CoreAlign: Core-based Global Alignment for Protein-Protein Interaction Networks

Ahmed El-Sawy
Computer Science Department
Faculty of Computers & Artificial Intelligence
Benha University
Benha, Egypt

Mahmoud Mousa
Computer Science Department
Faculty of Computers & Artificial Intelligence
Benha University
Benha, Egypt

Ahmed Hassan
Computer Science Department
Faculty of Computers & Artificial Intelligence
Benha University
Benha, Egypt

Sammer Kamal
Computer Science Department
Faculty of Computers & Artificial Intelligence
Benha University
Benha, Egypt

ABSTRACT

Biological network alignment aims to find similar functional and topological regions to guide the transfer of biological knowledge of cellular functioning from known, well-studied species to unknown ones. The proposed aligner (CoreAlign) relays on the structural of the Protein-Protein Interactions (PPI) network by using network decomposition of what is called shells or internal network cores. The proposed aligner searches the space of each core to build the Alignment. CoreAlign has been compared with many aligners and it has competitive results among these aligners in either topological or biological measures.

Keywords

Protein-protein interactions, PPI, network alignment, protein function, network decomposition.

1. INTRODUCTION

Nowadays, the real-world phenomenon of binding proteins to each other plays a powerful roll in configuring how cells work together to perform various functions. This progress helps in disease and drug discovery and gives a new insight into evolutionary relationships of different species. The huge number of these binding events can be modeled as Protein-Protein Interaction (PPI) networks. The PPI networks can be represented as unweighted and undirected graphs $G(V, E)$ to present the much-interconnected nature of the biological processes. For each PPI network, proteins are modeled as the vertices (V) of the graph and interactions and relationships between different proteins in the network are represented by graph edges (E).

The prediction of protein functions in unknown species becomes a challenging problem since some of them cannot be studied easily by experiments besides ethical constraints especially when dealing with human diseases. The main goal of PPI network alignment is to try to solve this problem. PPI network alignment tries to help in predicting unknown functions from known protein functions of the well-studied species that became a model like yeast, worm or fly(Faisal, Meng, Crawford, & Milenković, 2015). It also facilitates discovering evolutionary and functionally conserved complexes and pathways.

PPI network alignment can be divided into local aligners or global aligners. Local aligners such as PathBlast (Kelley et al., 2004), NetworkBlast (Sharan et al., 2005), MaWISh(Koyutürk et al., 2006), and AlignNemo (Ciriello, Mina, Guzzi, Cannataro, & Guerra, 2012). The purpose of these aligners is to detect small, multiple and dense subnetworks with a similar structure corresponding to a motif protein complexes or a pathway between input networks. These aligners are many-to-many node mapping since a node can be mapped to many different nodes.

Global network alignment (GNA) is concerned with finding a mapping between input networks that maximizes the overall similarity. Many of global aligners called pairwise aligners are one-to-one node mapping such as IsoRank (Singh, Xu, & Berger, 2007), GRAAL family aligners: {GRAAL(Kuchaiev, Milenković, Memišević, Hayes, & Pržulj, 2010), C-GRALL(Memišević & Pržulj, 2012), H-GRAAL (Milenković, Ng, Hayes, & Pržulj, 2010), MI-GRAAL (Kuchaiev & Pržulj, 2011), and L-GRAAL(Malod-Dognin & Pržulj, 2015)}, NETAL(Neyshabur, Khadem, Hashemifar, & Arab, 2013), HubAlign(Hashemifar & Xu, 2014), ModuleAlign(Hashemifar, Ma, Naveed, Canzar, & Xu, 2016), MAGNA(Saraph & Milenković, 2014) and its extended framework MAGNA ++(Vijayan, Saraph, & Milenković, 2015).

Many network aligners combine network topology with external biological information such as the similarities of protein sequences such as BLAST bit scores or E-values in the cost function to improve their results by taking the advantage of both(Clark & Kalita, 2014). Using Topology only may mislead the solution, because the current datasets are noisy and incomplete. This problem makes the actual complexes may appear disconnected. So combing biological information can be more helpful especially when dealing with closely related species (Clark & Kalita, 2014).

Global network aligners can be categorized into two different approaches. The first approach such as the GRAAL family, NETAL, HubAlign, and ModuleAlign can be called traditional methods. These aligners are two-step methods because they estimate pairwise node similarity between the input networks using a node cost function, then an alignment strategy is used to find an alignment with high score taking into account the overall aligned node similarity. The second approach such as MAGNA and MAGNA ++ are called recent aligners they are search-based aligners since they focus on maximizing the actual alignment quality.

