BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–3

Supplementary Material:
L-GRAAL: Lagrangian Graphlet-based Network Aligner

Noël Malod-Dognin* and Nataša Pržulj
Department of Computing, Imperial College London, United Kingdom

1 METHOD

1.1 Solving the relaxed problem LR(λ)

In our approach, LR(λ) is solved by a double bipartite matching algorithm, where *Local* bipartite matchings are used to find, for each possible node mapping $i \leftrightarrow k$, the best sets of edge mappings having $i \leftrightarrow k$ as tail-node mapping. Then a *Global* bipartite matching finds the best set of node mappings according to the previously found sets of edge mappings.

Specifically, for a given node mapping $i \leftrightarrow k$, a *local* problem consists of finding a set of edge mappings $(i, j) \leftrightarrow (k, l)$, $k < l$, such that the corresponding head-nodes mappings $j \leftrightarrow l$ form a 1-to-1 matching and such that the contribution of these edge mappings to LR(λ)'s objective function, denoted by $Local(ik)$, is maximum. We recall that the contribution of an edge mapping $(i, j) \leftrightarrow (k, l)$ into LR(λ)'s objective function is $e^{\lambda}(i, j, k, l)$ (from (12) in the main text). This corresponds to the following IP program:

$$Local(ik) = \max_{y} \sum_{j,l} e^{\lambda}_{i,j,k,l} \times y_{ijkl},$$

subject to constraints (10) and (11) from the main text. This problem can be rephrased as a maximum weighted bipartite matching problem between the neighbours of i (i.e., all possible j) and the neighbours of k (i.e., all possible l, $k < l$), where the weight of mapping j to k is e^{λ}_{ijkl}. Denoting the maximum degree of a node in N_1 and N_2 by d, this matching problem can be solved in $O(d^3)$ time using, for example, the Hungarian algorithm or the successive shortest paths approach.

The *global problem* consists of finding a set of node mappings and the corresponding edge mappings that have maximum contribution to the objective function of LR(λ). The contribution of a node mapping $i \leftrightarrow k$ is $n^{\lambda}(i, k)$ (see (12) in the main document), and the contribution of the edge mappings connected to $i \leftrightarrow k$ is $Local(ik)$ (as previously found by solving the local problem). This corresponds to the following IP program:

$$Global = \max_{x} \sum_{i,k} (n^{\lambda}(i, k) + Local(ik)) \times x_{ik},$$

subject to (4), (5), and (7) from the main text. Again, this problem can be rephrased as a maximum weighted bipartite matching problem between the nodes in V_1 and the nodes in V_2, where the weight of mapping node i to node k is $n^{\lambda}(i, k) + Local(ik)$. This can be computed in $O(|V|^3)$ time. Thus, solving LR(λ) is done in $O(|V|^3 + |V|^2 d^3)$ time.

1.2 Solving the Lagrangian dual problem

The main relation between IP and LR(λ) is that LR(λ) is an upper bound of IP for any values of λ, i.e., $IP \leq LR(\lambda), \quad \forall \lambda \in \mathcal{R}^{+,0}$. Also, LR($\lambda$)'s solution, $(\vec{x}, \vec{y})$, can be used to create a lower bound on IP, denoted by $lb(\lambda)$, by simply selecting the edge mappings, $\vec{y'}$, that are adjacent to the selected node mappings $\vec{x}$. In order to improve the bounds, or eventually to solve IP, we need to solve its Lagrangian dual problem (LD), which is the minimization of LR(λ) over λ: $LD = \min_{\lambda} LR(\lambda)$.

