Global alignment of protein–protein interaction networks by graph matching methods

Mikhail Zaslavskiy[1,2,3], Francis Bach[4] and Jean-Philippe Vert[1,2,3,*]

[1]Centre for Computational Biology, Mines ParisTech, 35 rue Saint-Honoré, Fontainebleau, F-77300, [2]Institut Curie, [3]INSERM, U900, Paris, F-75248 and [4]INRIA-WILLOW project, Ecole Normale Supérieure, Paris, France

ABSTRACT

Motivation: Aligning protein–protein interaction (PPI) networks of different species has drawn a considerable interest recently. This problem is important to investigate evolutionary conserved pathways or protein complexes across species, and to help in the identification of functional orthologs through the detection of conserved interactions. It is, however, a difficult combinatorial problem, for which only heuristic methods have been proposed so far.
Results: We reformulate the PPI alignment as a graph matching problem, and investigate how state-of-the-art graph matching algorithms can be used for that purpose. We differentiate between two alignment problems, depending on whether strict constraints on protein matches are given, based on sequence similarity, or whether the goal is instead to find an optimal compromise between sequence similarity and interaction conservation in the alignment. We propose new methods for both cases, and assess their performance on the alignment of the yeast and fly PPI networks. The new methods consistently outperform state-of-the-art algorithms, retrieving in particular 78% more conserved interactions than IsoRank for a given level of sequence similarity.
Availability: All data and codes are freely and publicly available upon request.
Contact: jean-philippe.vert@mines-paristech.fr

1 INTRODUCTION

Protein–protein interactions (PPIs) play a central role in most biological processes. Recent years have witnessed impressive progresses towards the elucidation of large-scale PPI networks in various organisms, thanks in particular to the development of high-throughput experimental techniques such as yeast two-hybrid (Fields and Song, 1989) or co-immunoprecipitation followed by mass spectrometry (Aebersold and Mann, 2003). As the amount of PPI network data increases, computational methods to analyze and compare them are also being developed at a fast pace. In particular, comparative PPI network analysis across species has already provided insightful views of similarities and differences between species at the systemic level (Sharan *et al.*, 2005; Suthram *et al.*, 2005) and helped in the identification of functional orthologs (Bandyopadhyay *et al.*, 2006).

Comparing PPI networks usually involves some form of *network alignment*, i.e. the identification of pairs of homologous proteins from two different organisms, such that PPIs are conserved between matched pairs. The rationale behind this notion is that a protein and its functional orthologs are likely to interact with proteins in their respective network that are themselves functional orthologs. Hence, while direct sequence homology alone is often not sufficient to identify functional orthologs within paralogous families (Sjölander, 2004), the use of PPI information can help in the disambiguation of functional orthologs within clusters of homologous sequences, such as those produced by the Inparanoid algorithm (Remm *et al.*, 2001). This approach has been investigated in particular by Bandyopadhyay *et al.* (2006). Conversely, network alignment can also be a valuable approach to validate PPI conserved across multiple species and detect evolutionary conserved pathways or protein complexes (Kelley *et al.*, 2003; Sharan *et al.*, 2005).

Several methods have been proposed to perform *local* network alignment (LNA) of PPI networks, i.e. to identify subsets of matching pairs of proteins with conserved subgraphs of interactions. These methods include PathBLAST (Kelley *et al.*, 2003, 2004) and NetworkBLAST (Sharan *et al.*, 2005), which adapt the ideas of the BLAST algorithm to the search for local alignments between graphs, the method of Koyutürk *et al.* (2006), inspired by biological models of deletion and duplication, Graemlin (Flannick *et al.*, 2006), which uses networks of modules to infer the alignment, or the Bayesian approach of Berg and Lässig (2006). Less attention has been paid to the problem of *global* network alignment (GNA), i.e. the search for a global correspondence between most or all vertices of two networks that again matches similar proteins and leads to conserved interactions. Notable exceptions include the Markov random field (MRF)-based method of Bandyopadhyay *et al.* (2006) and the IsoRank algorithm (Singh *et al.*, 2008), which formulates the problem as an eigenvalue problem.

While LNA procedures can detect multiple, unrelated matched regions between networks, and can in particular match a given protein of a network to several proteins of the other network in different local matchings, GNA seeks the best consistent matching across all nodes simultaneously. This can be a desirable property for many applications, such as functional ortholog identification. On the other hand, from a computational point of view, GNA is arguably more difficult than LNA since it must find a solution among all possible global matchings. In fact, as we explain below, it is natural to reformulate GNA as weighted graph matching problem, a problem for which no polynomial time algorithm is known. Solving the general GNA problem therefore must involve some sort of approximate or heuristic method, such as IsoRank.

Following this line of thought, we propose here to formulate explicitly GNA as a graph matching problem, and investigate the use of modern state-of-the-art exact and approximate methods to solve it. While no exact solution of the graph matching optimization problem can be found in general, we show that in certain cases, if 'enough constraints' are put on the possible protein associations, and if the

---

*To whom correspondence should be addressed.

PPI networks are 'not too dense' (these notions being rigorously defined in Section 3.2), then an exact solution can be found efficiently by a new message passing (MP) algorithm. Interestingly, this case arises in particular in the functional ortholog detection problem between yeast and fly investigated by Bandyopadhyay *et al.* (2006), where matching pairs are constrained to belong to clusters of proteins produced by the Inparanoid algorithm and the PPI networks of both species are not too dense. On these data, we are therefore able to find a matching that conserves more interactions than the solutions found by MRF (Bandyopadhyay *et al.*, 2006) as well as a version of IsoRank adapted to this situation (Singh *et al.*, 2008), and we are in fact certain that our solution is optimal in the sense that it produces the largest possible number of conserved interactions. Interestingly, the resulting alignment retrieves 13% more HomoloGene pairs than the alignments of MRF and 5% more than that of IsoRank, suggesting that maximizing the number of conserved interactions indeed improves functional orthology disambiguation. When the GNA is more complex, e.g. matched pairs are not limited to belong to the same Inparanoid clusters, or the PPI networks have more edges, then our MP algorithm cannot be used and the optimal matching cannot be found in reasonable time anymore. In that case, we propose to use a recent state-of-the-art approximate methods for graph matching (Zaslavskiy *et al.*, 2008b), which tracks a path of solutions for a family of relaxed problems, as well as a new, faster and more direct gradient-based method, which bears similarities with the IsoRank method. Like IsoRank, these methods have a free parameter to balance the trade-off between matching similar proteins, on the one hand, and producing an alignment with many conserved interactions, on the other hand. We test them on the global unconstrained alignment of the fly and yeast networks, and show that for a given level of mean sequence similarity between matched proteins, our new method retrieves 78% more conserved interactions than IsoRank.

