Expert Systems With Applications 230 (2023) 120671

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

# Boosting-based ensemble of global network aligners for PPI network alignment

Manuel Menor-Flores, Miguel A. Vega-Rodríguez *

*Escuela Politécnica, Universidad de Extremadura* [1]*, Campus Universitario s/n, 10003 Cáceres, Spain*

ARTICLE INFO

*Keywords:*
Boosting-based ensemble
Boosting and bagging
Machine learning
Protein–protein interaction
Network alignment

ABSTRACT

The number of investigations attempting to align protein–protein interaction (PPI) networks has increased with the growth of studies focused on collecting PPI data. These works aim to identify conserved areas between species that are difficult to differentiate due to speciation. However, there is no standard approach to align PPI networks, and global aligners encounter difficulties in constructing alignments with high biological and structural quality. To address this issue, we propose an innovative ensemble technique that combines the strengths of aligners in the PPI network alignment field while avoiding their weaknesses. This approach reduces the spread of dispersion in so different individual global aligners and contributes to achieving a global standard that produces alignments of higher quality. This is possible thanks to the two branches composing our ensemble that aim to improve alignments in terms of biological or structural quality. In addition to a new heuristic replacing the second-level aligner in the biological quality-focused branch. Our approach achieves alignments of higher quality, as demonstrated through experiments with 10 different scenarios involving real data from 5 species. Our solutions outperform other individual aligners and ensemble techniques, like bagging, in terms of biological and structural quality. Moreover, the time required to perform the ensemble is minimal compared to that of individual aligners.

## 1. Introduction

There is currently a large amount of work focused on obtaining protein–protein interaction (PPI) data of different species through methods such as co-immunoprecipitation (Lin & Lai, 2017), yeast two-hybrid (Bacon et al., 2021), and others (Sharma et al., 2018). Thanks to these studies, it has been found that PPI networks govern most of the biological processes in species. For this reason, the alignment of PPI networks plays an important role in gaining a deeper understanding of inter-species biological transfer which has many applications in areas like drug discovery (Athanasios, Charalampos, Vasileios, & Ashraf, 2017), the study of diseases (Apostolakou, Sula, Nastou, Nasi, & Iconomidou, 2021), or phylogenetic tree reconstruction (Kuchaiev & Pržulj, 2011).

The main focus of PPI network alignment is the identification of shared information between the two networks. For this purpose, PPI network alignments have to be made in such a way as to ensure that they are meaningful. In general, the alignment of PPI networks, commonly with different sizes, involves the following steps: node mapping, edge mapping, and scoring. While node mapping aims to identify corresponding proteins between the two PPI networks, edge mapping focuses on identifying corresponding links between proteins in the two PPI networks. Once that node and edge mappings are identified, a scoring system is utilized to determine the extent of information shared between the aligned networks. Specifically, there exist two different approaches when aligning PPI networks (Gong, Peng, Ma, & Huang, 2016): local and global network alignment. On the one hand, Local Network Alignment (LNA) is focused on the identification of highly conserved and, more likely, small structures between the networks (Ma & Liao, 2020). For this purpose, it can produce many-to-many protein mappings between the networks. Therefore, a protein from one network can be aligned to more than one protein from the other network. On the other hand, Global Network Alignment (GNA) is committed to find large conserved regions between the two networks aligned (Guzzi & Milenković, 2018). To do so, it constructs one-to-one mappings between the proteins of both PPI networks, seeking to align the biggest number of proteins from the smaller network to the proteins from the bigger one. In this article, we will focus on global network alignment

Over time, many GNA studies (Ibragimov, Malek, Guo, & Baumbach, 2013; Memišević & Pržulj, 2012; Patro & Kingsford, 2012) have only focused on the biological or structural quality for the construction of

* Corresponding author.
*E-mail addresses:* mmenorf@unex.es (M. Menor-Flores), mavega@unex.es (M.A. Vega-Rodríguez).

[1] https://ror.org/0174shg90

https://doi.org/10.1016/j.eswa.2023.120671
Received 1 June 2022; Received in revised form 29 March 2023; Accepted 30 May 2023
Available online 5 June 2023

the alignments. This, in turn, has led to aligners experiencing problems such as not agreeing on most of the protein–protein mappings for the same alignment scenario or producing alignments in small regions of the solution space (Clark & Kalita, 2015). To tackle this, previous work in the field recommends focusing not only on the structural or biological quality of the alignment but on both of them (Elmsallati, Msalati, & Kalita, 2018; Ma et al., 2021; Mamano & Hayes, 2017).

However, despite the expectation that proteins' biological functions are closely associated with their set of partners involved (interaction structure), the reality is that, among species, biologically similar proteins differ significantly in their interaction structures. As some works highlight (Meng et al., 2022; Wang, Atkinson, & Hayes, 2022), this could be mainly due to the insufficient, noisy, and biased PPI data gathered so far. For this purpose, performing PPI network alignment while simultaneously maximizing biological and structural qualities is a challenging task since both qualities conflict more than expected. As a result, an increase in the biological quality leads to a decrease in the structural quality, and vice versa. To solve this problem, in the last few years, it has been observed an increase in the number of aligners proposing highly differentiated techniques and, therefore, obtaining alignments of very different biological and structural quality.

In this sense, an interesting line of research would be to develop an ensemble with some of the existing aligners in the area. The main idea is to combine their strengths and avoid their weaknesses in order to produce alignments of higher biological and structural quality. In general, ensembles are capable of decreasing the spread of dispersion among individuals that vary significantly. If well calibrated, ensembles manage to combine individuals in such a way that they cooperate with each other to achieve an improved solution (Asur, Ucar, & Parthasarathy, 2007). Therefore, in the PPI network alignment field, the use of an ensemble technique of existing PPI network aligners will surely aid in attaining a global standard that generates alignments with improved quality. Although this is a good idea, to our best knowledge, it has only been explored once in the scientific literature. More specifically, Manners, Elmsallati, Guzzi, Roy, and Kalita (2017) used a bagging-based ensemble which is one of the most known ensemble techniques (Wen & Hughes, 2020). From the PPI network alignment point of view, bagging is based on the idea of combining alignments from different aligners through different systems (mostly voting systems). In addition to bagging, one of the most popular ensemble techniques is boosting (Wen & Hughes, 2020). However, it has not been used in the area of PPI network alignment. For this reason, we propose a novel boosting-based ensemble approach of global network aligners that has not been previously used in the area in order to obtain alignments of higher biological and structural quality. Conversely to bagging, boosting first obtains an alignment from an aligner and then tries to improve its quality using another aligner. The main difference between bagging and boosting is that in the bagging case all the aligners are at the same level and their alignments are combined to produce one final solution. On the other side, in our boosting proposal, an aligner at top-level produces first an alignment and a second aligner (second-level) takes the previous alignment as input and improves it.

Apart from that, attempts have been made using the unification technique (Malod-Dognin, Ban, & Pržulj, 2017; Manners et al., 2017). However, it is not formally correct since there are relationships, which are not one-to-one, between proteins involved in the global alignment, which also means that the existing standard quality metrics in the field cannot be used to determine the quality of the final alignment (Chen et al., 2021).

Specifically, the main contributions and novelties of this work can be summarized as:

- Use and detailed explanation, for the first time in the field, of a boosting-based ensemble technique for current PPI network aligners showcasing its ability to achieve higher quality solutions.
- Comprehensive study of all the proposed aligners found in the scientific literature, assessing their availability, if they can be executed, the quality of their results in comparison with other aligners, and their execution times.
- First time in the area of a comparison between a boosting-based ensemble, individual aligners and different bagging-based ensemble methods, including the only one found in the area (Manners et al., 2017), showing that their results are not sufficiently competitive in contrast to our boosting-based ensemble results.
- Validation of the results, performing experiments with 10 different alignment scenarios using real data from PPI networks of 5 species.
- Runtime evaluation verifying that the time required to perform the proposed ensemble is minimal when compared to that required by current individual aligners.

The sections into which this article is divided are: Section 2 details the comprehensive study performed about the PPI network aligners in the scientific literature, in order to select those that will compose the proposed ensemble. Later, Section 3 describes how to measure the quality of resulting alignments, different techniques of bagging-based ensemble with voting, and our boosting-based ensemble approach. In Section 4, we show the datasets of PPI networks used in the experiments, the configurations of the individual aligners composing our ensemble, and the evaluation of the results in terms of runtime, and biological and structural quality comparing them with other tools and ensemble techniques (such as bagging) in the area. Finally, Section 5 reveals the conclusions obtained from this work and some future research lines are proposed.

## 2. Related work

Although the PPI network alignment research line is considered young and far from being solved (Clark & Kalita, 2015), there is already a large number of publications in the field. Thus, it is not surprising that several surveys are summarizing the progress in this area (Elmsallati, Clark, & Kalita, 2016; Guzzi & Milenković, 2018).

Among all the existing publications we can highlight the following works proposing global network aligners: BEAMS (Alkan & Erten, 2014), a global many-to-many aligner which relies on the use of a heuristic method based on a backbone extraction and a merge strategy. The GRAAL family of aligners: GRAAL (Kuchaiev, Milenković, Memišević, Hayes, & Pržulj, 2010) and C-GRAAL (Memišević & Pržulj, 2012) that use only topological information to construct the alignments, L-GRAAL (Malod-Dognin & Pržulj, 2015), adopting a heuristic search based on integer programming and Lagrangian relaxation, and MI-GRAAL (Kuchaiev & Pržulj, 2011), the only aligner in the GRAAL family introducing biological information to obtain the alignments. The GEDEVO group of aligners: GEDEVO (Ibragimov, Malek, et al., 2013) that applies the Graph Edit Distance (GED) approach to model one PPI network into another, CytoGEDEVO (Malek, Ibragimov, Albrecht, & Baumbach, 2016), a Cytoscape app to extend the previous GEDEVO aligner with graphical and functional features, and GEDEVO-M (Ibragimov, Malek, Baumbach, & Guo, 2014) which allows the topological multiple one-to-one alignment of PPI networks through the GED method too. DualAligner (Seah, Bhowmick, & Dewey, 2014) that proposes a new PPI network alignment technique that matches proteins with high data confidence based on biological and structural data. FastAlign (Kollias, Sathe, Mohammadi, & Grama, 2013) is committed to making protein pairings of involved PPI networks by a matrix-based greedy approach and an auction-based matching algorithm. Fuse (Gligorijević, Malod-Dognin, & Pržulj, 2016), a two-step network alignment algorithm that computes similarity scores by the Non-negative Matrix Tri-Factorization method and identifies clusters of proteins through its maximum weight k-partite matching approximation algorithm. GHOST (Patro & Kingsford, 2012) which combines a seed-and-extend global alignment phase with a local-search procedure. GREAT (Crawford & Milenković, 2015) that first aligns optimally edges to improve the node cost function used to then align the rest of nodes of both PPI networks. HubAlign (Hashemifar & Xu, 2014) that employs a minimum-degree heuristic algorithm using the biological and structural data of proteins. IBNAL (Elmsallati et al., 2018) that makes use of a clique-based index to hasten the alignment process. IsoRankN (Liao, Lu, Baym, Singh, & Berger, 2009), a multiple PPI network aligner based on the spectral clustering of proteins.

The MAGNA family of aligners with: MAGNA (Saraph & Milenković, 2014) based on a genetic algorithm that uses a novel function for crossover, its subsequent updating MAGNA++ (Vijayan, Saraph, & Milenković, 2015) that improves MAGNA by simultaneously maximizing measures of edge conservation, speeding up the algorithm and providing a user-friendly interface, and the multiple PPI network alignment version MultiMAGNA++ (Vijayan & Milenković, 2018). ModuleAlign (Hashemifar, Ma, Naveed, Canzar, & Xu, 2016) that first computes a homology score between proteins with sequence and both local and global information to then align proteins with the highest score. MONACO (Woo & Yoon, 2021), a highly adaptable aligner that constructs alignments via the optimal matching of neighbouring proteins near to a focal one. NABEECO (Ibragimov, Martens, Guo, & Baumbach, 2013), the first bee colony optimization algorithm for PPI network alignment. Natalie 2.0 (El-Kebir, Heringa, & Klau, 2015) where its authors state that they improved a Lagrangian relaxation approach by applying an integer linear programming formulation. The first server for PPI network alignment proposed in the area, NETAL (Neyshabur, Khadem, Hashemifar, & Arab, 2013) which was, in part, because of the great results obtained by the application of its greedy algorithm. NetCoffee (Hu, Kehr, & Reinert, 2014), a multiple PPI network aligner which makes use of a triplet approach to construct a set of weighted bipartite graphs to, afterwards, maximize a target function through simulated annealing. NSD (Kollias, Mohammadi, & Grama, 2012) which creates PPI network alignments by preprocessing each input network individually and searching link patterns in pairwise similarity scores. The only multi-objective approach in the field, OptNetAlign (Clark & Kalita, 2015) where its authors improved the well-known multi-objective genetic algorithm NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002) by adding crossover and mutation operators alongside a local-search operator based on hill climbing. PINALOG (Phan & Sternberg, 2012) that identifies communities in the input PPI networks to find seed protein pairs and uses a novel neighbourhood similarity score to construct the alignments. The local optimization based tool PISwap (Chindelevitch, Ma, Liao, & Berger, 2013) where alignments are first modelled taking only biological data into account, but afterwards, it includes structural information by compensating conserved interactions for mapping proteins whose biological sequences are not similar at all. SAlign (Ayub, Haider, & Naveed, 2020) which incorporates structure and sequence information to calculate biological scores instead of only sequence information. SAlign also uses the topological information of the network. The stochastic algorithm SANA (Mamano & Hayes, 2017) that implements a metaheuristic search based on simulated annealing. SMAL (Dohrmann & Singh, 2016), an innovative web server for the exploratory analysis of multiple alignments relative to a specific one. SMETANA (Sahraeian & Yoon, 2013) that first uses a semi-Markov random walk model to compute the probabilistic similarity between nodes in the PPI network to be aligned, and then, it enhances the previous probabilities by adding local and cross-species information through two types of probabilistic consistency transformations. Finally, it aligns both PPI networks by using a greedy approach based on the precomputed probabilities. SPINAL (Aladağ & Erten, 2013), a heuristic algorithm divided into two phases: it first creates similarity scores based on local neighbourhood matchings and then it iteratively grows a locally improved solution subset. TAME (Mohammadi, Gleich, Kolda, & Grama, 2017) that proposes a triangular alignment method to maximize the node correctness of the alignments. And, finally, WAVE (Sun, Crawford, Tang, & Milenković, 2015) where its authors introduce a novel measure for edge conservation that, at the same time, favours node correctness.

