

Multiple Protein Network Alignment Using Pair-wise IBNAL Aligner

Mubarak Parekh and Jugal Kalita

Abstract—Protein-protein interaction and their networks solve the problem of finding biological correspondences between species through their PPI network alignment. IBNAL is a clique-based index network aligner which finds conserved network patterns across pairs of networks.

We present a global network alignment of multiple protein networks using IBNAL pair-wise aligner. Our algorithm presents four novel multiple alignment ways based on the number of cliques and nodes present in the networks to be aligned- Ascending and Descending sequence alignment, Consecutive network alignment and High-low network alignment. Our results confirm that ascending alignment of the networks which are aligned from highest number of cliques to lowest number of cliques produces a better aligner than other alignment ways. Using our results we also show that as the number of alignment pair increases the quality of resulting alignment decreases.

Index Terms—cliques, subordinate nodes, multiple network alignment, pair-wise network alignment

I. INTRODUCTION

Protein-protein interaction (PPI) networks are organized representation indicating interaction between the proteins across multiple species. Interpreting this data is an integral part of the inference of cellular machinery. PPI networks comprises of multiple nodes and edges. Each node in the network represents a set of protein and each edge designates a direct physical interaction between the proteins. A comparative approach for analyzing PPI networks calls for algorithms for protein network alignment. Alignments between these networks help gain functional understanding such as evolutionary conserved pathways, protein complexes and functional orthologs.

Previous research algorithms were local network aligners which searched for small, but highly conserved sub-networks [8]. The local network aligners were also used to duplicate sub-networks by using a node from a given network to map to several nodes of the other network to produce one-to-many or many-to-many multiple mappings. These multiple mappings indicate gene duplication, which often biologically improbable. Hence, an alternate to local network aligners known as global network aligners were developed and introduced to perform an overall comparison of the input networks and produce one-to-one mappings between the nodes of the two networks [2].

The solution to the network alignment problem involves identifying network regions that are well preserved and similar in their sequence and interaction pattern across two or more species. The result will be a network alignment graph with nodes and edges representing conserved PPI's across the investigated species.

Network alignment in pairs is difficult to compute due to NP-completeness of the underlying sub-graph isomorphic problem. Several alignment heuristics have been proposed for solving the pairwise network alignment such as Network-BLAST [7], PINALOG [6], NETAL [5] and IBNAL using global network alignment. However, its extension to more than two network has proved difficult due to the exponential growth of the alignment graph. With the availability of PPI networks of multiple species came the multiple network alignment problem, where given networks, aligning them means to group together the proteins that are evolutionary or functionally conserved between the networks. There are several multiple alignment algorithms available today such as Fuse [4], Graemlin [3] and BEAMS [1].

In this paper we present different multiple protein network aligners based on pair-wise IBNAL aligner. Our multiple PPI network aligner uses total number of cliques and subordinate nodes present in the network to perform multiple pair-wise alignment of networks. Our four different multiple aligners include: Ascending sequence aligner, Descending sequence aligner, Consecutive pair aligner and High-Low pair aligner.

II. METHOD

In this section, we will discuss our four different multiple aligners, implementation details and the datasets used for evaluating our aligners.

A. Multiple Network Aligners

We perform multiple alignment of networks using four different multiple aligners. These aligners combines pairs of network for alignment based on the total number of cliques and total number of subordinate nodes present in the aligning networks. Figure 1 shows four multiple aligners which are individually discussed below.

Ascending Sequence Multiple Alignment: This multiple network aligner first arranges networks in ascending order based on the total number of cliques present in each network. Then the aligner sequentially aligns the networks as seen in Figure 1 (a). It takes first two networks A and B in the arrangement and aligns them to produce resulting alignment R1. Since this is a sequence alignment, the result R1 is then aligned to network C to produce alignment R2. Similarly all networks in the arrangement are sequentially aligned to the result obtained from the previous alignment. The last network alignment produces the final alignment result.