Many methods have been proposed so far for solving Lagrangian dual problem (Guignard, 2003). Here, we choose the sub-gradient descent (Held *et al.*, 1974) because of our large number of Lagrangian multipliers. The sub-gradient descent is an iterative method which generate a sequence of Lagrangian multiplier vectors $\lambda(0), \lambda(1), \lambda(2), \ldots$, starting from $\lambda(0) = 0$, as follows:

$$\lambda^{ijl}_{E_1}(t+1) = \max(0, \lambda^{row}_{ijl}(t) - \frac{\alpha \times (UB - LB)}{||g(\lambda(t))||^2} g(\lambda^{ijl}_{E_1}(t))),$$

$$\lambda^{kjl}_{E_2}(t+1) = \max(0, \lambda^{col}_{kjl}(t) - \frac{\alpha \times (UB - LB)}{||g(\lambda(t))||^2} g(\lambda^{kjl}_{E_2}(t))),$$

where UB is the smallest upper bound on IP found so far (i.e., the smallest value of LR(λ)), LB is the largest lower bound on IP found so far (i.e., the largest value of $lb(\lambda)$), $g(\lambda^{ijl}_{E_1}(t)) = x_{jl} - \sum_{k,k<l} y_{ijkl}$ is the sub-gradient vector component associated to the corresponding relaxed constraint (8) from the main text, $g(\lambda^{kjl}_{E_2}(t)) = x_{jl} - \sum_{i} y_{ijkl}$ is the sub-gradient vector component associated to the corresponding relaxed constraint (9) from the main text, $||g(\lambda(t))||$ is the number of non-zero sub-gradient vector components, and α is the step size. In our implementation, the step-size is initialised with $\alpha = 1$, but is divided by 1.3 every five consecutive iterations that do not improve the bounds on IP and is similarly multiplied by 1.3 every five consecutive iterations that improve the bounds.

A solution of LD is an optimal solution of IP if the corresponding sub-gradient vector components are all equal to 0, but the process can be stopped earlier if $UB = LB$.

1.3 Extending seed alignments

As presented in the main document, the Lagrangian relaxation-based solver is used to generate a suite of seed alignments, which are optimized over the *selected node mappings* having protein similarities higher than a given threshold (see the main text). Supplementary Algorithm 1 presents the greedy heuristic that we use to extend each seed alignment, f, by using all possible node

*to whom correspondence should be addressed

© Oxford University Press 2005.

mappings, i.e., without being restricted to these selected node mappings.

Supplementary Algorithm 1. Between two networks, N_1 and N_2, the *Extend* function heuristically refines a seed alignment, f, so that its score, $S(f)$, measured by using L-GRAAL's scoring function (see eq. 2 in the main document), is maximized. Note that $f(u) = \emptyset$ means that $u \in V_1$ is not aligned yet, and $f^{-1}(v) = \emptyset$ means that $v \in V_2$ is not aligned yet.

```
Extend( $N_1 = (V_1, E_1)$, $N_2 = (V_2, E_2)$, $f$)
//Step 1: Remove non-contributing node-mappings $u \leftrightarrow v$.
for $u \leftrightarrow v \in f$ do
    if $S(f \backslash \{u \leftrightarrow v)\}) \geq S(f)$ then
        $f \leftarrow f \backslash \{u \leftrightarrow v\}$ (i.e., set $f(u) = \emptyset$)
//Step 2: Maximally extend $f$
for $u \in V_1$ such that $f(u) = \emptyset$ do
    Find $v \in V_2$ s.t. $f^{-1}(v) = \emptyset$ and $v$ =argmax $S(f \bigcup \{u \leftrightarrow v\})$
    $f \leftarrow f \bigcup \{u \leftrightarrow v\}$ (i.e., set $f(u) = v$)
//Step 3: Greedy local search
for $u \in V_1$ do
    $f' \leftarrow f \backslash \{u \leftrightarrow v\}$
    Find $v' \in V_2$ s.t. $f'^{-1}(v') = \emptyset$ and $v'$ =argmax $S(f' \bigcup \{u \leftrightarrow v'\})$
    if $S(f) < S(f' \bigcup \{u \leftrightarrow v'\})$ then
        $f \leftarrow f' \bigcup \{u \leftrightarrow v'\}$
Return $f$
```

Step one, which removes node mappings that do not contribute to the score of the alignment, is needed because such node mappings may be included in the seed alignments (the repaired solutions from the Lagrangian relaxation-based solver) when we use topological similarity only: this is because when $\alpha = 0$, the node mappings do not contribute directly to the objective function (their weights are all zero because $\alpha = 0$), but the edges adjacent to such nodes contribute to the relaxed solution. Then, because the edge mappings chosen to be in the relaxed solution might be infeasible (when only one of their two end-node mapping are in the alignment) such infeasible edge mapping are removed when creating the repaired solution. If the repairing process removes all the edge mappings that are adjacent to the node mapping, this node mapping does not contribute to the alignment's score any more.