Table 1 shows a summary of all the aligners considered to form part of the ensemble. A total of 36 aligners were studied. In this table, the aligners are listed in alphabetical order, including their corresponding reference in the scientific literature. Furthermore, the table also specifies the link from which they can be downloaded and the reason why they have been discarded to be part of the ensemble (if this is the case). Aligners with a blank discard reason (highlighted in grey shading) indicate those finally selected to be part of the ensemble. All the aligners have been executed on a PC with 8 GB of RAM. Therefore, aligners requiring more than 8 GB of RAM were discarded. It is clearly preferable that all the aligners in the ensemble can be executed in the same operating system. Hence, since most of the aligners were available for Linux-based systems we discarded aligners whose authors only provided Windows executables. Moreover, we only retained aligners producing complete pairwise global alignments, referring as complete pairwise global alignments to those pairing all the nodes of the smaller PPI network to nodes of the bigger PPI network. But, in addition to having all nodes of the smaller PPI network paired, we also need these pairings between nodes to occur on an one-to-one basis. Therefore, aligners giving one-to-many mappings between nodes were also discarded. Apart from that, aligners producing alignments limited to specific species were discarded since our ensemble is focused on the construction of global alignments of any kind of species. On the other side, a set of quality metrics were established to determine whether the alignments of certain aligners were of sufficient quality to be part of the ensemble. In this sense, aligners with runtime 30 times longer than the fastest aligner or whose alignments did not exceed 10% of the maximum (best) values of structural or biological quality were discarded. Also, aligners of the same family were reduced to the one (in that family) obtaining the best results.

To our best knowledge, there is only one approach (Manners et al., 2017) which uses a bagging-based ensemble technique of PPI network aligners. In particular, it is focused on finding majority node mappings among alignments produced from individual PPI network aligners (voting-based strategy) to later combine these most common protein alignments into one final PPI network alignment. In addition, attempts have been made using different ensemble techniques with great results in related topics to PPI network alignment. More concretely, Wang, Ma, and Wang (2022) proposed an ensemble learning framework for detecting protein complexes in PPI networks that outperformed related works in the field. It was achieved through the integration of a voting regression model and structural modularity together with the development of a novel graph heuristic search. Liu et al. (2021) presented an innovative ensemble method that combined three deep learning methods (FoldNet, SSAfold, and DeepFR) focused on the optimization of protein fold recognition. As they stated, this combination helped to improve the results since the ensemble had the ability to combine them in ways that complemented each other. And Asur et al. (2007) developed an ensemble of various clustering techniques such as Principal Component Analysis-based and soft consensus clustering. As a result, biologically significant functional clusterings and multiple functional associations for proteins were discovered.

## 3. Methodology

Since the origin of the PPI network alignment research line, multiple approaches, such as single-objective heuristic optimization (Hashemifar & Xu, 2014; Neyshabur et al., 2013), multi-objective heuristic optimization (Clark & Kalita, 2015) or numerical optimization (El-Kebir et al., 2015), have been proposed to solve this problem of conflicting objectives (biological and structural quality). However, no global consensus has yet been established on which approach (aligner) is preferred since their resulting alignments present a great diversity of trade-offs in

**Table 1**

Comprehensive study of current aligners in the PPI network alignment field. A total of 36 aligners were studied. For every aligner, the table indicates its name and reference, the link from which it can be downloaded, and the reason why it has been discarded (if it is the case). Aligners highlighted in grey shading indicate those finally selected to be part of the ensemble.

| Aligner | Link | Discard reason |
|---|---|---|
| BEAMS (Alkan & Erten, 2014) | http://webprs.khas.edu.tr/∼cesim/BEAMS.tar.gz | High runtime |
| C-GRAAL (Memišević & Pržulj, 2012) | http://www0.cs.ucl.ac.uk/staff/natasa/C-GRAAL/index.html | |
| CytoGEDEVO (Malek et al., 2016) | http://cytogedevo.compbio.sdu.dk | Executable not working |
| DualAligner (Seah et al., 2014) | https://github.com/trove2017/DualAligner/releases | Limited to specific species |
| FastAlign (Kollias et al., 2013) | https://github.com/shmohammadi86/fastAlign | High runtime |
| Fuse (Gligorijević et al., 2016) | http://www0.cs.ucl.ac.uk/staff/natasa/FUSE/ | Only Windows executable |
| GEDEVO (Ibragimov, Malek, et al., 2013) | http://gedevo.mpi-inf.mpg.de/ | High runtime |
| GEDEVO-M (Ibragimov et al., 2014) | http://gedevo.mpi-inf.mpg.de/multiple-network-alignment/ | High runtime |
| GHOST (Patro & Kingsford, 2012) | https://github.com/Kingsford-Group/ghost2 | High runtime |
| GRAAL (Kuchaiev et al., 2010) | http://www0.cs.ucl.ac.uk/staff/natasa/GRAAL/index.html | Worse quality results than C-GRAAL |
| GREAT (Crawford & Milenković, 2015) | https://www3.nd.edu/∼cone/GREAT/ | More than 8 GB of RAM needed |
| HubAlign (Hashemifar & Xu, 2014) | http://ttic.uchicago.edu/∼hashemifar/software/HubAlign.zip | |
| IBNAL (Elmsallati et al., 2018) | http://www.cs.uccs.edu/∼linclab/IBNAL/Documentation.html | No complete pairwise global alignments |
| IsoRankN (Liao et al., 2009) | http://cb.csail.mit.edu/cb/mna/ | No one-to-one mappings between nodes |
| L-GRAAL (Malod-Dognin & Pržulj, 2015) | http://www0.cs.ucl.ac.uk/staff/natasa/L-GRAAL/index.html | Only Windows executable |
| MAGNA (Saraph & Milenković, 2014) | http://www3.nd.edu/∼cone/NA/MAGNA.zip | Poor quality results |
| MAGNA++ (Vijayan et al., 2015) | http://nd.edu/∼cone/MAGNA++/ | Poor quality results |
| MI-GRAAL (Kuchaiev & Pržulj, 2011) | http://www0.cs.ucl.ac.uk/staff/natasa/MI-GRAAL/index.html | High runtime |
| ModuleAlign (Hashemifar et al., 2016) | http://ttic.uchicago.edu/∼hashemifar/ModuleAlign.html | |
| MONACO (Woo & Yoon, 2021) | https://github.com/bjyoontamu/MONACO | No complete pairwise global alignments |
| MultiMAGNA++ (Vijayan & Milenković, 2018) | http://nd.edu/∼cone/multiMAGNA++/ | High runtime |
| NABEECO (Ibragimov, Martens, et al., 2013) | http://nabeeco.mpi-inf.mpg.de/ | High runtime |
| Natalie 2.0 (El-Kebir et al., 2015) | https://github.com/ls-cwi/natalie | No complete pairwise global alignments |
| NETAL (Neyshabur et al., 2013) | http://bioinf.modares.ac.ir/software/netal/ | |
| NetCoffee (Hu et al., 2014) | http://code.google.com/p/netcoffee/ | High runtime |
| NSD (Kollias et al., 2012) | https://github.com/shmohammadi86/NSD | No complete pairwise global alignments |
| OptNetAlign (Clark & Kalita, 2015) | http://github.com/crclark/optnetaligncpp/ | |
| PINALOG (Phan & Sternberg, 2012) | http://www.sbg.bio.ic.ac.uk/∼pinalog/ | No complete pairwise global alignments |
| PISwap (Chindelevitch et al., 2013) | http://groups.csail.mit.edu/cb/piswap/webserver/ | No complete pairwise global alignments |
| SAlign (Ayub et al., 2020) | https://github.com/cbrl-nuces/SAlign | |
| SANA (Mamano & Hayes, 2017) | https://github.com/waynebhayes/SANA | Poor quality results |
| SMAL (Dohrmann & Singh, 2016) | http://haddock6.sfsu.edu/smal/ | Web server not working |
| SMETANA (Sahraeian & Yoon, 2013) | https://github.com/bjyoontamu/SMETANA | More than 8 GB of RAM needed |
| SPINAL (Aladağ & Erten, 2013) | http://code.google.com/p/spinal/ | Only Windows executable |
| TAME (Mohammadi et al., 2017) | https://github.com/shmohammadi86/TAME | Poor quality results |
| WAVE (Sun et al., 2015) | http://nd.edu/∼cone/WAVE/WAVE.zip | |

terms of biological and structural quality. To this end, an interesting proposal is the ensemble of current aligners with the aim of combining their strengths while avoiding their weaknesses. In this way, the use of the ensemble technique could improve the performance over any of the individual aligners composing the ensemble (Wu, Mallipeddi, & Suganthan, 2019). Even so, the ensemble of current aligners has very scarcely been used in the PPI network alignment field. With only one work in the scientific literature (Manners et al., 2017), implementing a bagging-based ensemble which, in fact, uses one of the most common systems to combine solutions (in bagging-based ensemble), the majority voting system (van Erp, Vuurpijl, & Schomaker, 2002).

In this article, we contrast the only ensemble proposal made in the literature (Manners et al., 2017) with our new ensemble proposal based on boosting. Unlike bagging, our boosting approach does not combine the complete pairwise global alignments of the aligners (all aligners at the same level) through any voting system, but rather obtains a complete pairwise global alignment with one aligner (top-level aligner) and improves it with a different aligner (second-level aligner).

### 3.1. *Measuring the quality of an alignment*

To contrast both ensemble techniques, we used the biological and structural quality of the resulting alignments. Structural quality refers to the number of overlapped (conserved) edges in the PPI networks involved in the alignment. On the other side, the biological quality alludes to the number of orthologous proteins matched in the alignment. The biological and structural qualities have been measured with the *GOC* (Gene Ontology Consistency) and $S^3$ (Symmetric Substructure Score) metrics, respectively.

#### 3.1.1. *Gene Ontology Consistency (GOC)*

The Gene Ontology (GO) Consortium is a non-profit organism committed to the collection of functions of genes of different species. In fact, it is the world's largest source of information about the biological functions of proteins, containing data in terms of cellular components, molecular functions, and biological processes (Peng, Lu, Xue, Wang, & Shang, 2019). More specifically, the biological functions of any protein of a PPI network are differentiated by unique identifiers known as GO terms. Thus, given an alignment $f$ of two given graphs, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ representing the two PPI networks where $V_x$ refers to the proteins (vertices or nodes) of the network $x$ and $E_x$ refers to the edges (relationships) between proteins in the network $x$: we can easily measure the biological quality of the alignment of two proteins $u \in V_1$ and $v \in V_2$ (as long as $u$ and $v$ are paired, that is, $f(u) = v$) by using these GO terms. Bearing in mind that the cardinality (number of proteins) $|V_1|$ and $|V_2|$ of two PPI networks can be different, we always refer to $G_1$ as the PPI network with fewer proteins ($|V_1| \leq |V_2|$). Moreover, this difference in the cardinality of two PPI networks can cause loss of information about those proteins of $G_2$ that could not be aligned which is crucial to further improve the quality of the alignments. Particularly in contexts such as this one in which an ensemble of individual aligners is being performed to improve the quality of their alignments. For this reason, we introduce the concept of 'dummy' nodes which consists of a set of fictitious nodes added to the smaller PPI network ($G_1$) to equalize the network sizes ($|V_1| = |V_2|$). In this way, proteins of $G_2$ not aligned to proteins of $G_1$ will be aligned to these 'dummy' nodes. Of course, these last alignments will not be considered when measuring any of the qualities of the alignment (biological or structural) since they are not real alignments. Regarding the measurement of the biological quality, we used the Jaccard index metric known as the Gene Ontology Consistency ($GOC$). It is based on the idea of measuring the set of GO terms shared between two matched proteins ($GO(u) \cap GO(v)$) with respect to the union of all their GO terms ($GO(u) \cup GO(v)$). Eq. (1) guides the computation for the $GOC$ metric, where $|\cdot|$ is the cardinality (number of elements) of the set.

$$GOC = \sum_{u \in V_1 : \exists v \in V_2 : f(u)=v} \frac{|GO(u) \cap GO(v)|}{|GO(u) \cup GO(v)|}. \tag{1}$$

Hence, higher $GOC$ values indicate that a greater number of biological functions are conserved among the proteins matched, and therefore, a higher biological quality of alignment.

