_i^a$ ($S_i^b$) on $G_a$ ($G_b$).

The *pairwise local network alignment problem* consists of finding all the correspondences $S_i$ (henceforth referred to as solutions or clusters) that satisfy/maximize two criteria:

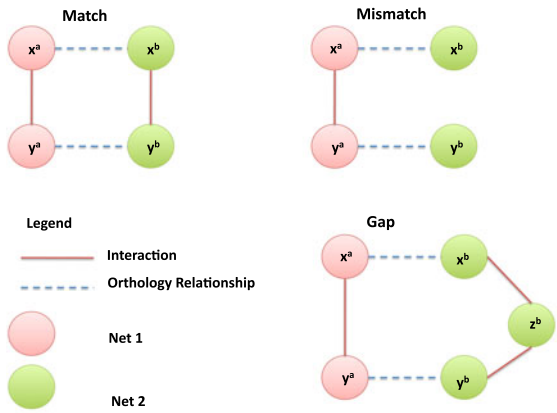



Fig. 1. Examples of match, mismatch, and gap.

- *similarity criterion*. $G_i^a$ is topologically similar to $G_i^b$, according to some measure
- *model criterion*. $G_i^a$ and $G_i^b$, or at least their common components, have meaningful interactions patterns, according to the selected module model.

The similarity criterion guarantees that matched subgraphs are topologically similar. The model criterion drives the analysis toward the identification of specific topologies, and depends on the specific module to be uncovered (i.e., protein complex, linear pathway). Both requirements must be satisfied simultaneously: two subgraphs might be topologically similar by chance, but the common topology might not be meaningful. Moreover, the model criterion usually influences the design of the similarity criterion.

The network alignment can be extended to the simultaneous alignment of multiple networks. Given $n$ graphs $G_1$, $G_2, \ldots, G_n$, a solution $S_i$ is defined as

$$S_i = \{(x^1, x^2, \ldots, x^n)\} \mid x^j \in V_j \cup \{\eta\}. \tag{2}$$

All the induced subgraphs $\{G_i^1, G_i^2, \ldots, G_i^n\}$ must satisfy similarity and model criteria, as in the pairwise version of the problem.

These formulations of network alignment support a node-centric perspective, since they describe alignments in terms of correspondences between proteins, rather than between protein interactions. They can be easily reformulated by replacing nodes with edges in the definition of $S_i$:

$$S_i = \{(x^a, y^b) \mid x^a \in E_a \cup \{\eta\},\ y^b \in E_b \cup \{\eta\}\}. \tag{3}$$

In this work we adopt a node-centric perspective, describing alignments as sets of orthologous proteins.

### 2.1 Similarity Criteria

Given two PINs $G_a = \{V_a, E_a\}$ and $G_b = \{V_b, E_b\}$, two pairs of ortholog proteins $(x^a, x^b)$ and $(y^a, y^b)$, with $x^a, y^a \in V_a$, and $x^b, y^b \in V_b$, are involved in a *conserved interaction* if

$$(x^a, y^a) \in E_a \quad AND \quad (x^b, y^b) \in E_b. \tag{4}$$

An interaction either present (conserved) or absent in both $G_a$ and $G_b$ between two corresponding orthologs is commonly referred to as a *match* (Fig. 1):

$$\begin{gathered}(x^a, y^a) \in E_a,\ (x^b, y^b) \in E_b \\ OR \\ (x^a, y^a) \notin E_a,\ \ (x^b, y^b) \notin E_b.\end{gathered} \tag{5}$$

On the contrary, there is a *mismatch* if an interaction is present in just one of the two networks:

$$\begin{gathered}(x^a, y^a) \in E_a, (x^b, y^b) \notin E_b \\ OR \\ (x^a, y^a) \notin E_a, (x^b, y^b) \in E_b.\end{gathered} \tag{6}$$

Given two subgraphs $S_i^a$ and $S_i^b$, induced on $G_a$ and $G_b$ by a putative solution $S_i = \{(x_i^a, y_i^b),\ i = 1, \ldots, n\}$, a simple similarity criterion would be requiring a perfect match between all the edges of $S_i^a$ and $S_i^b$. This criterion corresponds to the exact graph isomorphism test. More formally,

$$\begin{gathered}\left(x_i^a, x_j^a\right) \in E_a \Leftrightarrow \left(y_i^b, y_j^b\right) \in E_b \\ \forall \left(x_i^a, y_i^a\right), \left(x_j^b, y_j^b\right) \in S_i.\end{gathered} \tag{7}$$

In a real scenario this criterion is too strict, both for computational and biological reasons. As noted in [13] and [14], the knowledge about PINs is not complete. Current PINs present high values of missing or wrong interactions, and the isomorphism test is extremely sensible to both false positives and false negatives. The use of more flexible criteria is motivated also by biological reasons. In some cases, a (small) component of a conserved module might differ between different species. An interaction in one organism might work through an additional bridge protein in another (situation commonly referred to as a *gap*, as in Fig. 1). A rigid criterion is unable to deal with this situation.

More relaxed criteria allow gaps and mismatches between the induced subgraphs. Indeed, existing algorithms deal with missing interactions by introducing less restrictive similarity criteria, i.e., by verifying whether corresponding orthologs are at a distance less than or equal to $k$ in the original PINs, instead of checking only for perfect matches.

### 2.2 Model Criteria

The discovery of all conserved modules presents some difficulties due to the following reasons:

- some types of modules are not easy to define topologically (i.e., alternative routes in pathways),
- some interactions, such as protein-ligand and protein-DNA interactions, and not represented in PINs (i.e., transcriptional regulation),
- protein-protein interaction networks are quite noisy, thus impairing the high quality identification of some types of modules (i.e., linear pathways).

Many paper describing local network alignment algorithms focus on protein complexes, since they are somehow easier to study due to their topological properties. It has indeed been suggested that protein complexes are represented in PINs by densely connected subgraphs [5], and a simple model criteria requiring induced subgraphs to be densely connected should be able to identify them. More flexible models have been considered as well, such as requiring members of induced subgraphs to be more connected between each other than to the rest of the network. This approach is in line with recent findings on the modularity and the organization of complexes, suggesting that complexes in PPI networks consist of a core and (many) attachments [10]. The core is defined as a small group of proteins that are functionally similar and have highly correlated transcriptional profiles. The core is surrounded by less strongly connected proteins, defined attachments. Attachments can bind to multiple complexes, allowing the same complex to perform several potential functions. Flexible models are able to separate core components in different solutions that overlap in the presence of shared attachments [15].

### 2.3 Local Network Alignment Algorithms

Several algorithms have been proposed to detect modules in PINs [3] (synopsis available in Supplementary Table S1, available online). Most network alignment algorithms follow one of these two paradigms:

1) mine-and-merge: first analyze each PIN separately, and then project solutions reciprocally from a PIN to the others [16], [17];
2) merge-and-mine: analyze PINs together, usually after the alignment or merging in a single graph [15], [18], [11], [19], [12], [20].

In general, merge-and-mine algorithms are more complicated due to difficulties in formulating and accounting for approximate matches, and the existence of multiple mappings between proteins in different species [16]. Moreover, they are computationally expensive, since in order to merge the input networks it is necessary to compare their topologies. Mine and merge analysis alleviates this problem by avoiding one-to-one comparison of network topologies [16]. On the other side, mine-and-merge approaches do not completely exploit the presence of redundant information in the two networks. They might be more sensible to noise, and miss protein complexes not well topologically represented in the networks.

A common issue of several algorithms is the identification of ortholog pairs. In the general definition of local network alignment, any protein of $G_1$ can be matched to any protein of $G_2$. Without any additional information on the possible orthologs, local network alignment problem is extremely complex, since all the possible combinations of protein pairs should be considered. Therefore most algorithms require as additional input data a list of putative orthologs $O = \{(x, y), x \in V_1, y \in V_2\}$. This set is usually determined by sequence similarity, using tools such as BLAST [21] or BLAST+ [22].