2 CONSTRAINED AND BALANCED GNA PROBLEMS

In this section, we set the notations and formalize two variants of the GNA problems. We represent a PPI network describing the interactions among N proteins of an organism as an undirected simple graph $G = (V_G, E_G)$, where $V_G = (v_1, \ldots, v_N)$ is a finite set of N vertices representing the N proteins, and $E_G \subset V_G \times V_G$ is the set of edges representing the pairs of interacting proteins. Each such graph (or network) can equivalently be represented by a symmetric $N \times N$ adjacency matrix A_G where $[A_G]_{ij} = [A_G]_{ji} = 1$ if protein v_i interacts with protein v_j and 0 otherwise.

Given two graphs G and H representing the PPI networks of two species, the GNA problem is, roughly speaking, to find a correspondence between the vertices of G and the vertices of H that matches similar proteins and enforces as much as possible the conservation of interactions between matched pairs in the two graphs. To formalize this, let us assume that G and H have the same number N of vertices, and that we are looking for a bijection between the vertices of G and the vertices of H. Although this may sound at first sight a strong assumption, given that PPI networks usually do not have the same size, and that we may not want to match all proteins of each network, both limitations can be addressed by adding dummy nodes (with no connection) to each graph in order to ensure that they finally have the same size. In a complete matching of such graphs with dummy nodes, matching a protein to a dummy node simply means that in the GNA the protein is not matched. G and H being assumed to have the same number of vertices, a matching of their vertices is now simply a permutation π of $\{1, \ldots, N\}$, which associates the i-th vertex of H with the $\pi(i)$-th vertex of G. Equivalently, the permutation π can be represented by a $N \times N$ permutation matrix P, i.e. a binary matrix whose (i,j)-th entry is equal to 1 if and only if $\pi(i) = j$ (i.e. when the i-th vertex of H is matched to the j-th vertex of G). We denote by $\mathcal{P} = \{P \in \{0,1\}^{N \times N} : P1_N = 1_N, P^T 1_N = 1_N\}$ the set of permutation matrices, where 1_N is the N-dimensional vectors whose entries are all equal to 1.

The number of interactions conserved by a permutation π is the number of pairs (i,j) that are connected in H, and such that their corresponding vertices $\pi(i)$ and $\pi(j)$ are also connected in G. Let us denote the number of such interactions conserved by the permutation encoded in the permutation matrix P by $J(P)$. In order to express $J(P)$, we can observe that if we apply the permutation encoded by P to the vertices of H, we obtain a new graph isomorphic to H which we denote by $P(H)$. It is easy to see that the adjacency matrix of the permuted graph, $A_{P(H)}$, is simply obtained from A_H by the equality $A_{P(H)} = PA_H P^T$ (Umeyama, 1988). As a result, $J(P)$ is simply obtained as half the number of entries that are simultaneously equal to 1 in both binary matrices A_G and $PA_H P^T$ (each conserved interaction results in two identical entries, by symmetry of the adjacency matrices). Hence we obtain the following expression for $J(P)$:

$$J(P) = \frac{1}{2} \sum_{i,j=1}^{N} [A_G]_{ij} [PA_H P^T]_{ij} = \frac{1}{2} \text{tr}(A_G^T PA_H P^T). \quad (1)$$

Besides the number of conserved interactions, a good GNA should match proteins with similar sequences. We consider here two possible formulations of this objective.

- *Constrained GNA*. Here, we assume that a pre-processing of the protein sequences has produced a set of candidate matched pairs $\mathcal{A} \subset V_H \times V_G$, and we simply wish to disambiguate the matching using PPI information, if some proteins have several candidate matchings. This is, for example, the formulation proposed by Bandyopadhyay *et al.* (2006), where a first clustering of all proteins sequences is performed to define a collection of protein clusters with the Inparanoid algorithm, and the pairs matched between the yeast and fly proteome are constrained to belong to the same cluster. Such constraints can be directly encoded as constraints over the permutation matrix P, by imposing $P_{ij} = 0$ if the i-th vertex of the first graph and the j-th vertex of the second graph are not allowed to match. We are then looking for a solution in the set of matrices $\mathcal{P}_{\mathcal{A}} = \{P \in \mathcal{P} : \forall (i,j) \in 1, N^2 \setminus \mathcal{A}, P_{ij} = 0\}$, and it is then natural to look for the permutation compatible with the constraints with the largest number of conserved interactions, i.e. to solve:

$$\max_{P \in \mathcal{P}_{\mathcal{A}}} J(P). \quad (2)$$

- *Balanced GNA*. An interesting property of constrained GNA is that, by reducing the search space to $\mathcal{P}_{\mathcal{A}}$, it can result in a tractable optimization problem (as shown for example in Section 3.2). On the other hand, in some cases one may

want to accept matching between less similar vertices if it leads to an important increase in the number of conserved interactions. In other words, one would like to be able to automatically *balance* the matching of similar vertices with the conservation of interactions, as advocated by Singh *et al.* (2008) and implemented by IsoRank. This can be formalized by assuming that a $N \times N$ matrix of similarities between vertices C is given (e.g. derived from pairwise sequence similarity scores), and by trying to maximize the total similarity between matched pair. C_{ij} denoting the similarity between the i-th vertex of G and the j-th vertex of H, the total similarity between pairs matched by a permutation matrix π is simply

$$S(P) = \sum_{i=1}^{N} C_{\pi(i),i} = \mathrm{tr}\left(PC\right). \tag{3}$$

In order to find a balance between matching similar pairs [large $S(P)$] and having many conserved interactions [large $J(P)$], we propose to consider the following optimization problem:

$$\max_{P \in \mathcal{P}} \lambda J(P) + (1-\lambda)S(P), \tag{4}$$

where $\lambda \in [0,1]$ controls the trade-off between both objectives. $\lambda = 1$ corresponds to the maximization of $J(P)$ only, i.e. to find a good topological matching of the graphs independently of the similarity between matched pairs, while $\lambda = 0$ amounts to focus only on the similarity between proteins and finding a matching which maximized the mean sequence similarity, without using PPI information.

When $\lambda > 0$, the balanced GNA problem (4) is equivalent to a general graph matching problem, discussed in Section 3.1, which is known to be computationally intractable in general. The constrained GNA (2) can be seen as a particular case of the balanced GNA, by taking the similarity function equal to 0 between two vertices allowed to match and $-\infty$ for two vertices not allowed to match. Indeed, in that case (4) is equivalent to minimizing $J(P)$ over the set of matrices P for which $S(P)$ is finite, that is exactly the set $\mathcal{P}_{\mathcal{A}}$ of (2). While indeed general graph matching methods to solve (4) can be applied to solve (2), we show in the next section that in some cases there exists a simple polynomial time algorithm to solve (2) directly even for large non-sparse graphs.

