Lagrangian Relaxation Applied to Sparse Global Network Alignment

Mohammed El-Kebir[1,2,3], Jaap Heringa[2,3,4], and Gunnar W. Klau[1,3]

[1] Centrum Wiskunde & Informatica, Life Sciences Group, Science Park 123,
1098 XG Amsterdam, The Netherlands
{m.el-kebir,gunnar.klau}@cwi.nl
[2] Centre for Integrative Bioinformatics VU (IBIVU), VU University Amsterdam,
De Boelelaan 1081A, 1081 HV Amsterdam, The Netherlands
heringa@few.vu.nl
[3] Netherlands Institute for Systems Biology, Amsterdam, The Netherlands
[4] Netherlands Bioinformatics Centre, The Netherlands

Abstract. Data on molecular interactions is increasing at a tremendous pace, while the development of solid methods for analyzing this network data is lagging behind. This holds in particular for the field of comparative network analysis, where one wants to identify commonalities between biological networks. Since biological functionality primarily operates at the network level, there is a clear need for topology-aware comparison methods. In this paper we present a method for global network alignment that is fast and robust, and can flexibly deal with various scoring schemes taking both node-to-node correspondences as well as network topologies into account. It is based on an integer linear programming formulation, generalizing the well-studied quadratic assignment problem. We obtain strong upper and lower bounds for the problem by improving a Lagrangian relaxation approach and introduce the software tool NATALIE 2.0, a publicly available implementation of our method. In an extensive computational study on protein interaction networks for six different species, we find that our new method outperforms alternative state-of-the-art methods with respect to quality and running time. An extended version of this paper including proofs and pseudo code is available at http://arxiv.org/pdf/1108.4358v1.

1 Introduction

In the last decade, data on molecular interactions has increased at a tremendous pace. For instance, the STRING database [24], which contains protein protein interaction (PPI) data, grew from 261,033 proteins in 89 organisms in 2003 to 5,214,234 proteins in 1,133 organisms in May 2011, more than doubling the number of proteins in the database every two years. The same trends can be observed for other types of biological networks, including metabolic, gene-regulatory, signal transduction and metagenomic networks, where the latter can incorporate the excretion and uptake of organic compounds through, for example, a microbial community [21, 12]. In addition to the plethora of experimentally derived
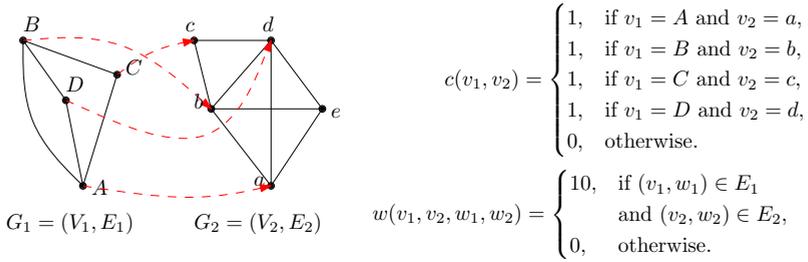
network data for many species, also the structure and behavior of molecular networks have become intensively studied over the last few years [2], leading to the observation of many conserved features at the network level. However, the development of solid methods for analyzing network data is lagging behind, particularly in the field of comparative network analysis. Here, one wants to identify commonalities between biological networks from different strains or species, or derived form different conditions. Based on the assumption that evolutionary conservation implies functional significance, comparative approaches may help (i) improve the accuracy of data, (ii) generate, investigate, and validate hypotheses, and (iii) transfer functional annotations. Until recently, the most common way of comparing two networks has been to solely consider node-to-node correspondences, for example by finding homologous relationships between nodes (e.g. proteins in PPI networks) of either network, while the topology of the two networks has not been taken into account. Since biological functionality primarily operates at the network level, there is a clear need for topology-aware comparison methods. In this paper we present a network alignment method that is fast and robust, and can flexibly deal with various scoring schemes taking both node-to-node correspondences as well as network topologies into account.

Previous Work. Network alignment establishes node correspondences based on both node-to-node similarities and conserved topological information. Similar to sequence alignment, *local* network alignment aims at identifying one or more shared subnetworks, whereas *global* network alignment addresses the overall comparison of the complete input networks.