More recently semantic similarity (SS) (i.e., functional similarity among proteins derived from existing biological ontologies) [23] has been used to define the input sets of orthologs as well [24]. The definition of solution $S_i$ can be reformulated to take into account the additional input information as

$$S_i = \{(x^a, y^b) \mid (x^a, y^b) \in O\}. \tag{8}$$

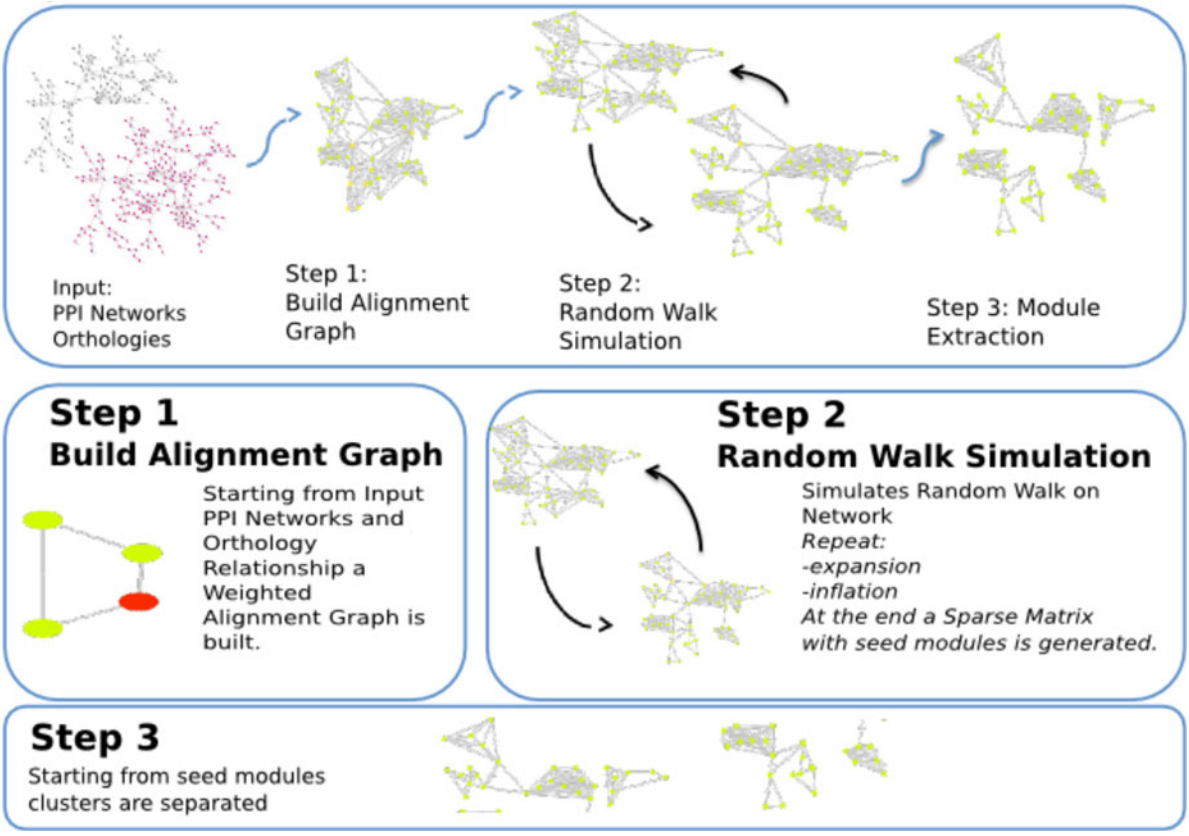



Fig. 2. AlignMCL pipeline.

Orthologies are not required to be exclusive (a protein from $V_1$ can correspond to several proteins in $V_2$, and vice versa). It is worth noting that some algorithms have been designed to work entirely on topological data, and they often produce global alignments [25], [26], [27].

Several merge-and-mine algorithms follow a similar processing scheme: merge all input data (PINs, putative homologies) in a single weighted undirected graph, generally referred to as *alignment graph*, and then apply a mining heuristic on top of it to identify conserved modules. This two step procedure is functional to split the problem in two parts, the first constraining the alignment following the similarity criterion, and the second identifying conserved modules according to the model criterion. The alignment graph serves the following purposes:

- expose structures common to different PINs, and discard components without any correspondence between the two networks,
- reduce the complexity of the alignment problem to the analysis of a single graph,
- to de-noise the input PINs.

## 3 AlignMCL

This work introduces AlignMCL, a local network alignment algorithm following the merge-and-mine approach. The pipeline is illustrated in Fig. 2 and in Algorithm 1. Input data are merged together into an alignment graph (merge step), afterwards analyzed to identify conserved modules (mine step). This section discusses the rationale behind the selected model of alignment graph, the mining heuristic, and the implementation.

**Algorithm 1** AlignMCL

**Require:** $G_1, G_2$ // two PPI networks
**Require:** $H$ // protein orthologs
**Require:** $T_p$ // pruning threshold
**Require:** $I$ // MCL inflation level

*Merge Step (Build Alignment Graph):*
$UG \leftarrow$ Build Union Graph$(G_1, G_2, H)$
$AG_{raw} \leftarrow$ Create Raw Alignment Graph $(UG)$
$AG \leftarrow$ Prune $(AG_{raw}, T_p)$

*Mine Step: (Random Walk Simulation and Module Extraction.)*
$S \leftarrow MCL(AG, I)$ *Markov Clustering on AG*

**return** $S$

### 3.1 Merge Step-Alignment Graph Model

Several models of alignment graph have been proposed, their design driven by different topological and similarity criteria. In its simplest formulation, nodes of an alignment graph correspond to pairs of putative orthologs, and edges represent potentially conserved interactions (Supplementary Fig. S1, available online). The presence of noise in input PINs, both in terms of missing or wrong interactions, negatively influences the identification of conserved interactions

(a proof-of-concept example is illustrated in Supplementary Fig. S2, available online).

To deal with false negatives, some authors introduced relaxed definitions of potentially conserved interaction. NetworkBlast, for instance, allows two nodes of the alignment graph (that is, two pairs of orthologs) to be connected if proteins are at a distance less than or equal to $k$ (usually $k = 3$) in one of the original PINs, and are direct neighbors in the other [18], [4]. If not carefully tuned, this approach might introduce many unreliable links in the alignment graph, leading to incorrect solutions even for small values of $k$. First of all, current PINs are affected by high levels of false positives. This means that two proteins are likely to be connected by a bridge protein through false interactions. Moreover, the model is likely to provide worse and worse results as available PINs get more and more complete. In an ideal case with no noise, several wrong interactions would be added as conserved. We indeed verified that NetworkBlast produces extremely dense alignment graphs with many misleading edges on the up-to-date PINs used in our assessment.

A more constrained model of alignment graph is proposed in MaWish [11], where the $k$ threshold is set to 2. The model proposed by MaWish employs a strategy to weight the edges of the alignment graph, based on the similarity scores of the putative orthologs. While effective, this model may be too strict, leading to small conserved structures and failing in recovering larger complexes on older and sparser PINs [15].

NetAligner [19] adopts a two-steps procedure for building the alignment graph. First, a high constrained alignment graph containing only conserved and likely-conserved interactions is built. This first graph is used to divide the network in promising seeds. Then, the alignment graph is extended predicting potential conserved interactions, overcoming the issue of the false negatives in the input data. This procedure carefully limits the amount of new interactions that can be added, forbidding the creation of new intra-seed interactions. Even NetAligner considers the shortest path distance to decide whether or not to add a new edge in the alignment graph.