3 METHODS

In this section, we present methods to solve both the constrained GNA problem (2) and the balanced GNA problem (4). Since any algorithm to solve the balanced GNA problem can also solve the constrained GNA, as explained in the previous section, we start by describing methods to solve the balanced GNA problem.

3.1 Algorithms for the balanced GNA problem

The balanced GNA problem (4) is a general graph matching problem, which is known to be a difficult combinatorial problem. While some methods based on incomplete enumeration may be applied to search for an exact optimal solution in the case of small or sparse graphs, only approximate algorithms that usually find non-optimal solutions but are more scalable can be used for large non-sparse graph matching. Many such approximate algorithms have been proposed, see e.g. the review of Conte *et al.* (2004). They include in particular spectral methods (Caelli and Kosinov, 2004; Singh *et al.*, 2008; Umeyama, 1988), or methods based on a relaxation of the optimization

problem (4) (Almohamad and Duffuaa, 1993; Gold and Rangarajan, 1996). They differ mainly on their scalability, and on the accuracy of the solution found. For example, a comparison of several such methods was carried out recently (Zaslavskiy *et al.*, 2008b, c).

Based on these observation, we propose here to use state-of-the-art graph matching methods to balanced GNA for PPI networks. In particular, we focus on the PATH algorithm (Zaslavskiy *et al.*, 2008b), which was shown to provide state-of-the-art performance in various graph matching benchmark. We also propose a new and simpler gradient ascent method, similar in spirit to the graduated assignment (GA) algorithm (Gold and Rangarajan, 1996). As a benchmark, we consider the IsoRank method, which can be thought of as a particular spectral method for graph alignment, and which is currently the method of choice for balanced GNA of PPI networks. We now briefly describe these methods.

- *PATH method.* The PATH algorithm is based on two relaxations of (4), one concave and one convex, over the set of doubly stochastic matrices (Zaslavskiy *et al.*, 2008b). The method starts by solving the convex relaxation, and then iteratively solves a linear combination of the convex and concave relaxations by gradually increasing the weight of the concave relaxation and following the path of solutions thus created. It finishes when the solution reaches a corner of the set of doubly stochastic matrices, i.e. when the solution is a permutation matrix in $\mathcal{P}$. On several benchmarks, the PATH method was shown to be state-of-the-art in accuracy, and can easily process graphs with a few thousands vertices in a few hours on a modern desktop computer.

- *GA method.* We propose a new, simple gradient method based on a relaxation of (4) over the set of doubly stochastic matrices. Although the function to be maximized is not concave [because of the term $J(P)$], we simply start from an initial solution and iteratively choose a new permutation matrix in the direction of the gradient of the objective function. This approach may be relevant if we can start from a 'good' initial solution, i.e., if we solve a constrained GNA (2) where the constraints are strong enough. The gradient of $S(P)$ in (3) is equal to S, the gradient of $J(P)$ in (1) at a matrix P_n is equal to $A_G^T P_n A_H$. Hence we propose to iteratively update the permutation matrix following the rule $P_{n+1} \leftarrow \mathrm{argmax}_{P \in \mathcal{P}} \mathrm{tr}\left([\lambda A_G^T P_n A_H + (1-\lambda)C]P\right)$, which can be found efficiently by the Hungarian algorithm (Kuhn, 1955).

- *IsoRank method.* The idea of the IsoRank algorithm is to use the following recursive formula (Singh *et al.*, 2008)

$$R(i,j) = \sum_{v \in N(i)} \sum_{u \in N(j)} \frac{1}{|N(u)||N(v)|} R(u,v), \ i \in V_G, \ j \in V_H, \tag{5}$$

where $N(i)$ denotes the set of neighbors of i, V_G denotes the set of vertices of graph G and element $R(i,j)$ represents the similarity between vertex i of graph G and vertex j of graph H. In the case of PPI networks, it represents the 'likelihood' that proteins i and j are functional orthologs. The recursive formula says that the more i and j have similar neighbors, the greater is the similarity measure between i and j. To estimate R, Singh *et al.* (2008) propose to use the power method to iteratively update R according to:

$$R \leftarrow AR/||AR||, \tag{6}$$

where A is the $N^2 \times N^2$ matrix defined as:

$$A(i,j)(u,v) = \frac{1}{|N(u)||N(v)|}.$$

To take into account the information on protein sequence similarities encoded by matrix C, the following modification of (5) is used

$$R = \lambda AR + (1-\lambda)C, \tag{7}$$

where λ has the same interpretation as in (4).

3.2 Algorithms for the constrained GNA problem

As explained in Section 2, all methods for solving the balanced GNA problem (4) can also be used to solve the constrained GNA problem (2), by using a particular similarity function to enforce the constraints. Hence a first series of methods to solve (2) are the constrained version of IsoRank, GA and PATH, described in the previous section. In addition to these three methods, we consider two additional approaches specifically dedicated to the constrained GNA problem: the MRF method of Bandyopadhyay *et al.* (2006), and a new method based on MP which we propose to find the global optimum of (2) when the graphs are not too dense.

- *MRF method.* To solve ambiguous assignments in Inparanoid clusters with more than two proteins, Bandyopadhyay *et al.* (2006) propose to use the information on protein interactions, by choosing the assignments that maximize the number of conserved interactions between two species. For that purpose they use the following probabilistic model. They associate a binary variable z_{ij} to each possible protein ortholog pair (f_i, y_j) (here f_i and y_j denote fly and yeast proteins from the same Inparanoid cluster), where $z_{ij} = 1$ means that f_i and y_j are functional orthologs. Two variables z_{ij} and z_{kt} are connected if at least one pair of proteins (f_i, f_k) or (y_j, y_t) is connected in its PPI network, and the other one has a common neighbor (or is also connected). Let $N(ij)$ denote the set of indices connected to z_{ij}. Then the probability law of z_{ij} is modeled by:

$$P(z_{ij}|z_{N(ij)}) = \frac{1}{1 + exp\{-\alpha - \beta \sum_{kt \in N(ij)} z_{kt}\}}. \quad (8)$$

The interpretation of this formula is that z_{ij} has more chances to be equal to one when the number of neighbors equal to one is large. When there are only two proteins in cluster f_i and y_j then by definition $z_{ij} = 1$. If f_i and y_j are from different clusters then also by definition $z_{ij} = 0$. The parameters α and β are estimated on the basis of training data, then a Gibbs sampling is performed to define the value of unknown variables z on the test set. We refer to Bandyopadhyay *et al.* (2006) for more details on this method.