In addition to the $GOC$ metric, there are other methods for calculating the biological quality of PPI network alignments. In particular, we can highlight GOGO (Zhao & Wang, 2018) and G-SESAME (Wang, Du, Payattakool, Yu, & Chen, 2007), two graph-based evaluation methods using the Gene Ontology graph to determine the biological quality of the alignments.

### 3.1.2. *Symmetric Substructure Score ($S^3$)*

The Symmetric Substructure Score ($S^3$) is a commonly used metric to measure the structural quality of PPI network alignments (Clark & Kalita, 2015; Hashemifar et al., 2016; Hashemifar & Xu, 2014; Sun et al., 2015). Part of its recognition is due to the improvement process effectuated to obtain the $S^3$ metric. Starting from the Edge Correctness ($EC$) metric, this improvement process involved its refinement into a better metric, the Induced Conserved Structure ($ICS$). And again, a posterior improvement of the $ICS$ metric gave rise to the $S^3$ metric. An important aspect to highlight is that the formulas for the above 3 metrics ($EC$, $ICS$, and $S^3$) share the same numerator, which represents the number of overlapped (conserved) edges between the alignment $f(E_1)$ and the second network ($E_2$), as indicated in Eq. (2).

$$conserved(f) = |f(E_1) \cap E_2|. \tag{2}$$

In this sense, the improvement of these metrics was made by introducing changes in their respective denominators. Starting from $EC$, it simply divides the numerator by the number of edges of the first network, that is, by $|E_1|$. However, Patro and Kingsford (2012) found that the previous $EC$ metric was not able to penalize alignments from sparse regions of the first network to dense ones of the second network and created the $ICS$ metric to solve this problem. In this case, $ICS$ divides the numerator (Eq. (2)) by $|E_{G_2[f(V_1)]}|$, where $|E_{G_2[f(V_1)]}|$ refers to those edges in the $G_2$ network whose nodes are aligned to $G_1$ nodes. But, although $ICS$ penalizes sparse-to-dense alignments it was not able to identify dense-to-sparse alignments to penalize them too. To this end, a stronger metric is $S^3$, which was introduced to find an appropriate solution to the dense-to-sparse and sparse-to-dense alignment problem. With this purpose, $S^3$ divides the numerator (Eq. (2)) by $|E_1|+|E_{G_2[f(V_1)]}|-|f(E_1) \cap E_2|$ which simply implicates subtracting the number of conserved edges in the alignment ($|f(E_1) \cap E_2|$) to the sum of the previous $EC$ and $ICS$ denominators. Therefore, the $S^3$ metric is calculated through Eq. (3).

$$S^3 = \frac{|f(E_1) \cap E_2|}{|E_1| + |E_{G_2[f(V_1)]}| - |f(E_1) \cap E_2|}. \tag{3}$$

As expected, values after computing the $S^3$ score of an alignment are in the [0,1] range, with a 0 value indicating that 0% of the edges are conserved and a 1 value indicating that 100% of the edges are conserved. Consequently, higher $S^3$ values denote higher structural quality of the alignments.

### 3.1.3. *Combining both metrics: confidence*

In this article, we introduce the *confidence* metric to combine the biological and structural quality of the alignments. For comparison purposes, the *confidence* metric allows us to easily determine which PPI network alignment is better than others. Since both biological and structural qualities are equally important (Clark & Kalita, 2015), we combined both by giving equal weight to them (0.5). In this way, we can state that alignments with greater *confidence* values are preferred over others. As said, structural quality is measured by using $S^3$ and biological quality is assessed by using $GOC$. Also, as explained before, $S^3$ values are in the [0,1] range while $GOC$ values are not. Therefore, if we do not normalize both values of $GOC$ and $S^3$, the biological quality would interfere in the structural one when determining which alignment is better than the other. For this reason and prior to the combination of both quality metrics, we performed a min–max normalization approach. In this sense, for a particular alignment scenario, each of the qualities of an alignment was subtracted by the minimum value and divided by the maximum value minus the minimum. Eq. (4) shows how to compute the *confidence* metric of an alignment $f$ given its normalized structural and biological qualities ($S^3_{norm}$ and $GOC_{norm}$ respectively).

$$confidence(f) = 0.5 * S^3_{norm} + 0.5 * GOC_{norm}. \tag{4}$$

## 3.2. *Ensemble based on bagging with voting*

As explained before, the only ensemble technique proposed so far in the PPI network alignment field is a bagging-based ensemble with a majority voting system (Manners et al., 2017). However, apart from this majority voting system, other voting systems can be used in conjunction with the bagging-based ensemble technique. In particular, van Erp et al. (2002) contains a compendium of the most common voting systems. On account of this work, we have made a review of all of them using those that can be applied to the ensemble of PPI network aligners. All of these voting techniques are detailed in the following subsections. It is important to highlight that, in all these ensembles based on bagging with voting, every aligner in the ensemble generates a complete pairwise global alignment, whose confidence metric can be computed.

### 3.2.1. *Plurality*

The plurality voting system is based on the idea of giving one vote to each protein from $G_2$ (second PPI network) aligned with a specific protein from $G_1$ (first PPI network). Each vote is indicated by one of the aligners composing the ensemble. In this way, each alignment (aligner) gives its vote as to which protein from $G_2$ has to be aligned with each protein from $G_1$. The complete pairwise global alignment resulting from the ensemble is constructed by matching all the $G_1$ proteins to the $G_2$ proteins with the highest number of votes per $G_1$ protein. However, there may be situations where once the votes are counted there are two or more proteins in $G_2$ with the same maximum number of votes. And it is, in this case, when different variants of the plurality voting system arise for selecting one of the tied proteins of $G_2$. On the one hand, we explored the random plurality voting system (Plurality-Random) which states that when there is a tie in proteins with the highest number of votes, one of these proteins of $G_2$ is chosen at random. On the other hand, the confidence plurality voting system (Plurality-Confidence) chooses the protein of $G_2$ whose sum of the confidence values of the alignments that voted for it is higher.

### 3.2.2. *Majority*

The majority voting system is the only proposal found in the scientific literature (Manners et al., 2017) and it is commonly confused with the previous plurality method. In the majority voting system, again, each aligner has one vote per $G_2$ protein and they all indicate their votes in the same way as in the plurality method. However, the decision on which proteins of $G_2$ are aligned to proteins in $G_1$ is slightly different. In this case, it is necessary, for a candidate (any $G_2$ protein), to have obtained a majority of the votes to be selected (more than half of the votes). If no candidate fulfils the previous assumption, the final selected candidate (protein) is the one that comes from the alignment whose confidence is the highest.

### 3.2.3. Sum rule

In the sum rule case, all global aligners composing the ensemble indicate which is their preferred protein in $G_2$ to be aligned to a given protein of $G_1$ using their alignments likewise. But, instead of giving one vote to each $G_2$ candidate that they consider to be the best matching per $G_1$ protein, each global aligner assigns a confidence value to its candidate. As expected, the confidence value that each global aligner provides is the one of its constructed alignment. In this way, all confidence values given to a $G_2$ protein are added up and that protein with the highest total confidence value is the one chosen. In case of a tie in the maximum total confidence value among two or more proteins, one of them is randomly selected.

### 3.2.4. Product rule

In the same sense as the sum rule, all global aligners give a confidence value to a protein in $G_2$ through their alignments. And afterwards, the candidate with the highest final confidence value is selected. However, in the product rule case, confidence values are not added up but multiplied. This can result in aligners with a very high confidence value being severely affected by aligners with small confidence values. Again, in case of a tie in the confidence values given to several candidate proteins, the final protein is randomly selected among the tied candidates.

## 3.3. Our proposal: Ensemble based on boosting

In this subsection, we detail our ensemble proposal which is based on boosting. In contrast to the previously explained bagging techniques, boosting does not combine alignments making use of voting or other types of systems considering all the alignments at the same level. Instead, the boosting technique uses the output (alignment) of an aligner at top-level as input to another aligner at second-level to improve that alignment. For our particular case, we used a two-level boosting-based ensemble approach with two different branches. These two branches arise since we measure the quality of resulting alignments through two different metrics, $GOC$ and $S^3$. Therefore, we can improve any given alignment in two different ways, through its biological or structural quality. For a specific alignment scenario, that is, any alignment involving two PPI networks, the starting point of our boosting-based ensemble consists of obtaining the alignments of these two PPI networks by passing them through each of the 7 individual aligners finally selected in Section 2. When these alignments have been obtained, the $GOC$ and $S^3$ values for each of them are computed and normalized through the min–max normalization approach to later calculate their corresponding *confidence*. At this point, alignments are ordered by their *confidence* and the one with the highest *confidence* is selected to be the seed of the two boosting branches. The first branch tries to improve the seed alignment by selecting the proteins whose pairings do not reach a certain $GOC$ threshold and passes them through a second aligner focused on improving $S^3$. In this way, those pairings contributing less to $GOC$ will be changed to contribute more to $S^3$. The second branch performs the inverse operation, that is, it takes proteins of the seed alignment whose pairings do not reach a certain $S^3$ threshold and passes them through a second aligner focused on improving $GOC$. Therefore, those pairings contributing less to $S^3$ will be changed to contribute more to $GOC$. More exactly, in this second branch, we did not use a second aligner but implemented our own heuristic focused on improving $GOC$. Thus, once the alignment with the highest *confidence* has been improved through these two different branches we obtain two different alignments. Between these two alignments, the one with the highest *confidence* is selected as the final result of the boosting-based ensemble. Both branches of the proposed boosting-based ensemble technique are detailed in the following subsections.

### 3.3.1. Boosting $S^3$ in alignments with high GOC

Through this branch of our boosting-based ensemble, the seed alignment is improved in terms of $S^3$ by minimizing the loss in $GOC$. In order to improve the structural quality of an alignment while retaining high $GOC$ values, proteins of both PPI networks whose alignment is not a major improvement in terms of $GOC$ need to be identified. To then pass these proteins, as an input, to an aligner focused on obtaining high structural alignments. In this way, by combining the resulting alignment of this last aligner with the seed alignment we can obtain the desired results. Besides, determining a good aligner in a specific quality metric ($S^3$ or $GOC$) is direct since we record the maximum values from both $GOC$ and $S^3$ metrics to perform the previously explained min–max normalization approach. Therefore, the aligner constructing the alignment whose value for a specified quality metric is maximum will be selected as the second-level aligner of our proposed ensemble. For the case of this branch of the ensemble, the second-level aligner selected is the one constructing the alignment with the maximum $S^3$ value for each particular alignment scenario. Algorithm 1 shows the main steps performed to apply this improvement to the seed alignment. In addition, the corresponding flowchart of the boosting_s3 method can be found in Fig. 1.

To identify proteins involved in a poor biological alignment (low $GOC$) we have to first establish a criterion. In this sense, we use, as $GOC$ threshold, the average $GOC$ value that each alignment between two proteins implies (line 2). This is calculated by dividing the whole biological quality of the alignment by the number of proteins of the smallest network. It can serve as a threshold to identify low $GOC$ alignments since alignments of proteins not reaching this value will be considered biologically poor (their biological quality is below average). Once the threshold has been established, it checks all the pairings between two proteins composing the whole alignment one by one to see if the previous threshold is not reached and stores those proteins in *prot_net1_low_goc* and *prot_net2_low_goc*, which are initialized in lines 3 and 4. For that reason, it iterates over all the proteins of the biggest network (line 5).

**Algorithm 1** Pseudo-code of boosting type 1 - boosting_s3.

**Input:** $S$: alignment with the highest confidence, $size\_net1$: number of proteins of the smallest PPI network of the alignment, $size\_net2$: number of proteins of the biggest PPI network of the alignment, $\alpha$: threshold multiplication factor, $high\_s3\_aligner$: aligner constructing high $S^3$ alignments.