Some of the GRAAL family relays on the usage of the graphlets (section 2.3). GRAAL(Kuchaiev et al., 2010) used a greedily seed-and-extend strategy. H-GRAAL(Milenković et al., 2010) used the Hungarian algorithm to solve the maximum weight bipartite matching problem(West, 2001), while MI-GRAAL and C-GRAAL use the concept of shared network neighbors in their cost function. Another GNA method is NETAL(Neyshabur et al., 2013) which is a traditional pairwise aligner that iteratively updates its node similarity scores. MAGNA(Saraph & Milenković, 2014) and MAGNA ++ (Vijayan et al., 2015) use a genetic algorithm to directly optimize node or edge conservation during the construction of alignment.

The proposed algorithm (CoreAlign) is a pairwise global aligner that uses the concept of k-cores or k-shells network decomposition on the smallest PPI network. The search space of CoreAlign is the nodes of each shell that are ordered by betweenness centrality. The aligner then builds a similarity function between the given two networks using the graphlet degree vector (GDV) to construct the final alignment.

The rest of this paper is organized as the following: section 2 gives a related background such as network decomposition, betweenness centrality, and graphlets and orbits. Section 3 presents the proposed algorithm. Section 4 contains the results and discussion. Section 5 is the conclusion.

2. RELATED BACKGROUND

When visualizing biological networks, the existence of some proteins with high connectivity called hubs is observed. Also, there are some proteins with low connectivity called peripherals, and bottleneck proteins that can have any type of connectivity can exist in any PPI network. The hubs tend to be connected to each other. The bottlenecks have a high value of betweenness centrality (the number of shortest paths passing through a node). bottleneck proteins gain their importance by containing some of the topology and functionality of a biological network(Hashemifar & Xu, 2014).

2.1 Network Decomposition

K-core or k-shell decomposition is a common method that has been used to analyze social networks (Liu, Ren, Guo, & Chen, 2014) and to investigate proteins(Janjić & Pržulj, 2012). K-core decomposition of a network is a way to have an insight into its central structure by dividing a network into some shells or cores. The first core represents the whole network while the k-most core is a subgraph of the main network that consists of proteins that are intensively interconnected to each other. This core has only proteins with connections equal to or greater than k. The proteins which included in k-core usually tend to be hubs, but as illustrated in figure 1, it is not necessary to contain all the hubs or even the biggest hubs of the network in it. For example, there is a peripheral hub in the first core with a degree equals five but it is not as important as the hubs that exist in the fourth core with degrees equal four or five.

As illustrated in figure 1, all nodes in the given network have been assigned a core number. The first core represents the whole network. This core contains eight nodes in blue not concluded in the next core, which later will be referred to as the first core nodes. These nodes contain one peripheral hub with degree four and seven nodes with degree one. The second core has five nodes with degrees two and three. The third core has four nodes in black with degrees four and five. Finally seven nodes with degrees four, five and six exist in the fourth core, which is the k-most core.

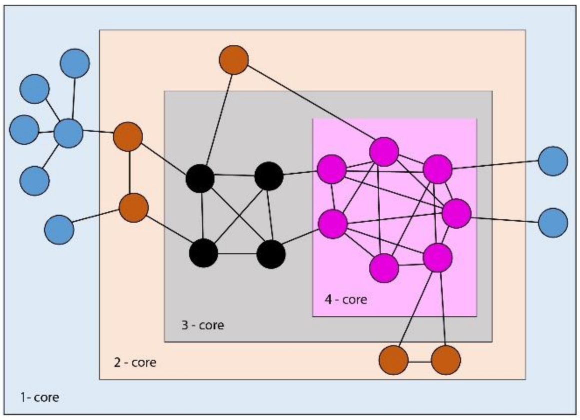


Fig 1: A four-level decomposition of a network. The 4-core is the most core of the given network.

Algorithm 1 shows the steps to extract the cores of a network and the nodes belonging to each core.

Algorithm 1: K-core network decomposition

Input: network G.

Output: all the cores of network G.

```
Step1: start with d=1 and C=1.
Step2: repeat until no proteins remained in the network.
    a.  Remove proteins with degree ≤ d.
    b.  Do the following if there are any removed proteins.
        i.   Omit the connections of excluded nodes.
        ii.  Assign them a core = C.
    c.  C=C+1.
    d.  d=d+1
```

Figure 2 shows a real network RNorvegicus (RN) which contains 1657 proteins and 2330 interactions. After Applying algorithm 1 (K-core network decomposition), the RN network can be divided into five cores. The 5-most core of this network has only 19 nodes and their inner interactions as shown in figure 3. The few interactions among the most core proteins link many of the hubs of the network, so it can represent the important nodes in the network.

