1 SUPPLEMENTARY INFORMATION

Algorithm 1 Align$(G_1, G_2, \alpha, \beta, \lambda)$

Require: $|V_1| \leq |V_2|$
$S \leftarrow$ SimilarityScore$(G_1, G_2, \alpha, \beta)$
$P \leftarrow 0$
$C_1 \leftarrow 0$
$L \leftarrow \emptyset$
for all $i \in V_1$ **do**
  for all $j \in V_2$ **do**
    $I(i,j) = \frac{\min\left\{\sum_{i' \in N(i)} \frac{1}{|N(i')|}, \sum_{j' \in N(j)} \frac{1}{|N(j')|}\right\}}{\max_{k \in V_1 \cup V_2}\{|N(k)|\}}$
    $A(i,j) = \lambda S(i,j) + (1-\lambda) I(i,j)$
  end for
end for
for all $n \in V_1$ **do**
  $(i', j') \leftarrow \text{argmax}_{(i',j') \in \{(i',j') | (i',j') \cup L = \emptyset\}} \{A(i', j')\}$
  Al$(i) \leftarrow j$
  $L \leftarrow L \cup \{i, j\}$
  for all $i' \in N(i)$ **do**
    $C_1(i') \leftarrow C_1(i') + \frac{1}{|N(i)|}$
  end for
  for all $j' \in N(j)$ **do**
    $C_2(j') \leftarrow C_2(j') + \frac{1}{|N(j)|}$
  end for
  for all $i' \in N(i)$ **do**
    for all $j' \in N(j)$ **do**
      $D(i', j') = D(i', j') + 1$
    end for
  end for
  $Q \leftarrow \{(i', j) | i' \in N(i) - L, j' \in N(j) - L'\}$
  for all $(i', j') \notin Q$ **do**
    $I(i,j) = \frac{\min\left\{\sum_{i' \in N(i)} \frac{1}{|N(i')|} - C_1(i), \sum_{j' \in N(j)} \frac{1}{|N(j')|} - C_2(j)\right\}}{\max_{k \in V_1 \cup V_2}\{|N(k)|\}} + D(i,j)$
    $A(i,j) = \lambda S(i,j) + (1-\lambda) I(i,j)$
  end for
end for
return Al

Algorithm 2 TopologicalScore(G_1, G_2)

$T \leftarrow 1$
for all $t \leftarrow 1$ **to** k **do**
  for all $i \in V_1$ **do**
    for all $j \in V_2$ **do**
      $T'(i,j) \leftarrow$ ComputeScore(G_1, G_2, i, j, T)
    end for
  end for
  $T \leftarrow T'$
end for
return T

Algorithm 3 ComputeScore(G_1, G_2, i, j, M)

$A \leftarrow \emptyset$
$s \leftarrow 0$
for all $i' \in N(i)$ **do**
  $i'' \leftarrow \text{argmax}_{j' \in N(i') - A}\{M(i', j')\}$
  $A \leftarrow A \cup \{j''\}$
  $s \leftarrow s + M(i', j'')$
end for
$s \leftarrow \frac{s}{\max\{N(i), N(j)\}}$
return s

Algorithm 4 BiologicalScore(G_1, G_2, β)

$C \leftarrow$ Biological Similarity
$B \leftarrow C$
for all $t \leftarrow 1$ **to** k **do**
  for all $i \in V_1$ **do**
    for all $j \in V_2$ **do**
      $B'(i,j) \leftarrow \beta C(i,j) + (1 - \beta)$ComputeScore$(G_1, G_2, i, j, B)$
    end for
  end for
  $B \leftarrow B'$
end for
return T

Algorithm 5 SimilarityScore$(G_1, G_2, \alpha, \beta)$

$T \leftarrow$ TopologicalScore(G_1, G_2)
$B \leftarrow$ BiologicalScore(G_1, G_2, β)
for all $i \in V_1$ **do**
  for all $j \in V_2$ **do**
    $S(i,j) \leftarrow \alpha T(i,j) + (1-\alpha) B(i,j)$
  end for
end for
return S