**Output:** $S_e$: output solution of the branch.

```
1:  $S_e \leftarrow S$
2:  $goc\_threshold \leftarrow get\_goc(S_e)/size\_net1$
3:  $prot\_net1\_low\_goc \leftarrow \emptyset$
4:  $prot\_net2\_low\_goc \leftarrow \emptyset$
5:  for $protein\_net2 \leftarrow 0$ to $size\_net2 - 1$ do
6:      if $aligned\_to\_dummy\_node(protein\_net2)$ then
7:          $prot\_net2\_low\_goc \leftarrow prot\_net2\_low\_goc \cup \{protein\_net2\}$
8:      else
9:          $protein\_net1 \leftarrow get\_aligned\_protein\_to(protein\_net2)$
10:         $goc\_value \leftarrow get\_goc\_pair(protein\_net1, protein\_net2)$
11:         if $goc\_value < \alpha * goc\_threshold$ then
12:             $prot\_net1\_low\_goc \leftarrow prot\_net1\_low\_goc \cup \{protein\_net1\}$
13:             $prot\_net2\_low\_goc \leftarrow prot\_net2\_low\_goc \cup \{protein\_net2\}$
14:         end if
15:     end if
16: end for
17: $S_p \leftarrow execute(high\_s3\_aligner, prot\_net1\_low\_goc, prot\_net2\_low\_goc)$
18: $S_e \leftarrow combine\_alignments(S_e, S_p)$
```

As previously stated, we introduced some fictitious nodes (‘dummy’ nodes) to be aligned with proteins of the biggest network that could

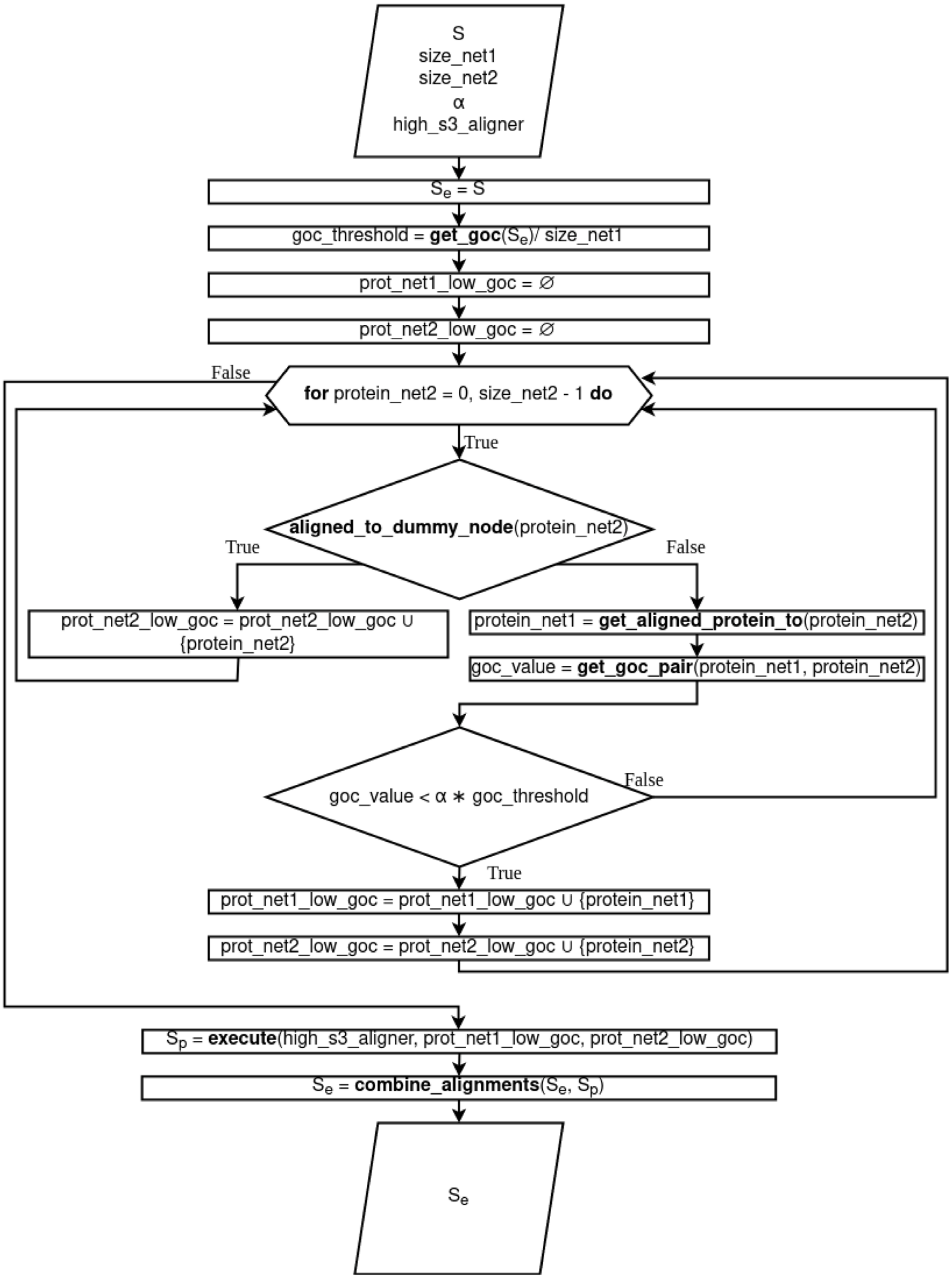
S
size_net1
size_net2
α
high_s3_aligner
$S_e$ = S
goc_threshold = **get_goc**($S_e$)/ size_net1
prot_net1_low_goc = ∅
prot_net2_low_goc = ∅
False
**for** protein_net2 = 0, size_net2 - 1 **do**
True
**aligned_to_dummy_node**(protein_net2)
True
False
prot_net2_low_goc = prot_net2_low_goc ∪ {protein_net2}
protein_net1 = **get_aligned_protein_to**(protein_net2)
goc_value = **get_goc_pair**(protein_net1, protein_net2)
goc_value < α ∗ goc_threshold
False
True
prot_net1_low_goc = prot_net1_low_goc ∪ {protein_net1}
prot_net2_low_goc = prot_net2_low_goc ∪ {protein_net2}
$S_p$ = **execute**(high_s3_aligner, prot_net1_low_goc, prot_net2_low_goc)
$S_e$ = **combine_alignments**($S_e$, $S_p$)
$S_e$

**Fig. 1.** Flowchart of boosting_s3 method.

not be matched with a protein of the smallest network. So, these false alignments, since they are fictitious, do not involve any improvement in the biological quality of the alignment. Therefore, if a given protein of $G_2$ network is aligned to a 'dummy' node (line 6) it will be stored in the *prot_net2_low_goc* set of proteins (line 7). In case this protein is not aligned to a 'dummy' node, it obtains the protein of $G_1$ network aligned with this one (line 9) to later evaluate the *GOC* value resulting from this alignment (line 10). Having this *GOC* value, by means of a simple comparison it can be determined whether the precalculated threshold is reached or not (line 11). Note that this threshold can be modified both upwards ($\alpha > 1$) and downwards ($\alpha < 1$). In essence, those proteins whose alignments do not reach the threshold will be stored at *prot_net1_low_goc* and *prot_net2_low_goc* sets of proteins (lines 12 and 13). In this way, after iterating over all the proteins, those two sets will contain all proteins involved in low biological alignments and they will be passed to the second-level aligner (line 17), that is, the one that constructs high structural alignments. After that, these high structural alignments are combined with the high biological pairings of the seed alignment to produce the final improved alignment (line 18).

### 3.3.2. Boosting GOC in alignments with high $S^3$

Conversely, the other branch of the proposed boosting-based ensemble is committed to improving the biological quality (*GOC*) of the seed alignment without destroying the conserved edges which means preserving the $S^3$ value. In this case, alignments between proteins that do not have their edges conserved will be the ones passed to the second-level aligner. However, now the second-level aligner is a heuristic developed by us. This heuristic along with the rest of steps of this other branch of the proposed boosting-based ensemble are detailed in Algorithm 2. Furthermore, Fig. 2 presents the flowchart of the boosting_goc method.

**Algorithm 2** Pseudo-code of boosting type 2 - boosting_goc.

**Input:** $S$: alignment with the highest confidence, $size\_net1$: number of proteins of the smallest PPI network of the alignment, $size\_net2$: number of proteins of the biggest PPI network of the alignment, $\alpha$: threshold multiplication factor.

**Output:** $S_e$: output solution of the branch.

```
 1: $S_e \leftarrow S$
 2: $conserved\_edges\_threshold \leftarrow get\_conserved\_edges(S_e)/size\_net1$
 3: $prot\_net1\_low\_s3 \leftarrow \emptyset$
 4: $prot\_net2\_low\_s3 \leftarrow \emptyset$
 5: for $protein\_net2 \leftarrow 0$ to $size\_net2 - 1$ do
 6:    if $aligned\_to\_dummy\_node(protein\_net2)$ then
 7:       $prot\_net2\_low\_s3 \leftarrow prot\_net2\_low\_s3 \cup \{protein\_net2\}$
 8:    else
 9:       $protein\_net1 \leftarrow get\_aligned\_protein\_to(protein\_net2)$
10:       $conserved\_edges \leftarrow get\_c\_edges\_pair(protein\_net1, protein\_net2)$
11:       if $conserved\_edges < \alpha * conserved\_edges\_threshold$ then
12:          $prot\_net1\_low\_s3 \leftarrow prot\_net1\_low\_s3 \cup \{protein\_net1\}$
13:          $prot\_net2\_low\_s3 \leftarrow prot\_net2\_low\_s3 \cup \{protein\_net2\}$
14:       end if
15:    end if
16: end for
17: for each $protein\_net1 \in prot\_net1\_low\_s3$ do
18:    $prot\_net2\_sorted \leftarrow sort\_by\_high\_goc(prot\_net2\_low\_s3, protein\_net1)$
19:    $protein\_net2 \leftarrow get\_highest\_goc\_prot(prot\_net2\_sorted)$
20:    $S_e \leftarrow align(S_e, protein\_net1, protein\_net2)$
21:    $prot\_net2\_low\_s3 \leftarrow erase\_protein(prot\_net2\_low\_s3, protein\_net2)$
22: end for
```

Again, some criterion has to be specified to select proteins of both networks involved in a low structural alignment. As explained in Section 3.1.2, in order to maximize the structural quality of a given alignment *f*, we need to increase the number of conserved edges which is, in fact, the numerator of the $S^3$ equation (Eq. (2)). Therefore, a good threshold for determining whether the alignment of two proteins is high in terms of structural quality is the average number of conserved edges of the whole alignment (line 2). For its calculation, it divides the total number of conserved edges of the seed alignment by the proteins of the smallest network. Those proteins not reaching this threshold will be considered structurally poor (their structural quality is below average), and they will be stored at *prot_net1_low_s3* and *prot_net2_low_s3* sets of proteins, which are initialized in lines 3 and 4. For that reason, it iterates over all $G_2$ proteins (line 5) checking if each protein is aligned to a 'dummy' node (line 6) in which case it will be stored at *prot_net2_low_s3* (line 7) since such an alignment with a fictitious node does not involve any conserved edge. On the contrary, if it is not aligned to a 'dummy' node, it obtains the $G_1$ protein aligned with this one (line 9) together with the number of conserved edges resulting from that alignment (line 10).

By checking if the number of conserved edges is lower than the established threshold (line 11), it can be determined whether these proteins have to be stored at *prot_net1_low_s3* and *prot_net2_low_s3* (lines 12 and 13). Here, the threshold value can be modified through the multiplication factor $\alpha$, both upwards ($\alpha > 1$) and downwards ($\alpha < 1$). In this way, after iterating over all $G_2$ proteins following these steps all proteins involved in a low structural alignment will be collected at *prot_net1_low_s3* and *prot_net2_low_s3* sets of proteins.

**Table 2**
Information of the PPI networks in the IsoBase database.

| Species | Abbreviation | Number of proteins | Number of edges |
|---|---|---|---|
| M. musculus (mouse) | mm | 623 | 679 |
| C. elegans (worm) | ce | 2995 | 6325 |
| D. melanogaster (fly) | dm | 7396 | 36017 |
| S. cerevisiae (yeast) | sc | 5524 | 100664 |
| H. sapiens (human) | hs | 10403 | 68228 |

At this point, it is where our proposed heuristic begins to leverage the previously gathered information to generate an enhanced alignment. The idea behind this is to align each $G_1$ protein stored at *prot_net1_low_s3* with the $G_2$ protein in *prot_net2_low_s3* that yields the highest possible biological quality alignment (line 20). To achieve this, the algorithm iterates over all $G_1$ proteins at *prot_net1_low_s3* (line 17) and uses each of them to sort the $G_2$ proteins from *prot_net2_low_s3*. Specifically, this sorting is based on ranking the $G_2$ proteins that generate a better alignment with the selected $G_1$ protein in terms of biological quality (line 18). Once sorted, the next step consists of selecting the $G_2$ protein producing the highest *GOC* possible, that is to say, the first $G_2$ protein in the previous rank (line 19) to finally align it with the selected $G_1$ protein (line 20). In the end (line 21), the previous $G_2$ protein is erased from *prot_net2_low_s3* to prevent its selection in the next iterations. In this way, it is ensured that future $G_2$ candidates to be aligned to other $G_1$ proteins have not been already aligned, thus avoiding duplicates in the resulting alignment.