- *MP method for exact optimization.* Although intractable in general, we now show that constrained GNA problem (2) can be solved exactly and efficiently in some cases, and propose a new, efficient algorithm based on MP for that purpose. More precisely, we consider the situation where the set of proteins have been clustered into a finite set of L groups $c_1, \ldots, c_L$, which form a partition of $V_G \cup V_H$, and where only proteins within the same group can be matched.[1] This situation, illustrated in Figure 1, represents for example the problem investigated by Bandyopadhyay *et al.* (2006), where proteins of two organisms are first clustered by the Inparanoid algorithm, and functional orthologs are searched within clusters. Let us now consider the L clusters as vertices of a graph, and connect two clusters c_i and c_j if they contain proteins of both organisms that interact in their respective PPI network. For example, in Figure 1, c_1 and c_2 are connected because c_1 contains f_1 from the first organism and y_1 from the second organism, which interact with f_5 and y_3, respectively, both in c_2. The reason why we introduce this graph of clusters is that it allows to decompose the choice of a global matching P into local matchings within each cluster, the dependency between the local choices being described by the edges of the graph. For example, if a cluster is isolated, then the choice of the matching within this cluster has no influence over the total number of conserved interactions apart from interactions within this cluster. In other words, the local matching within an isolated cluster can be optimized independently from the others. On the other hand, if a cluster is connected to other clusters, then changing the matching within this cluster can affect the total number of interactions between proteins of
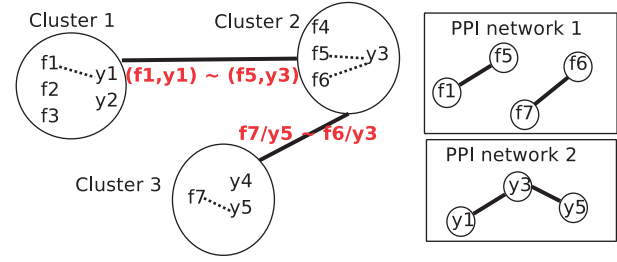
---

[1]Technically, we add dummy nodes in each cluster to obtain the same number of proteins of each species in each cluster.

Fig. 1. Inparanoid cluster network. Two clusters are connected if there exist at least one pair of proteins in one cluster, and one pair of proteins in the other cluster, which may produce a conserved interaction.

different clusters, and the matchings between connected clusters must be chosen synchronously to optimize the total number of conserved interactions.

More formally, if we denote the permutation P restricted to the L clusters by $P_1, \ldots, P_L$, then an important property is that the total number of interactions conserved by P decomposes as:

$$J(P) = \sum_{i=1}^{L} J_1(P_i) + \sum_{i \sim j} J_2(P_i, P_j), \quad (9)$$

where $J_1(P_i)$ denotes the number of conserved interactions within c_i, $J_2(P_i, P_j)$ denotes the number of conserved interactions between c_i and c_j and $i \sim j$ means that c_i is connected to c_j.

While maximizing (9) remains a challenging optimization problem in general, it may be optimized efficiently if the graph of clusters has a particular structure, e.g. if many nodes are isolated or if it contains no loop. For example, Figure 2a shows the graph of clusters for the problem of fly/yeast protein alignment investigated by Bandyopadhyay *et al.* (2006). Interestingly, this graph has no loop. In this case, we can maximize (9) by a particular MP algorithm (Jordan, 2001). The idea of the MP algorithm is similar to the Viterbi algorithm (Viterbi, 1973) widely used to optimize functions over linear graphs, such as finding the most likely set of hidden states in a hidden Markov model (Durbin *et al.*, 1998). Here we describe how to apply MP on a graph without loop to optimize (9). First, we note that each of the permutations involving proteins within a connected component of the graph can be optimized independently from each other, so we just consider a single connected component without loop, i.e. a tree $\mathcal{T}$ of clusters. We choose a vertex of $\mathcal{T}$ that we call root, which allows to define the directions up (towards the root) or down (away from the root) when moving on edges of the graph. Each cluster c_i except the root has a unique parent cluster, namely, the connected cluster in the direction of the root. The clusters connected to a cluster c that are not its parent are called its children and are denoted $ch(c)$. To each node c of $\mathcal{T}$, we associate a vector $u_c \in \mathbb{R}^{\mathcal{P}_c}$, where $\mathcal{P}_c$ is the set of possible local matchings within c, i.e., the set of possible P_c's. The MP algorithm to solve (9) is then a recursive algorithm, which starts from the leaves up to the root in a first phase (the 'forward' step) to find the optimal value of the functional, and then downwards from the root to leaves (the 'backward' step) to find the solution which achieves the optimal value. The forward step at node c solves, for any $P_c \in \mathcal{P}_c$:

$$u_c(P_c) = J_1(P_c) + \sum_{c' \in ch(c)} \max_{P_{c'} \in \mathcal{P}_{c'}} [u_{c'}(P_{c'}) + J_2(P_c, P_{c'})]. \quad (10)$$

At the end of the forward step, the maximum value of the vector u at the root is equal to the maximal value of $J(P)$, and the local permutation which achieves this maximum is the optimal local permutation. In the backward step, the optimal local matching of the children of a cluster are obtained by recovering the local permutations $P_{c'}$ which achieved the optimal value in (10) for the optimal permutation of the parent cluster.
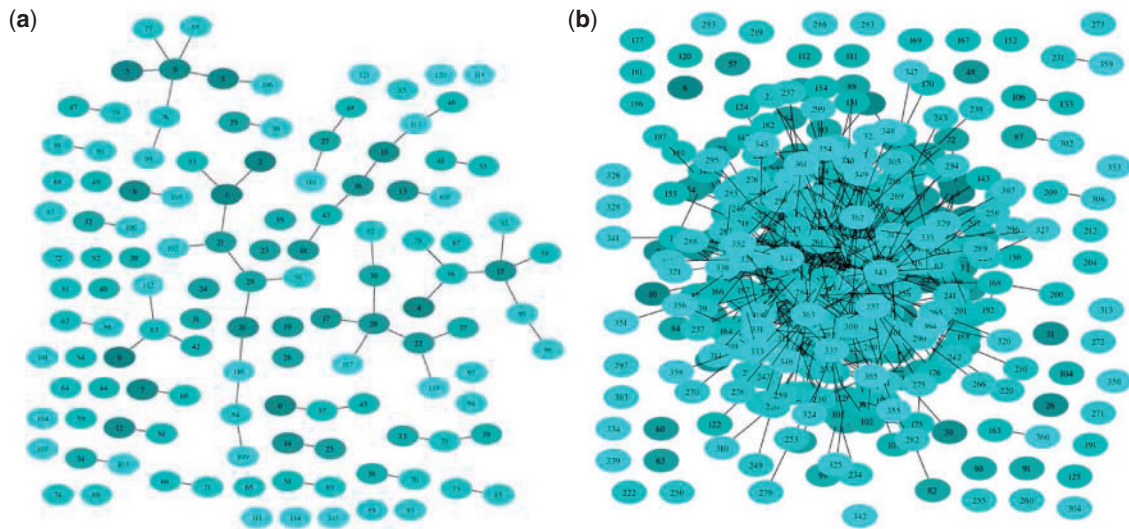
Fig. 2. Inparanoid cluster networks. (**a**) The case of the benchmark data used in Bandyopadhyay *et al.* (2006). (**b**) The case of generalized interactions (1–4), see text.