1.4 Differences between L-GRAAL and NATALIE

Since L-GRAAL and NATALIE both use integer programming and Lagrangian relaxation to optimize their objective functions, we briefly explain here how the two methods differ.

The two approaches start with the same modelling of node and edge mappings: node mappings $i \leftrightarrow k$ are represented with boolean variables x_{ik}, edge mappings $(i, j) \leftrightarrow (k, l)$ with boolean variables y_{ijkl}, and the relationships between an edge mapping and its two end-node mappings are first represented by the two constraints:

$$x_{ik} \leq y_{ikjl},$$
$$x_{jl} \leq y_{ikjl}.$$

Then, to apply different relaxation schemes, the two methods alter the above model in different ways. NATALIE applies so-called cost split technique: variables representing the edge mappings are duplicated (mapping edge (i, j) with edge (k, l) is represented by two variables, y_{ijkl} and z_{ijkl}), each copy being bound to a different end-node mapping, and the validity of the alignment then being guaranteed by the equality between y and z variables:

$$x_{ik} \leq y_{ikjl},$$
$$x_{jl} \leq z_{ikjl},$$
$$y_{ijkl} = z_{ijkl}.$$

Natalie relaxes and tries to repair the edge equalities, using subgradient and dual-ascent techniques. In the case of a dense network, the number of relaxed constraints in NATALIE's scheme is upper-bounded by n^4 (where n is the number of nodes in the network).

In L-GRAAL, we first rewrite constraints $x_{jl} \leq y_{ikjl}$ to reduce their numbers, and then relax them. In our approach, the number of relaxed constraints is upper-bounded by n^3. Since the efficiency of dual solvers is strongly dependant on the number of relaxed constraints, our relaxation scheme is favourable. Of lesser importance, NATALIE doubles the number of variables representing edge-mapping, which can be an issue for general purpose solvers.

Finally, NATALIE only optimizes the alignments over the sequence similar node mappings, but never tries to extend the alignments using non-sequence related proteins. This means in particular that NATALIE will never uncover functionally similar proteins that are not sequence related.

1.5 Statistical significance of Edge Correctness

When aligning two networks $N_1 = (V_1, E_1)$ and $N_2 = (V_2, E_2)$, under the standard model of sampling without replacement, the probability p of obtaining at least k common edges by chance is the tail of the hyper-geometric distribution:

$$p(k) = \sum_{i=k}^{m_2} \frac{\binom{m_2}{i}\binom{M-m_2}{m_1-i}}{\binom{M}{m_1}},$$

where $m_1 = |E_1|$, $m_2 = |E_2|$, and $M = |V_2| \times (|V_2| - 1)/2$ is the number of node pairs in N_2 (Pržulj *et al.*, 2004).

2 SUPPLEMENTARY RESULTS

2.1 Additional semantic similarity results

We detail here the semantic similarity results that are obtained at the interaction level.

As presented in supplementary Fig. 2, HUBALIGN, L-GRAAL, and SPINAL best map together interactions that are involved in similar biological processes, in similar molecular functions, and that are localised in similar cellular regions. When using GO-BP, the average semantic similarity of the interaction mapping is 1.09

for HUBALIGN, 1.08 for L-GRAAL and 1.04 for SPINAL. When using GO-BP, the average semantic similarity of the interaction mapping is 0.42 for HUBALIGN, 0.38 for L-GRAAL, and 0.37 for SPINAL. Finally, when using GO-BP, the average semantic similarity of the interaction mapping is 0.64 for HUBALIGN, 0.60 for L-GRAAL, and 0.50 for SPINAL.