Descending Sequence Multiple Alignment: This multiple network aligner first arranges networks in descending order

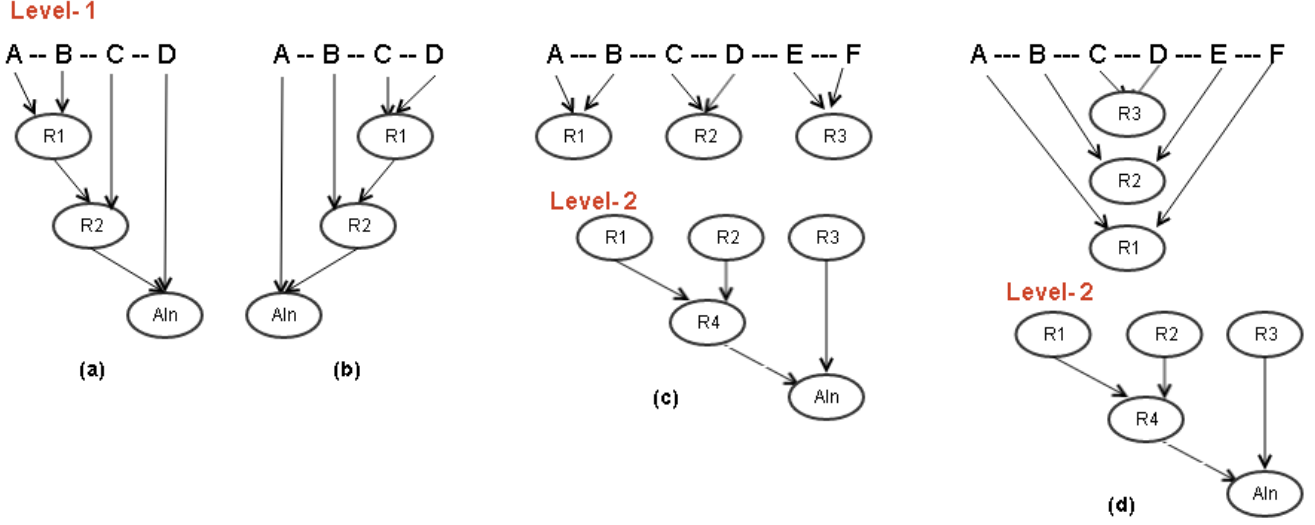


Fig. 1. Four Multiple Aligners. (a) Ascending Sequence Alignment, (b) Descending Sequence Alignment, (c) Consecutive Pair Alignment and (d) High-Low Pair Alignment

based on the total number of cliques present in each network. After the arrangement of networks, as seen in Figure 1 (b), alignment starts from lowest number of clique network to highest number of clique network such that D and C are aligned first to produce R1. It performs further alignments sequentially as done in Ascending aligner where the result obtained from two networks is aligned to third network and so on.

Consecutive Pair Multiple Alignment: This multiple network aligner works in two levels of alignment. First, it arranges networks in descending order based on the total number of cliques present in each network. as seen in Figure 1 (c) it takes consecutive pair of networks and aligns them to produce different results such that A and B produce R1, C and D produce R2 and so on. In the second level, it takes these results and arranges them in ascending order based on the subordinate nodes present in each network. These arranged results are then sequentially aligned to produce final alignment.

High-Low Pair Multiple Alignment: This multiple network aligner also works in two level alignment. In level 1, it arranges networks in ascending order depending on the total number of cliques in the networks. As seen in Figure 1 (d) to perform alignment, it takes network with highest number of cliques to align with lowest number of cliques and produce resulting alignment R1. Similarly, for next alignment result R2, it chooses the next lowest and highest number of clique networks. This continues until all results from level 1 is obtained. Level 2 alignment arranges all results obtained from level 1 in ascending order of total number of subordinate nodes and then perform alignment of networks sequentially as done in previous aligners.

B. Implementation

Implementation of multiple alignment using above discussed aligners is done in multiple steps. Algorithm 1 shows

the overall implementation of these steps with different aligners.