We note that it is also possible to use the MP algorithm on graphs that are not trees, but which have a small tree-width value (Jordan, 2001). Roughly speaking it means that the graph of clusters is not a tree, we may transform it into a tree by grouping together clusters. If the size of these cluster groups is not very large, then the exact optimization may still be feasible.

4 DATA

In order to compare the performance of the different graph matching methods, we performed several experiments aiming at aligning the PPI networks of the yeast *Saccharomyces cerevisiae* and of the fly *Drosophila melanogaster*, as already investigated by Bandyopadhyay *et al.* (2006) and Singh *et al.* (2008). We downloaded all necessary data from the Supplementary Material of Bandyopadhyay *et al.* (2006) (http://www.cellcircuits. org/Bandyopadhyay2006). The yeast PPI network contains 4389 proteins and 14 319 pairwise interactions, while the fly network contains 7038 proteins and 20 720 interactions. In addition, we also retrieved the set of Inparanoid clusters used by Bandyopadhyay *et al.* (2006), consisting in 2244 cluster covering 2834 yeast proteins and 3881 fly proteins. The majority of these clusters (1552) contains only two proteins (one from fly, one from yeast), while the remaining 692 cluster contain at least two proteins from the same species and one from the other species. Those 692 clusters are called ambiguous in Bandyopadhyay *et al.* (2006), since they do not allow to associate a single protein from the fly to a single protein from the yeast as functional orthologs.

5 RESULTS

We wish to investigate two different questions: (i) compare the ability of the different methods to find alignment with many conserved interactions, and (ii) assess whether conserving more interactions really helps in retrieving more functional orthologs. While the first question can be answered without ambiguity by counting the number of conserved interactions found by the different methods in different settings, the second one, as we will see, remains

difficult to answer due to the lack of large-scale and curated ground truth.

We performed three sets of experiments, in order to compare the different methods in different settings and to test different formulations of the GNA problem. In the first set of experiments, we reproduce the problem studied by Bandyopadhyay *et al.* (2006), where the goal is to disambiguate functional orthologs within Inparanoid clusters using PPI information. This is a particular instance of the constrained GNA problem which turns out to be amenable to exact optimization by the MP method. In the second set of experiments, we generalize the benchmark problem of Bandyopadhyay *et al.* (2006) by adding second-order interactions between proteins in order to account for possible noise in the interaction data or protein duplications. In that case, we are again confronted with a constrained GNA problem, but the increased number of interactions makes its exact minimization intractable and only approximate methods for constrained GNA can be applied. Finally, in a third set of experiments, we discard the knowledge of Inparanoid clusters and directly search a global alignment which balances the similarity between aligned proteins and the number of conserved interactions. This is then an instance of the balanced GNA problem. In all cases, we assess the number of conserved interactions captured by the different methods, as an indicator of how well they solve the GNA problem. Furthermore, since the final objective of PPI network alignment is to match functional orthologs, we assess for each method how many matched pairs are present in the HomoloGene database, a set of curated functional orthologous pairs based on the comparison of the protein as well as the DNA sequence which we consider here as a 'gold standard' for disambiguation purpose.

5.1 Disambiguation of functional orthologs within Inparanoid clusters

The goal of this experiment is to use PPI GNA to select functional orthologs between the yeast and the fly for proteins with several homologs. More precisely, all proteins sequences are first clustered

Table 1. Performance of the different methods for constrained GNA on the benchmark of Bandyopadhyay *et al.* (2006)

Algorithm	MP	MRF	IsoRank	GA	PATH
Number of conserved interactions	**238**	233	228	**238**	**238**
Number of HomoloGene pairs (121 cl.)	41	36	39	41	41
Timing (s)	1–2	10	1–2	1–2	80–100

Each algorithm is evaluated by the number of conserved interactions, number of recovered HomoloGene pairs and the running time. The number of recovered HomoloGene pairs is counted only in 121 ambiguous Inparanoid clusters where PPI data may be used. The data in bold are significant because they correspond to the absolute maximum of the conserved interaction number. The MP (message passing) algorithm is known to be exact, GA and PATH are in bold since they produce the same number of conserved interactions as the MP algorithm.

into groups by the Inparanoid algorithm (Brein *et al.*, 2005), and only proteins from the same cluster can be considered as protein functional orthologs. Then each GNA algorithm tries to find an association of protein functional orthologs which maximizes the total number of conserved interactions. In other words, we try to solve the constrained GNA (2), where the constraints are provided by the Inparanoid clusters. A priori, the most natural definition of 'conserved interaction' for the alignment $(f_1 - y_1)$ and $(f_2 - y_2)$ (where f_1 and f_2 are fly's proteins, and y_1 and y_2 are yeast's proteins) is the following:

1. f_1 interacts with f_2, and y_1 interacts with y_2 in their respective PPI networks.

However, this strict notion of conserved interaction leads to a very small number of potentially conserved interactions. To have more potential interactions, Bandyopadhyay *et al.* (2006) generalized this definition by adding the following two cases, which additionally allow to account for possible duplication or fusion events in the two proteomes:

2. f_1 interacts with f_2 in the fly PPI network, and y_1 has a common neighbor with y_2 in the yeast PPI networks;

3. f_1 has a common neighbor with f_2 in the fly PPI network, and y_1 interacts with y_2 in the yeast PPI networks.

To be able to compare the results of different algorithms, we use this exact definition of conserved interactions (Cases 1–3). Figure 2a presents the network of Inparanoid clusters (as explained in Figure 1) used in Bandyopadhyay *et al.* (2006), where only non-isolated ambiguous clusters are shown. As can be easily seen, this network which contains 121 ambiguous clusters has no loop, which implies that we can use the MP method to find the optimal alignment with the largest number of conserved interactions. Although we know how to solve the problem exactly in this case with the MP method, it is instructive to compare also the results of the different approximate algorithms for constrained GNA, namely, MRF and the constrained versions of IsoRank, GA and PATH. To construct the alignment made by the MRF method (Bandyopadhyay *et al.*, 2006), we downloaded the result file (http://www.cellcircuits.org/Bandyopadhyay2006/data/Bandyopadhyay_results.xls) with probabilities for all possible protein association, and we extracted the one-to-one alignment by taking the most probable pairs. The results of the PATH, GA and IsoRank algorithms were obtained with the GraphM package (Zaslavskiy *et al.*, 2008a).