Over the last years a number of methods have been proposed for both global and local network alignment, for example PATHBLAST [14], NETWORKBLAST [22], MAWISH [16], GRAEMLIN [8], ISORANK [23], GRAAL [17], and SUBMAP [4]. PATHBLAST heuristically computes high-scoring similar paths in two PPI networks. Detecting protein complexes has been addressed with NETWORK-BLAST by Sharan et al. [22], where the authors introduce a probabilistic model and propose a heuristic greedy approach to search for shared complexes. Koyutürk et al. [16] use a more elaborate scoring scheme based on an evolutionary model to compute local pairwise alignments of PPI networks. The ISORANK algorithm by Singh et al. [23] approaches the global alignment problem by preferably matching nodes which have a similar neighborhood, which is elegantly solved as an eigenvalue problem. Kuchaiev et al. [17] take a similar approach. Their method GRAAL matches nodes that share a similar distribution of so-called graphlets, which are small connected non-isomorphic induced subgraphs.

In this paper we focus on pairwise global network alignment, where an alignment is scored by summing up individual scores of aligned node and interaction pairs. Among the above mentioned methods, ISORANK and GRAAL use a scoring model that can be expressed in this manner.

Contribution. We present an algorithm for global network alignment based on an integer linear programming (ILP) formulation, generalizing the well-studied quadratic assignment problem (QAP). We improve upon an existing Lagrangian

$$c(v_1, v_2) = \begin{cases} 1, & \text{if } v_1 = A \text{ and } v_2 = a, \\ 1, & \text{if } v_1 = B \text{ and } v_2 = b, \\ 1, & \text{if } v_1 = C \text{ and } v_2 = c, \\ 1, & \text{if } v_1 = D \text{ and } v_2 = d, \\ 0, & \text{otherwise.} \end{cases}$$

$$w(v_1, v_2, w_1, w_2) = \begin{cases} 10, & \text{if } (v_1, w_1) \in E_1 \\ & \text{and } (v_2, w_2) \in E_2, \\ 0, & \text{otherwise.} \end{cases}$$

Fig. 1. Example of a network alignment. With the given scoring function, the alignment has a score of $4 + 40 = 44$.

relaxation approach presented in previous work [15] to obtain strong upper and lower bounds for the problem. We exploit the closeness to QAP and generalize a dual descent method for updating the Lagrangian multipliers to the generalized problem. We have implemented the revised algorithm from scratch as the software tool NATALIE 2.0. In an extensive computational study on protein interaction networks for six different species, we compare NATALIE 2.0 to GRAAL and ISORANK, evaluating the number of conserved edges as well as functional coherence of the modules in terms of GO annotation. We find that NATALIE 2.0 outperforms the alternative methods with respect to quality and running time. Our software tool NATALIE 2.0 as well as all data sets used in this study are publicly available at http://planet-lisa.net.

2   Preliminaries

Given two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, an *alignment* $a : V_1 \rightarrow V_2$ is a *partial injective function* from V_1 to V_2. As such we have that an alignment relates every node in V_1 to at most one node in V_2 and that conversely every node in V_2 has at most one counterpart in V_1. An alignment is assigned a real-valued *score* using an additive scoring function s defined as follows:

$$s(a) = \sum_{v \in V_1} c(v, a(v)) + \sum_{\substack{v, w \in V_1 \\ v < w}} w(v, a(v), w, a(w)) \tag{1}$$

where $c : V_1 \times V_2 \rightarrow \mathbb{R}$ is the score of aligning a pair of nodes in V_1 and V_2 respectively. On the other hand, $w : V_1 \times V_2 \times V_1 \times V_2 \rightarrow \mathbb{R}$ allows for scoring topological similarity. The problem of global pairwise network alignment (GNA) is to find the highest scoring alignment a^*, i.e. $a^* = \arg\max s(a)$. Figure 1 shows an example.

NP-hardness of GNA follows by a simple reduction from the decision problem CLIQUE, which asks whether there is a clique of cardinality at least k in a given simple graph $G = (V, E)$ [13]. The corresponding GNA instance concerns the alignment of the complete graph of k vertices $K_k = (V_k, E_k)$ with G using the

scoring function $s(a) = |\{(v, w) \in E_k \mid (a(v), a(w)) \in E\}|$. Since an alignment is injective, there is a clique of cardinality at least k if and only if the cost of the optimal alignment is $\binom{k}{2}$. The close relationship of GNA with the quadratic assignment problem is more easily observed when formulating GNA as a mathematical program. Throughout the remainder of the text we use dummy variables $i, j \in \{1, \ldots, |V_1|\}$ and $k, l \in \{1, \ldots, |V_2|\}$ to denotes nodes in V_1 and V_2, respectively. Let C be a $|V_1| \times |V_2|$ matrix such that $c_{ik} = c(i, k)$ and let W be a $(|V_1| \times |V_2|) \times (|V_1| \times |V_2|)$ matrix whose entries w_{ikjl} correspond to interaction scores $w(i, k, j, l)$. Now we can formulate GNA as