## 4. Results

Before presenting the results obtained by our boosting-based ensemble, we detail the datasets used in the experiments, as well as the configuration of the individual aligners composing the ensemble. In order to further clarify the advantages gained by using the proposed boosting-based ensemble, the results will be shown in comparison to the results of other tools such as the individual aligners, the only bagging-based ensemble proposed in the area (Manners et al., 2017), and other bagging-based ensembles with voting implemented. In addition to this, we also show how little extra runtime it takes to add the ensemble to the individual aligners compared to the runtime that these individual aligners already require.

### 4.1. Datasets

In this subsection, we detail the datasets used in the experiments. In particular, we employed PPI networks of 5 species (Saccharomyces cerevisiae (sc), Drosophila melanogaster (dm), Caenorhabditis elegans (ce), Mus musculus (mm), and Homo sapiens (hs)), giving us a total of 10 different alignment scenarios (ce-dm, ce-hs, ce-sc, dm-hs, mm-ce, mm-dm, mm-hs, mm-sc, sc-dm, sc-hs). All this information has been extracted from the IsoBase database (Park, Singh, Baym, Liao, & Berger, 2011), which is very well-known and used in the field, containing real data (no synthetic one) of PPI networks. Such a collection of data that IsoBase contains has been obtained by the combination of information from 3 different databases: the Database of Interacting Proteins (DIP) (Salwinski et al., 2004), the Database of Protein, Genetic and Chemical Interactions (BioGRID) (Oughtred et al., 2019), and the Human Protein Reference Database (HPRD) (Keshava Prasad et al., 2009). In more detail, Table 2 shows some information of the PPI networks in the IsoBase database: the species of each PPI network, its colloquial name (in brackets), its abbreviation, the number of proteins composing that PPI network, and the number of edges in that PPI network.

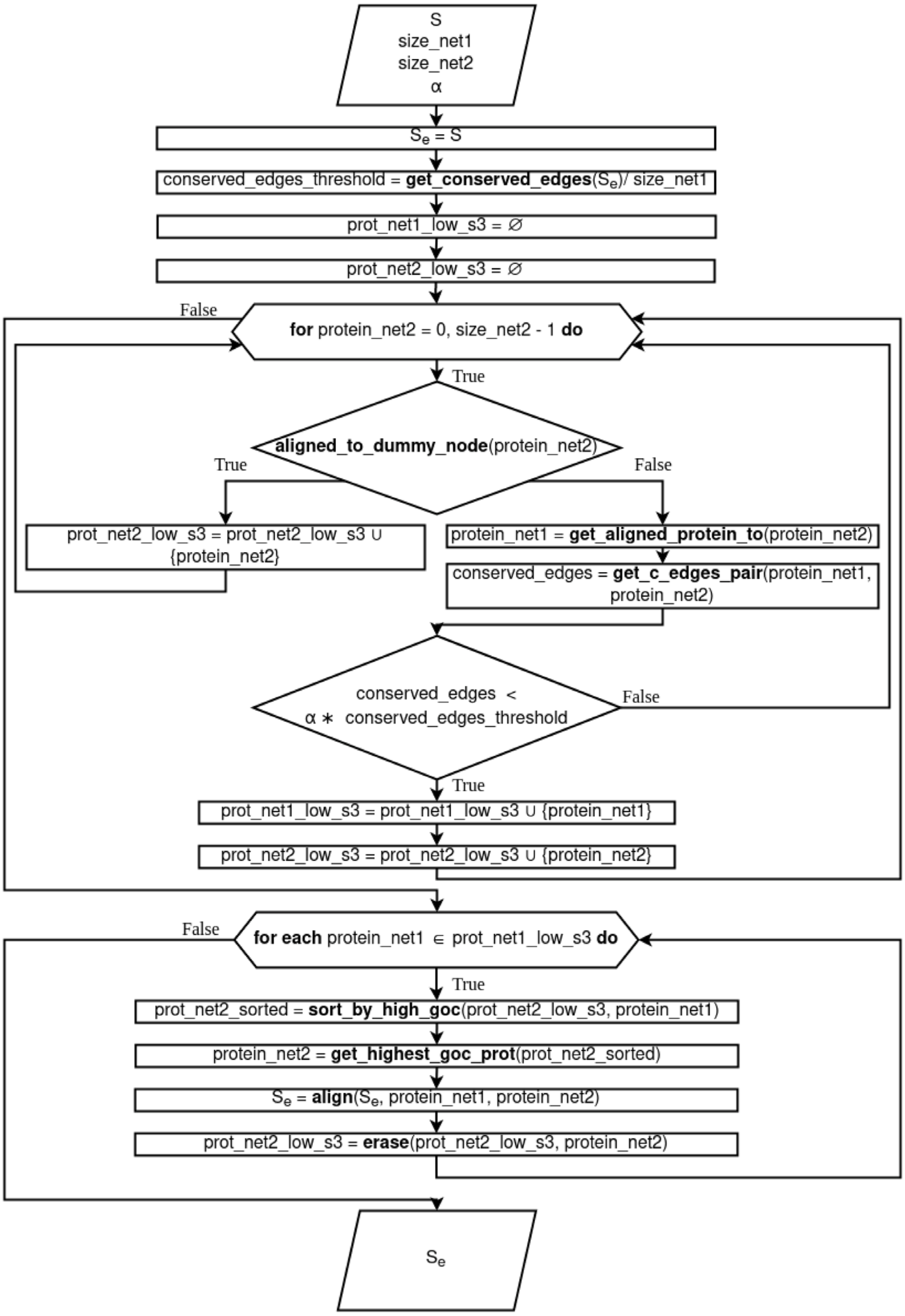
S
size_net1
size_net2
α
S_e = S
conserved_edges_threshold = get_conserved_edges(S_e)/ size_net1
prot_net1_low_s3 = ∅
prot_net2_low_s3 = ∅
False
for protein_net2 = 0, size_net2 - 1 do
True
aligned_to_dummy_node(protein_net2)
True
False
prot_net2_low_s3 = prot_net2_low_s3 ∪ {protein_net2}
protein_net1 = get_aligned_protein_to(protein_net2)
conserved_edges = get_c_edges_pair(protein_net1, protein_net2)
conserved_edges < α ∗ conserved_edges_threshold
False
True
prot_net1_low_s3 = prot_net1_low_s3 ∪ {protein_net1}
prot_net2_low_s3 = prot_net2_low_s3 ∪ {protein_net2}
False
for each protein_net1 ∈ prot_net1_low_s3 do
True
prot_net2_sorted = sort_by_high_goc(prot_net2_low_s3, protein_net1)
protein_net2 = get_highest_goc_prot(prot_net2_sorted)
S_e = align(S_e, protein_net1, protein_net2)
prot_net2_low_s3 = erase(prot_net2_low_s3, protein_net2)
S_e

**Fig. 2.** Flowchart of boosting_goc method.

### 4.2. Experimental settings

As explained in Section 2, from the wide range of studied aligners, we have selected 7 of them to form the ensemble. For this purpose, they have all been run with the default parameters proposed by their authors. In particular, Table 3 shows the user-specified execution parameters for each of them.

In the case of OptNetAlign (Clark & Kalita, 2015), the *timelimit* parameter has been set to the average runtime (in minutes) of all other 6 aligners for each one of the alignment scenarios. In this way, independently of the values of the other parameters (*popsize*, *generations*, and *hillclimbiters*), OptNetAlign will finish when the *timelimit* is reached.

Regarding the proposed ensemble, we only had to configure the $\alpha$ parameter, which is the threshold multiplication factor explained in Sections 3.3.1 and 3.3.2. In order to configure the $\alpha$ value, a parametric study was conducted in which $\alpha$ values in the range [0, 3.5] in steps of 0.25 were tested. We considered the mm-ce (low complexity), ce-sc (medium complexity), and dm-hs (high complexity) alignment scenarios for the parametric study of the $\alpha$ parameter. After this, we obtained the best results when the $\alpha$ parameter was equal to 1.25. For this reason, all the experiments were performed by setting the $\alpha$ value to 1.25.

Finally, all the experiments have been executed on a PC with an Intel Core i7-5500 CPU at 2.4 GHz and 8 GB of RAM under the Ubuntu 20.04 operating system.

**Table 3**
User-specified parameters for the execution of the 7 global aligners composing the ensemble.

| Aligner | Parameter | Value |
|---|---|---|
| C-GRAAL (Memišević & Pržulj, 2012) | No user-specified parameters | |
| HubAlign (Hashemifar & Xu, 2014) | *l* (edge and node weights) | 0.1 |
| | *a* (sequence and topological similarities) | 0.7 |
| ModuleAlign (Hashemifar et al., 2016) | *a* (topological and homological similarities) | 0.5 |
| NETAL (Neyshabur et al., 2013) | *a* (edge and node weights) | 0.0001 |
| | *b* (similarity and interaction scores) | 0 |
| | *c* (contribution of neighbours) | 1 |
| | *i* (number of iterations) | 2 |
| OptNetAlign (Clark & Kalita, 2015) | *popsize* | 200 |
| | *generations* | 1000000000 |
| | *hillclimbiters* | 10000 |
| | *timelimit* | avg_runtime |
| SAlign (Ayub et al., 2020) | *l* (node score weight) | 0.1 |
| | *a* (topological and biological scores) | 0.1 |
| | *d* (degree threshold) | 10 |
| | *t* (sequence and structure weights) | 0.7 |
| | *n* (alignment number) | 1 |
| WAVE (Sun et al., 2015) | No user-specified parameters | |

**Table 4**
Ce-dm scenario: confidence, $S^3$, and *GOC* values of our boosting-based ensemble, different bagging-based ensembles, and the individual aligners. The best aligner is highlighted in grey shading.

| Aligner | Confidence | $S^3$ | *GOC* |
|---|---|---|---|
| **Our proposal** | | | |
| Boosting-based ensemble | 0.626288 | 0.381901 | 261.768000 |
| **Bagging-based ensemble with voting** | | | |
| Plurality-Random | 0.178040 | 0.051458 | 179.577000 |
| Plurality-Confidence | 0.501188 | 0.236415 | 290.808000 |
| Majority (Manners et al., 2017) | 0.537736 | 0.274709 | 286.432000 |
| SumRule | 0.534767 | 0.263186 | 294.997000 |
| ProductRule | 0.514943 | 0.266631 | 273.739000 |
| **Individual aligners** | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 0.181790 | 0.167400 | 69.817300 |
| HubAlign (Hashemifar & Xu, 2014) | 0.263705 | 0.266996 | 46.574800 |
| ModuleAlign (Hashemifar et al., 2016) | 0.174302 | 0.184414 | 46.454300 |
| NETAL (Neyshabur et al., 2013) | 0.500000 | 0.513999 | 18.849600 |
| OptNetAlign (Clark & Kalita, 2015) | 0.537736 | 0.274709 | 286.432000 |
| SAlign (Ayub et al., 2020) | 0.384889 | 0.374370 | 51.191500 |
| WAVE (Sun et al., 2015) | 0.553191 | 0.100664 | 470.230000 |

### 4.3. Evaluation of our proposal: Ensemble based on boosting

This subsection evaluates the quality of the alignments produced by the proposed boosting-based ensemble technique. To this end, we compare these alignments with those obtained by the individual aligners composing the ensemble as well as with those generated by the only ensemble proposal in the area (Manners et al., 2017) (bagging-based ensemble with voting). Apart from that, we have implemented other well-known bagging-based ensemble with voting techniques extracted from van Erp et al. (2002) for comparison purposes. To make all these comparisons, we have used the previously explained confidence metric (Section 3.1.3) in which the structural and biological qualities of the alignments are fairly combined into one metric. In this way, alignments with the highest confidence values are preferred as they have the best structural and biological quality as a whole. In this sense, Tables 4–13 detail the confidence, $S^3$, and *GOC* values obtained by all the aligners (individual ones and based on ensemble) for each of the 10 different alignment scenarios. In every scenario (table), the best aligner is highlighted in grey shading.

After analysing these 10 tables, it can be seen how our boosting-based ensemble technique obtains the best results in most of the alignment scenarios (7 out of 10): wining in 4 alignment scenarios by a comfortable margin (ce-dm, mm-dm, sc-dm, and sc-hs) and also winning, albeit more narrowly, in other 3 more scenarios (ce-hs, dm-hs, and mm-ce). Taking into account the previous results, we can consider that the boosting-based ensemble improves the results of the individual aligners. In addition to this, one important aspect to highlight is the confidence results obtained by the bagging-based ensemble with majority voting, the only ensemble proposed in the field (Manners et al., 2017). It cannot obtain the best result in any of the alignment scenarios, only producing the second best result in the ce-hs and ce-sc scenarios. Moreover, when the results of the other bagging-based ensembles are analysed, it can be concluded that the bagging-based ensembles are not able to improve the result of the best individual aligner, showing the superiority of the boosting-based ensemble.

Apart from this, it is evident that although the ensemble treats all aligners equally, some aligners have generally been more competitive and have achieved closer results to our ensemble's best performance. In particular, the individual aligners that have performed the best, in order, are: OptNetAlign (Clark & Kalita, 2015), WAVE (Sun et al., 2015), and NETAL (Neyshabur et al., 2013). In any case, on average, incorporating a second-level aligner into our ensemble has resulted in an improvement of 8.27%, 15.30%, and 16.78% with respect to OptNetAlign, WAVE, and NETAL individual aligners, respectively.