Models proposed in NetworkBlast, MaWish, and NetAligner share the use of the shortest path distance to infer new connections. A natural extension to these models would be considering the number of paths connecting two pairs of proteins, rather than their distance. This is the idea on which is based AlignNemo [15]. For each pair of orthologs, the number of short paths (of length at most 2) connecting them is used to evaluate how likely the orthologs are connected in both the species. In this process, AlignNemo also takes into consideration the degree of each protein, penalizing paths passing through hubs and high-degree proteins. The impact of false positives on the construction of the alignment graph is greatly reduced, since it is unlikely that many false interactions consistently form short redundant paths between two proteins.

Almost all the proposed models associate a score to each edge and node of the alignment graph. AlignNemo assigns higher scores to edges involving protein pairs connected by multiple short paths, and MaWish assigns edge scores relying on the sequence similarity of the connected proteins. NetAligner, instead, is based on an evolutionary and probabilistic model: interacting proteins evolve at rates significantly closer than expected by chance. Thus, edge scores in the alignment graph represent probabilities, and are influenced by the difference of evolutionary distances between the corresponding ortholog pairs.

AlignMCL adopts the alignment graph model proposed with AlignNemo. The model was selected after performing some preliminary tests (on updated PINs) that support the observations reported in this section. Details about the model and its construction steps (creation of the union graph, determination of the raw alignment graph, and the pruning step) are available in Section *Alignment Graph model* of the supplementary materials, available online.

### 3.2 Mine Step - Detecting Conserved Modules

Many algorithms start the detection of conserved modules by selecting promising starting regions (generally referred to as seeds) and extending them in consecutive steps. The procedures try to maximize/minimize an ad hoc cost function based on node and edge scores, and terminate when the solution reaches a local optimum or some terminating condition is met. For instance, NetworkBlast maximizes a likelihood ratio that represents the probability of current solution being not drawn at random, under the hypothesis that protein complexes are densely connected (tend to form cliques). MaWish, instead, addresses the network alignment identifying the maximum weight induced subgraphs in the alignment graph, selecting those subgraphs that exceed a predefined score.

Both NetworkBlast and MaWish add/remove a single node from the current solution at each step. In some cases, employing a strategy that considers multiple nodes at once, instead of greedily trying to add single nodes to an expanding solution, provides better results [15], [19]. Indeed, during the expansion some proteins might be not "enough connected" to the current seed when considered alone, but present a significant net of connections if considered together.

To circumvent the problem, NetAligner identifies solutions in a single step, identifying the solutions in the connected components of the final alignment graph. AlignNemo is still based on the idea of expanding promising seeds, but considers groups of proteins at each step. More precisely, it tries to add the 4-subgraphs that surround the current solution. This allows to explore the network context of a solution beyond its immediate neighbors.

The size of PPIs databases has increased considerably in recent years, with some PINs doubling the number of interactions. Not all the alignment algorithms are able to work on the new data sets, such as those based on the enumeration of graphlets, or on too relaxed definitions of conserved interactions. The mining heuristic implemented in AlignNemo, for instance, is not scalable on the size of current PINs, since it requires all four-subgraphs to be enumerated. Large execution times and high levels of memory consumption are required by AlignNemo to process large networks.

After examining different solutions, we decided to use a more scalable approach for mining the alignment graph, based on Markov Clustering.

#### 3.2.1 Markov CLustering

Markov CLustering [28], [29] is a well known algorithm used to find clusters on graphs, robust to noise and graph alterations. Brohée and van Helden demostrated in an extensive comparison [30] that MCL outperforms other clustering algorithms, such as MCODE [5], RNSC [31] and Super Paramagnetic Clustering [32], in different conditions and using suboptimal parameters. More recently, MCL has been employed in a mine-and-merge alignment algorithm [17] to identify protein complexes on single PINs. However, to the best of our knowledge, MCL has never been used on alignment graphs.

The rationale behind MCL is quite simple. A possible way to define a module within a network is as a collection of nodes that are more connected with each other than to the others. It follows that a random walk starting in any of these nodes is more likely to stay within the cluster rather than to travel between clusters. By simulating many random walks starting from the various nodes, it is possible to identify flows of random walks that tend to gather in specific regions of the graph (the modules). MCL is an iterative algorithm that simulates random walks using Markov chains.

The simulation is performed by iteratively applying two main operations, usually referred to as expansion and inflation. The expansion step simulates, for each node of the graph, a stochastic flow spreading out from the node toward all the other nodes. The step is performed by repeatedly multiplying the normalized adjacency matrix of the graph by itself. Nodes connected by multiple (and shorter) paths will be the endpoints of stronger flows. The inflation step introduces a modification into the process, enhancing flows within clusters and weakening inter-cluster flows. This is done by stopping the expansion step, and computing the Hadamard power of the adjacency matrix.

The initial distribution of flows becomes more and more non-uniform as the process is repeated, terminating when a steady state is reached. The resulting adjacency matrix is a very sparse matrix where modules can be easily identified. A more detailed mathematical formulation of the steps performed to extract the modules is provided in the supplementary materials, available online. The complete description of the algorithm can be found in [28].

The most important parameter of MCL is the Inflation level (I), that is the exponent used in the Hadamard powering operation. As a rule of thumb, the higher the inflation parameter the smaller the average dimension of clusters, since the inflation step will increasingly penalize weaker flows as the inflation level increases. MCL supports weighted input graphs. Edges' weights are taken into account when the first stochastic matrix used in the iterative process is defined. When MCL is applied to an alignment graph the edges weights, representing the likelihood of ortholog pairs to be interacting, are taken into account.

### 3.3 Implementation

AlignMCL is implemented in two components. The first component processes input PINs and orthologies to create the alignment graph. The Java-based implementation used by AlignNemo to build the alignment graphs is too slow and requires too much memory to deal with PINs available nowadays. Therefore, it has been fully reimplemented in Python and further enhanced. The new version is able to process all the alignments, requiring less than half an hour (in the worst case) per alignment on a standard laptop equipped with 4 GB of ram. Current implementation runs on Python 2.x and requires the igraph library.

The second component of AlignMCL is the MCL clustering engine. We decided to rely on the MCL implementation by van Dongen [28], selecting it for its speed, robustness and reliability. Software is available at http://sites.google.com/site/alignmcl.

## 4 Assessment Guidelines

A plethora of indices has been used to evaluate the ability of algorithms to select orthologs, interactions, and functional modules (i.e., complexes or pathways) conserved in different species. Previous works combined such indices in more or less extensive assessments, selecting the more appropriate to the proposed analysis. For instance, counting the number of interactions of input networks recapitulated by alignment's solutions is fit for evaluating a global network alignment [33].

Considering the local network alignment problem, an ideal assessment consists of measuring the ability of alignment's solutions to recapitulate a set of a-priori known conserved modules. Unfortunately, knowledge on protein modules varies a lot across different species, and some modules do not have a (known) counterpart in other species. Several strategies have been proposed to evaluate alignment algorithms despite this lack of knowledge. More in general, an alignment can be analyzed from an intra-species or an inter-species point of view. An intra-species agreement tells, for instance, whether the proteins collected in a single solution belong to a known module. Instead, an inter-species assessment tries to understand whether the putative orthologs within a single cluster share some functional roles. Even though the inter-species evaluation is more appropriate, an intra-species analysis might be preferable when there is not much information available on one of the two aligned species. Indeed, most works assessed algorithms performance relying mainly on intra-species analysis [18], [12], [20]. It is our opinion that an alignment algorithm should be validated in both the senses. In this work we here follow a similar approach to those proposed in Pache and Aloy [19].

Supplementary Table S2, available online, summarizes the assessments used in previous works for evaluating local network alignment algorithms.

### 4.1 Intra-Species Assessment: Comparison with Known Modules

Given a (partial) knowledge of modules present in at least one of the aligned organisms, it is rather intuitive thinking of evaluating to which extent solutions provided by an algorithm resemble them.

