Supplementary Methods and Results for:

R.M. Desai, W.J.R. Longabaugh, and W.B. Hayes, *BioFabric Visualization of Network Alignments*.

1) Link and Node Groups With Blue Nodes

In the manuscript, for simplicity, we limited the discussion to the common case of aligning of network G_1 onto network G_2 when *every* node in G_1 is aligned onto a node in G_2. In the nomenclature we have introduced, this is an alignment "without any blue nodes". Tables 1 and 2 in the manuscript enumerate the possible link and node groups for these alignments. However, VISNAB is capable of handling alignments where unaligned blue nodes *are* permitted. In that case, the five link groups expand to seven, adding in groups three and four, which account for the case where blue edges are incident on blue nodes. Supplemental Table 1 enumerates all possible link groups in the presence of blue nodes in the alignment.

Link Group	Edge Color	Endpoint 1	Endpoint 2	Symbol
1	Purple	Purple	Purple	P
2	Blue	Purple	Purple	pBp
3	Blue	Purple	Blue	pBb
4	Blue	Blue	Blue	bBb
5	Red	Purple	Purple	pRp
6	Red	Purple	Red	pRr
7	Red	Red	Red	rRr

Supp. Table 1: Expansion of Link Groups When Unaligned (Blue) Nodes are Present

In a similar fashion, the twenty node groups that can be present in an alignment without blue nodes expands to forty possible groups when blue nodes are allowed. Note how, for example, Node Group 3 (present in Table 2 of the manuscript) splits into three distinct groups (3, 4, and 5) with blue nodes, since blue edges can be incident on blue nodes as well as purple nodes. In

a similar fashion, groups 7, 11, 14, 19, 23, 27, and 30 all split as well into three distinct groups. Groups 33 to 36 are introduced as well to account for blue nodes being present.

Node Group	Node Color	Incident Edges	Symbol
1	Purple	None	(P:0)
2	Purple	Purple	(P:P)
3	Purple	pBp	(P:pBp)
4	Purple	pBb	(P:pBb)
5	Purple	pBp, pBb	(P:pBp/pBb)
6	Purple	pRp	(P:pRp)
7	Purple	Purple, pBp	(P:P/pBp)
8	Purple	Purple, pBb	(P:P/pBb)
9	Purple	Purple, pBp, pBb	(P:P/pBp/pBb)
10	Purple	Purple, pRp	(P:P/pRp)
11	Purple	pBp, pRp	(P:pBp/pRp)
12	Purple	pBb, pRp	(P:pBb/pRp)
13	Purple	pBp, pBb, pRp	(P:pBp/pBb/pRp)
14	Purple	Purple, pBp, pRp	(P:P/pBp/pRp)
15	Purple	Purple, pBb, pRp	(P:P/pBb/pRp)
16	Purple	Purple, pBp, pBb, pRp	(P:P/pBp/pBb/pRp)
17	Purple	pRr	(P:pRr)
18	Purple	Purple, pRr	(P:P/pRr)
19	Purple	pBp, pRr	(P:pBp/pRr)
20	Purple	pBb, pRr	(P:pBb/pRr)
21	Purple	pBp, pBb, pRr	(P:pBp/pBb/pRr)
22	Purple	pRp, pRr	(P:pRp/pRr)
23	Purple	Purple, pBp, pRr	(P:P/pBp/pRr)
24	Purple	Purple, pBb, pRr	(P:P/pBb/pRr)
25	Purple	Purple, pBp, pBb, pRr	(P:P/pBp/pBb/pRr)
26	Purple	Purple, pRp, pRr	(P:P/pRp/pRr)
27	Purple	pBp, pRp, pRr	(P:pBp/pRp/pRr)
28	Purple	pBb, pRp, pRr	(P:pBb/pRp/pRr)
29	Purple	Blue, pRp, pRr	(P:pBp/pBb/pRp/pRr)
30	Purple	Purple, pBp, pRp, pRr	(P:P/pBp/pRp/pRr)
31	Purple	Purple, pBb, pRp, pRr	(P:P/pBb/pRp/pRr)
32	Purple	Purple, pBp, pBb, pRp, pRr	(P:P/pBp/pBb/pRp/pRr)
33	Blue	pBb	(B:pBb)
34	Blue	bBb	(B:bBb)
35	Blue	pBb, bBb	(B:pBb/bBb)
36	Blue	None	(B:0)
37	Red	pRr	(R:pRr)
38	Red	rRr	(R:rRr)
39	Red	pRr, rRr	(R:pRr/rRr)
40	Red	None	(R:0)