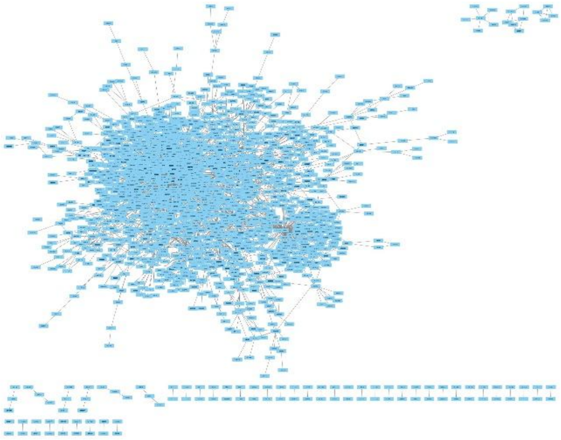


Fig 2: The RNorvegicus (RN) PPI Network.

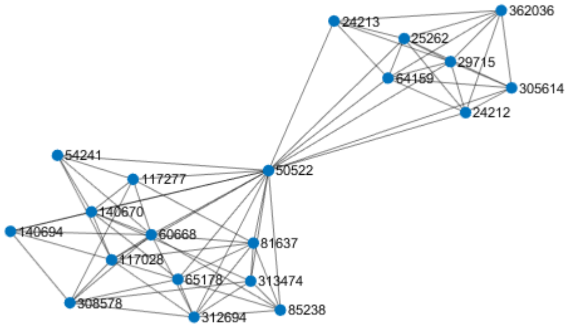


Fig 3: The 5-most core of RN network.

2.2 Betweenness Centrality

Betweenness centrality (B_{cen}) measures how often each node from a given graph appears on the shortest path between two nodes in that graph. Because of the existence of different shortest paths between any given two nodes in a graph, the betweenness centrality of node u is given by:

$$B_{cen}(\boldsymbol{u}) = \sum_{s,t\neq u} \frac{\boldsymbol{n_{st}}\ (\boldsymbol{u})}{\boldsymbol{N_{st}}} \tag{1}$$

Where $n_{st}(u)$ is the number of shortest paths from s to t passing through node u, and N_{st} is the count of shortest paths from node s to node t. If the graph is undirected, the equation is divided by two(MathWorks, n.d.).

2.3 Graphlets and Orbits

Graphlets are induced subgraphs, which are non-isomorphic and small connected as shown in figure 4. Graphlets are used as a topological measure to find the similarity between nodes in different networks (Kuchaiev et al., 2010) by using the graphlet degree vector (GDV) of a node or node signature.

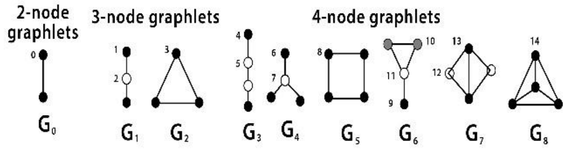


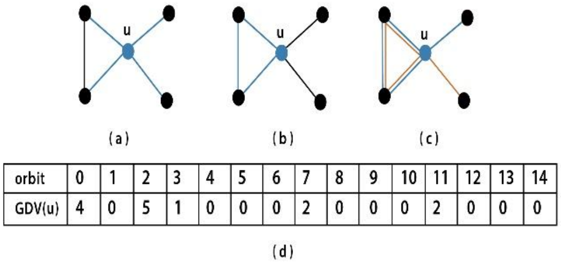
Fig 4: The 2-4 node graphlets, G0, G1, G2 … G8, and their corresponding 14 automorphism orbits. Nodes of the same orbit have the same shade.

As shown in figure 5, the GDV of a node represents the shape of the neighborhood around that node. The GDV of two different nodes can be used to compute how topologically they are similar. Equation 2 is one of many forms that can be used to compute the topological similarity T between any two nodes in the given networks.

$$T(u,v) = \frac{1}{15}\sum_{i=0}^{14} \frac{2 \times \min\left(d_u^i, d_v^i\ \right)}{\min\left(d_u^i, d_v^i\ \right) + \max\left(d_u^i, d_v^i\ \right)} \tag{2}$$

Where T is the similarity between node u and v, and d_u^i and d_v^i are the occurrence of orbit i around node u respectively.

As represented in figure 5, node u has four direct neighbors, which called its degree represented by orbit 0 from G0. G1 graphlet contains two orbits: orbit 1 and orbit 2. Orbit 1 equals zero since it does not exist around node u, while orbit 2 occurred five times. The next graphlet G2 contains only orbit 3, which equals one since node u only found once shaping a triangle. The occurrence of each orbit is computed to express each node neighborhood.



orbit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
GDV(u)	4	0	5	1	0	0	0	2	0	0	0	2	0	0	0

Fig 5: An illustration of the signature of node u, (a) the degree of node u represented by orbit 0 from G0, (b) orbit 3 from G2, (c) orbit 11 from G6, and (d) the complete 2-4 GDV of node u.

3. PROPOSED WORK

The proposed algorithm (CoreAlign) strategy is to search for suitable seeds to initiate the alignment and then extend the alignment by searching neighborhoods of these seeds.