Given a solution $S_i$ and a known module $M_j$, there are several ways to compare them. A simple approach would be looking for a summary agreement, such as the number or ratio of proteins in overlap between $S_i$ and $M_i$ [15], [19]. A finer measure would consider the internal connections between the proteins as well, as done in Pache and Aloy

work [19]. In this work we opted for the simpler strategy. In fact, data sets of known modules are not always annotated with fine information on their internal topology. Moreover, solutions provided by current alignment algorithms are often not enough specific to reconstruct the internal topology with high quality, mainly due to the noise of input data. Therefore, a fine comparison might not be suited for most of the cases (indeed previous works relied on the simpler strategy).

In general, the quality of the overlap between two sets (a solution and a module, for instance) can be quantified through precision ($\pi$, also called Positive Predictive Value) and recall ($\rho$, also known as Sensitivity). Precision represents the percentage of proteins in the solution that are also present in the module, while recall measures the percentage of proteins in the module that are in common with the solution. The two measures can be integrated into the F-index function, defined as the harmonic mean of precision and recall as proposed by Pache and Aloy in [19].

More formally, given two sets $S_i$ and $M_j$, with $|S_i| > 0$ and $|M_j| > 0$,

$$\pi = \frac{|M_j \cap S_i|}{|S_i|}, \ \rho = \frac{|M_j \cap S_i|}{|M_j|}, \text{F-index} = \frac{2\pi\rho}{\pi + \rho}. \quad (9)$$

The F-index ranges in the interval [0, 1], with 1 corresponding to perfect agreement.

The comparison of two sets ($S_i$, $M_j$) can be extended to the comparison of a set of solutions $\{S_i\}$ and a set of known modules $\{M_j\}$. Given a solution $S_i$ and a set of a-priori known modules $M = \{M_j\}$ for species $a$, the best matching module for solution $S_i$ is defined as

$$B_{S_i} = \underset{j}{argmax}\, \text{F-index}\ (S_i, M_j). \quad (10)$$

Given a set of solutions $S = \{S_1, S_2, \ldots\}$, the vector of best matching solutions for $S$ is defined as

$$BestMatch(S, M) = \{B_{S_1}, B_{S_2}, \ldots\}, \quad (11)$$

with the corresponding F-indices

$$F(S, M) = \big\{\text{F-index}\ (S_1, B_{S_1}),\ \text{F-index}\ (S_2, B_{S_2}), \ldots\big\}. \quad (12)$$

Two algorithms can be compared by considering the corresponding vectors of best matching modules for the solutions provided.

### 4.2 Inter-Species Assessment: Employing Semantic Similarity

In general, modules are groups of interacting proteins that share common functions or play similar biological roles. For instance, a biological pathway is a number of biochemical steps, linked together, that perform a process inside cells.

GO functional enrichment [34] has been used to evaluate the significant presence of common functions in the solutions. Functional enrichment generally considers proteins from the same organism, and the inter-species comparison is usually performed by checking for common enriched functions. This approach has some drawbacks, since in general similar functions are considered as not corresponding at all. Moreover, there are some biases introduced by the size of the assessed sets (see [34] for a complete discussion).

We already proposed the use of Semantic Similarity [15] to address these problems. SS measures are able to quantify the functional similarity of pairs of proteins/genes, comparing the GO terms that annotate them. Thus, there are no constraints on the minimum set size [23]. Since proteins within the same pathway are involved in the same biological process, they are likely to have high semantic similarity. In a similar way, protein belonging to the same complex, and in the same solution should have a semantic similarity significantly higher than random expectation.

Given a solution $S_k$, its inter-species semantic similarity $SS_i(S_k)$ is defined as

$$SS_i(S_k) = \frac{\sum_{x_i \in S_k^1} \sum_{y_j \in S_k^2} SS(x_i, y_j)}{|S_k^1||S_k^2|}, \quad (13)$$

where $SS(x_i, y_i)$ is the semantic similarity between proteins $x_i$ and $y_j$.

Note that in general $|S_k^{1,2}| \leq |S_k|$, since a protein can appear in more than one association.

The inter-species semantic similarity can be directly used to compare the quality of different solutions and, by extension, algorithms. It is worth noting, however, that smaller solutions are more likely to have higher SS scores [15], [35].

Real alignments can be compared against random ones, in order to prove their statistical significance. Given a solution $S_i$, we can test the null hypothesis $H_0^1$: *the inter-species semantic similarity $SS_i(S_i)$ is drawn from the background distribution*, where the background distribution can be estimated from the $SS_i$ of random solutions. As usual, the hypothesis can be rejected if the results p-value is lower than a given threshold, usually set to $0.05$ or $0.001$.

This approach is useful to prioritize or filter the solutions in a post-processing step. However, if the purpose is to validate the entire alignment, this approach presents two issues. First, instead of a single p-value, many are returned, and merging them is not straightforward. Second, the p-values need to be corrected for multiple hypothesis testing.

Let $A = \{A_1, A_2, \ldots A_n\}$ be the set of solutions of a given alignment problem. Let's define

$$SS_i(A) = \{SS_i(A_1), SS_i(A_2), \ldots SS_i(A_n)\}, \quad (14)$$

as the semantic similarity profile of $A$. We would like to test the following null hypothesis $H_0^2$: *the inter-species semantic similarity profile $SS(A)$ of alignment $A$ has the same distribution of the similarity profiles of random alignments*. Algorithm 2 evaluates a single p-value for the whole alignment, testing the statistical significance of the entire alignment.

A non-parametric test is used in Algorithm 2, since the distribution of inter-species semantic similarity scores does not follow a normal distribution (data not shown).

### 4.3 Intra-Species Assessment with Semantic Similarity

It is straightforward to define an intra-species agreement based on semantic similarity.

Given a solution $S_k$, the intra-species semantic similarity of $S_k$ is separately defined on the two species as

$$SS_1(S_k) = \frac{\sum_{x_i \in S_k^1} \sum_{y_j \neq x_i \in S_k^1} SS(x_i, y_j)}{|S_k^1 \| S_k^1 - 1|}, \quad (15)$$

and

$$SS_2(S_k) = \frac{\sum_{x_i \in S_k^2} \sum_{y_j \neq x_i \in S_k^2} SS(x_i, y_j)}{|S_k^2 \| S_k^2 - 1|}. \quad (16)$$

It is straightforward to extend the statistical test performed by Algorithm 2 on $SS_i$ to $SS_{1,2}$.

**Algorithm 2** Test $H_0^2$

1: $Size(A) = (|A_1|, |A_2|, ..., |A_n|)$ *Build size profile*
2: Generate 1000 random solutions $R_i$ such that:
- $|R_i| = n$,
- $Size(R_i) = Size(A)$.

$R_i$ *is a group of random sets* $\{R_{i,j} \ : \ |R_{i,j}| = |A_i|\}$
3: Evaluate $SS_i(A)$
4: Evaluate $SS_i(R_i)\ \forall R_i$
5: Merge all $SS_i(R_i)$ into a single vector $SS_i(R)$
6: Compare $SS_i(A)$ and $SS_i(R)$ with the Mann-Whitney (Wilcoxon rank-sum) test and estimate the p-value $P_{SS_i}(A)$
7: Reject the null hypothesis $H_0^2$ if $P_{SS_i}(A) \leq 0.05$

### 4.4 General Applicability and Robustness

Interactomes differ significantly from each other both in terms of completeness and reliability, as some species are more studied than others. Knowledge about protein interactions is non-uniform [13], [14], and current PINs present high values of missing or wrong interactions. Y2H screens have false negative rates ranging from 43 to 71 percent, and TAP has false negative rates of 15-50 percent [36]. False positive rates for Y2H can be as high as 64 percent, increasing to 77 percent in TAP experiments [36]. Newer high throughput screening techniques, instead, are characterized by lower false positive and false negative rates [37]. Some of the currently available PPI data sets contain only interactions from high-quality experiments, while others mix interaction data provided by experiments with different reliability. Considering the high variability, evaluating the performance of an algorithm on different PINs is fundamental to avoid overfitting and prove its general applicability. However, general applicability and robustness of alignment algorithms are two aspects often overlooked (synopsis available in Supplementary Table S2, available online), as performance are usually evaluated on few alignments. In the context of this work an algorithm is robust if it provides similar results regardless the data sets used as input data. The availability of different PINs for the same organism allows to investigate the robustness of alignment algorithms.