### 4.4. Runtime evaluation

After verifying that the proposed boosting-based ensemble obtains the best results (alignments), the following question may arise: What is the extra time needed when using this ensemble approach? To this

**Table 5**
Ce-hs scenario: confidence, $S^3$, and *GOC* values of our boosting-based ensemble, different bagging-based ensembles, and the individual aligners. The best aligner is highlighted in grey shading.

| Aligner | Confidence | $S^3$ | *GOC* |
|---|---|---|---|
| **Our proposal** | | | |
| Boosting-based ensemble | 0.754861 | 0.183382 | 402.705000 |
| **Bagging-based ensemble with voting** | | | |
| Plurality-Random | 0.130733 | 0.036036 | 103.514000 |
| Plurality-Confidence | 0.704751 | 0.242339 | 303.046000 |
| Majority (Manners et al., 2017) | 0.752953 | 0.263635 | 315.557000 |
| SumRule | 0.741584 | 0.257244 | 314.068000 |
| ProductRule | 0.747455 | 0.262683 | 312.552000 |
| **Individual aligners** | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 0.216794 | 0.107965 | 89.621900 |
| HubAlign (Hashemifar & Xu, 2014) | 0.243737 | 0.159535 | 54.231000 |
| ModuleAlign (Hashemifar et al., 2016) | 0.212987 | 0.124177 | 69.513400 |
| NETAL (Neyshabur et al., 2013) | 0.500776 | 0.351233 | 37.457000 |
| OptNetAlign (Clark & Kalita, 2015) | 0.752845 | 0.263635 | 315.478000 |
| SAlign (Ayub et al., 2020) | 0.398944 | 0.266058 | 53.964100 |
| WAVE (Sun et al., 2015) | 0.000000 | 0.008874 | 36.888900 |

**Table 6**
Ce-sc scenario: confidence, $S^3$, and *GOC* values of our boosting-based ensemble, different bagging-based ensembles, and the individual aligners. The best aligner is highlighted in grey shading.

| Aligner | Confidence | $S^3$ | *GOC* |
|---|---|---|---|
| **Our proposal** | | | |
| Boosting-based ensemble | 0.784581 | 0.200748 | 294.527000 |
| **Bagging-based ensemble with voting** | | | |
| Plurality-Random | 0.203050 | 0.018319 | 183.402000 |
| Plurality-Confidence | 0.595379 | 0.133970 | 270.408000 |
| Majority (Manners et al., 2017) | 0.793749 | 0.232959 | 248.724000 |
| SumRule | 0.787539 | 0.228859 | 251.026000 |
| ProductRule | 0.767405 | 0.224131 | 244.604000 |
| **Individual aligners** | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 0.111045 | 0.046965 | 72.594200 |
| HubAlign (Hashemifar & Xu, 2014) | 0.171963 | 0.068548 | 80.226700 |
| ModuleAlign (Hashemifar et al., 2016) | 0.193750 | 0.055712 | 116.279000 |
| NETAL (Neyshabur et al., 2013) | 0.254714 | 0.128249 | 41.341000 |
| OptNetAlign (Clark & Kalita, 2015) | 0.794788 | 0.234110 | 247.585000 |
| SAlign (Ayub et al., 2020) | 0.204125 | 0.070184 | 100.076000 |
| WAVE (Sun et al., 2015) | 0.544768 | 0.037640 | 391.159000 |

**Table 7**
Dm-hs scenario: confidence, $S^3$, and *GOC* values of our boosting-based ensemble, different bagging-based ensembles, and the individual aligners. The best aligner is highlighted in grey shading.

| Aligner | Confidence | $S^3$ | *GOC* |
|---|---|---|---|
| **Our proposal** | | | |
| Boosting-based ensemble | 0.747214 | 0.144663 | 786.224000 |
| **Bagging-based ensemble with voting** | | | |
| Plurality-Random | 0.219209 | 0.025016 | 473.865000 |
| Plurality-Confidence | 0.665840 | 0.092801 | 921.467000 |
| Majority (Manners et al., 2017) | 0.743616 | 0.114578 | 944.694000 |
| SumRule | 0.675190 | 0.094317 | 930.306000 |
| ProductRule | 0.640172 | 0.106507 | 799.167000 |
| **Individual aligners** | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 0.317450 | 0.111020 | 182.236000 |
| HubAlign (Hashemifar & Xu, 2014) | 0.446638 | 0.158082 | 161.062000 |
| ModuleAlign (Hashemifar et al., 2016) | 0.401582 | 0.143203 | 160.028000 |
| NETAL (Neyshabur et al., 2013) | 0.500000 | 0.192226 | 71.634700 |
| OptNetAlign (Clark & Kalita, 2015) | 0.743849 | 0.114787 | 943.974000 |
| SAlign (Ayub et al., 2020) | 0.405024 | 0.140739 | 179.863000 |
| WAVE (Sun et al., 2015) | 0.640619 | 0.072042 | 989.094000 |

**Table 8**
Mm-ce scenario: confidence, $S^3$, and *GOC* values of our boosting-based ensemble, different bagging-based ensembles, and the individual aligners. The best aligner is highlighted in grey shading.

| Aligner | Confidence | $S^3$ | *GOC* |
|---|---|---|---|
| **Our proposal** | | | |
| Boosting-based ensemble | 0.629158 | 0.461704 | 76.869100 |
| **Bagging-based ensemble with voting** | | | |
| Plurality-Random | 0.206467 | 0.123636 | 47.164600 |
| Plurality-Confidence | 0.560055 | 0.267915 | 90.528500 |
| Majority (Manners et al., 2017) | 0.621555 | 0.285932 | 99.016900 |
| SumRule | 0.599807 | 0.294454 | 94.024500 |
| ProductRule | 0.519998 | 0.263279 | 84.050900 |
| **Individual aligners** | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 0.206764 | 0.161447 | 42.163100 |
| HubAlign (Hashemifar & Xu, 2014) | 0.303868 | 0.408558 | 26.337600 |
| ModuleAlign (Hashemifar et al., 2016) | 0.115988 | 0.225806 | 17.476700 |
| NETAL (Neyshabur et al., 2013) | 0.500000 | 0.786408 | 10.582500 |
| OptNetAlign (Clark & Kalita, 2015) | 0.580447 | 0.578352 | 52.646500 |
| SAlign (Ayub et al., 2020) | 0.453178 | 0.676611 | 16.962800 |
| WAVE (Sun et al., 2015) | 0.623420 | 0.287234 | 99.173200 |

**Table 9**
Mm-dm scenario: confidence, $S^3$, and *GOC* values of our boosting-based ensemble, different bagging-based ensembles, and the individual aligners. The best aligner is highlighted in grey shading.

| Aligner | Confidence | $S^3$ | *GOC* |
|---|---|---|---|
| **Our proposal** | | | |
| Boosting-based ensemble | 0.552460 | 0.541943 | 61.514200 |
| **Bagging-based ensemble with voting** | | | |
| Plurality-Random | 0.161431 | 0.120051 | 34.995000 |
| Plurality-Confidence | 0.414847 | 0.222555 | 71.229800 |
| Majority (Manners et al., 2017) | 0.482881 | 0.256852 | 80.202300 |
| SumRule | 0.457023 | 0.259506 | 75.043500 |
| ProductRule | 0.431014 | 0.248941 | 71.329600 |
| **Individual aligners** | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 0.198413 | 0.189278 | 34.234700 |
| HubAlign (Hashemifar & Xu, 2014) | 0.284650 | 0.454902 | 20.848800 |
| ModuleAlign (Hashemifar et al., 2016) | 0.093393 | 0.197256 | 13.594800 |
| NETAL (Neyshabur et al., 2013) | 0.500000 | 0.963768 | 4.635280 |
| OptNetAlign (Clark & Kalita, 2015) | 0.483720 | 0.638158 | 37.863100 |
| SAlign (Ayub et al., 2020) | 0.333714 | 0.575441 | 16.642000 |
| WAVE (Sun et al., 2015) | 0.483922 | 0.257509 | 80.325000 |

**Table 10**
Mm-hs scenario: confidence, $S^3$, and *GOC* values of our boosting-based ensemble, different bagging-based ensembles, and the individual aligners. The best aligner is highlighted in grey shading.

| Aligner | Confidence | $S^3$ | *GOC* |
|---|---|---|---|
| **Our proposal** | | | |
| Boosting-based ensemble | 0.545308 | 0.520468 | 95.665500 |
| **Bagging-based ensemble with voting** | | | |
| Plurality-Random | 0.465949 | 0.131504 | 227.184000 |
| Plurality-Confidence | 0.517926 | 0.178915 | 231.339000 |
| Majority (Manners et al., 2017) | 0.544557 | 0.503193 | 102.830000 |
| SumRule | 0.526511 | 0.194579 | 228.614000 |
| ProductRule | 0.571846 | 0.632492 | 59.543900 |
| **Individual aligners** | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 0.427923 | 0.139212 | 205.683000 |
| HubAlign (Hashemifar & Xu, 2014) | 0.419788 | 0.204619 | 173.318000 |
| ModuleAlign (Hashemifar et al., 2016) | 0.091315 | 0.104878 | 60.019400 |
| NETAL (Neyshabur et al., 2013) | 0.490820 | 0.642674 | 16.447600 |
| OptNetAlign (Clark & Kalita, 2015) | 0.614231 | 0.652733 | 70.954000 |
| SAlign (Ayub et al., 2020) | 0.432162 | 0.414404 | 87.865400 |
| WAVE (Sun et al., 2015) | 0.547833 | 0.157289 | 255.027000 |

end, Tables 14 and 15 detail the runtime in seconds of each of the parts composing the ensemble: the individual aligners at top-level, the aligner or our heuristic at second-level, and the set of ensemble steps. This information is given for the 10 alignment scenarios. The

**Table 11**
Mm-sc scenario: confidence, $S^3$, and *GOC* values of our boosting-based ensemble, different bagging-based ensembles, and the individual aligners. The best aligner is highlighted in grey shading.

| Aligner | Confidence | $S^3$ | *GOC* |
|---|---|---|---|
| **Our proposal** | | | |
| Boosting-based ensemble | 0.593721 | 0.453106 | 65.775600 |
| **Bagging-based ensemble with voting** | | | |
| Plurality-Random | 0.231103 | 0.030672 | 58.231500 |
| Plurality-Confidence | 0.366860 | 0.178571 | 62.471200 |
| Majority (Manners et al., 2017) | 0.670102 | 0.647235 | 52.848300 |
| SumRule | 0.597720 | 0.541613 | 54.283100 |
| ProductRule | 0.678034 | 0.674961 | 50.461300 |
| **Individual aligners** | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 0.168703 | 0.031443 | 46.791300 |
| HubAlign (Hashemifar & Xu, 2014) | 0.205220 | 0.108832 | 42.739900 |
| ModuleAlign (Hashemifar et al., 2016) | 0.082262 | 0.064738 | 26.494200 |
| NETAL (Neyshabur et al., 2013) | 0.231950 | 0.335837 | 16.255700 |
| OptNetAlign (Clark & Kalita, 2015) | 0.691241 | 0.688498 | 50.991300 |
| SAlign (Ayub et al., 2020) | 0.188839 | 0.112722 | 39.227700 |
| WAVE (Sun et al., 2015) | 0.546582 | 0.091958 | 107.072000 |

**Table 12**
Sc-dm scenario: confidence, $S^3$, and *GOC* values of our boosting-based ensemble, different bagging-based ensembles, and the individual aligners. The best aligner is highlighted in grey shading.

| Aligner | Confidence | $S^3$ | *GOC* |
|---|---|---|---|
| **Our proposal** | | | |
| Boosting-based ensemble | 0.725832 | 0.102185 | 664.033000 |
| **Bagging-based ensemble with voting** | | | |
| Plurality-Random | 0.226223 | 0.015995 | 419.146000 |
| Plurality-Confidence | 0.575780 | 0.041243 | 804.343000 |
| Majority (Manners et al., 2017) | 0.700083 | 0.063617 | 859.603000 |
| SumRule | 0.659077 | 0.055674 | 844.804000 |
| ProductRule | 0.644420 | 0.059469 | 799.039000 |
| **Individual aligners** | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 0.282664 | 0.077820 | 129.043000 |
| HubAlign (Hashemifar & Xu, 2014) | 0.192754 | 0.010698 | 399.833000 |
| ModuleAlign (Hashemifar et al., 2016) | 0.390269 | 0.085792 | 246.402000 |
| NETAL (Neyshabur et al., 2013) | 0.500000 | 0.137128 | 102.492000 |
| OptNetAlign (Clark & Kalita, 2015) | 0.465150 | 0.074569 | 430.382000 |
| SAlign (Ayub et al., 2020) | 0.272782 | 0.057134 | 240.002000 |
| WAVE (Sun et al., 2015) | 0.700881 | 0.063756 | 859.983000 |