The aligner needs a network file for each of the two networks, and any similarity score file where larger values indicate a higher similarity between proteins of the given networks. This similarity can be any topological or biological or a combination of both such as GDV signature, E-value, etc.

In CoreAlign, The topological similarity between protein u from the first network and protein v from the second network is measured using a 2-4 node graphlet degree signature according to equation 2.

For the biological similarity, a pre-introduced measure used in L-GRAAL aligner (Malod-Dognin & Pržulj, 2015) is used. It is formulated as the following:

$$B(u,v) = \frac{SeqSim(u,v\ )}{max_{i,j}\ SeqSim(i,j\ )} \tag{3}$$

The CoreAlign objective function, S, combines both of the above topological and biological similarities when mapping proteins between two networks, based on a balancing parameter $\propto \in [0,1]$. For topological similarity, only $\propto = 1$ is used, while for biological similarity only $\propto = 0$ is used. The proposed algorithm similarity function is computed according to equation 4.

$$S\big(u, f(u)\big) = S(u,v) = \propto \times\ T(u,v) + (1 - \propto) \times B(u,v) \tag{4}$$

The proposed algorithm (CoreAlign):

Input: *G1*, *G2*, similarity scores $S(u,v), \forall\ u \in V1, v \in V2$

Step 1: for the smallest network *G1*, run the K-core algorithm to assign each node with a core number, C^i.

Step 2: compute betweenness centrality (B_{cen}) for the nodes of both networks by equation 1.

Step3: start with C^i = k-most core.

```
    a.  Repeat until no unaligned nodes $U \in\ C^i$ exist.
        i.  Find seed $u \in U$, where $B_{cen}(u) = \max\ (B_{cen}(U))$.
    ii.   Find if there is already aligned neighbors $u'$, where $u' = N(u) \in C^i$ $or\ greater\ C$.
    iii.  If $u'$ is not empty and $f(u') \in V2$ still have unaligned neighbors $N(f(u'))$.
        1.  Find $v = \max\ (S(u, N(f(u')))$
        2.  Align node $u$ to node $v$.
    iv.   Else
        1.  find $v = \max\ (S(u, v_k))$,where $v_k$: $all\ unaligned\ nodes\ in\ V2$
        2.  Align node $u$ to $v$.
    v.    Repeat if node $u$ has unaligned neighbors $u''$
        1.  Findnode $x \in u''$, where $B_{cen}(x) = \max\ (B_{cen}(N(u))$.
        2.  Do the same as step ii and iii from step a.
  b.  $C^i = C^i - 1$
```

After reading the networks and having the final similarity scores, S, CoreAlign runs algorithm 1 to compute how many cores exist in the smallest network *G1*. K-core network decomposition algorithm assigns a core number to each node in the network. The nodes of each core are then ordered by betweenness centrality (B_{cen}).

The betweenness centrality, B_{cen} is computed for the given networks *G1* and *G2* according to equation 1, CoreAlign uses B_{cen} to select the seeds from each core of the smaller network *G1*. The betweenness centrality is also used to order the neighbors of these seeds to choose the next node to be aligned.

To construct the final solution, the proposed aligner starts from the k-most core. A seed is chosen to initiate the alignment. This seed *u* is the node with the maximum betweenness centrality value in the current core only. A suitable unaligned node *v* from the network *G2* is chosen based on the previously computed similarity scores between the two networks. The two nodes *u* and *v* from *G1* and *G2* are aligned to each other.

CoreAlign greedily searches the neighbors of seed *u* that belong to the current core to align first based on the betweenness centrality. To find a suitable match for each node *x* in the neighborhood of seed *u*, the aligner identifies already aligned neighbors of node *x*. Then it looks up to their matched nodes from the second network *G2*, after that, it finds their unaligned neighbors in *G2* to choose a suitable unaligned node y among them. The chosen node *y* has the maximum similarity value with node *x*, otherwise, choose node y from the remained unaligned nodes in *G2* based on the similarity function S.

After aligning all the nodes of the current core, the aligner steps into the lower core and so on until the whole network nodes in all cores of *G1* is aligned to unique nodes in *G2*.

4. RESULTS AND DISCUSSION

4.1 Datasets

For evaluating the resulting alignments the BioGRID dataset is used. Its PPI networks are manually curated obtained from yeast two-hybrid and affinity capture, so interactions may be direct or indirect. It consists of eight PPI networks: HSapiens (13276 proteins and 110528 interactions), DMelanogaster (7937 proteins and 34753 interactions, SCerevisiae (5831 proteins and 77149 interactions), AThaliana (5897 proteins and 13381 interactions), MMusculus (4370 proteins and 9116 interactions), CElegans (3134 proteins and 5428 interactions), SPombe (1911 proteins and 4711 interactions) and RNorvegicus (1657 proteins and 2330 interactions).