Table 1 presents the results of all algorithms on this benchmark, in terms of conserved interactions, number of HomoloGene pairs and

running time. We know that the MP algorithm produces the maximal possible value (238 in this case), and an interesting observation is that the GA and the PATH algorithms reach this maximum, while the MRF (233) and the IsoRank (228) algorithms do not. All methods are comparable in terms of CPU time, except for MRF which is one order of magnitude slower on this dataset. Although the differences in number are slight, with only 2% more conserved interactions for MP/GA/PATH than for MRF, and 4% more than for IsoRank, this nevertheless confirms that even on this relatively easy optimization problem neither MRF nor IsoRank finds the optimal solution, which can be found by other methods at no additional computational cost.

Figure 3a and b show some examples where the MRF assignment and the assignment made by the MP, PATH and GA algorithms are different, and illustrate how these differences influence the total number of conserved interactions. For instance, in the Inparanoid cluster 1113, the MRF algorithm associate the fly protein skpA to the yeast protein skp1, while the MP algorithm prefers the assignment skpF to skp1. In the later case, we lose one conserved interaction with pair ago-cdc4, but we gain two new conserved interactions with (vha36 and vm28) and (ef2b and eft2). In another example, shown in Figure 3b, the MP algorithm proposes a different association for the yeast protein act1 in the 94th Inparanoid cluster. This assignment results in two lost and three gained conserved interactions. From a biological point of view, the assignment of the fly protein act87e to act1 proposed by the MRF algorithm seems to be worse that the assignment (act5c and act1) proposed by the MP algorithm. Indeed, although proteins act5c and act87e are very similar (being both from the actine family), it is known that act1 and act5c participate together to the INO80 protein complex (which exhibits chromatin remodeling activity and $3'$ to $5'$ DNA helicase activity), while act87e does not.

In order to assess more systematically and quantitatively whether differences in the number of conserved interactions lead to significant differences in number of correctly assigned functional orthologous pairs, we counted how many pairs in each alignment is reported as functional orthologous in the HomoloGene database, considered here as a 'gold standard'. As shown in Table 1, the number of HomoloGene pairs in each alignment also differs between the different methods, ranging from 36 for MRF to 39 for IsoRank and 41 for MP/GA/PATH. Interestingly, we observe that the methods MP, GA and PATH, which retrieve the largest number of conserved interaction, also result in the largest number HomoloGene pairs (41), which represents a relative increase of 13% compared to MRF (36), and of 5% compared to IsoRank. To illustrate the differences between the methods, Table 2 lists the HomoloGene pairs found by MRF and not MP/GA/PATH, and vice versa. Interestingly, a new method for PPI network alignment was published recently
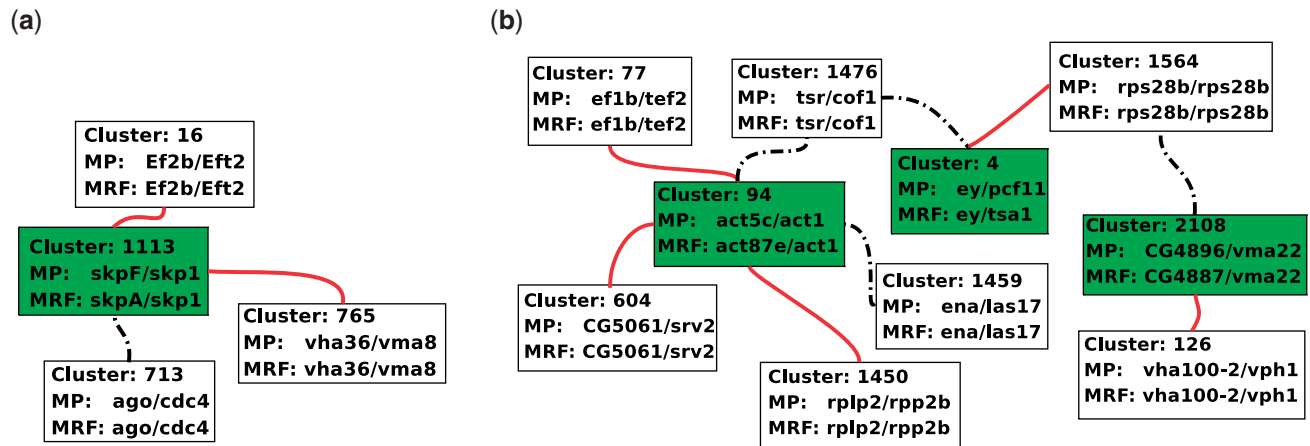
Fig. 3. Illustration of difference between MRF and MP alignment. Each box represents an Inparanoid cluster, white unfilled boxes represent clusters where MP and MRF assignments are the same. Red solid lines represent interactions conserved by MP alignment and not by MRF, black dotted lines represent interactions conserved by MRF and not by MP.

Table 2. HomoloGene orthologs found by the MP method and not by MRF and vice versa

MP		MRF
(TfIIA-S, TOA2)	(RPL23, RPL23A)	(Pros35, PRE5)
(CG13890, ECI1)	(Gapdh1, TDH1)	(Rab11, Ypt31)
(TfIIS, DST1)	(Rpt4, Rpt4)	(Rps26, Rps26A)
(Ef1gamma, TEF4)	(act5c, act1)	(CG6523, YDR098C)
(Glut1, YBR241C)	(Sir2, hst1)	(CG8690, YBR299W)

(Yosef *et al.*, 2008), which detects 37 HomoloGene orthologs on the same set of proteins. This puts its between MRF and IsoRank according to this criterion.

The validity of taking HomoloGene as a 'gold standard' for assessing the number of correctly assigned homologous pairs remains, however, subject to discussion. Indeed, although HomoloGene clusters are defined using a variety of evidences, they are mainly driven by sequence similarity. To illustrate this, we assessed the performance of a simple alignment method that matches pairs within an ambiguous cluster by maximizing the total sequence similarity over matched pairs. This method does not use any PPI information for the matching. The resulting alignment has only 184 conserved interaction, which is not surprisingly much worse than all methods which take PPI into account. However, the resulting matched pairs contain 43 HomoloGene pairs, which is more than all methods taking into account PPI. This shows that the number of HomoloGene pairs as an indicator should be taken with caution, since it favors methods which focus on matching proteins based on sequence similarity only.