$$\max_x \quad \sum_{i,k} c_{ik} x_{ik} + \sum_{\substack{i,j \\ i<j}} \sum_{\substack{k,l \\ k \neq l}} w_{ikjl} x_{ik} x_{jl} \tag{IQP}$$

$$\text{s.t.} \quad \sum_l x_{jl} \leq 1 \qquad \forall j \tag{2}$$

$$\sum_j x_{jl} \leq 1 \qquad \forall l \tag{3}$$

$$x_{ik} \in \{0, 1\} \qquad \forall i, k \tag{4}$$

where the decision variable x_{ik} indicates whether the i-th node in V_1 is aligned with the k-th node in V_2. The above formulation shares many similarities with Lawler's formulation [19] of the QAP. However, instead of finding an assignment we are interested in finding a matching, which is reflected in constraints (2) and (3) being inequalities rather than equalities. As can be seen in (1) we only consider the upper triangle of W rather than the entire matrix. An analogous way of looking at this, is to consider W to be symmetric. This is usually not the case for QAP instances. In addition, due to the fact that biological input graphs are typically sparse, we have that W is sparse as well. These differences allow us to come up with an effective method of solving the problem as we will see in the following.

3   Method

The relaxation presented here follows the same lines as the one given by Adams and Johnson for the QAP [1]. We start by linearizing (IQP) by introducing binary variables y_{ikjl} defined as $y_{ikjl} := x_{ik} x_{jl}$ and constraints $y_{ikjl} \leq x_{jl}$ and $y_{ikjl} \leq x_{ik}$ for all $i \leq j$ and $k \neq l$. If we assume that all entries in W are positive, we do not need to enforce that $y_{ikjl} \geq x_{ik} + x_{jl} - 1$. In Section 5 we will discuss this assumption. Rather than using the aforementioned constraints, we make use of a stronger set of constraints which we obtain by multiplying constraints (2) and (3) by x_{ik}:

$$\sum_{l \neq k} y_{ikjl} = \sum_{l \neq k} x_{ik} x_{jl} \leq \sum_l x_{ik} x_{jl} \leq x_{ik}, \quad \forall i, j, k, \ i < j \tag{5}$$

$$\sum_{j > i} y_{ikjl} = \sum_{j > i} x_{ik} x_{jl} \leq \sum_j x_{ik} x_{jl} \leq x_{ik}, \quad \forall i, k, l, \ k \neq l \tag{6}$$

We proceed by splitting the variable y_{ikjl} (where $i < j$ and $k \neq l$). In other words, we extend the objective function such that the counterpart of y_{ikjl} becomes y_{jlik}. This is accomplished by rewriting the dummy constraint in (6) to $j \neq i$. In addition, we split the weights: $w_{ikjl} = w_{jlik} = (w'_{ikjl}/2)$ where w'_{ikjl} denotes the original weight. Furthermore, we require that the counterparts of the split decision variables assume the same value, which amounts to

$$\max_{x,y} \quad \sum_{i,k} c_{ik}x_{ik} + \sum_{\substack{i,j \\ i<j}}\sum_{\substack{k,l \\ k\neq l}} w_{ikjl}y_{ikjl} + \sum_{\substack{i,j \\ i>j}}\sum_{\substack{k,l \\ k\neq l}} w_{ikjl}y_{ikjl} \qquad \text{(ILP)}$$

s.t. (2), (3) and (4)

$$\sum_{\substack{l \\ l\neq k}} y_{ikjl} \leq x_{ik} \qquad\qquad \forall i,j,k, \ i \neq j \qquad (7)$$

$$\sum_{\substack{j \\ j\neq i}} y_{ikjl} \leq x_{ik} \qquad\qquad \forall i,k,l, \ k \neq l \qquad (8)$$

$$y_{ikjl} = y_{jlik} \qquad\qquad \forall i,j,k,l, \ i < j, k \neq l \qquad (9)$$

$$y_{ikjl} \in \{0,1\} \qquad\qquad \forall i,j,k,l, \ i \neq j, k \neq l \qquad (10)$$

We can solve the continuous relaxation of (ILP) via its Lagrangian dual by dualizing the linking constraints (9) with multiplier λ:

$$\min_{\lambda} \quad Z_{\mathrm{LD}}(\lambda) \ , \qquad\qquad (\text{LD})$$

where $Z_{\mathrm{LD}}(\lambda)$ equals

$$\max_{x,y} \quad \sum_{i,k} c_{ik}x_{ik} + \sum_{\substack{i,j \\ i<j}}\sum_{\substack{k,l \\ k\neq l}} (w_{ikjl}+\lambda_{ikjl})y_{ikjl} + \sum_{\substack{i,j \\ i>j}}\sum_{\substack{k,l \\ k\neq l}} (w_{ikjl}-\lambda_{jlik})y_{ikjl}$$

s.t.   (2), (3), (4), (7), (8), (10)