The preliminary step involves the clique extraction which is an offline step from multiple alignment of networks. We use Bron-Kerbosch algorithm in Java for maximal clique finding to extract all cliques from a given network. We then use one of our multiple aligners which arranges files in a particular arrangement based on total number of cliques and performs alignments of network in pair-wise. Pair-wise alignment is done using IBNAL aligner []. IBNAL uses similarity matrix and subordinate-clique index. It first aligns two nodes which have lowest similarity score and then aligns cliques that are attached to these two nodes and have the same clique size. Thus the alignment of nodes continues by finding the nodes that have next lowest similarity score.

After each pair alignment, the resulting alignment file which contains nodes from two networks is converted into a network file. This step is essential in multiple alignment of networks as the resulting alignment file cannot be used for next pair alignment directly. Algorithm 2 explains the conversion of alignment file to network file. The network aligned is basically extracted from the parent network file from either of the two aligned networks.

Further alignment continues as per the multiple aligner used by the user. Ascending and Descending aligners performs alignment using only one level of alignment whereas Consecutive and High-Low aligner performs alignment using 2 level of alignment.

C. Experimental Evaluation

All experiments were performed on Windows operating system, a 2.4GHz Intel(R) Core(TM)i5 processor and with 6GB main memory.

To evaluate the quality of each of our multiple aligner, we use two datasets- Isobase [] and NAPAbench suite datasets. Isobase uses four different well-know PPI networks of species-

Algorithm 1: Pseudo Code for Multiple Network Aligner

```

F[]  $\leftarrow$  Save all network files names
nF  $\leftarrow$  Num of networks to be aligned
nC[]  $\leftarrow$  Save num of cliques of all networks

function SeqMultAln(F[], nF, nC)
  initialize x;
  F[]  $\leftarrow$  arrange networks ascending/descending based
    numOfClqs nC[];
  // LEVEL 1
  while  $x < nF$  do
    IBNAL.pairAln(N1, N2);
    createNetworkFile(N1, N2);
    x = x+2;
  end

function ConsMultAln(F[], nF, nC)
  initialize x;
  F[]  $\leftarrow$  arrange networks ascending based nC[];
  // LEVEL 1
  while  $x < nF$  do
    if  $x$  is odd one out in alignment then
      | add x to LEVEL 2
    end
    else
      IBNAL.pairAln(N1, N2);
      createNetworkFile(N1, N2);
      RF[]  $\leftarrow$  new result alignment network file name
      RS[]  $\leftarrow$  save num of subordinate nodes of result
        alignment
      x = x+2;
    end
  end
  //LEVEL 2
  RF[]  $\leftarrow$  arrange networks ascending based respective
    subordinate nodes in RS[]
  nRF  $\leftarrow$  num of new alignment files
  SeqMultAln(RF[], nRF, nRC);

function HighLowMultAln(F[], nF, nC)
  low  $\leftarrow$  0;
  high  $\leftarrow$  nF-1;
  F[]  $\leftarrow$  arrange networks ascending based nC[];
  // LEVEL 1
  while low < high do
    if  $x$  is odd one out in alignment then
      | add x to LEVEL 2
    end
    else
      IBNAL.pairAln(N1, N2);
      createNetworkFile(N1, N2);
      RF[]  $\leftarrow$  new result alignment network file name
      RS[]  $\leftarrow$  save num of subordinate nodes of result
        alignment
      low++; high--;
    end
  end
  //LEVEL 2
  RF[]  $\leftarrow$  arrange networks ascending based respective
    subordinate nodes in RS[]
  nRF  $\leftarrow$  num of new alignment files
  SeqMultAln(RF[], nRF, nRC);

```

Algorithm 2: Pseudo Code for Creating a Network File

```

function CreateNetworkFile(N1, N2)
  Create file N1-N2.net
  Open file N1-N2.aln
  while line is not null do
    | M[]  $\leftarrow$  node of N1
  end
  Open file N1.net
  while line is not null do
    if both nodes of the edge E match nodes in M[] then
      | N1-N2.net  $\leftarrow$  E form N1.net
    end
    N[]  $\leftarrow$  node of N1
  end