Supp. Table 2: Expansion of Node Groups When Unaligned (Blue) Nodes are Present

2) Jaccard Similarity With Blue Nodes

Again, for simplicity, the manuscript only discussed the definition of our Jaccard Similarity (JS) score when there are no blue nodes present in the alignment. When blue nodes are not allowed, then for every node in G_1, we can find (using the correct alignment) where that node is *supposed* to go in G_2, and (using the given alignment) where it *actually* ends up in G_2. These two nodes in G_2 can then be compared to create the JS score for the node.

However, when blue nodes are allowed, there are four possible cases that can arise instead of one:

1. The node is supposed to be aligned, and it is (the case described above) ("purple node stays purple")
2. The node is supposed to be aligned, and it is not ("purple node turns to blue")
3. The node is not supposed to be aligned, and it is ("blue node turn to purple")
4. The node is not supposed to be aligned, and it is not ("blue node stays blue")

VISNAB handles case 4 by simply assigning a score of 1.0 to the node, since it is correctly left unaligned. To deal with cases 2 and 3, VISNAB instead compares two nodes in network G_1. Specifically, for case 2, if a node a in G_1 is supposed to (using the correct alignment) be aligned to node n in G_2, but is instead unaligned, we look to see which node b in G_1 is aligned (using the given alignment) to node n in G_2. We then create the JS score for node a by comparing the neighborhoods of a and b in G_1. If there is no node b (when nothing is aligned to node n in G_2, i.e. it is "red"), then the JS score for node a is 0.0. Case 3 is handled analogously, again comparing two nodes a and b in G_1 to obtain a JS score, with a 0.0 assigned if there is no matching node in G_1.

For some network $G = (V,E)$, let $N_G(z_1) = \{z_2 \in V : (z_1, z_2) \in E\}$ be the neighborhood of node z_1 in G. For nodes $x,y \in V$, let $N_G(x,y)$ be the neighborhood of x disregarding y, and let i_{xy} be a corrective term accounting for a possible edge between the two. Accordingly, if $y \in N_G(x)$ then $N_G(x,y) = N_G(x) - y$ and $i_{xy} = 1$, else $N_G(x,y) = N_G(x)$ and $i_{xy} = 0$. Our extended JS definition $\sigma_G : V \times V \to [0,1]$ between two nodes is defined as:

$$\sigma_G(x,y) = \frac{|N_G(x,y) \cap N_G(y,x)| + i_{xy}}{|N_G(x,y) \cup N_G(y,x)| + i_{xy}}$$

Note that when x and y are both singletons, we define $\sigma_G(x,y) \equiv 1.0$ to avoid dividing by zero.

$$f_a(u) = \begin{cases} \sigma_{G_2}(a(u), a_c(u)) & \text{if } a(u) \text{ and } a_c(u) \text{ are defined} \\ \sigma_{G_1}(u, a_c^{-1}(a(u))) & \text{if } a(u) \text{ is defined} \\ \sigma_{G_1}(u, a^{-1}(a_c(u))) & \text{if } a_c(u) \text{ is defined} \\ 1.0 & \text{if } a(u) \text{ and } a_c(u) \text{ are undefined} \end{cases}$$

Given node sets $V_a, V_c \subseteq V_1$, an alignment $a : V_a \to V_2$, and the correct alignment $a_c : V_c \to V_2$, our JS measure for the given alignment a, with respect to the correct alignment a_c, is defined as:

$$JS(a) = \frac{1}{|V_1|} \sum_{u \in V_1} f_a(u)$$

If the correct alignment is provided, the user can choose to have correctly and incorrectly aligned nodes laid out separately in different node groups. The user can choose the criterion for the correct alignment to be based either on the traditional NC measure, or on our JS measure. If JS is chosen the user can set the threshold value $\beta \in [0,1]$, so a node in the form of $u::v$ or $u::$ is denoted correct if $f_a(u) \geq \beta$.