Now that the linking constraints have been dualized, one can observe that the remaining constraints decompose the variables into $|V_1||V_2|$ disjoint groups, where variables across groups are not linked by any constraint, and where each group contains a variable x_{ik} and variables y_{ikjl} for $j \neq i$ and $l \neq k$. Hence, we have

$$Z_{\mathrm{LD}}(\lambda) \ = \ \max_{x} \quad \sum_{i,k} [c_{ik} + v_{ik}(\lambda)]x_{ik} \quad \text{s.t.} \quad (2), (3) \text{ and } (4) \qquad (\text{LD}_\lambda)$$

which corresponds to a maximum weight bipartite matching problem on the so-called *alignment graph* $G_m = (V_1 \cup V_2, E_m)$. In the general case G_m is a complete bipartite graph, i.e. $E_m = \{(i,k) \mid i \in V_1, v_2 \in V_2\}$. However, by exploiting biological knowledge one can make G_m more sparse by excluding biologically-unlikely edges (see Section 4). For the global problem, the weight of a matching edge (i,k) is set to $c_{ik} + v_{ik}(\lambda)$, where the latter term is computed as

$$v_{ik}(\lambda) = \max_y \sum_{\substack{j \\ j>i}} \sum_{\substack{l \\ l \neq k}} (w_{ikjl} + \lambda_{ikjl})y_{ikjl} + \sum_{\substack{j \\ j<i}} \sum_{\substack{l \\ l \neq k}} (w_{ikjl} - \lambda_{jlik})y_{ikjl} \qquad (\text{LD}_\lambda^{ik})$$

$$\text{s.t.} \quad \sum_{\substack{l \\ l \neq k}} y_{ikjl} \leq 1 \qquad\qquad\qquad \forall j, \; j \neq i \quad (11)$$

$$\sum_{\substack{j \\ j \neq i}} y_{ikjl} \leq 1 \qquad\qquad\qquad \forall l, \; l \neq k \quad (12)$$

$$y_{ikjl} \in \{0,1\} \qquad\qquad\qquad \forall j, l. \quad (13)$$

Again, this is a maximum weight bipartite matching problem on the same alignment graph but excluding edges incident to either i or k and using different edge weights: the weight of an edge (j,l) is $w_{ikjl} + \lambda_{ikjl}$ if $j > i$, or $w_{ikjl} - \lambda_{jlik}$ if $j < i$. So in order to compute $Z_{\text{LD}}(\lambda)$, we need to solve a total number of $|V_1||V_2| + 1$ maximum weight bipartite matching problems, which, using the Hungarian algorithm [18,20] can be done in $O(n^5)$ time, where $n = \max(|V_1|, |V_2|)$. In case the alignment graph is sparse, i.e. $O(|E_m|) = O(n)$, $Z_{\text{LD}}(\lambda)$ can be computed in $O(n^4 \log n)$ time using the successive shortest path variant of the Hungarian algorithm [7]. It is important to note that for any λ, $Z_{\text{LD}}(\lambda)$ is an upper bound on the score of an optimal alignment. This is because any alignment a is feasible to $Z_{\text{LD}}(\lambda)$ and does not violate the original linking constraints and therefore has an objective value equal to $s(a)$. In particular, the optimal alignment a^* is also feasible to $Z_{\text{LD}}(\lambda)$ and hence $a^* \leq Z_{\text{LD}}(\lambda)$. Since the two sets of problems resulting from the decomposition both have the integrality property [6], the smallest upper bound we can achieve equals the linear programming (LP) bound of the continuous relaxation of (ILP) [9]. However, computing the smallest upper bound by finding suitable multipliers is much faster than solving the corresponding LP. Given solution (x, y) to $Z_{\text{LD}}(\lambda)$, we obtain a lower bound on $s(a^*)$, denoted $Z_{\text{lb}}(\lambda)$, by considering the score of the alignment encoded in x.

3.1   Solving Strategies

In this section we will discuss strategies for identifying Lagrangian multipliers λ that yield an as small as possible gap between the upper and lower bound resulting from the solution to $Z_{\text{LD}}(\lambda)$.