```

Yeast, Human, Fly and Worm with total number of cliques-101957, 24018, 3150 and 340 respectively. It should be noted that the IBNAL alignment ignores all cliques of size 2 and all cliques greater than size 11. The clique number presented here excludes cliques of size 2 and 11. NAPAbench is a synthetic benchmark which is constructed based on a comprehensive scheme for generating evolutionary related families of synthetic protein-protein interaction (PPI) networks. We use 3 families from each- DMC-set and DMR-set for our experiments.

III. RESULTS AND DISCUSSION

In this section, we present our results obtained on different alignment ways used for multiple alignment of networks. Along with results, we discuss the criteria used during alignment and its effects on alignment results. Table I and Table II shows number of cliques and nodes for NAPAbench respectively. Table III and Table IV shows number of cliques and nodes for Real data respectively.

A. Alignment is affected by number of cliques in the network

On average in the resulting alignment, ascending alignment finds 94 cliques and 846 nodes and descending alignment finds 71 cliques and 840 nodes. The difference between the two results shows that ascending alignment performs better than descending alignment. If the first pair of networks in the multiple alignment both has high number of cliques, the resulting alignment which is used in further alignment of networks will keep the number of cliques consistently high. As opposed to ascending alignment, descending alignment starts the alignment with lowest number of cliques and so the final alignment results in comparatively less cliques and less nodes in the alignment. The results of descending alignment is only 1% less than ascending alignment which is not a significant difference. This indicates that the resulting alignment is slightly affected from arranging networks in ascending and descending number of cliques in the networks.

B. Multiple Alignment is affected by number of Levels of Alignment

Ascending and Descending sequence alignment is performed in one level where as Consecutive and High-Low align-

TABLE I
NUMBER OF CLIQUES ALIGNED USING FOUR DIFFERENT MULTIPLE ALIGNERS USING NAPABENCH

	5-way						8-way					
	DMC-set			DMR-set			DMC-set			DMR-set		
	f1	f2	f3	f1	f2	f3	f1	f2	f3	f1	f2	f3
AscAln	110	82	135	141	92	88	50	75	64	90	153	51
DscAln	85	60	62	127	95	136	48	56	49	45	46	44
ConAln	24	27	30	116	57	72	25	29	32	58	49	36
HLAln	63	29	30	135	73	122	26	41	38	49	33	39

TABLE II
NUMBER OF NODES ALIGNED USING FOUR DIFFERENT MULTIPLE ALIGNERS USING NAPABENCH

	5-way						8-way					
	DMC-set			DMR-set			DMC-set			DMR-set		
	f1	f2	f3	f1	f2	f3	f1	f2	f3	f1	f2	f3
AscAln	830	797	820	950	946	939	712	705	747	899	910	893
DscAln	831	807	809	955	954	969	722	655	691	865	925	901
ConAln	411	455	448	645	659	706	616	602	706	823	871	870
HLAln	803	770	763	947	932	959	622	633	675	878	811	876

TABLE III
NUMBER OF CLIQUES ALIGNED USING FOUR DIFFERENT MULTIPLE ALIGNERS USING REAL DATA

Aligner	3-way			4-way
	ce-dm-hs	ce-dm-sc	dm-hs-sc	ce-dm-hs-sc
AscAln	333	309	14425	309
DscAln	f	572	f	f
ConAln	160	550	5181	232
HLAln	118	101	2355	f