3) Creation of the Correct Network Alignment

To create the "correct" alignment used in the case studies, we wanted to create two networks where all nodes in the smaller network had one and only one known matching node in the larger network. One consequence of this approach is that our correct alignment did not have any blue nodes. These case studies use two different protein-protein interaction datasets. The larger "SC" network, from *S. cerevisiae*, contains 5,831 nodes and 77,149 edges, and was originally obtained from BioGRID (v3.2.101, June 2013) (Chatr-aryamontri *et al.*, 2013). The smaller "Yeast2" network, also from *S. cerevisiae,* has 2,390 nodes and 16,127 edges. It was originally generated from data in Collins *et al.* (2007) and used in Kuchaiev *et al.* (2010). Both networks were previously used in Mamano & Hayes (2017).

Nodes in Yeast2 are tagged with a variety of gene symbols (e.g. *PSY4*), secondary identifiers, and synonyms, while nodes in SC were tagged with ENTREZ IDs (e.g. 852234). In order to generate the "correct" alignment file, it was necessary to find the mapping from the former to the latter. To do this, we first used the YeastMine API (Balakrishnan *et al.*, 2012; Smith *et al.*, 2012) at https://yeastmine.yeastgenome.org/, provided by the Saccharomyces Genome Database (SGD) (Cherry *et al.*, 1998), in order to generate a mapping from the node names to the SGD IDs that we could then feed to the DAVID web tool (Huang *et al.*, 2009, 2009). With a Java program employing libraries provided by `org.intermine`, we downloaded (02/11/18) tuples for `Gene.primaryIdentifier`, `Gene.secondaryIdentifier`, `Gene.symbol`, and `Gene.synonyms.value` for `Gene.organism.shortName="S. cerevisiae"`, for all entries in the lists `Verified_ORFs`, `Dubious_ORFs` and `ALL_Verified_Uncharacterized_Dubious_ORFs`. Three remaining genes *YAR010C*, *YBR012W-B*, and *YHL009W-B* were not in any of these lists and were explicitly queried.

For each gene in Yeast2, we then matched the node name to a `Gene.synonyms.value`, and from this obtained a list of one or more `Gene.primaryIdentifiers`. In the cases where there was more than one, we chose the `Gene.primaryIdentifier` that mapped to a `Gene.symbol` that matched the `Gene.synonyms.value`. For example, synonym *MSL1* mapped to SGD IDs S000004374 and S000001448. However, while the former SGD ID mapped to gene symbol *NAM2*, the latter mapped to *MSL1*, and thus was selected. With one exception, this approach resulted in an unambiguous mapping of all Yeast2 node names to SGD IDs. The exception was for gene names EFG1 and YGR272C; the latter was merged into the former, giving both names the same SGD ID (S000007608). Thus, node YGR272C was dropped:

Gene Name	SGD ID
YGR272C	S000007608

These SGD IDs were then uploaded as a gene list to DAVID at https://david.ncifcrf.gov/conversion.jsp (DAVID 6.8, accessed 02/18/18). Since DAVID has restrictions on large-scale queries through their web API, this was done manually. Upon uploading the list, DAVID's Gene List Manager was not able to identify five IDs, so these nodes were dropped as well:

Gene Name	SGD ID
IMD1	S000000095
YNL276C	S000005220
YDR133C	S000002540
YDL026W	S000002184
YAR075W	S000002145

We instructed the tool to convert SGD_IDs to ENTREZ_GENE_IDs, and downloaded the result. Thus, we had a mapping of 2,384 of the nodes in Yeast2 to ENTREZ IDs. However, not all of these ENTREZ IDs are present as nodes in the larger SC network. In order to create a correct alignment with no blue nodes, we then pruned the Yeast2 network to remove the small number of nodes that could not be mapped onto the SC network. This resulted in an additional five nodes that needed to be dropped:

Gene Name	ENTREZ ID
ATM1	855347
PHM8	856759
PUT1	850833
CTM1	856509
SBE2	851953

Thus, we created a "Yeast2-reduced" network consisting of 2,379 nodes and 16,063 edges, which was used in the case studies.