## 5 Experiments

AlignMCL was compared to the state-of-the-art algorithms MaWish [11], NetAligner [19] and NetworkBlast [4], following the indications from the previous section. All the algorithms require two PINs and a set of putative orthologs as input data. Graemlin [12] and Graemlin 2.0 [20] have not been considered because they require additional data to learn some alignment parameters, and these are not readily available for all the organisms. The parameters used for each alignment algorithm are reported in the supplementary material, available online.

In this section the results of an extensive assessment based on 114 alignments between several PINs of five different species are described. The comparison not only demonstrates that AlignMCL outperforms the other algorithms, but also that it is more stable when different PINs of the same organisms are used.

NetworkBlast was unable to deal with the size and complexity of current PINs. We tried to run NetworkBlast on a Linux CentOS Cluster, equipped with two Intel Xeon processors and 8 GB of RAM, and for some alignments it required more than a week to complete. In all the alignments, when able to conclude the computation, it produced few random solutions. The alignment graph built by NetworkBlast was analyzed to understand the reasons of this behavior. When aligning yeast and fly PINs it produced alignment graphs with almost 1 million of edges between 7,000 nodes. This confirms our hypothesis that it adopts a definition of alignment graph too relaxed for current PINs, resulting in an extremely dense alignment graph. NetworkBlast's results are not shown in the following comparisons.

In [15] it had been suggested that MaWish generated small solutions due to a rather strict definition of alignment graph. In this assessment, instead, MaWish performed quite well, producing sounds solutions.

### 5.1 Input Data Set

#### 5.1.1 Protein Interaction Networks

We built an extensive data set including 20 PINs from five of the most studied species: Drosophila Melanogaster (fly), Saccaromices Cerevisiae (yeast), Homo Sapiens (human), Cenhorabditis Elegans (worm), and Mus Musculus (mouse)—for a total of 114 different alignments. Up-to-date PINs have been downloaded from i2d [38], DroID [39], Hint [40], HIPPIE [41], WID [42], and DIP [43] databases. Edges in all the networks are weighted according to the reliability scores provided by the input data sets.

Since current PINs are still incomplete, integrating multiple data sources to build more complete and reliable networks can have beneficial effects on the analysis [44]. Indeed two of the PINs selected in this analysis, i2d and DroID, integrate several data sets including IntAct, BIND, MINT, and BioGRID. For each species all PINs were further integrated in a *merged* network. Statistics for the 20 PINs are reported in Supplementary Table S3, available online.

A comparison in terms of number of interactions and network density is shown in Fig. 3. Network density $d$ is a global graph statistics defined as

$$d(G) = \frac{2|E|}{|V|(|V|-1)}. \quad (17)$$

PINs vary a lot in terms of number of interactions, even for the same organism: human PINs range from $\approx$ 4k (DIP) to $\approx$ 185k interactions (merged). A similar trend is observable for yeast and fly. DIP networks count the lowest numbers of

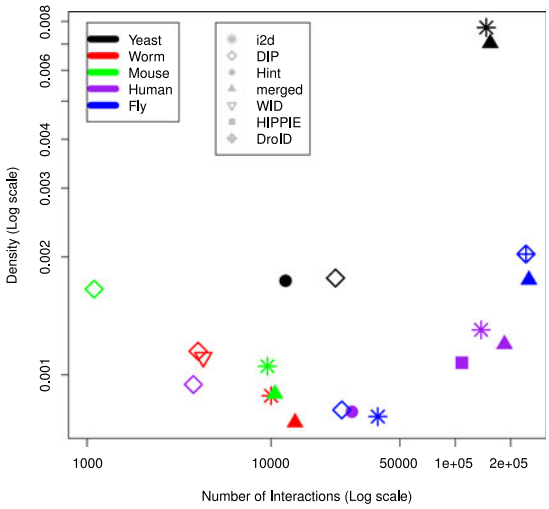



Fig. 3. Log-plot of density and number of interactions of each PIN used in the assessment.

interactions, with the exception of yeast. On the contrary, i2d networks are the most complete but for fly. Density values range between 0.001 and 0.002, with the only exception of yeast i2d network ($\approx$ 0.008). For human, mouse and worm, densities are comparable between the different PINs. In general, the lower number of proteins in DIP networks is responsible for their slightly higher levels of density.

The distribution of interaction reliabilities in these PPI networks is considerably shifted compared to the networks in NetAligner work. In particular, the PPI networks contain a significant number of interactions with a reliability of 1. To avoid introducing a potential systematic disadvantage for NetAligner, we downscaled the PPI reliability scores by multiplying them for the factor $\alpha$. NetAligner performance improve on rescaled network, with best results for $0.8 \leq \alpha \leq 1.0$. In the comparison with other algorithms we used the scaled networks ($\alpha = 0.9$) for NetAligner.

#### 5.1.2 Orthologs

We downloaded a comprehensive set of putative orthology associations between proteins of different organisms from the Integrative Ortholog Prediction Tool (DIOPT) [45] (statistics available in Supplementary Table S4, available online).

Some algorithms (i.e., NetAligner) require BLAST data in addition or in substitution of the data provided by DIOPT. The complete sequence data set for the five species have been downloaded from the NCBI website [46], and a BLAST sequence alignment between the proteins of the different species has been performed. The standard parameters reported in BLAST documentation were used. The same set of orthologs was used for all the algorithms. BLAST data were used only to annotate orthologs coming from DIOPT. This process is necessary, since NetAligner works by measuring the evolutionary divergence between pairs of interacting orthologs.

#### 5.1.3 Known Protein Complexes, Gene Ontology and GO Annotations

For each species a data set of known complexes was selected as benchmark data set for the intra-species assessment based on the comparison with known modules. The 408 complexes from CYC2008 [47], a comprehensive catalogue of complexes derived from small scale experiments and literature mining, have ben used to evaluate alignments involving yeast organism. For fly we considered the 556 complexes from DPIM data set [48], and for human and mouse complexes were extracted from CORUM database [49] (1685 and 439 complexes, respectively). All the data sets are updated to 2012. For fly, out of the original 556 complexes, only the 153 complexes with a functional p-value lower than $10^{-3}$ were considered [48].

TABLE 1
Statistics of Data Sets of Known Protein Complexes Used as Gold Standard

| Species | Dataset | Raw Complexes | Merged Complexes |
|---|---|---|---|
| Human | CORUM | 1685 | 606 |
| Yeast | CYC2008 | 408 | 345 |
| Mouse | CORUM | 439 | 248 |
| Fly | DPIM (DroID) | 556 | 153 |

Within each data set there are several complexes with similar biological functions and highly overlapping with each other. This might lead to a biased evaluation, since a solution can overlap with more than a known complex, and therefore be counted more than once. Moreover, these overlapping complexes are often quite small (2-4 proteins). We decided to merge these complexes together. More in detail, FastSemsSim was used to evaluate a quantitative measure of functional similarity between overlapping complexes. Afterwards, complexes have been clustered together using ClusterMaker [50]. This process produced a smaller number of complexes, as shown in Table 1. The performance of the algorithms were evaluated on the original sets of complexes as well. Results (not shown) are less clear, but the only significant difference is that some solutions encompass several small complexes (2-4 proteins) partially overlapping.