TABLE IV
NUMBER OF NODES ALIGNED USING FOUR DIFFERENT MULTIPLE ALIGNERS USING REAL DATA

	3-way			4-way
	ce-dm-hs	ce-dm-sc	dm-hs-sc	ce-dm-hs-sc
	2984	874	1748	874
	f	1778	f	f
	2911	814	1662	875
	2922	816	1830	f

ment is performed in two levels. The former two alignment ways have higher results in synthetic data than the later two as seen from tables of NAPAbench and Isobase data. Resulting alignment from IBNAL pair alignment decreases the number of cliques and number of nodes as it aligns more networks to the resulting alignment. This is due to IBNAL alignment nature of selecting nodes randomly during alignment process. Since the Consecutive and High-Low alignment takes place in 2 levels and has more number of pair-wise alignment than Ascending and Descending alignment ways, the results in terms of number of cliques and nodes are considerably less. On average, Consecutive alignment results in 23% less cliques and 51% less nodes and High-Low alignment results in 5% less cliques and 40% less nodes as compared to the highest results of Ascending alignment.

C. Overall Comparison Between Multiple Aligners

We have seen multiple alignment of networks are affected by number of cliques and levels of alignment done on the given set of networks. The Table I and Table II shows experiments performed on synthetic data. It shows different aligners having differences between the results based on the discussion of

cliques and level of alignment. Below discussion is based on average number of cliques and nodes aligned.

- AscAln has the highest results with 94 cliques and 845 nodes aligned as the alignment is done arranging networks starting with highest number of cliques to lowest number of clique network and also there are no levels in this alignment.
- DscAln results are slightly less than AscAln with 71 cliques and 840 nodes aligned due to the arrangement of networks in descending order of number of cliques.
- HLAln has 850 nodes aligned but only 56 cliques due to 2 levels of alignment.
- ConAln has the lowest results with 46 number of cliques and 651 total nodes aligned due to both the arrangement of networks based on cliques and 2 level of alignment.

The Table III and Table IV shows experiments performed on real data. Although some of the alignment resulted in to a failure shown as (f) in tables, the other alignments obtained show similar results to the synthetic data results. AscAln ranks first, DscAln second, HLAln third and lowest results are obtained from ConAln.

IV. CONCLUSION

We perform multiple network alignment using pair-wise aligner- IBNAL, an index based aligner. We present four different multiple network aligners: Ascending and Descending network aligner, Consecutive network aligner and High-Low network aligner. These aligners are developed based on the number of cliques and subordinate nodes present in the protein networks. These aligners perform alignment on level 1 and level 2. We obtained best results with the ascending and descending aligner which aligns networks from highest number of clique to lowest number of cliques. Performance of other aligners show that as the number of alignment between networks increases the quality of resulting alignment decreases. Multiple alignment with pair-wise alignment starting from network with more cliques to network with less cliques and alignment done with least number of pair-wise alignment can result in a better alignment quality.

REFERENCES

- [1] F. Alkan and C. Erten. Beams: backbone extraction and merge strategy for the global many-to-many alignment of multiple ppi networks. *Bioinformatics*, 30(4):531–539, 2014.
- [2] C. Clark and J. Kalita. A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics*, 30(16):2351–2359, 2014.
- [3] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou. Graemlin: general and robust alignment of multiple large interaction networks. *Genome research*, 16(9):1169–1181, 2006.
- [4] V. Gligorijević, N. Malod-Dognin, and N. Pržulj. Fuse: multiple network alignment via data fusion. *Bioinformatics*, page btv731, 2015.
- [5] B. Neyshabur, A. Khadem, S. Hashemifar, and S. S. Arab. Netal: a new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics*, 29(13):1654–1662, 2013.
- [6] H. T. Phan and M. J. Sternberg. Pinalog: a novel approach to align protein interaction networks implications for complex detection and function prediction. *Bioinformatics*, 28(9):1239–1245, 2012.
- [7] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker. Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Sciences of the United States of America*, 102(6):1974–1979, 2005.
- [8] B. S. Srinivasan, A. F. Novak, J. A. Flannick, S. Batzoglou, and H. H. McAdams. Integrated protein interaction networks for 11 microbes. In *Annual International Conference on Research in Computational Molecular Biology*, pages 1–14. Springer, 2006.