4) Detailed Description of the Node Assignment Algorithm for the Node and Link Group Layout

The node assignment algorithm for the Node and Link Group Layout is a multi-queue breadth first search graph traversal. While a typical breadth first search utilizes a single queue, our multi-queue approach uses one queue for each node group, and the queues are processed in the order listed in Table 2 of the manuscript. The traversal starts on the node of highest degree in the first queue; its neighbor nodes are then visited in order of decreasing degree. If a newly visited node is in the current node group, it will be placed onto the current queue; if it is not, it will be placed onto the queue of its node group. The traversal is finished with a queue when every node in that node group has been visited. If the queue is empty but there still are unvisited nodes in that group, the highest degree node from the set of unvisited nodes of that group is added to the queue; after the queue is traversed, if there still are unvisited nodes in the group, this step is repeated until all nodes in the group are visited. Once finished, the traversal moves to the next queue. If a queue is empty when first evaluated, the node of highest degree in that queue's node group is added.

5) Full Table of All Alignment Scores for Mixtures of Importance and Symmetric Substructure Score

Supplemental Table 3 lists the scores for the ten-hour SANA (Mamano & Hayes, 2017) runs between Yeast2K-Reduced and SC, in which we used combinations of Importance (I) (Hashemifar and Xu, 2014) and Symmetric Substructure Score (S^3) (Saraph and Milenković, 2014) in the objective function. Note that all these scores, with the exception of Resnik, are available using the *Alignment Measures* tool in VISNAB. The Resnik scores (Resnik, 1995; Lord *et al.*, 2003a,b) shown here are the means of the non-zero, non-"None" values computed separately using FastSemSim (Guzzi 2012), incorporating Gene Ontology (GO) terms (Ashburner *et al*., 2000; The Gene Ontology Consortium, 2019) downloaded in February 2019.

Alignment	NGS	LGS	NC	JS	S^3	Resnik
Correct	1.00	1.00	1.00	1.00	0.25	9.63
$1.0 * I$	0.61	0.79	0.00042	0.021	0.0043	3.16
$.001 * S^3 + .999 * I$	0.88	0.86	0.00042	0.024	0.17	3.48
$.003 * S^3 + .997 * I$	0.88	0.86	0.00042	0.021	0.18	3.39
$.005 * S^3 + .995 * I$	0.72	0.85	0.00	0.025	0.10	3.27
$.01 * S^3 + .99 * I$	0.88	0.86	0.00	0.024	0.19	3.32
$.03 * S^3 + .97 * I$	0.87	0.80	0.018	0.057	0.27	3.62
$.05 * S^3 + .95 * I$	0.73	0.53	0.022	0.063	0.49	3.44
$.1 * S^3 + .9 * I$	0.67	0.48	0.017	0.067	0.54	3.50
$1.0 * S^3$	0.64	0.46	0.021	0.069	0.55	3.61

Supp. Table 3: All Alignment Scores for Case Study III

6) Table of Node Group Sizes for Case III

Supplemental Table 4 provides the number of nodes in each node group for the four alignments discussed in Case III. Asterisks show the four largest node groups per alignment, which are labeled prominently in the Figure 4 in the manuscript. As called out in the text, of the 716 nodes in the top 13 rows above group **(P:P/pRp/pRr)** for the mixed alignment, 544 (76%) have no incident **pRr** edges.

Symbol	**Correct**	**All Importance**	**Mixed**	**All S3**
(P:0)	0	0	0	0
(P:P)	2	0	23	16
(P:pBp)	0	0	0	0
(P:pRp)	0	0	0	0
(P:P/pBp)	2	0	59	23
(P:P/pRp)	52	0	62	3
(P:pBp/pRp)	32	403 *	207	0
(P:P/pBp/pRp)	27	9	193	1
(P:pRr)	0	0	0	0
(P:P/pRr)	5	0	31	439
(P:pBp/pRr)	5	35	98	303
(P:pRp/pRr)	0	0	0	0
(P:P/pBp/pRr)	1	9	43	662 *
(P:P/pRp/pRr)	981 *	0	530 *	157

(P:pBp/pRp/pRr)	310	1677 *	82	0
(P:P/pBp/pRp/pRr)	962 *	246	1051 *	775 *
(R:pRr)	578 *	843 *	752 *	31
(R:rRr)	209	58	0	1087 *
(R:pRr/rRr)	2665 *	2551 *	2700 *	2334 *
(R:0)	0	0	0	0

Supp. Table 4: Sizes of all Node Groups in Case III

7) Percentage of Purple Nodes Without and With Incident pRr Edges between Correct and Mixed Alignments

The manuscript discussion of Figure 4 notes that while there are *more* **pRr** edges in the mixed alignment compared to the correct alignment, those edges are concentrated across a *smaller fraction* of the purple nodes in that mixed alignment. Supplemental Table 5 compares the percentages of all purple nodes without [**(P:*)**] and with [**(P:*/pRr)**] **pRr** incident edges, between the correct and mixed alignments.