5.2 Disambiguation of Inparanoid clusters with second-order interactions

The idea of Bandyopadhyay *et al.* (2006), to expand the natural notion of conserved interaction (Case 1) to Cases 2 and 3, aims to take into account second-order interactions, that is, when two proteins do not interact directly to each other have a common

neighbor. Another natural generalization of the notion of conserved interaction is then the following case:

> 4. f_1 has a common neighbor with f_2, and y_1 has a common neighbor with y_2, in their respective PPI networks.

Adding interactions according to this rule makes the problem computationally more difficult, since ambiguous clusters become more connected. Indeed, while we were able to solve the original problem exactly with the MP algorithm, the network of Inparanoid clusters when Cases 1–4 are included takes the form presented in Figure 2b. Contrary to the previous network (Cases 1–3 in Figure 2a), the new network has loops and is not amenable to exact optimization with the MP procedure. Only approximate algorithms can be applied in this case.

In order to compare all methods (except MP) in this new setting, we re-implemented the MRF algorithm with the new data. The estimated values of the model parameters [see details in Bandyopadhyay *et al.* (2006)] are $\alpha = 0.51$ and $\beta = -6.87$. We used the same training and test data as those used used in Bandyopadhyay *et al.* (2006) to estimate them. Then, we estimated the probabilities of being protein orthologs for potential pairs of proteins by Gibbs sampling, and obtained a one-to-one alignment based on the most probable associations.

Table 3 shows the results obtained by the different graph matching algorithms. Although we do not know the maximum number of interactions that can be conserved in this case, we observe again that PATH and GA find solutions with 3–4% more interactions conserved than MRF and IsoRank. There is no clear difference in the number of HomoloGene pairs between the different methods, and the addition of second-order interactions has no obvious effects on this indicator neither: it leads to a gain of three pairs for MRF, but to a loss of one pair for IsoRank and PATH, and to no change for GA.

5.3 Global PPI network alignment by balancing sequence and interaction conservation

In this last series of experiments, we consider the problem proposed by Singh *et al.* (2008), for which IsoRank reflects the state-of-the-art: find a global PPI alignment by balancing the sequence

Table 3. Performance of the different methods for constrained GNA on the benchmark of Bandyopadhyay *et al.* (2006) with second-order interactions added

Algorithm	MRF	IsoRank	GA	PATH
Number of conserved interactions	1112	1101	1140	1143
Number of HomoloGene pairs (121 cl.)	39	38	41	40
Number of HomoloGene pairs (602 cl.)	172	167	172	166
Timing (s)	623	31	372	1542

The number of recovered HomoloGene pairs is counting on the 121 Inparanoid clusters from the previous section as well as on the new 602 ambiguous Inparanoid clusters that have second-order interaction with other Inparanoid clusters.
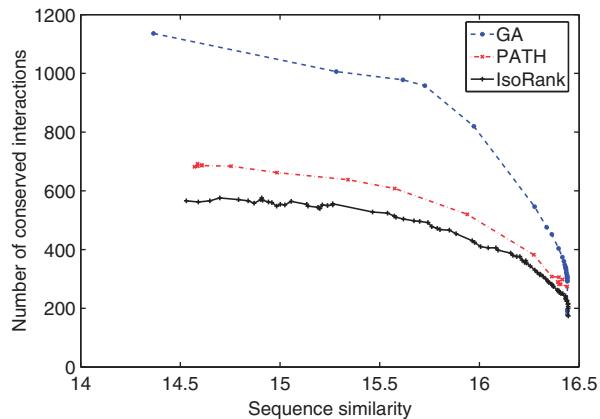
Fig. 4. Algorithm performance comparison. Number of conserved interaction $J(P)$ versus sequence similarity $S(P)$.

similarity in matched pairs with the total number of conserved interactions, allowing in particular matches between proteins in different Inparanoid clusters if they allow an increased number of conserved interactions. For this application, we can only compare the three methods for balanced GNA, namely, IsoRank, GA and PATH. The trade-off between matching proteins with similar sequences and matching with a lot of conserved interactions is controlled by the parameter λ in (4) and (7). The greater the λ, the more attention we pay to the sequence similarity and the less to the number of conserved interactions. For each method, by varying λ, we therefore obtain a family of alignments with different compromise found between the number of conserved interactions $J(P)$ (4) and the summary sequence similarity score $S(P)$ (4).

Figure 4 shows the different trade-offs that are found by the different methods. For a given level of average sequence similarity, we wish to have the largest possible number of conserved pairs. We observe that over all the range of average sequence similarity, the GA algorithms clearly outperforms PATH, which itself outperforms IsoRank. For example, for the trade-off parameter choice advocated by (Singh *et al.*, 2008) for IsoRank ($\lambda = 0.6$), IsoRank finds an alignment with 566 conserved interactions, corresponding to an average sequence similarity score in the matched pairs of 15.26. At this level of average sequence similarity, PATH and GA find alignments with, respectively, 678 and 1006 interactions, which corresponds to relative improvements of, respectively, 20 and 78%.

Again, there is still only limited objective evidence that optimizing the number of conserved interactions leads to better matching

in terms of functional orthology detection. As an attempt to test this fact, we first counted, for each alignment, the number of HomoloGene pairs in the alignment. However, we observed that, for each method, this number increases monotonically when more weight is given to sequence similarity as opposed to interaction conservation. This again highlights the limitation of this criterion, which is optimized by construction when sequences are optimally matched in terms of similarity. We then attempted to compare the different alignments in terms of mean similarity between Gene Ontology (GO) annotations of matched pairs. In order to compare GO annotations of two proteins, we tested the method presented by Singh *et al.* (2008) to compute the functional coherence of a pair. However, we were not able to observe any clear difference between the methods, or between the different parameter choice for each individual method. The maximum mean functional coherence over the choice of the trade-off parameter is 0.519, 0.509 and 0.522 for IsoRank, GA and PATH, respectively. However, the fluctuations of this score when the parameters change are so large that these maximum values are not significantly different. This is due to the fact that the number of annotated proteins remains limited, and that they are rarely annotated with such precision that it is possible to clearly differentiate true functional orthologs from spurious ones (Bandyopadhyay *et al.*, 2006). For example, when we estimate the functional score of a given alignment, there is rarely >15–20% of pairs with GO annotations.