**Table 13**
Sc-hs scenario: confidence, $S^3$, and *GOC* values of our boosting-based ensemble, different bagging-based ensembles, and the individual aligners. The best aligner is highlighted in grey shading.

| Aligner | Confidence | $S^3$ | *GOC* |
|---|---|---|---|
| **Our proposal** | | | |
| Boosting-based ensemble | 0.639735 | 0.059522 | 1045.310000 |
| **Bagging-based ensemble with voting** | | | |
| Plurality-Random | 0.221993 | 0.029681 | 504.440000 |
| Plurality-Confidence | 0.492392 | 0.055779 | 801.756000 |
| Majority (Manners et al., 2017) | 0.598823 | 0.080559 | 805.888000 |
| SumRule | 0.562658 | 0.072241 | 803.689000 |
| ProductRule | 0.536525 | 0.072980 | 749.572000 |
| **Individual aligners** | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 0.196542 | 0.067862 | 160.205000 |
| HubAlign (Hashemifar & Xu, 2014) | 0.172410 | 0.026292 | 439.051000 |
| ModuleAlign (Hashemifar et al., 2016) | 0.338392 | 0.085353 | 286.610000 |
| NETAL (Neyshabur et al., 2013) | 0.500000 | 0.145197 | 119.977000 |
| OptNetAlign (Clark & Kalita, 2015) | 0.532594 | 0.079828 | 689.007000 |
| SAlign (Ayub et al., 2020) | 0.232250 | 0.057604 | 306.117000 |
| WAVE (Sun et al., 2015) | 0.602494 | 0.081320 | 806.759000 |

**Table 14**
Runtime evaluation (Part 1). Runtime in seconds of the ensemble components: the individual aligners at top-level, the aligner or our heuristic at second-level, and the set of ensemble steps (Ensemble).

| Component of the ensemble | ce-dm | ce-hs | ce-sc | dm-hs | mm-ce |
|---|---|---|---|---|---|
| **Top-level aligners** | | | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 674.597 | 1296.563 | 327.394 | 5534.927 | 11.483 |
| HubAlign (Hashemifar & Xu, 2014) | 73.856 | 110.160 | 95.509 | 707.249 | 2.245 |
| ModuleAlign (Hashemifar et al., 2016) | 1461.010 | 2797.830 | 3572.980 | 21779.900 | 28.726 |
| NETAL (Neyshabur et al., 2013) | 24.978 | 52.597 | 64.115 | 228.397 | 1.006 |
| OptNetAlign (Clark & Kalita, 2015) | 569.428 | 1029.805 | 1076.219 | 6445.734 | 74.515 |
| SAlign (Ayub et al., 2020) | 171.087 | 187.581 | 109.958 | 864.039 | 5.847 |
| WAVE (Sun et al., 2015) | 273.194 | 280.857 | 204.706 | 2483.094 | 6.514 |
| **Ensemble** | | | | | |
| Ensemble of *boosting_s3* | 0.109 | 0.181 | 0.274 | 0.401 | 0.020 |
| Ensemble of *boosting_goc* | 0.107 | 0.180 | 0.273 | 0.400 | 0.020 |
| **Second-level aligners** | | | | | |
| 2nd-level aligner in *boosting_s3* | 20.872 | 43.083 | 942.611 | 174.389 | 1.197 |
| 2nd-level heuristic in *boosting_goc* | 4.447 | 6.503 | 2.396 | 12.527 | 0.282 |

**Table 15**
Runtime evaluation (Part 2). Runtime in seconds of the ensemble components: the individual aligners at top-level, the aligner or our heuristic at second-level, and the set of ensemble steps (Ensemble).

| Component of the ensemble | mm-dm | mm-hs | mm-sc | sc-dm | sc-hs |
|---|---|---|---|---|---|
| **Top-level aligners** | | | | | |
| C-GRAAL (Memišević & Pržulj, 2012) | 64.556 | 194.576 | 31.548 | 2353.036 | 5495.754 |
| HubAlign (Hashemifar & Xu, 2014) | 5.782 | 9.280 | 6.067 | 541.237 | 727.645 |
| ModuleAlign (Hashemifar et al., 2016) | 493.894 | 1741.920 | 2810.940 | 17093.600 | 19140.300 |
| NETAL (Neyshabur et al., 2013) | 3.398 | 6.368 | 7.133 | 370.295 | 724.089 |
| OptNetAlign (Clark & Kalita, 2015) | 136.874 | 478.990 | 691.434 | 4619.119 | 6146.218 |
| SAlign (Ayub et al., 2020) | 14.603 | 23.044 | 11.154 | 512.798 | 650.190 |
| WAVE (Sun et al., 2015) | 14.515 | 22.022 | 12.137 | 988.862 | 1272.757 |
| **Ensemble** | | | | | |
| Ensemble of *boosting_s3* | 0.055 | 0.105 | 0.137 | 0.535 | 0.686 |
| Ensemble of *boosting_goc* | 0.056 | 0.105 | 0.136 | 0.533 | 0.701 |
| **Second-level aligners** | | | | | |
| 2nd-level aligner in *boosting_s3* | 11.385 | 515.496 | 651.078 | 203.331 | 393.701 |
| 2nd-level heuristic in *boosting_goc* | 1.853 | 0.700 | 0.282 | 3.913 | 11.814 |

decomposition of the runtime in parts shows that the global runtime is actually governed by the runtime of the individual aligners and not by the ensemble itself.

More specifically, it is important to note that the ensemble runtime of the two different branches (Ensemble of *boosting_s3* and Ensemble of *boosting_goc*) is very small in comparison to the runtime of the individual aligners. In fact, the individual aligners are the part governing the runtime of all the necessary steps to execute the ensemble. Therefore, it can be concluded that the addition of the ensemble runtime to the individual aligner runtime entails a minimal increase. Also, it can be seen that the runtime values of the aligners at second-level are smaller than the ones of many top-level individual aligners. The reason behind this is that the input PPI networks of the second-level aligners are made up of a subset of proteins from the PPI networks aligned by the top-level aligners that did not reach a specified threshold. Hence, the alignment of smaller PPI networks is faster. Moreover, another significant observation is the comparison of the runtime values of our heuristic at second-level in *boosting_goc* with regard to the runtime values of the second-level aligner in *boosting_s3*, being our heuristic on average 356 times faster.

## 5. Conclusions and future work

In the last years, there has been a great deal of work trying to align PPI networks with the aim of identifying complexes evolutionarily conserved between species. From all this work, no standard has emerged to state what is the best way to align two PPI networks since their alignments result in a series of very different alignments in terms of biological and structural quality. For this reason, in this paper, we propose and detail a boosting-based ensemble of existing global aligners intending to combine their virtues while avoiding their disadvantages. Specifically, the ensemble technique has been very scarcely used in the area, with only one paper using a bagging-based ensemble of global aligners together with a majority voting system (Manners et al., 2017). Within our innovative boosting-based ensemble strategy, we implemented two different branches for the improvement of the alignments obtained by the individual aligners. Being each of them focused on either biological or structural quality improvement of the resulting alignments. For one of these improvement branches, in particular, for the one that improves the biological quality of the alignments, we have implemented and explained our own heuristic to replace the second-level aligner of the boosting-based ensemble.

Previous to the implementation of the boosting-based ensemble, the starting point was to select the current best global PPI network aligners. To this end, we conducted a comprehensive study of all the individual aligners found in the area (a total of 36 different aligners) evaluating their availability, requirements, runtime, and the quality of their obtained alignments among other aspects. In addition to this, apart from the only bagging-based ensemble proposal made in the PPI network alignment field, we performed an exhaustive study of other different bagging-based ensemble with voting techniques for comparison purposes.

The experimentation has been carried out with real data of 5 species which resulted in 10 different alignment scenarios of very varied PPI networks with very different number of proteins and edges (interactions) among them. The main conclusions of the experimentation were:

- We proved the higher quality of the alignments from the proposed boosting-based ensemble technique since it obtained better results in 7 out of 10 alignment scenarios.
- The experimental evaluation was exhaustive, because we compared the results of our boosting-based ensemble with the ones obtained by the individual aligners composing the ensemble, the only ensemble proposed in the related work, and different bagging-based ensembles with voting.
- Furthermore, we corroborated that the extra time needed when executing this new boosting-based ensemble was not influenced by the ensemble itself, but by the individual aligners composing it.

As future work, we will study the performance of the individual aligners composing the ensemble from a multi-objective viewpoint. We will evaluate how the solution space is covered by the different trade-offs between $S^3$ and $GOC$ metrics that the different aligners can provide. In this sense, the variability of the aligners' results when their configurations are changed will be also analysed. Furthermore, we will conduct a study on the noise robustness of the different aligners, and how an ensemble technique could improve this robustness regarding the ones of the individual aligners.

## CRediT authorship contribution statement

**Manuel Menor-Flores:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Visualization. **Miguel A. Vega-Rodríguez:** Conceptualization, Methodology, Formal analysis, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgements

This work was partially funded by Junta de Extremadura (Spain) and the ERDF (European Regional Development Fund, EU), grant GR18090 provided to the research group TIC015. This work was also partially funded by the MCIU (Ministry of Science, Innovation and Universities, Spain), the AEI (State Research Agency, Spain), and the ERDF (European Regional Development Fund, EU), under the contract PID2019-107299GB-I00/AEI/10.13039/501100011033 (Multi-HPC-Bio project).

## References

Aladağ, A. E., & Erten, C. (2013). SPINAL: scalable protein interaction network alignment. *Bioinformatics*, *29*(7), 917–924. http://dx.doi.org/10.1093/bioinformatics/btt071.

Alkan, F., & Erten, C. (2014). BEAMS: backbone extraction and merge strategy for the global many-to-many alignment of multiple PPI networks. *Bioinformatics*, *30*(4), 531–539. http://dx.doi.org/10.1093/bioinformatics/btt713.

Apostolakou, A. E., Sula, X. K., Nastou, K. C., Nasi, G. I., & Iconomidou, V. A. (2021). Exploring the conservation of Alzheimer-related pathways between H. sapiens and C. elegans: a network alignment approach. *Scientific Reports*, *11*, 4572. http://dx.doi.org/10.1038/s41598-021-83892-9.

Asur, S., Ucar, D., & Parthasarathy, S. (2007). An ensemble framework for clustering protein–protein interaction networks. *Bioinformatics*, *23*(13), i29–i40. http://dx.doi.org/10.1093/bioinformatics/btm212.

Athanasios, A., Charalampos, V., Vasileios, T., & Ashraf, G. M. (2017). Protein-protein interaction (PPI) network: recent advances in drug discovery. *Current Drug Metabolism*, *18*(1), 5–10. http://dx.doi.org/10.2174/138920021801170119204832.

Ayub, U., Haider, I., & Naveed, H. (2020). SAlign–a structure aware method for global PPI network alignment. *BMC Bioinformatics*, *21*, 500. http://dx.doi.org/10.1186/s12859-020-03827-5.

Bacon, K., Blain, A., Bowen, J., Burroughs, M., McArthur, N., Menegatti, S., et al. (2021). Quantitative yeast–yeast two hybrid for the discovery and binding affinity estimation of protein–protein interactions. *ACS Synthetic Biology*, *10*(3), 505–514. http://dx.doi.org/10.1021/acssynbio.0c00472.

Chen, H., Li, F., Wang, L., Jin, Y., Chi, C.-H., Kurgan, L., et al. (2021). Systematic evaluation of machine learning methods for identifying human–pathogen protein–protein interactions. *Briefings in Bioinformatics*, *22*(3), bbaa068. http://dx.doi.org/10.1093/bib/bbaa068.

Chindelevitch, L., Ma, C.-Y., Liao, C.-S., & Berger, B. (2013). Optimizing a global alignment of protein interaction networks. *Bioinformatics*, *29*(21), 2765–2773. http://dx.doi.org/10.1093/bioinformatics/btt486.

Clark, C., & Kalita, J. (2015). A multiobjective memetic algorithm for PPI network alignment. *Bioinformatics*, *31*(12), 1988–1998. http://dx.doi.org/10.1093/bioinformatics/btv063.

Crawford, J., & Milenković, T. (2015). GREAT: GRaphlet Edge-based network alignmenT. In *2015 IEEE international conference on bioinformatics and biomedicine* (pp. 220–227). IEEE, http://dx.doi.org/10.1109/BIBM.2015.7359684.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. http://dx.doi.org/10.1109/4235.996017.

Dohrmann, J., & Singh, R. (2016). The SMAL web server: global multiple network alignment from pairwise alignments. *Bioinformatics*, *32*(21), 3330–3332. http://dx.doi.org/10.1093/bioinformatics/btw402.

El-Kebir, M., Heringa, J., & Klau, G. W. (2015). Natalie 2.0: sparse global network alignment as a special case of quadratic assignment. *Algorithms*, *8*(4), 1035–1051. http://dx.doi.org/10.3390/a8041035.

Elmsallati, A., Clark, C., & Kalita, J. (2016). Global alignment of protein-protein interaction networks: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *13*(4), 689–705. http://dx.doi.org/10.1109/TCBB.2015.2474391.