Alignment	**(P:*)**	**(P:*/pRr)**
Correct Alignment	4.83%	95.17%
Mixed Alignment	22.87%	77.13%

Supp. Table 5: Comparison of Purple Node Fractions Without and With pRr Edges

8) Alignment Cycle Layout With Blue Nodes

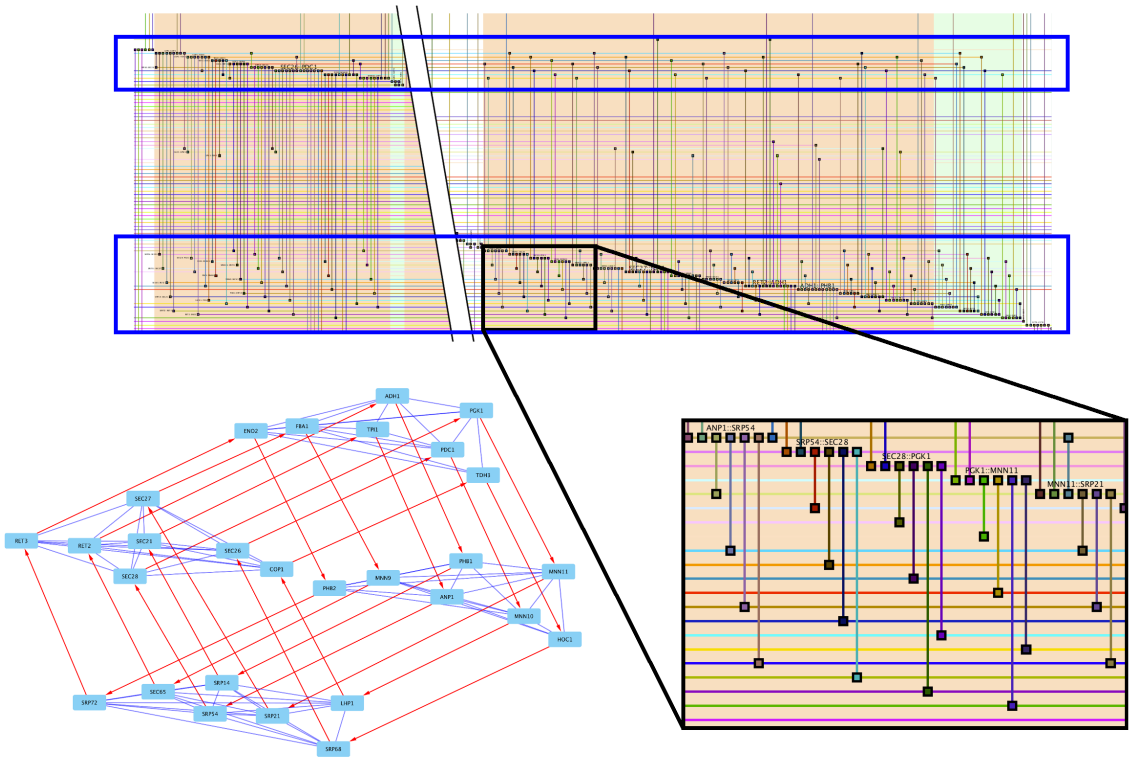
When unaligned blue nodes are not allowed, there are four cases that must be handled by the Alignment Cycle layout, and these are indicated by a checkmark in the rightmost column of Supplemental Table 6. When unaligned blue nodes are present, there are nine path types that must be handled. In the table, a network with nodes {A, B, C, … L} has been aligned onto a network with nodes {1, 2, 3, …12}. Note that purple node runs can extend for any length of nodes, as shown by the ⋯, but the matches and alignments given in this table are for the cases where there are none of these extra nodes. The Alignment Cycle layout will order the nodes in the path and cycle cases so that misaligned nodes are laid out next to their correct partners; see case 9 in particular to see this pattern.