Subgradient Optimization. We start by discussing subgradient optimization, which is originally due to Held and Karp [10]. The idea is to generate a sequence $\lambda^0, \lambda^1, \ldots$ of Lagrangian multiplier vectors starting from $\lambda^0 = \mathbf{0}$ as follows:

$$\lambda_{ikjl}^{t+1} = \lambda_{ikjl}^t - \frac{\alpha \cdot (Z_{\text{LD}}(\lambda) - Z_{\text{lb}}(\lambda))}{\|g(\lambda^t)\|^2} g(\lambda_{ikjl}^t) \qquad \forall i, j, k, l, \; i < j, k \neq l \quad (14)$$

where $g(\lambda_{ikjl}^t)$ corresponds to the subgradient of multiplier λ_{ikjl}^t, i.e. $g(\lambda_{ikjl}^t) = y_{ikjl} - y_{jlik}$, and α is the step size parameter. Initially α is set to 1 and it is halved

if neither $Z_{\text{LD}}(\lambda)$ nor $Z_{\text{lb}}(\lambda)$ have improved for over N consecutive iterations. Conversely, α is doubled if M times in a row there was an improvement in either $Z_{\text{LD}}(\lambda)$ or $Z_{\text{lb}}(\lambda)$ [5]. In case all subgradients are zero, the optimal solution has been found and the scheme terminates. Note that this is not guaranteed to happen. Therefore we abort the scheme after exceeding a time limit or a pre-specified number of iterations. In addition, we terminate if α has dropped below machine precision.

Dual Descent. In this section we derive a dual descent method which is an extension of the one presented in [1]. The dual descent method takes as a starting point the dual of $Z_{\text{LD}}(\lambda)$:

$$Z_{\text{LD}}(\lambda) = \min_{\alpha,\beta} \quad \sum_i \alpha_i + \sum_k \beta_k \tag{15}$$

$$\text{s.t.} \quad \alpha_i + \beta_k \geq c_{ik} + v_{ik}(\lambda) \qquad \forall i, k \tag{16}$$

$$\alpha_i \geq 0 \qquad \forall i \tag{17}$$

$$\beta_k \geq 0 \qquad \forall k \tag{18}$$

where the dual of $v_{ik}(\lambda)$ is

$$v_{ik}(\lambda) = \min_{\mu,\nu} \quad \sum_{\substack{j \\ j \neq i}} \mu_j^{ik} + \sum_{\substack{l \\ l \neq k}} \nu_l^{ik} \tag{19}$$

$$\text{s.t.} \quad \mu_j^{ik} + \nu_l^{ik} \geq w_{ikjl} + \lambda_{ikjl} \qquad \forall j, l, \; j > i, l \neq k \tag{20}$$

$$\mu_j^{ik} + \nu_l^{ik} \geq w_{ikjl} - \lambda_{jlik} \qquad \forall j, l, \; j < i, l \neq k \tag{21}$$

$$\mu_j^{ik} \geq 0 \qquad \forall j \tag{22}$$

$$\nu_l^{ik} \geq 0 \qquad \forall l. \tag{23}$$

Suppose that for a given λ^t we have computed dual variables (α, β) solving (15) with objective value $Z_{\text{LD}}(\lambda^t)$, as well as dual variables (μ^{ik}, ν^{ik}) yielding values $v_{ik}(\lambda)$ to linear programs (19). The goal now is to find λ^{t+1} such that the resulting bound is better or just as good, i.e. $Z_{\text{LD}}(\lambda^{t+1}) \leq Z_{\text{LD}}(\lambda^t)$. We prevent the bound from increasing, by ensuring that the dual variables (α, β) remain feasible to (15). This we can achieve by considering the slacks: $\pi_{ik}(\lambda) = \alpha_i + \beta_k - c_{ik} - v_{ik}(\lambda)$. So for (α, β) to remain feasible, we can only allow every $v_{ik}(\lambda^t)$ to increase by as much as $\pi_{ik}(\lambda^t)$. We can achieve such an increase by considering linear programs (19) and their slacks defined as

$$\gamma_{ikjl}(\lambda) = \begin{cases} \mu_j^{ik} + \nu_l^{ik} - w_{ikjl} + \lambda_{ikjl}, & \text{if } j > i, \\ \mu_j^{ik} + \nu_l^{ik} - w_{ikjl} - \lambda_{jlik}, & \text{if } j < i, \end{cases} \qquad \forall j, l, \; j \neq i, l \neq k, \tag{24}$$

and update the multipliers in the following way.

Lemma 1. *The adjustment scheme below yields solutions to linear programs* (19) *with objective values $v_{ik}(\lambda^{t+1})$ at most $\pi_{ik}(\lambda^t) + v_{ik}(\lambda^t)$ for all i, k.*

$$\lambda_{ikjl}^{t+1} = \lambda_{ikjl}^t + \varphi_{ikjl} \left[\gamma_{ikjl}(\lambda^t) + \tau_{ik} \left(\frac{1}{2(n_1 - 1)} + \frac{1}{2(n_2 - 1)} \right) \pi_{ik}(\lambda^t) \right]$$
$$- \varphi_{jlik} \left[\gamma_{jlik}(\lambda^t) + \tau_{jl} \left(\frac{1}{2(n_1 - 1)} + \frac{1}{2(n_2 - 1)} \right) \pi_{jl}(\lambda^t) \right] \quad (25)$$

for all $j, l, i < j, k \neq l$, where $n_1 = |V_1|$, $n_2 = |V_2|$, and $0 \leq \varphi_{ikjl}, \tau_{jl} \leq 1$ are parameters.