We did not consider known dimeric complexes in the assessment, as done in [15] and [51], since our purpose is to discover conserved structures that induce complex topologies on the input PINs. Therefore, no conclusions can be drawn on which algorithm should be used when uncovering conserved dimeric compexes is the primary purpose.

The Gene Ontology and GO annotations used for semantic similarity-based analysis were downloaded from the Gene Ontology website on October 2012.

### 5.2 Parameter Tuning

The most important parameters of AlignMCL are the *pruning threshold*, used in the pruning step of the alignment graph, and the *inflation*, that regulates the MCL-based clustering algorithm. The impact of varying the pruning threshold has been widely tested, and it has been shown that it does not affect the alignment outcome when kept between 0.3 and 0.7 [15]. In this work a pruning threshold of 0.5 has been used.

The inflation parameter was tuned following the procedure described in [51]. Top performance are achieved when inflation ranges between 2.4 and 3.0, and the algorithm is quite stable within this range (Supplementary Fig. S3, available online). Inflation levels below 2.4 determine a quick

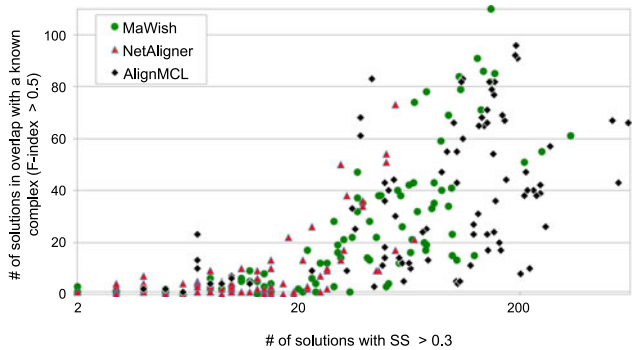



Fig. 4. Combined view of inter-species semantic similarity and complex coverage. Each point represent an alignment, its position denoting the number of solutions matching a known complex with Findex $>0.5$ (*y*-axis) and the number of solutions with semantic similarity scores $>0.3$ (*x*-axis).

degradation of performance, that instead slowly decrease with inflation levels beyond 3.0. An inflation level of 2.8 has been used in this work. It differs from the level used in [17] for the mine-and-merge algorithm (1.8).

The set of parameters used for MaWish and NetAligner in this work are reported in the supplementary material, available online.

### 5.3 Performance Assessment

All the algorithms were evaluated in terms of inter- and intra-species semantic similarity and known complexes coverage. Scatter plot in Fig. 4 combines the number of solutions matching known complexes with F-index $>0.5$ and inter-species SS $>0.3$.

Most AlignMCL's alignments concentrate in the top-right area, while MaWish's ones are more scattered; this is in line with the observation that MaWish behaves well on most networks, but has some problems in dealing with the sparsest ones. NetAligner's solutions are clearly relegated to the bottom-left part of the plot.

A finer comparison is presented in Supplementary Fig. S4, available online. In most cases AlignMCL achieves the better results. AlignMCL and MaWish have similar trends for fly-mouse, human-mouse, human-worm, human-fly, and yeast-worm alignments, consistently improving and worsening in the same cases. A weaker correlation is noticeable for fly-worm and mouse-yeast alignments.

Focusing on the overlap with known complexes, AlignMCL outperforms MaWish and NetAligner in most cases (Supplementary Fig. S5, available online). Best results are consistently achieved by all the algorithms when considering more complete PINs, such as i2d or DroID; similar results were obtained when considering merged networks (Supplementary Fig. S6, available online).

We verified that alignments provided by AlignMCL are sound in terms of semantic similarity, employing the strategy described in the previous section. FastSemSim[1] was used to evaluate the inter-species semantic similarity for all the solutions, selecting SimGIC [52] as semantic measure. According to our analysis, the null hypothesis $H_0^2$ can be rejected for all the alignments, but yeast-DIP versus

1. http://sourceforge.net/p/fastsemsim/.

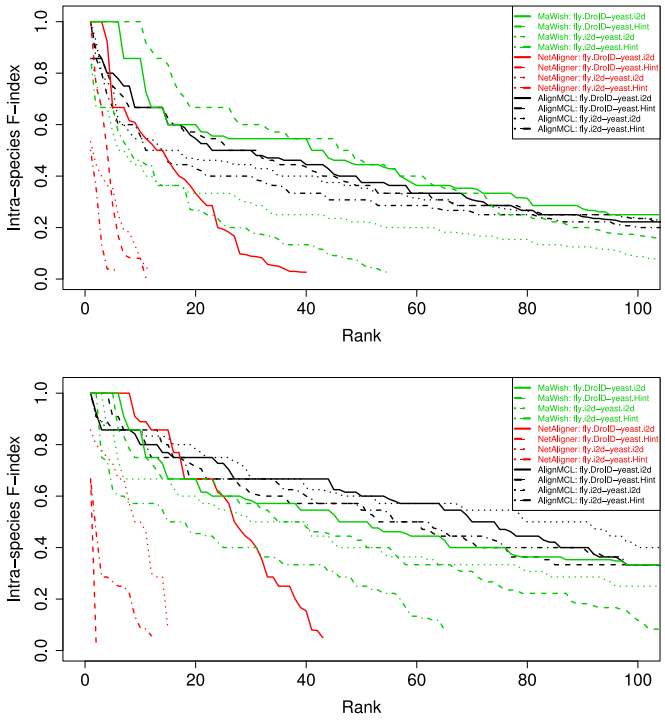



Fig. 5. Complex overlap variability between fly-yeast alignments using different PINs. Upper and lower plots focus on yeast and fly side, respectively. For each algorithm, top 100 solutions (ranked by F-index) are shown. AlignMCL's performances are similar across alignments in both species, while Mawish and NetAligner show higher variability.

mouse-DIP, and yeast-DIP versus fly-DIP. This is in line with the results of the complex overlap assessment, where DIP PINs produced the worst results. It was similarly verified that also the intra-species semantic similarity of our alignments is significant.

### 5.4 Robustness Assessment

To assess the algorithm robustness we considered the degree of variation of alignments quality between alignments involving the same pair of organisms, both in terms of complex overlap and inter-/intra-species SS. DIP networks have not been considered, since all the algorithms return low-quality solutions on them. The quality variability in fly-yeast alignments when considering different input PINs is represented in Fig. 5. Top 100 solutions from AlignMCL cover known complexes with similar quality, regardless the PINs being used. The quality of MaWish results is more variable, with few high quality solutions in the alignment of i2d fly and Hint yeast PINs. A similar trend emerges when evaluating inter-species SS, as show in Fig. 6 for human-fly alignments, and in Supplementary Fig. S7, available online, for fly-yeast alignments. Similar conclusions can be drawn from alignments involving other species (supplementary plots can be found in the supporting website). NetAligner's solutions show a higher degree of variability, suggesting a stronger sensibility to input PINs.

## 6 Conclusions

We introduced AlignMCL, a novel tool for the local alignment of protein-protein interactions networks. Its mining

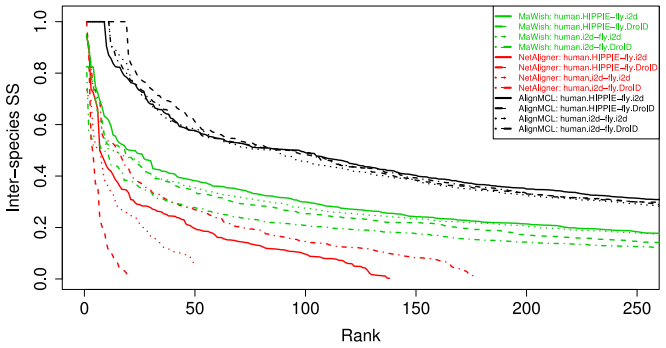



Fig. 6. Semantic similarity variability between human-fly alignments using different PINs. For each algorithm, top 250 solutions (ranked by inter-species SS) are shown. AlignMCL outperforms MaWish in all the alignments. Both algorithms are more stable than NetAligner.

strategy, based on Markov clustering, is able to identify conserved protein modules without imposing rigid constraint on their topology. Performance and stability of AlignMCL have been tested on a novel extensive assessment based on 114 alignments between five organisms. There is a positive correlation between AlignMCL and MaWish results, suggesting that both algorithms select similar conserved patterns. In general AlignMCL is more stable, while MaWish and NetAligner results show greater variance.