2.2 Balancing sequence and topological information

In the main document, we comment on how the topological and biological quality of the alignments change when α, the parameter that balances topological and sequence information, varies in [0,1]. These changes are presented in supplementary Figure 5. Note that these alignments are obtained when using $-\log$ of blast's evalues as sequence similarity.

2.3 Predicting protein interactions

In the main document, we present the number of protein interactions that can be predicted from L-GRAAL's alignment of yeast and human PPI networks, when using a sequence identity threshold between the mapped proteins of 70%, threshold for which the mapped proteins are expected to share the same functions. Supplementary Figure 4 extends these results for more lenient sequence identity thresholds. In particular, we can predict 24,147 protein interactions with a sequence identity threshold of 30%, for which 90% of the mapped proteins are expected to be homologous (Rost, 1999). Among these 24,147 predicted interactions, 2,273 (10.6%) are also predicted in the Interologous Interaction Database (I2D ver. 2.3)(Brown and Jurisica, 2007), which validates our approach.

3 COMPARISON OF NETWORK ALIGNERS ON THE NAPA BENCHMARK

Since no GO term annotation is available on the NAPA benchmark, we cannot apply semantic similarity-based measures on this dataset. However, the equivalence classes of all nodes are known. Thus, we evaluate the biological quality of the alignments by their node correctness (NC), which is the percentage of the nodes from the smaller network that are mapped with nodes from the same equivalence classes.

GHOST, SPINAL and L-GRAAL have the largest edge-correctness between the aligned networks, with edge-correctness of 76.3% for GHOST, 74.4% for SPINAL and 72.7% for L-GRAAL (see the left panel of supplementary Fig. 5). NATALIE, L-GRAAL and GHOST best map sparse regions with sparse regions and dense regions with dense regions, with symmetric sub-structures score of 64.0% for NATALIE, 61.4% for L-GRAAL and 61.3% for GHOST (see the middle panel of supplementary Fig. 5). Finally, NATALIE, GHOST and SPINAL uncover the largest connected common sub-networks, L-GRAAL being the 4^{th} with LCC of 74.6%, NATALIE, GHOST and SPINAL leading at 79.3%, 77.9%, and 76.1%, respectively (see the right panel of Supplementary Fig. 5).

As already observed by Clark and Kalita (2014), the behaviour of network aligners on the NAPA benchmark is different than on real PPI networks. Although L-GRAAL still performs well on the NAPA Benchmark, HUBALIGN, which performs very well on real PPI networks obtains poor results on the NAPA benchmark. Methods such as NATALIE and GHOST achieve much better performances on the NAPA benchmark than on real PPI networks.

REFERENCES

Brown, K. R. and Jurisica, I. (2007). Unequal evolutionary conservation of human protein interactions in interologous networks. *Genome biology*, **8**(5), R95.

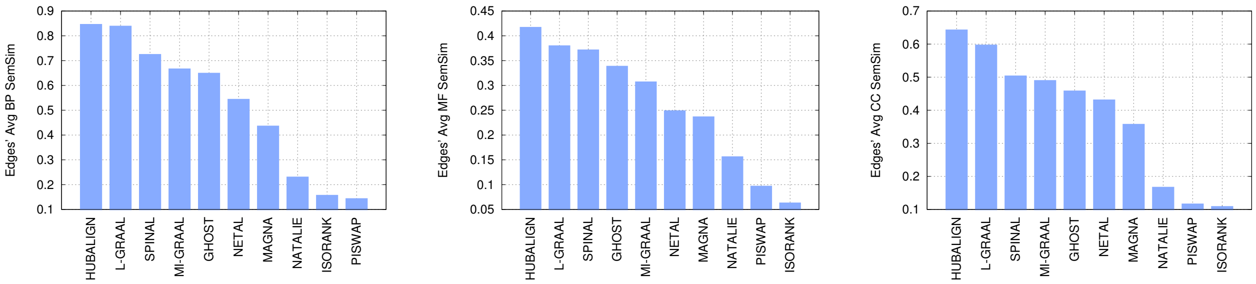
Clark, C. and Kalita, J. (2014). A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics*, page btu307.