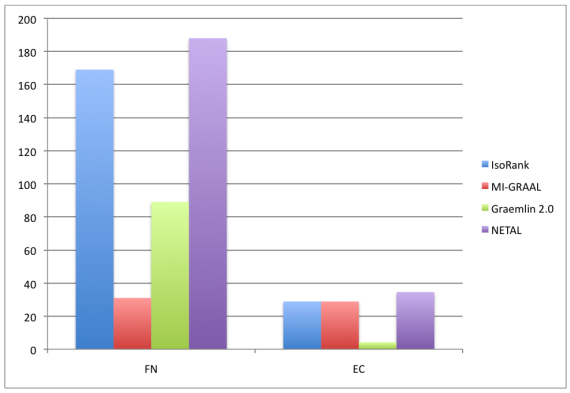

Fig. 1. FN and EC in the yeast-fly alignment, NETAL in comparison with Graemlin, MI-GRAAL and IsoRank

1.1 Alignment results for EN and FN for tools Graemlin, MI-Graal, IsoRank, NetAl

Graemlin 2.0 requires a training set to learn the parameters for the alignment. Thus comparison with Graemlin can only be made

using the data set in the package, which provides PPINs of fly and yeast (denoted as fly2 and yeast2) Phan *et al*., 2012. Therefore, to compare NETAL with Graemlin 2.0, we used the data of yeast and fly PPI network provided in the Graemlin 2.0 package. The yeast PPI network has 4,905 proteins and 16,999 interactions, while the fly network has 7,056 proteins and 21,822 interactions. Graemlin 2.0 was applied to the input networks using the trained parameters (from the provided dataset) and the default options. As you see in Figure 1, our algorithm outperform the others.

1.2 Time complexity

Let $n = \max\{|V_1|, |V_2|\}$ and $m = \max\{|E_1||E_2|\}$ and . Procedure *Compute Score* requires $O(|N(i)||N(j)|)$ number of operations for two nodes i and j. Therefore, calculating topological score matrix and biological score matrix can be done in:

$$2k \sum_{i \in V_1} \sum_{j \in V_2} |N(i)||N(j)| = 8k|E_1||E_2| = O(m^2) \qquad (1)$$

Clearly, having both topology and biological score matrices, computing similarity score matrix takes $O(n^2)$ time. Additionally, based on formula **??** and **??**, computing interaction score matrix and alignment score matrix require $O(n^2)$ time. Constructing priority queue takes $O(n^2 \log n)$. So, complexity of the phase of computing all mentioned matrices is $O(n^2 \log n + m^2)$. In the second phase, selecting the best pair node from the priority queue takes $O(1)$ time. Updating C_1, C_2 and D matrices requires $O(m)$, $O(m)$ and $O(mn)$, respectively. After aligning each pair nodes i and j, the number of elements of matrices I and A that are needed to be updated is equal to the number of pair nodes that one of them is adjacent to i or j i.e. $|N(i)||V_2|+|N(j)||V_1|-|N(i)||N(j)|$. Since priority queue has $|V_1||V_2|$ members, updating each of its values takes $O(\log(|V_1||V_2|))$ time. Thus, the time needed for updating the priority queues is:

$$\sum_{i \in V_1, j \in N(i)} \left(|N(i)||V_2| + |N(j)||V_1| - |N(i)||N(j)| \log(|V_1||V_2|)\right) = O(mn \log n) \qquad (2)$$

Therefore, the time complexity of NETAL is equal to $O(n^2 \log(n) + m^2 + mn \log(n))$ that is much better than other existing algorithms.

REFERENCES

Phan HTT *et al.* (2012) PINALOG: a novel approach to align protein interaction networksimplications for complex detection and function prediction. *Bioinformatics 2012*,28:1239-1245.