The current version of AlignMCL has two main limitations. First, the predicted complexes are represented as sets of aligned proteins, but no information on the putative conserved interactions is provided. Second, AlignMCL is not calibrated to identify dimeric complexes, since they do not induce unique-enough topologies in the alignment graph. These issues will be addressed in the future development of AlignMCL. Another interesting extension would be combining the Markov clustering engine with global network alignment algorithms. One of the purposes of global network alignment is to find the best superimposition of two or more PINs. It is possible to reinterpret the output of such algorithms as alignment graphs, and apply MCL to extract conserved modules. An advantage of this strategy is that many global network alignment algorithms do no require putative orthologs as input data, potentially providing more extensive results.

The MCL implementation used in this work performs a hard-clustering of the alignment graph that produces non overlapping clusters. Even though our assessment proved the quality of the solutions of MCL, a soft-clustering approach that allows solutions to overlap might perform even better. The hard-clustering constraint is not related to the flow simulation process performed by MCL: the raw solutions produced at the end of the expansion-inflation iteration can overlap, and the overlap is actually removed in a postprocessing step. In this work we purposely selected the classic hard-clustering version, mainly for its reported performance and simplicity. Indeed, our objective was to apply and assess a relatively simple and general approach to the local network alignment problem. More recently, however, soft-clustering variants of MCL have been proposed and implemented [53], paving the way to possible improvements of AlignMCL.

## References


[1] P. Bertolazzi, M. E. Bock, and C. Guerra, "On the functional and structural characterization of hubs in protein-protein interaction networks," *Biotechnol. Adv.*, vol. 31, no. 2, pp. 274–286, 2013.

[2] A.-L. Barabási, "The network takeover," *Nature Phys.*, vol. 8, no. 1, pp. 14–16, Dec. 2011.

[3] J. Ji, A. Zhang, C. Liu, X. Quan, and Z. Liu, "Survey: Functional module detection from protein-protein interaction networks," *IEEE Trans. Knowledge Data Eng.*, vol. 26, no. 2, pp. 261–277, Feb. 2014. DOI: 10.1109/TKDE.2012.225

[4] R. Sharan and T. Ideker, "Modeling cellular machinery through biological network comparison," *Nature Biotechnol.*, vol. 24, no. 4, pp. 427–33, 2006.

[5] G. Bader and C. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics*, vol. 4, no. 1, p. 2, 2003.

[6] A. D. King, "Graph clustering with restricted neighbourhood search," Ph.D. dissertation, Dept. Comput. Sci., Univ. of Toronto, ON, Canada, 2004.

[7] B. Kelley, R. Sharan, R. Karp, T. Sittler, D. Root, B. Stockwell, and T. Ideker, "Conserved pathways within bacteria and yeast as revealed by global protein network alignment," in *Proc Nat. Academy Sci. USA*, vol. 100, no. 20, pp. 11394–11399, 2003.

[8] E. Hirsh and R. Sharan, "Identification of conserved protein complexes based on a model of protein network evolution," *Bioinformatics*, vol. 23, no. 2, pp. e170–176, Jan. 2007.

[9] J. Berg and M. Lässig, "Cross-species analysis of biological networks by Bayesian alignment," in *Proc Nat. Academy Sci. USA*, vol. 103, no. 29, pp. 10967–10972, 2006.

[10] A.-C. Gavin, P. Aloy, P. Grandi, R. Krause, M. Boesche, M. Marzioch, C. Rau, L. J. Jensen, S. Bastuck, B. Dümpelfeld, A. Edelmann, M.-A. Heurtier, V. Hoffman, C. Hoefert, K. Klein, M. Hudak, A.-M. Michon, M. Schelder, M. Schirle, M. Remor, T. Rudi, S. Hooper, A. Bauer, T. Bouwmeester, G. Casari, G. Drewes, G. Neubauer, J. M. Rick, B. Kuster, P. Bork, R. B. Russell, and G. Superti-Furga, "Proteome survey reveals modularity of the yeast cell machinery," *Nature*, vol. 440, no. 7084, pp. 631–636, Mar. 2006.

[11] M. Koyutürk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama, "Pairwise alignment of protein interaction networks," *J. Comput. Biol. A J. Comput. Molecular Cell Biol.*, vol. 13, no. 2, pp. 182–199, Mar. 2006.

[12] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou, "Graemlin: General and robust alignment of multiple large interaction networks," *Genome Res.*, vol. 16, no. 9, pp. 1169–1181, Sep. 2006.

[13] E. de Silva, T. Thorne, P. Ingram, I. Agrafioti, J. Swire, C. Wiuf, and M. P. H. Stumpf, "The effects of incomplete protein interaction data on structural and evolutionary inferences," *BMC Biol.*, vol. 4, p. 39, Jan. 2006.

[14] L. Hakes, J. W. Pinney, D. L. Robertson, and S. C. Lovell, "Protein-protein interaction networks and biology–what's the connection?" *Nature Biotechnol.*, vol. 26, no. 1, pp. 69–72, Jan. 2008.

[15] 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, p. e38107, Jan. 2012.

[16] S. Erten, X. Li, G. Bebek, J. Li, and M. Koyutürk, "Phylogenetic analysis of modularity in protein interaction networks," *BMC Bioinformatics*, vol. 10, p. 333, Jan. 2009.

[17] P. Jancura, E. Mavridou, E. Carrillo-de Santa Pau, and E. Marchiori, "A methodology for detecting the orthology signal in a PPI network at a functional complex level," *BMC Bioinformatics*, vol. 13, no. Suppl 10, article S18, Jan. 2012.

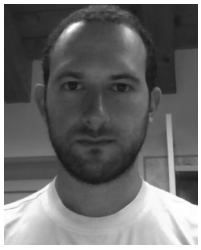
[18] R. Sharan, T. Ideker, B. Kelley, R. Shamir, and R. Karp, "Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data," *J. Comput. Biol.*, vol. 12, no. 6, pp. 835–846, 2005.

[19] R. A. Pache and P. Aloy, "A novel framework for the comparative analysis of biological networks," *PLoS ONE*, vol. 7, no. 2, p. e31220, Feb. 2012.

[20] 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.

[21] S. F. Altschul, G. Warren, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *J. Molecular Biol.*, vol. 215, no. 3, pp. 403–410, 1990.