4.2 Evaluation Metrics

Several metrics can be used to evaluate alignment quality and similarity. They are divided into two categories: topological similarity metrics and biological similarity metrics. The use of both metrics becomes important since topological quality is not sufficient in all cases(Elmsallati, Clark, & Kalita, 2016).

To assess the topological similarity, edge correctness (*EC*) and symmetric substructure score (*S^3*) is used. Previously edge correctness (*EC*) was used widely, but recently symmetric substructure score (*S^3*) becomes the latest evaluation metric. The S^3 measure was proposed in MAGNA method (Saraph & Milenković, 2014).

For the second category: Biological similarity or functional coherence (*FC*), the KEGG Orthology (*KO*) annotations (Kanehisa, Goto, Sato, Furumichi, & Tanabe, 2012) retrieved via the UniProt mapping service (Bateman et al., 2017) is used.

Edge correctness (*EC*) is the percentage of the number of edges in the first network that has been preserved by the alignment.

$$EC = \frac{|f(E_1)|}{\min(\,|E_1|, |E_2|)} \times 100\% \qquad (5)$$

Where $f(E_1)$ is the conserved edge and it is formulated as:

$$f(E_1) = \{(u, v) \in E_1\ \cap (f(u), f(v)) \in E_2\ \} \qquad (6)$$

Symmetric substructure score (*S^3*) was introduced to overcome the failure of *EC* to measure the quality of sparse regions in the first network to dense regions in the second network. *S^3* considers mapping sparser regions to denser one and also mapping denser to sparser regions.

$$S^3 = \frac{|f(E_1)|}{|E_1| - |f(E_1)| + |E_2(G_2[f(V_1)])|} \times 100\% \qquad (7)$$

KEGG Orthology (*KO*) annotations are used for the integration of pathways and genomic information. It is known that orthologous genes are assigned the same *KO* annotation. The KO measure is computed as the number of shared *KO* annotations between the first network and aligned proteins from the second network.

$$KO = \sum\nolimits_{|V1|} 1\,, iff\ KO(u) = KO\big(f(u)\big) \qquad (8)$$

4.3 Evaluation of Experiment

The proposed algorithm (CoreAlign) is compared with several algorithms that are popular and available such as NETAL (Neyshabur et al., 2013), ModuleAlign (Hashemifar, Ma, & Naveed, 2016), MAGNA++ (Vijayan et al., 2015) and HubAlign (Hashemifar & Xu, 2014). The comparison shows the effectiveness of the proposed algorithm CoreAlign. The compared methods are run with their initial or recommended setting. NETAL used two iterations to compute similarities and 0.01 for the weight of interaction and similarity score. MAGNA++ is set to optimize *S^3* with 2000 in the population size and run it over 15000 generations. HubAlign is run with default $\propto = 0.7$. ModuleAlign uses the parameter $\propto = 0.3$.

CoreAlign is run for SCerevisiae (yeast) and HSapiens (human) PPI networks using topological similarity only. As illustrated later, CoreAlign is tested on the remaining six pairs of PPI networks topologically and biologically as well.

Figure 6 represents the edge correctness (*EC*) of the alignments generated by the previously mentioned aligners. It shows that CoreAlign results are comparable to ModuleAlign over different species. CoreAlign results are better than MAGNA++ and HubAlign which has the lowest scores. NETAL has the highest scores since it uses topological information only when construction the alignment.

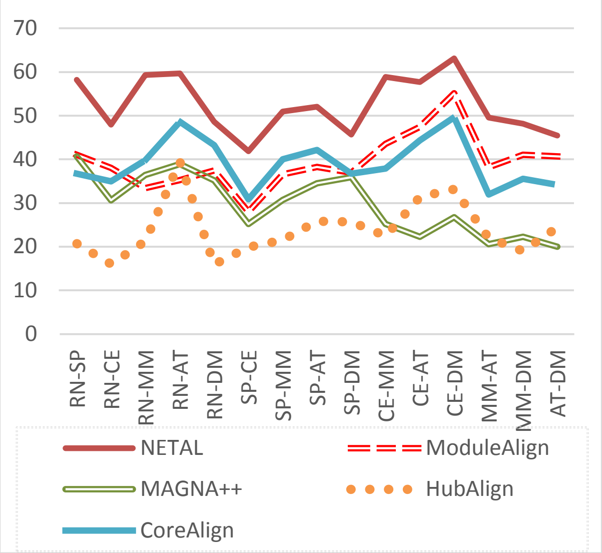


Fig 6: *EC* of NETAL, ModuleAlign, MAGNA ++, HubAlign, and CoreAlign.