Guignard, M. (2003). Lagrangean relaxation. *TOP*, **11**(2), 151–200.

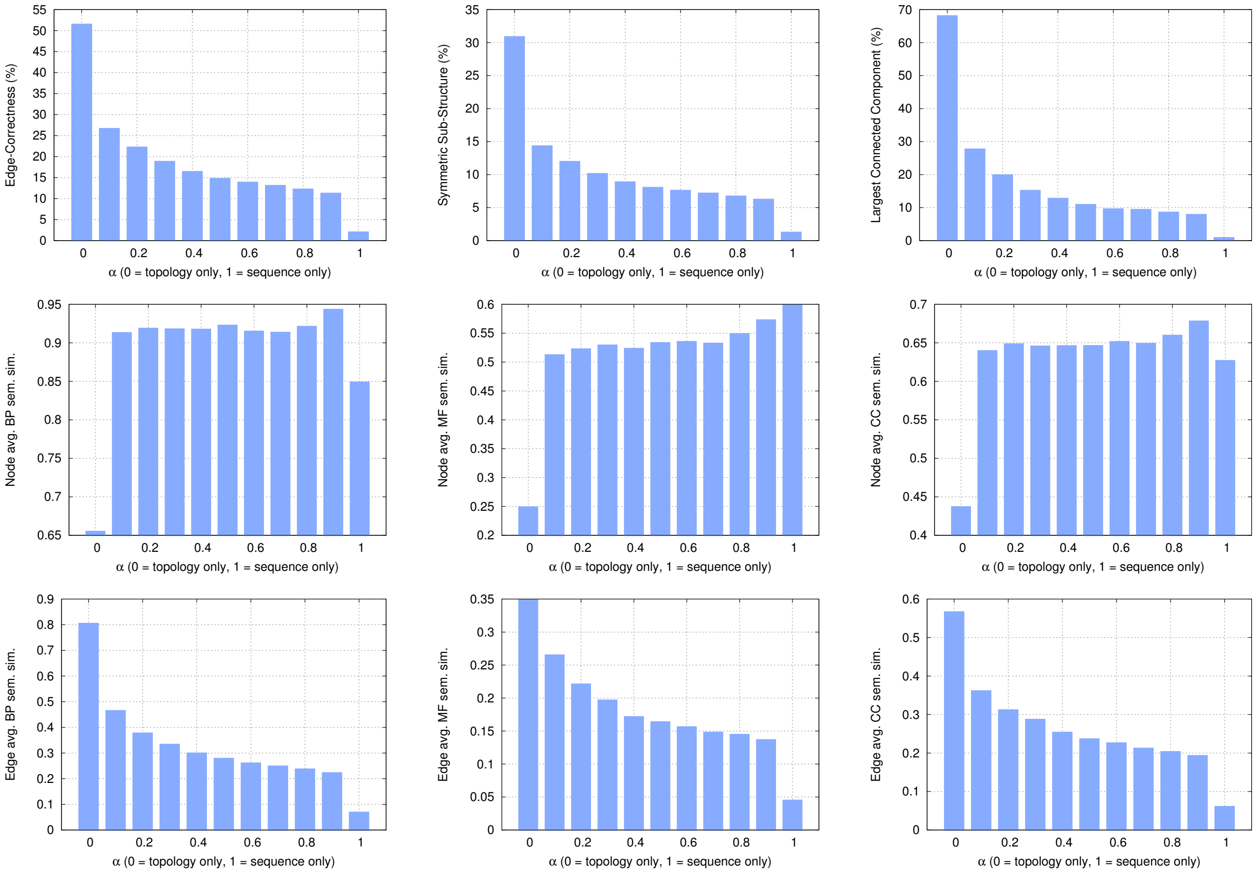
Held, M., Wolfe, P., and Crowder, H. (1974). Validation of subgradient optimization. *Mathematical Programming*, **6**(1), 62–88.