We use $\varphi = 0.5$, $\tau = 1$, and perform the dual descent method L successive times.

Overall Method. Our overall method combines both the subgradient optimization and dual descent method. We do this performing the subgradient method until termination and then switching over to the dual descent method. This procedure is repeated K times.

We implemented NATALIE in C++ using the LEMON graph library (http://lemon.cs.elte.hu). The successive shortest path algorithm for maximum weight bipartite matching was implemented and contributed to LEMON. Special care was taken to deal with the inherent numerical instability of floating point numbers. Our implementation supports both the GraphML and GML graph formats. Rather than using one big alignment graph, we store and use a different alignment graph for every local problem (LD$_\lambda^{ik}$). This proved to be a huge improvement in running times, especially when the global alignment graph is sparse. NATALIE is publicly available at http://planet-lisa.net.

## 4   Experimental Evaluation

From the STRING database v8.3 [24], we obtained PPI networks for the following six species: *C. elegans* (cel), *S. cerevisiae* (sce), *D. melanogaster* (dme), *R. norvegicus* (rno), *M. musculus* (mmu) and *H. sapiens* (hsa). We only considered interactions that were experimentally verified. Table 1 in the appendix shows the sizes of the networks. We performed, using the BLOSUM62 matrix, an all-against-all global sequence alignment on the protein sequences of all $\binom{6}{2} = 15$ pairs of networks. We used affine gap penalties with a gap-open penalty of 2 and a gap-extension penalty of 10. The first experiment in Section 4.1 compares the raw performance of IsoRank, GRAAL and NATALIE in terms of objective value. In Section 4.2 we evaluate the biological relevance of the alignments produced by the three methods. All experiments were conducted on a compute cluster with 2.26 GHz processors with 24 GB of RAM.

4.1   Edge-Correctness

The objective function used for scoring alignments in GRAAL counts the number of mapped edges. Such an objective function is easily expressible in our framework using $s(a) = |\{(v, w) \in E_1 \mid (a(v), a(w)) \in E_2\}|$ and can also be modeled using the IsoRank scoring function. In order to compare performance of the methods across instances, we normalize the scores by dividing by $\min(|E_1|, |E_2|)$. This measure is called the edge-correctness by Kuchaiev et al. [17].

Table 1. Characteristics of input networks considered in this study. The columns contain species identifier, number of nodes in the network, number of annotated nodes thereof, and number of interactions.

species	nodes	annotated	interactions
cel (c)	5,948	4,694	23,496
sce (s)	6,018	5,703	131,701
dme (d)	7,433	6,006	26,829
rno (r)	8,002	6,786	32,527
mmu (m)	9,109	8,060	38,414
hsa (h)	11,512	9,328	67,858
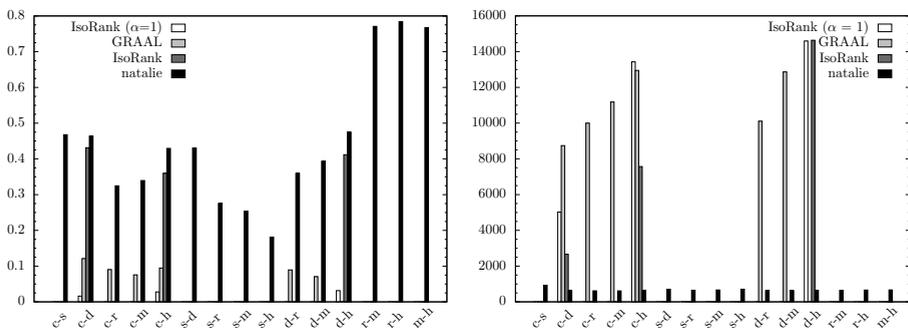
Fig. 2. Performance of the three different methods for the all-against-all species comparisons (15 alignment instances). Missing bars correspond to exceeded time/memory limits or software crashes. Left plot: Edge correctness, right plot: running times (sec.).