6 DISCUSSION

We presented two general formulations for the GNA problem. The constrained GNA formulation corresponds to a situation where we have a strong a priori about which pairs can be matched. In the balanced GNA problem, we replace the binary constraints on which pairs are allowed by a more global objective function that balances the matching of similar proteins with the conservation of interactions, with a parameter to smoothly control the trade-off between these two contradictory goals. While MRF and IsoRank are popular methods for these two formulations, we proposed in this article new methods which lead to significantly better alignments, when we assess the quality of an alignment in terms how many conserved interactions are retrieved. In particular, the MP method, when it is applicable, finds the optimal solution of a constrained GNA problem, and the GA method provides consistently good results in both cases. The question of which formulation is the best for a given application and dataset, between the constrained and balanced GNA, remains largely open and worth further systematic investigations. Regarding the relative performance of the different methods in terms of how many conserved interactions they find,

we observed that the MP/GA/PATH methods outperform MRF and IsoRank in both situations. This is not so surprising given that, once the problem is explicitly stated as a graph matching problems, it makes sense to use methods borrowing ideas and techniques from state-of-the-art graph matching approaches. The impressive performance of GA compared to PATH in the balanced GNA experiment (Fig. 4) is more surprising, given the good performance of PATH on a number of other benchmarks (Zaslavskiy *et al.*, 2008c). We believe that this weakness of PATH is due to the large difference in the number of nodes between the two networks. Indeed, the resulting large number of dummy nodes that must be added generate singularities in the convex relaxation in the PATH algorithm.

The GNA problems we studied have several extensions. First, it may be interesting to consider alignment of weighted PPI networks with weights representing, for instance, experimental evidence of interaction existence. Interestingly, the PATH, GA and IsoRank algorithm can be applied directly to a weighted network, by just replacing the binary graph adjacency matrix by a real-valued matrix. Another relevant extension is the alignment of multiple PPI networks, corresponding to more than two species, via pairwise comparisons as it was presented by Singh *et al.* (2008). Finally, it may be relevant in some cases to match one protein of one species with several proteins of the other species, to account for possible duplications or fusion events. An interesting property of the PATH algorithm is the fact that estimate a permutation matrix by first solving a relaxed problem. The solution of the relaxed problem is a doubly stochastic matrix whose entries can be interpreted as probabilities for proteins to be functional orthologs (Zaslavskiy *et al.*, 2008c) . Therefore, in order to allow many-to-many assignments of proteins, we could use the solution of the convex relaxation.

Finally, although progresses in graph alignment algorithms can be monitored by objective quantitative measures such as the number of conserved interactions, their biological relevance remains difficult to assess. In particular, for the detection of functional orthologs, it is apparent that current GO annotations or curated databases of functional orthologs are either biased by construction (e.g. HomoloGene), or not precise enough and too scarce for systematic evaluation (e.g. GO annotations). We believe we are reaching a point where more experimental validations are needed. On the other hand, there are many other possible applications for efficient graph matching algorithms scaling to large biological networks, such as phylogenetic comparison of sets of networks, detection of new conserved pathways or curation of PPI data. We expect the methods proposed in this article to have a direct impact in these applications.

Conflict of Interest: none declared.

REFERENCES

Aebersold,R. and Mann,M. (2003) Mass spectrometry-based proteomics. *Nature*, **422**, 198–207.

Almohamad,H. and Duffuaa,S. (1993) A linear programming approach for the weighted graph matching problem. *IEEE Trans. Inform. Theor.*, **15**, 522–525.

Bandyopadhyay,S. *et al.* (2006) Systematic identification of functional orthologs based on protein network comparison. *Genome Res.*, **16**, 428–435.

Berg,J. and Lässig,M. (2006) Cross-species analysis of biological networks by bayesian alignment. *Proc. Natl Acad. Sci. USA*, **103**, 10967–10972.

Brein,K. *et al.* (2005) Inparanoid: a comprehensive database of eukaryothic orthologs. *Nucleic Acids Res.*, **33**, D476–D480.

Caelli,T. and Kosinov,S. (2004) An eigenspace projection clustering method for inexact graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, **26**, 515–519.

Conte,D. *et al.* (2004) Thirty years of graph matching in pattern recognition. *Intern. J. Pattern Recognit. Artif. Intell.*, **18**, 265–298.

Durbin,R. *et al.* (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, NY.

Fields,S. and Song,O. (1989) A novel genetic system to detect protein-protein interactions. *Nature*, **340**, 245–246.

Flannick,J. *et al.* (2006) Graemlin: general and robust alignment of multiple large interaction networks. *Genome Res.*, **16**, 1169–1181.

Gold,S. and Rangarajan,A. (1996) A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, **18**, 377–388.

Jordan,M. (ed.) (2001) *Learning in Graphical Models*. The MIT Press, Cambridge.

Kelley,B. *et al.* (2003) Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc. Natl Acad. Sci. USA*, **100**, 11394–11399.

Kelley,B. *et al.* (2004) PathBLAST: a tool for alignment of protein interaction networks. *Nucleic Acids Res.*, **32**, W83–W88.

Koyutürk,M. *et al.* (2006) Pairwise alignment of protein interaction networks. *J. Comput. Biol.*, **13**, 182–199.

Kuhn,H.W. (1955) The Hungarian method for the assignment problem. *Nav. Res.*, **2**, 83–97.

Remm,M. *et al.* (2001) Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *J. Mol. Biol.*, **314**, 1041–1052.

Sharan,R. *et al.* (2005) Conserved patterns of protein interaction in multiple species. *Proc. Natl Acad. Sci. USA*, **102**, 1974–1979.

Singh,R. *et al.* (2008) Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proc. Natl Acad. Sci. USA*, **105**, 12763–12768.

Sjölander,K. (2004) Phylogenomic inference of protein molecular function: advances and challenges. *Bioinformatics*, **20**, 170–179.

Suthram,S. *et al.* (2005) The plasmodium protein network diverges from those of other eukaryotes. *Nature*, **438**, 108–112.

Umeyama,S. (1988) An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, **10**, 695–703.

Viterbi,A. (1973) Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inform. Theor.*, **13**, 260–269.

Yosef,N. *et al.* (2008) Improved network-based identification of protein orthologs. *Bioinformatics*, **24**, i200–i206.

Zaslavskiy,M. *et al.* (2008a) GRAPHM: graph matching package. Available at http://cbio.ensmp.fr/graphm (last accessed date March 2009).

Zaslavskiy,M. *et al.* (2008b) A path following algorithm for graph matching. In *Image and Signal Processing, Proceedings of the 3rd International Conference, ICISP 2008*, Elmoataz,A. *et al.* eds, Vol. 5099 of *LNCS*, Springer, Berlin/Heidelberg, pp. 329–337.

Zaslavskiy,M. *et al.* (2008c). A path following algorithm for the graph matching problem. *Technical Report 00232851, HAL*. Mines ParisTech.