In figure 7, the S^3 results of all mentioned aligners are given. NETAL has the best results. After NETAL, CoreAlign results are comparable to MAGNA++ which was set to optimize S^3 measure. CoreAlign results exceed HubAlign results in overall species. ModuleAlign has the lowest scores.

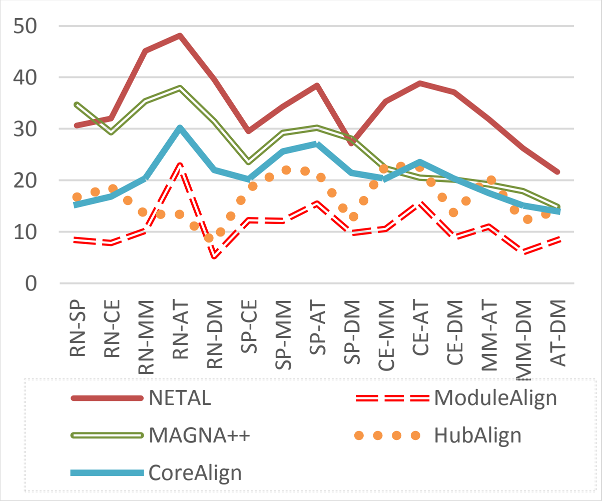


Fig 7: S^3 of NETAL, ModuleAlign, MAGNA ++, HubAlign, and CoreAlign.

The biological measure, *KO* is presented in figure 8. CoreAlign outperforms all aligners except HubAlign. NETAL and MAGNA++ fail to have biological fit since they rely on the topological information only. HubAlign which has the highest scores in the *KO* metric fails to have sufficient topological fit.

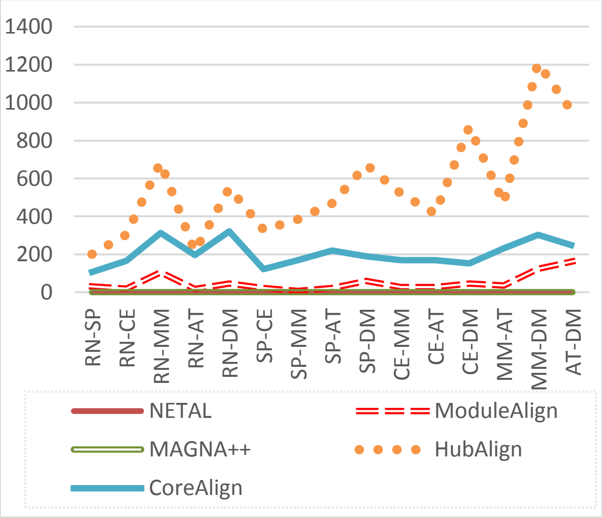


Fig 8: *KO* of NETAL, ModuleAlign, MAGNA ++, HubAlign, and CoreAlign.

The results of the aligners show that a high topological fit can be achieved at the expense of a better biological fit and the verse is true. NETAL focuses on maximizing the topological fit but failed to have acceptable results for function coherence. CoreAlign results are found good enough for both topological and biological assessments over the different species.

4.3.1 Evaluation of parameter ∝

The proposed aligner, CoreAlign, makes use of the parameter ∝ to balance between topological similarity (*T*) and biological similarity (*B*). The effect of using the parameter ∝ on the quality of network alignment is studied. CoreAlign is applied to the same dataset with different values of ∝ between 0 and 1 and the change on *EC*, S^3 and *KO* measures are reported.

It is obvious that changing ∝ to combine both topological and biological information has a little influence on topological metrics such as *EC* and S^3. It is also noticed that changing the parameter ∝ has an impact on biological metric *KO* as illustrated in figure 11.

Figure 9 shows the effect of changing the parameter ∝ on the edge correctness (*EC*) metric. The impact of decreasing ∝ from 1 to 0.1 is not noteworthy. The *EC* decreases only when ∝ = 0 which represents the use of the biological similarity only.

The influence of the parameter ∝ on S^3 is represented by figure 10. The same as the *EC* measure, no remarkable change is observed when ∝ changed from 1 to 0.1, but when using only biology information ∝ =0, the S^3 decreases.