Pržulj, N., Corneil, D., and Jurisica, I. (2004). Modeling interactome: Scale-free or geometric? *Bioinformatics*, **20**, 3508–3515.

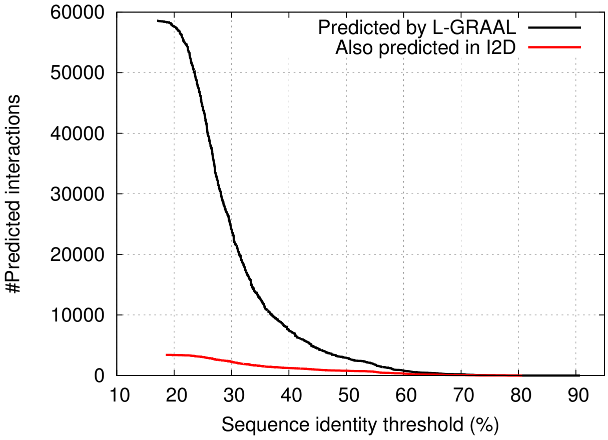
Rost, B. (1999). Twilight zone of protein sequence alignments. *Protein Engineering*, **12**(2), 85–94.



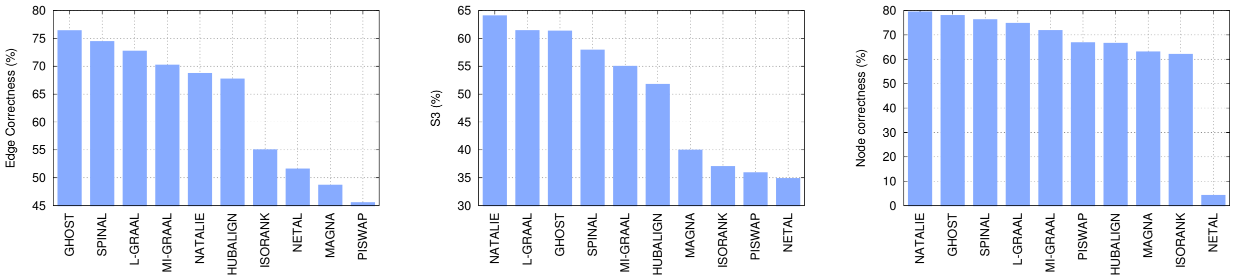
Supplementary Figure 2. Network aligners (x-axis) are compared according to the average semantic similarity of their interaction mappings (y-axis), when using GO-BP (left panel), GO-MF (middle panel), and GO-CC (right panel).



Supplementary Figure 3. Effect of α (x-axis) on the quality of L-GRAAL's alignments, as indicated by the average value of the different scoring schemes (y-axis). **Top:** Topological quality of the alignments, as measured by edge-correctness (EC, left panel), by symmetric sub-structure score (S3, middle panel), and by largest connected component (LCC, right panel). **Middle:** Biological quality of the protein mappings, as measured by the average semantic similarity using GO-BP (left panel), GO-MF (middle panel), and GO-CC (right panel) of the aligned proteins. **Bottom:** Biological quality of the interaction mappings, as measured by the average semantic similarity using GO-BP (left panel), GO-MF (middle panel), and GO-CC (right panel) of the aligned interactions.



Supplementary Figure 4. The number of predicted interactions (y-axis) as a function of the minimum sequence identity between the aligned yeast-human proteins (x-axis). We add in red the number of predicted interactions that are also predicted in I2D database.



Supplementary Figure 5. Network aligners (x-axis) are compared according to the average of the best scores (y-axis) that they achieve on the 30 pairs of PPI networks from the NAPA benchmark. **Left:** when the topological quality of the alignments is measured by edge-correctness (EC). **Middle:** when the topological quality of the alignments is measured by symmetric sub-structure score (S3). **Right:** when the biological quality of the alignments is measured by the node correctness (NC).