[22] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. L. Madden, "BLAST+: Architecture and applications," *BMC Bioinformatics*, vol. 10, article 421, pp. 1–12, Jan. 2009.
[23] P. Guzzi, M. Mina, C. Guerra, and M. Cannataro, "Semantic similarity analysis of protein data: Assessment with biological features and issues," *Briefings Bioinformatics*, vol. 13, no. 5, pp. 569–585, 2012.
[24] W. Ali and C. M. Deane, "Functionally guided alignment of protein interaction networks for module detection," *Bioinformatics*, vol. 25, no. 23, pp. 3166–3173, Dec. 2009.
[25] R. Singh, J. Xu, and B. Berger, "Global alignment of multiple protein interaction networks with application to functional orthology detection," in *Proc Nat. Academy Sci. USA*, vol. 105, no. 35, pp. 12 763–12 768, 2008.
[26] O. Kuchaiev and N. Pržulj, "Integrative network alignment reveals large regions of global network similarity in yeast and human," *Bioinformatics*, vol. 27, no. 10, pp. 1390–1396, May 2011.
[27] T. Milenković, W. Leong, and N. Pržulj, "Optimal network alignment with Graphlet degree vectors," *Cancer Informatics*, vol. 9, pp. 121–137, 2010.
[28] S. van Dongen, "Graph clustering by flow simulation," Ph.D. dissertation, Dept. of Informatics, Univ. of Utrecht, Utrecht, The Netherlands, 2000.
[29] A. J. Enright, S. Van Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic Acids Res.*, vol. 30, no. 7, pp. 1575–1584, Apr. 2002.
[30] S. Brohée and J. van Helden, "Evaluation of clustering algorithms for protein-protein interaction networks," *BMC Bioinformatics*, vol. 7, article 488, Jan. 2006.
[31] A. D. King, N. Przulj, and I. Jurisica, "Protein complex prediction via cost-based clustering," *Bioinformatics*, vol. 20, no. 17, pp. 3013–3020, Nov. 2004.
[32] M. Blatt, S. Wiseman, and E. Domary, "Superparamagnetic clustering of data," *Phys. Rev. Lett.*, vol. 76, no. 18, pp. 3251–3254, 1996.
[33] R. Patro and C. Kingsford, "Global network alignment using multiscale spectral signatures," *Bioinformatics*, vol. 28, no. 23, pp. 3105–3114, Dec. 2012.
[34] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Bioinformatics enrichment tools: Paths toward the comprehensive functional analysis of large gene lists," *Nucleic Acids Res.*, vol. 37, no. 1, pp. 1–13, Jan. 2009.
[35] X. Guo, R. Liu, C.D. Shriver, H. Hu, and M.N. Liebman, "Assessing semantic similarity measures for the characterization of human regulatory pathways," *Bioinformatics*, vol. 22, no. 8, pp. 967–973, Apr. 2006.
[36] A. M. Edwards, B. Kus, R. Jansen, D. Greenbaum, J. Greenblatt, and M. Gerstein, "Bridging structural biology and genomics: Assessing protein interaction data with known complexes," *Trends in Genetics*, vol. 18, no. 10, pp. 529–536, Oct. 2002.
[37] H. Yu and P. E. A. Braun, "High-quality binary protein interaction map of the yeast interactome network," *Science*, vol. 322, no. 5898, pp. 104–110, 2008.
[38] K. R. Brown and I. Jurisica, "Online predicted human interaction database," *Bioinformatics*, vol. 21, no. 9, pp. 461–470, May 2005.
[39] J. Yu, S. Pacifico, G. Liu, and R. Finley, "DroID: The drosophila interactions database, a comprehensive resource for annotated gene and protein interactions," *BMC Genomics*, vol. 9, no. 1, pp. 461–470, Oct. 2008.
[40] A. Patil and H. Nakamura, "Hint: A database of annotated protein-protein interactions and their homologs," *Biophysics*, vol. 1, pp. 21–24, 2005.
[41] M. H. Schaefer, J.-F. Fontaine, A. Vinayagam, P. Porras, E. E. Wanker, and M. A. Andrade-Navarro, "Hippie: Integrating protein interaction networks with experiment based quality scores," *PLoS ONE*, vol. 7, no. 2, p. e31826, 2012.
[42] N. Simonis, J.-F. F. Rual, A.-R. R. Carvunis, M. Tasan, I. Lemmens, T. Hirozane-Kishikawa, T. Hao, J. M. Sahalie, K. Venkatesan, F. Gebreab, S. Cevik, N. Klitgord, C. Fan, P. Braun, N. Li, N. Ayivi-Guedehoussou, E. Dann, N. Bertin, D. Szeto, A. Dricot, M.A. Yildirim, C. Lin, A.-S. S. de Smet, H.-L. L. Kao, C. Simon, A. Smolyar, J.S.S. Ahn, M. Tewari, M. Boxem, S. Milstein, H. Yu, M. Dreze, J. Vandenhaute, K. C. Gunsalus, M. E. Cusick, D. E. Hill, J. Tavernier, F. P. Roth, and M. Vidal, "Empirically controlled mapping of the caenorhabditis elegans protein-protein interactome network," *Nature Methods*, vol. 6, no. 1, pp. 47–54, Jan. 2009.
[43] I. Xenarios, D. W. Rice, L. Salwinski, M. K. Baron, E. M. Marcotte, and D. Eisenberg, "DIP: The database of interacting proteins," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 289–291, Jan. 2000.
[44] M. E. Futschik, G. Chaurasia, and H. Herzel, "Comparison of human protein-protein interaction maps," *Bioinformatics*, vol. 23, no. 5, pp. 605–611, 2007.
[45] Y. Hu, I. Flockhart, A. Vinayagam, C. Bergwitz, B. Berger, N. Perrimon, and S. E. Mohr, "An integrative approach to ortholog prediction for disease-focused and other functional studies," *BMC Bioinformatics*, vol. 12, article 357, Jan. 2011.
[46] NCBI, "National center for biotechnology information," http://www.ncbi.nlm.nih.gov/. (2014)
[47] S. Pu, J. Wong, B. Turner, E. Cho, and S. J. Wodak, "Up-to-date catalogues of yeast protein complexes," *Nucleic Acids Res.*, vol. 37, no. 3, pp. 825–831, Feb. 2009.
[48] K. Guruharsha, J.-F. Rual, B. Zhai, J. Mintseris, P. Vaidya, N. Vaidya, C. Beekman, C. Wong, D. Y. Rhee, O. Cenaj, E. McKillip, S. Shah, M. Stapleton, K. H. Wan, C. Yu, B. Parsa, J. W. Carlson, X. Chen, B. Kapadia, K. VijayRaghavan, S. P. Gygi, S. E. Celniker, R. A. Obar, and S. Artavanis-Tsakonas, "A protein complex network of drosophila melanogaster," *Cell*, vol. 147, no. 3, pp. 690–703, Oct. 2011.
[49] A. Ruepp, B. Waegele, M. Lechner, B. Brauner, I. Dunger-Kaltenbach, G. Fobo, G. Frishman, C. Montrone, and H.-Werner Mewes, "CORUM: the comprehensive resource of mammalian protein complexes—2009," *Nucl. Acids Res.*, (2010), vol. 38, no. suppl 1, pp. D497-D501, Nov. 2009. doi:10.1093/nar/gkp914
[50] J. H. Morris, L. Apeltsin, A. M. Newman, J. Baumbach, T. Wittkop, G. Su, G. D. Bader, and T. E. Ferrin, "ClusterMaker: A multi-algorithm clustering plugin for Cytoscape," *BMC Bioinformatics*, vol. 12, no. 1, article 436, Jan. 2011.
[51] M. Mina and H. P. Guzzi, "Alignmcl: Comparative analysis of protein interaction networks through Markov clustering," in *Proc. 5th Int. Workshop Biomolecular Netw. Anal.*, 2012, pp. 30–44.
[52] C. Pesquita, D. Faria, H. Bastos, A. E. N. Ferreira, A. O. Falcão, and F. M. Couto, "Metrics for GO based protein semantic similarity: A systematic evaluation," *BMC Bioinformatics*, vol. 9, no. Suppl 5, article S4, Jan. 2008.
[53] Y.-K. Shih and S. Parthasarathy, "Identifying functional modules in interaction networks through overlapping Markov clustering," *Bioinformatics*, vol. 28, no. 18, pp. i473–i479, Sep. 2012.



**Marco Mina** he received the research doctorate in information engineering from the University of Padova, Italy. He is a postdoc fellow at Fondazione Bruno Kessler (FBK), Trento, Italy. His research interests include biological networks, biomedical ontologies, and bioimaging.

**Pietro Hiram Guzzi** received the PhD degree. He is an assistant professor at the University of Catanzaro. His research interests include algorithms for analysis of microarray data, and of protein interaction networks. He is an editor of *SIGBIO Record*. He is an ACM member.