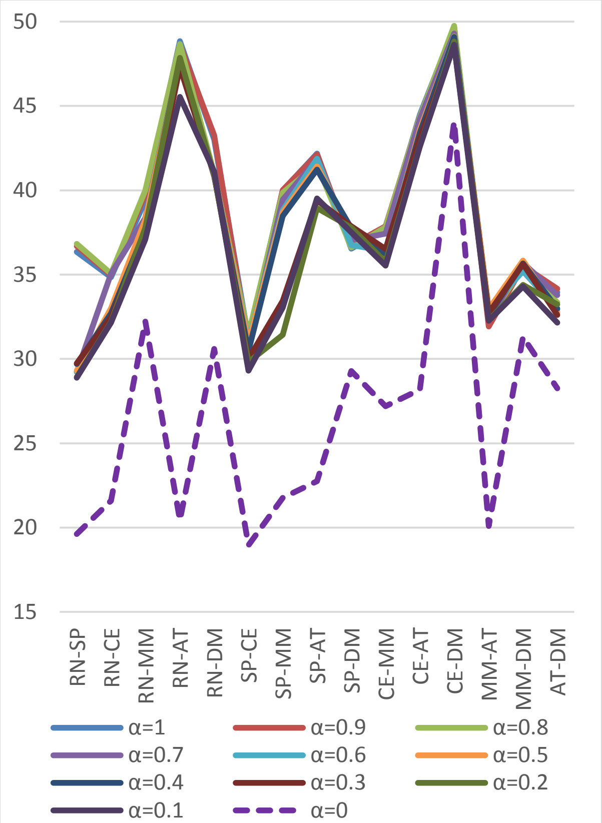


Fig 9: EC of CoreAlign over six species RNorvegicus (RV), SPombe (SP), CElegans (CE), MMusculus (MM), AThaliana (AT) and DMelanogaster (DM).

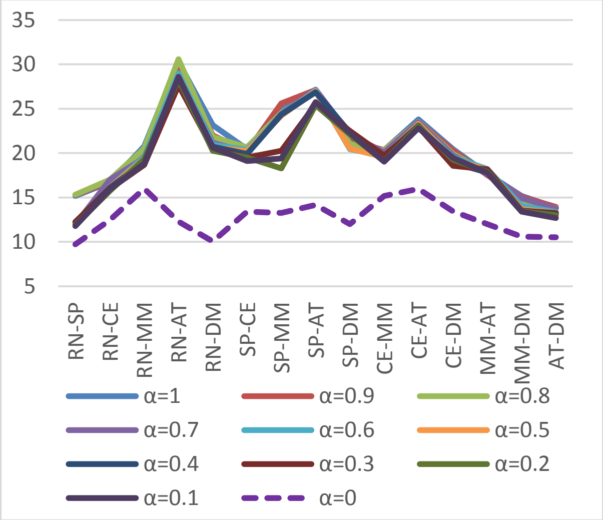


Fig 10: S3 of CoreAlign over six species RNorvegicus (RV), SPombe (SP), CElegans (CE), MMusculus (MM), AThaliana (AT) and DMelanogaster (DM).

As shown in figure 11, the effectiveness of changing the parameter $\propto$ on the biological measure *KO* is recorded. It is noticed that when decreasing $\propto$ to balance between the topological similarity and the biological similarity, the alignments become better. The more the use of sequence similarity, the better the results are achieved. The best score was obtained when $\propto = 0$.

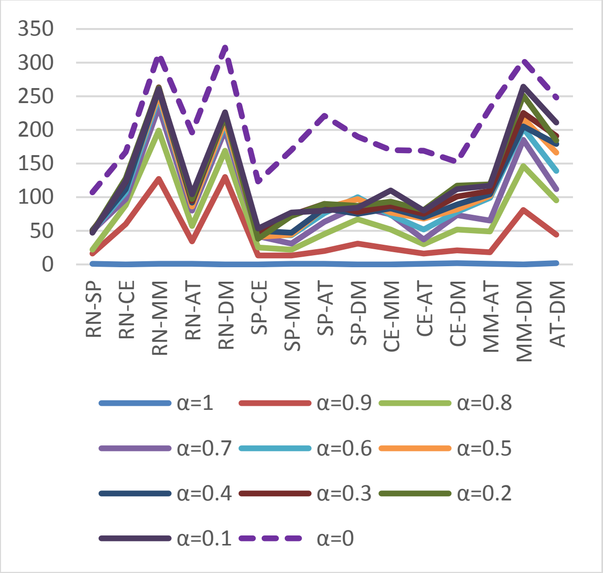


Fig 11: KO of CoreAlign over six species RNorvegicus (RV), SPombe (SP), CElegans (CE), MMusculus (MM), AThaliana (AT) and DMelanogaster (DM).

5. CONCLUSION

This paper introduces a global method called CoreAlign for aligning pairwise networks. It mixes the idea of network decomposition with some of the topological features such as betweenness centrality and graphlets. CoreAlign has been implemented and tested using some of the networks from the BioGRID dataset. The resultant alignments are evaluated by using different topological and biological measures. A comparison with current popular GNA methods shows that the proposed aligner outperforms many of them and can deal with large networks easily.