As mentioned in Section 3, our method benefits greatly from using a sparse alignment graph. To that end, we use the e-values obtained from the all-against-all sequence alignment to prohibit biologically unlikely matchings by only considering protein-pairs whose e-value is at most 100. Note that this only applies to NATALIE as both GRAAL and ISORANK consider the complete alignment graph. On each of the 15 instances, we ran GRAAL with 3 different random seeds and sampled the input parameter which balances the contribution of the graphlets with the node degrees uniformly within the allowed range of $[0, 1]$. As for ISORANK, when setting the parameter α—which controls to what extent topological similarity plays a role—to the desired value of 1, very poor results were obtained. Therefore we also sampled this parameter within its allowed range and re-evaluated the resulting alignments in terms of edge-correctness. NATALIE was run with a time limit of 10 minutes and $K = 3$, $L = 100$, $M = 10$, $N = 20$. For both GRAAL and ISORANK only the highest-scoring results were considered.

Figure 2 shows the results. ISORANK was only able to compute alignments for three out of the 15 instances. On the other instances ISORANK crashed, which may be due to the large size of the input networks. For GRAAL no alignments concerning *sce* could be computed, which is due to the large number of edges in the network on which the graphlet enumeration procedure choked: in 12 hours only for 3% of the nodes the graphlet degree vector was computed. As for the last

three instances, GRAAL crashed due to exceeding the memory limit inherent to 32-bit processes. Unfortunately no 64-bit executable was available. On the instances for which GRAAL could compute alignments, the performance—both in solution quality and running time—is very poor when compared to IsoRank and NATALIE. NATALIE outperforms IsoRank in both running time and solution quality.

4.2   GO Similarity

In order to measure the biological relevance of the obtained network alignments, we make use of the Gene Ontology (GO) [3]. For every node in each of the six networks we obtained a set of GO annotations (see Table 1 for the exact numbers). Each annotation set was extended to a multiset by including all ancestral GO terms for every annotation in the original set. Subsequently we employed a similarity measure that compares a pair of aligned nodes based on their GO annotations and also takes into account the relative frequency of each annotation [11]. Since the similarity measure assigns a score between 0 and 1 to every aligned node pair, the highest similarity score one can get for any alignment is the minimum number of annotated nodes in either of the networks. Therefore we can normalize the similarity scores by this quantity. Unlike the previous experiment, this time we considered the bitscores of the pairwise global sequence alignments. Similarly to IsoRank parameter α, we introduced a parameter $\beta \in [0, 1]$ such that the sequence part of the score has weight $(1 - \beta)$ and the topology part has weight β. For both IsoRank and NATALIE we sampled the weight parameters uniformly in the range $[0, 1]$ and showed the best result in Figure 3. There we can see that both NATALIE and IsoRank identify functionally coherent alignments.
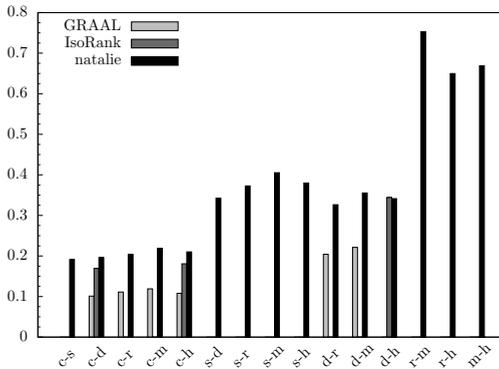

Fig. 3. Biological relevance of the alignments measured via GO similarity

5   Conclusion

Inspired by results for the closely related quadratic assignment problem (QAP), we have presented new algorithmic ideas in order to make a Lagrangian relaxation approach for global network alignment practically useful and competitive. In particular, we have generalized a dual descent method for the QAP.

We have found that combining this scheme with the traditional subgradient optimization method leads to fastest progress of upper and lower bounds.

Our implementation of the new method, NATALIE 2.0, works very well and fast when aligning biological networks, which we have shown in an extensive study on the alignment of cross-species PPI networks. We have compared NATALIE 2.0 to those state-of-the-art methods whose scoring schemes can be expressed as special cases of the scoring scheme we propose. Currently, these methods are ISORANK and GRAAL. Our experiments show that the Lagrangian relaxation approach is a very powerful method and that it currently outperforms the competitors in terms of quality of the results and running time.

Currently, all methods, including ours, approach the global network alignment problem heuristically, that is, the computed alignments are not guaranteed to be optimal solutions of the problem. While the other approaches are intrinsically heuristic—both ISORANK and GRAAL, for instance, approximate the neighborhood of a node and then match it with a similar node—the inexactness in our methods has two causes that we plan to address in future work: On the one hand, there may still be a gap between upper and lower bound of the Lagrangian relaxation approach after the last iteration. We can use these bounds, however, in a branch-and-bound approach that will compute provably optimal solutions. On the other hand, we currently do not consider the complete bipartite alignment graph and may therefore miss the optimal alignment. Here, we will investigate preprocessing strategies, in the spirit of [25], to safely sparsify the input bipartite graph without violating optimality conditions.