Elmsallati, A., Msalati, A., & Kalita, J. (2018). Index-based network aligner of protein-protein interaction networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *15*(1), 330–336. http://dx.doi.org/10.1109/TCBB.2016.2613098.

Gligorijević, V., Malod-Dognin, N., & Pržulj, N. (2016). Fuse: multiple network alignment via data fusion. *Bioinformatics*, *32*(8), 1195–1203. http://dx.doi.org/10.1093/bioinformatics/btv731.

Gong, M., Peng, Z., Ma, L., & Huang, J. (2016). Global biological network alignment by using efficient memetic algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *13*(6), 1117–1129. http://dx.doi.org/10.1109/TCBB.2015.2511741.

Guzzi, P. H., & Milenković, T. (2018). Survey of local and global biological network alignment: the need to reconcile the two sides of the same coin. *Briefings in Bioinformatics*, *19*(3), 472–481. http://dx.doi.org/10.1093/bib/bbw132.

Hashemifar, S., Ma, J., Naveed, H., Canzar, S., & Xu, J. (2016). ModuleAlign: module-based global alignment of protein–protein interaction networks. *Bioinformatics*, *32*(17), i658–i664. http://dx.doi.org/10.1093/bioinformatics/btw447.

Hashemifar, S., & Xu, J. (2014). HubAlign: an accurate and efficient method for global alignment of protein–protein interaction networks. *Bioinformatics*, *30*(17), i438–i444. http://dx.doi.org/10.1093/bioinformatics/btu450.

Hu, J., Kehr, B., & Reinert, K. (2014). NetCoffee: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks. *Bioinformatics*, *30*(4), 540–548. http://dx.doi.org/10.1093/bioinformatics/btt715.

Ibragimov, R., Malek, M., Baumbach, J., & Guo, J. (2014). Multiple graph edit distance: simultaneous topological alignment of multiple protein-protein interaction networks with an evolutionary algorithm. In *Proceedings of the 2014 annual conference on genetic and evolutionary computation* (pp. 277–284). http://dx.doi.org/10.1145/2576768.2598390.

Ibragimov, R., Malek, M., Guo, J., & Baumbach, J. (2013). GEDEVO: an evolutionary graph edit distance algorithm for biological network alignment. In *German conference on bioinformatics 2013, vol. 34* (pp. 68–79). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, http://dx.doi.org/10.4230/OASIcs.GCB.2013.68.

Ibragimov, R., Martens, J., Guo, J., & Baumbach, J. (2013). NABEECO: biological network alignment with bee colony optimization algorithm. In *Proceedings of the 15th annual conference companion on genetic and evolutionary computation* (pp. 43–44). http://dx.doi.org/10.1145/2464576.2464600.

Keshava Prasad, T. S., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., et al. (2009). Human protein reference database–2009 update. *Nucleic Acids Research*, *37*(suppl_1), D767–D772. http://dx.doi.org/10.1093/nar/gkn892.

Kollias, G., Mohammadi, S., & Grama, A. (2012). Network similarity decomposition (NSD): a fast and scalable approach to network alignment. *IEEE Transactions on Knowledge and Data Engineering*, *24*(12), 2232–2243. http://dx.doi.org/10.1109/TKDE.2011.174.

Kollias, G., Sathe, M., Mohammadi, S., & Grama, A. (2013). A fast approach to global alignment of protein-protein interaction networks. *BMC Research Notes*, *6*, 35. http://dx.doi.org/10.1186/1756-0500-6-35.

Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., & Pržulj, N. (2010). Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, *7*(50), 1341–1354. http://dx.doi.org/10.1098/rsif.2010.0063.

Kuchaiev, O., & Pržulj, N. (2011). Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, *27*(10), 1390–1396. http://dx.doi.org/10.1093/bioinformatics/btr127.

Liao, C.-S., Lu, K., Baym, M., Singh, R., & Berger, B. (2009). IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, *25*(12), i253–i258. http://dx.doi.org/10.1093/bioinformatics/btp203.

Lin, J.-S., & Lai, E.-M. (2017). Protein–protein interactions: co-immunoprecipitation. In *Bacterial protein secretion systems: Methods and protocols* (pp. 211–219). New York, NY: Humana Press, http://dx.doi.org/10.1007/978-1-4939-7033-9_17.

Liu, Y., Han, K., Zhu, Y.-H., Zhang, Y., Shen, L.-C., Song, J., et al. (2021). Improving protein fold recognition using triplet network and ensemble deep learning. *Briefings in Bioinformatics*, *22*(6), bbab248. http://dx.doi.org/10.1093/bib/bbab248.

Ma, C.-Y., & Liao, C.-S. (2020). A review of protein–protein interaction network alignment: From pathway comparison to global alignment. *Computational and Structural Biotechnology Journal*, *18*, 2647–2656. http://dx.doi.org/10.1016/j.csbj.2020.09.011.

Ma, L., Wang, S., Lin, Q., Li, J., You, Z., Huang, J., et al. (2021). Multi-neighborhood learning for global alignment in biological networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *18*(6), 2598–2611. http://dx.doi.org/10.1109/TCBB.2020.2985838.

Malek, M., Ibragimov, R., Albrecht, M., & Baumbach, J. (2016). CytoGEDEVO–global alignment of biological networks with cytoscape. *Bioinformatics*, *32*(8), 1259–1261. http://dx.doi.org/10.1093/bioinformatics/btv732.

Malod-Dognin, N., Ban, K., & Pržulj, N. (2017). Unified alignment of protein-protein interaction networks. *Scientific Reports*, *7*, 953. http://dx.doi.org/10.1038/s41598-017-01085-9.

Malod-Dognin, N., & Pržulj, N. (2015). L-GRAAL: Lagrangian graphlet-based network aligner. *Bioinformatics*, *31*(13), 2182–2189. http://dx.doi.org/10.1093/bioinformatics/btv130.

Mamano, N., & Hayes, W. B. (2017). SANA: simulated annealing far outperforms many other search algorithms for biological network alignment. *Bioinformatics*, *33*(14), 2156–2164. http://dx.doi.org/10.1093/bioinformatics/btx090.

Manners, H. N., Elmsallati, A., Guzzi, P. H., Roy, S., & Kalita, J. K. (2017). Performing local network alignment by ensembling global aligners. In *2017 IEEE International Conference on Bioinformatics and Biomedicine* (pp. 1316–1323). IEEE, http://dx.doi.org/10.1109/BIBM.2017.8217853.

Memišević, V., & Pržulj, N. (2012). C-GRAAL: Common-neighbors-based global GRAph ALignment of biological networks. *Integrative Biology*, *4*(7), 734–743. http://dx.doi.org/10.1039/c2ib00140c.

Meng, X., Li, W., Xiang, J., Bedru, H. D., Wang, W., Wu, F.-X., et al. (2022). Temporal-spatial analysis of the essentiality of hub proteins in protein-protein interaction networks. *IEEE Transactions on Network Science and Engineering*, *9*(5), 3504–3514. http://dx.doi.org/10.1109/TNSE.2022.3185717.

Mohammadi, S., Gleich, D. F., Kolda, T. G., & Grama, A. (2017). Triangular alignment (TAME): a tensor-based approach for higher-order network alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *14*(6), 1446–1458. http://dx.doi.org/10.1109/TCBB.2016.2595583.

Neyshabur, B., Khadem, A., Hashemifar, S., & Arab, S. S. (2013). NETAL: a new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics*, *29*(13), 1654–1662. http://dx.doi.org/10.1093/bioinformatics/btt202.

Oughtred, R., Stark, C., Breitkreutz, B.-J., Rust, J., Boucher, L., Chang, C., et al. (2019). The BioGRID interaction database: 2019 update. *Nucleic Acids Research*, *47*(D1), D529–D541. http://dx.doi.org/10.1093/nar/gky1079.

Park, D., Singh, R., Baym, M., Liao, C.-S., & Berger, B. (2011). IsoBase: a database of functionally related proteins across PPI networks. *Nucleic Acids Research*, *39*(suppl_1), D295–D300. http://dx.doi.org/10.1093/nar/gkq1234.

Patro, R., & Kingsford, C. (2012). Global network alignment using multiscale spectral signatures. *Bioinformatics*, *28*(23), 3105–3114. http://dx.doi.org/10.1093/bioinformatics/bts592.

Peng, J., Lu, G., Xue, H., Wang, T., & Shang, X. (2019). TS-GOEA: a web tool for tissue-specific gene set enrichment analysis based on gene ontology. *BMC Bioinformatics*, *20*, 572. http://dx.doi.org/10.1186/s12859-019-3125-6.

Phan, H. T. T., & Sternberg, M. J. E. (2012). PINALOG: a novel approach to align protein interaction networks–implications for complex detection and function prediction. *Bioinformatics*, *28*(9), 1239–1245. http://dx.doi.org/10.1093/bioinformatics/bts119.

Sahraeian, S. M. E., & Yoon, B.-J. (2013). SMETANA: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. *PLoS One*, *8*(7), Article e67995. http://dx.doi.org/10.1371/journal.pone.0067995.

Salwinski, L., Miller, C. S., Smith, A. J., Pettit, F. K., Bowie, J. U., & Eisenberg, D. (2004). The database of interacting proteins: 2004 update. *Nucleic Acids Research*, *32*(suppl_1), D449–D451. http://dx.doi.org/10.1093/nar/gkh086.

Saraph, V., & Milenković, T. (2014). MAGNA: maximizing accuracy in global network alignment. *Bioinformatics*, *30*(20), 2931–2940. http://dx.doi.org/10.1093/bioinformatics/btu409.

Seah, B.-S., Bhowmick, S. S., & Dewey, C. F., Jr. (2014). DualAligner: a dual alignment-based strategy to align protein interaction networks. *Bioinformatics*, *30*(18), 2619–2626. http://dx.doi.org/10.1093/bioinformatics/btu358.

Sharma, V., Ranjan, T., Kumar, P., Pal, A. K., Jha, V. K., Sahni, S., et al. (2018). Protein–protein interaction detection: methods and analysis. In *Plant Biotechnology* (pp. 391–411). Apple Academic Press, http://dx.doi.org/10.1201/9781315213743.

Sun, Y., Crawford, J., Tang, J., & Milenković, T. (2015). Simultaneous optimization of both node and edge conservation in network alignment via WAVE. In *Algorithms in bioinformatics* (pp. 16–39). Berlin, Heidelberg: Springer, http://dx.doi.org/10.1007/978-3-662-48221-6_2.

van Erp, M., Vuurpijl, L., & Schomaker, L. (2002). An overview and comparison of voting methods for pattern recognition. In *Proceedings eighth international workshop on frontiers in handwriting recognition* (pp. 195–200). IEEE, http://dx.doi.org/10.1109/IWFHR.2002.1030908.

Vijayan, V., & Milenković, T. (2018). Multiple network alignment via multi-MAGNA++. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *15*(5), 1669–1682. http://dx.doi.org/10.1109/TCBB.2017.2740381.

Vijayan, V., Saraph, V., & Milenković, T. (2015). MAGNA++: maximizing accuracy in global network alignment via both node and edge conservation. *Bioinformatics*, *31*(14), 2409–2411. http://dx.doi.org/10.1093/bioinformatics/btv161.

Wang, S., Atkinson, G. R. S., & Hayes, W. B. (2022). SANA: cross-species prediction of gene ontology GO annotations via topological network alignment. *Systems Biology and Applications*, *8*, 25. http://dx.doi.org/10.1038/s41540-022-00232-x.

Wang, J. Z., Du, Z., Payattakool, R., Yu, P. S., & Chen, C.-F. (2007). A new method to measure the semantic similarity of GO terms. *Bioinformatics*, *23*(10), 1274–1281. http://dx.doi.org/10.1093/bioinformatics/btm087.

Wang, R., Ma, H., & Wang, C. (2022). An ensemble learning framework for detecting protein complexes from PPI networks. *Frontiers in Genetics*, *13*, Article 839949. http://dx.doi.org/10.3389/fgene.2022.839949.

Wen, L., & Hughes, M. (2020). Coastal wetland mapping using ensemble learning algorithms: a comparative study of bagging, boosting and stacking techniques. *Remote Sensing*, *12*(10), 1683. http://dx.doi.org/10.3390/rs12101683.

Woo, H.-M., & Yoon, B.-J. (2021). MONACO: accurate biological network alignment through optimal neighborhood matching between focal nodes. *Bioinformatics*, *37*(10), 1401–1410. http://dx.doi.org/10.1093/bioinformatics/btaa962.

Wu, G., Mallipeddi, R., & Suganthan, P. N. (2019). Ensemble strategies for population-based optimization algorithms–A survey. *Swarm and Evolutionary Computation*, *44*, 695–711. http://dx.doi.org/10.1016/j.swevo.2018.08.015.

Zhao, C., & Wang, Z. (2018). GOGO: An improved algorithm to measure the semantic similarity between gene ontology terms. *Scientific Reports*, *8*, 15107. http://dx.doi.org/10.1038/s41598-018-33219-y.