6. REFERENCES

[1] Bateman, A., Martin, M. J., O'Donovan, C., Magrane, M., Alpi, E., Antunes, R., … Zhang, J. (2017). UniProt: The universal protein knowledgebase. Nucleic Acids Research, 45(D1), D158–D169. https://doi.org/10.1093/nar/gkw1099

[2] Ciriello, G., Mina, M., Guzzi, P. H., Cannataro, M., & Guerra, C. (2012). AlignNemo: A local network alignment method to integrate homology and topology. PLoS ONE, 7(6). https://doi.org/10.1371/journal.pone.0038107

[3] Clark, C., & Kalita, J. (2014). A comparison of algorithms for the pairwise alignment of biological networks. Bioinformatics, 30(16), 2351–2359. https://doi.org/10.1093/bioinformatics/btu307

[4] Elmsallati, A., Clark, C., & Kalita, J. (2016). Global Alignment of Protein-Protein Interaction Networks: A Survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 13(4), 689–705. https://doi.org/10.1109/TCBB.2015.2474391

[5] Faisal, F. E., Meng, L., Crawford, J., & Milenković, T. (2015). The post-genomic era of biological network alignment. Eurasip Journal on Bioinformatics and Systems Biology, 2015(1). https://doi.org/10.1186/s13637-015-0022-9

[6] Hashemifar, S., Ma, J., Naveed, H., Canzar, S., & Xu, J. (2016). ModuleAlign: Module-based global alignment of

protein-protein interaction networks. Bioinformatics, 32(17), i658–i664. https://doi.org/10.1093/bioinformatics/btw447

[7] Hashemifar, S., & Xu, J. (2014). HubAlign: An accurate and efficient method for global alignment of protein-protein interaction networks. Bioinformatics, 30(17), 438–444. https://doi.org/10.1093/bioinformatics/btu450

[8] Janjić, V., & Pržulj, N. (2012). The Core Diseasome. Molecular BioSystems, 8(10), 2614–2625. https://doi.org/10.1039/c2mb25230a

[9] Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., & Tanabe, M. (2012). KEGG for integration and interpretation of large-scale molecular data sets. Nucleic Acids Research, 40(D1), 109–114. https://doi.org/10.1093/nar/gkr988

[10] Kelley, B. P., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B. R., & Ideker, T. (2004). PathBLAST: A tool for alignment of protein interaction networks. Nucleic Acids Research, 32(WEB SERVER ISS.), 83–88. https://doi.org/10.1093/nar/gkh411

[11] Koyutürk, M., Kim, Y., Topkara, U., Subramaniam, S., Szpankowski, W., & Grama, A. (2006). Pairwise Alignment of Protein Interaction Networks. Journal of Computational Biology, 13(2), 182–199. https://doi.org/10.1089/cmb.2006.13.182

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

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

[14] Liu, J. G., Ren, Z. M., Guo, Q., & Chen, D. B. (2014). Evolution characteristics of the network core in the facebook. PLoS ONE, 9(8). https://doi.org/10.1371/journal.pone.0104028

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

[16] MathWorks. (n.d.). Measure node importance - MATLAB centrality. Retrieved April 22, 2019, from https://www.mathworks.com/help/matlab/ref/graph.centrality.html

[17] Memišević, V., & Pržulj, N. (2012). C-GRAAL: Common-neighbors-based global GRAph ALignment of biological networks. Integrative Biology (United Kingdom), 4(7), 734–743. https://doi.org/10.1039/c2ib00140c

[18] Milenković, T., Ng, W. L., Hayes, W., & Pržulj, N. (2010). Optimal network alignment with graphlet degree vectors. Cancer Informatics, 9, 121–137.

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

[20] Saraph, V., & Milenković, T. (2014). MAGNA: Maximizing Accuracy in Global Network Alignment. Bioinformatics (Oxford, England), 30(20), 2931–2940. https://doi.org/10.1093/bioinformatics/btu409

[21] Sharan, R., Suthram, S., Kelley, R. M., Kuhn, T., Mccuine, S., Uetz, P., … Ideker, T. (2005). Sharan_ConservedPatternsProtInterac_PNAS_2005. PNAS, 102, 1974–1979. Retrieved from sftp://cerca@192.168.2.5/home/cerca/Desktop/data/laptop_files/info/biologia/Interactome_bioinformatics/interologs/Sharan_ConservedPatternsProtInterac_PNAS_2005/Sharan_ConservedPatternsProtInterac_PNAS_2005.pdf%5Cnpapers2://publication/uuid/4104E593-512B-4

[22] Singh, R., Xu, J., & Berger, B. (2007). Research in Computational Molecular Biology. Research in Computational Molecular Biology, (April). https://doi.org/10.1007/978-3-540-71681-5

[23] Vijayan, V., Saraph, V., & Milenković, T. (2015). MAGNA11: Maximizing accuracy in global network alignment via both node and edge conservation. Bioinformatics, 31(14), 2409–2411. https://doi.org/10.1093/bioinformatics/btv161

[24] West, D. (2001). Introduction to Graph Theory (2nd ed.). Perntice Hall, Upper Saddle River.