The independence of the local problems $(\mathrm{LD}_\lambda^{ik})$ allows for easy parallelization, which, when exploited would lead to an even faster method. Another improvement in running times might be achieved when considering more involved heuristics for computing the lower bound, such as local search. More functionally-coherent alignments can be obtained when considering a scoring function where node-to-node correspondences are not only scored via sequence similarity but also for instance via GO similarity. In certain cases, even negative weights for topological interactions might be desired in which case one needs to reconsider the assumption of entries of matrix W being positive.

References

1. Adams, W.P., Johnson, T.: Improved linear programming-based lower bounds for the quadratic assignment problem. DIMACS Series in Discrete Mathematics and Theoretical Computer Science (1994)
2. Alon, U.: Network motifs: theory and experimental approaches. Nat. Rev. Genet. 8(6), 450–461 (2007)

3. Ashburner, M., Ball, C.A., Blake, J.A., et al.: Gene ontology: tool for the unification of biology. Nat. Genet. 25 (2000)

4. Ay, F., Kellis, M., Kahveci, T.: SubMAP: Aligning Metabolic Pathways with Subnetwork Mappings.. J. Comput. Biol. 18(3), 219–235 (2011)

5. Caprara, A., Fischetti, M., Toth, P.: A heuristic method for the set cover problem. Oper. Res. 47, 730–743 (1999)

6. Edmonds, J.: Path, trees, and flowers. Canadian J. Math. 17, 449–467 (1965)

7. Edmonds, J., Karp, R.M.: Theoretical improvements in algorithmic efficiency for network flow problems. J.ACM 19, 248–264 (1972)

8. Flannick, J., Novak, A., Srinivasan, B.S., McAdams, H.H., Batzoglou, S.: Graemlin: general and robust alignment of multiple large interaction networks.. Genome Res. 16(9), 1169–1181 (2006)

9. Guignard, M.: Lagrangean relaxation. Top 11, 151–200 (2003)

10. Held, M., Karp, R.M.: The traveling-salesman problem and minimum spanning trees: Part II. Math. Program 1, 6–25 (1971)

11. Jaeger, S., Sers, C., Leser, U.: Combining modularity, conservation, and interactions of proteins significantly increases precision and coverage of protein function prediction. BMC Genomics 11(1), 717 (2010)

12. Kanehisa, M., Goto, S., Hattori, M., et al.: From genomics to chemical genomics: new developments in KEGG. Nucleic Acids Res. 34, 354–357 (2006)

13. Karp, R.M.: Reducibility among combinatorial problems. In: Miller, R.E., Thatcher, J.W. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum Press (1972)

14. Kelley, B.P., Sharan, R., Karp, R.M., et al.: Conserved pathways within bacteria and yeast as revealed by global protein network alignment. P. Natl. Acad. Sci. USA 100(20), 11394–11399 (2003)

15. Klau, G.W.: A new graph-based method for pairwise global network alignment.. BMC Bioinform. 10(suppl.1), S59 (2009)

16. Koyutürk, M., Kim, Y., Topkara, U., et al.: Pairwise alignment of protein interaction networks.. J. Comput. Biol. 13(2), 182–199 (2006)

17. Kuchaiev, O., Milenkovic, T., Memisevic, V., Hayes, W., Przulj, N.: Topological network alignment uncovers biological function and phylogeny.. J. R. Soc. Interface 7(50), 1341–1354 (2010)

18. Kuhn, H.W.: The Hungarian method for the assignment problem. Nav. Res. Logist. Q 2(1-2), 83–97 (1955)

19. Lawler, E.L.: The quadratic assignment problem. Manage. Sci. 9(4), 586–599 (1963)

20. Munkres, J.: Algorithms for the assignment and transportation problems. SIAM J. Appl. Math. 5, 32–38 (1957)

21. Sharan, R., Ideker, T.: Modeling cellular machinery through biological network comparison.. Nat. Biotechnol. 24(4), 427–433 (2006)

22. Sharan, R., Ideker, T., Kelley, B., Shamir, R., Karp, R.M.: Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. J. Comput. Biol. 12(6), 835–846 (2005)

23. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks with application to functional orthology detection.. P. Natl. Acad. Sci. USA 105(35), 12763–12768 (2008)

24. Szklarczyk, D., Franceschini, A., Kuhn, M., et al.: The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. Nucleic Acids Res. 39, 561–568 (2010)

25. Wohlers, I., Andonov, R., Klau, G.W.: Algorithm engineering for optimal alignment of protein structure distance matrices. Optim. Lett. (2011)