#	Layout	Correct Alignment	Test Alignment	Type	No Blue
1	(A::)	A → Ø	A → Ø	Correct	
2	(::1)	Ø → 1	Ø → 1	Correct	✓
3	(B::2)	B ↔ 2	B → 2	Correct	✓
4	(C::3)	C → Ø; Ø → 3	C → 3	Path	
5	(::4) (D::)	4 ↔ D	Ø → 4; D → Ø	Path	
6	(::5) (E::6) ⋯	5 ↔ E; Ø → 6	Ø → 5; E → 6	Path	✓
7	(F::7) ⋯ (G::)	7 ↔ G; F → Ø	F → 7; G → Ø	Path	
8	(::8) (H::9) ⋯ (I::)	8 ↔ H; 9 ↔ I	Ø → 8; H → 9; I → Ø	Path	
9	(J::10) (K::11) ⋯ (L::12)	10 ↔ K; 11 ↔ L; 12 ↔ J	J → 10; K → 11; L → 12	Cycle	✓

Supp. Table 6: Alignment Cycle Layout Cases

9) The Four-Cluster Misalignment

In Case Study IV, we showed how the Alignment Cycle layout could be used to spot alignment problems such as two entire protein clusters being swapped. Supplemental Figure 1 shows a severe degeneracy for the same alignment run, where *four* separate protein clusters were misaligned in a cycle. The BioFabric depiction of this problem follows the same pattern shown in Figure 6B, but the successive edge wedges are even steeper here, and show a clear pattern of cycling between four distinct sets of node rows.



Supp. Figure 1: An even more striking misalignment, where four different protein complexes have been swapped in a round-robin fashion. The traditional node-link diagram is shown at the lower left with edges (colored blue) for the protein-protein interactions and directed edges (colored red) for the alignments. The four protein complexes clockwise from top (per SGD): 1) glycolysis and gluconeogenesis related genes, 2) mannosyltransferase complex and prohibitin complex, 3) signal recognition particle, and 4) the coatomer complex (COPI). The BioFabric layout on the top, shown in detail at the lower right, shows the distinct pattern displayed by this artifact, with adjacent edge wedges having edges cycling every fourth node. Note that a slice was removed from the upper view because the three separate cycles constituting this structure are not contiguous in the layout.

REFERENCES

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., Sherlock, G. (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, **25**(1), 25-29.

Balakrishnan, R., Park, J., Karra, K., Hitz, B. C., Binkley, G., Hong, E. L., Sullivan, J., Micklem, G., and Michael Cherry, J. (2012). Yeastmine-an integrated data warehouse for Saccharomyces cerevisiae data as a multipurpose tool-kit. *Database,* bar062.

Chatr-aryamontri, A., Breitkreutz, B. J., Heinicke, S., Boucher, L., Winter, A., Stark, C., Nixon, J., Ramage, L., Kolas, N., ODonnell, L., Reguly, T., Breitkreutz, A., Sellam, A., Chen, D., Chang, C., Rust, J., Livstone, M., Oughtred, R., Dolinski, K., and Tyers, M. (2013). The BioGRID interaction database: 2013 update. *Nucleic Acids Research*, **41**(D1), D816-D823.

Cherry, J., Adler, C., Ball, C., Chervitz, S., Dwight, S., Hester, E., Jia, Y., Juvik, G., Roe, T., Schroeder, M., Weng, S., and D., B. (1998). SGD: Saccharomyces Genome Database. *Nucleic Acids Research*, **26**(1), 73-79.

Collins, S. R., Kemmeren, P., Zhao, X.-C., Greenblatt, J. F., Spencer, F., Holstege, F. C. P., Weissman, J. S., and Krogan, N. J. (2007). Toward a comprehensive atlas of the physical interactome of saccharomyces cerevisiae. *Molecular and Cellular Proteomics*, **6**(3), 439-450.

Guzzi, P.H., Mina, M., Guerra, C., Cannataro, M. (2012). Semantic similarity analysis of protein data: assessment with biological features and issues. Briefings in bioinformatics, **13**(5), 569–585.

Hashemifar, S. and Xu, J. (2014). HubAlign: an accurate and efficient method for global alignment of proteinprotein interaction networks. *Bioinformatics*, **30**(17), i438-i444.

Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2009). Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature Protocols*, **4**(1), 44-57.

Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2009). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Res*, **37**(1), 1-13.

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.

Lord, P., Stevens, R. D., Brass, A., and Goble, C. A. (2003a). Semantic similarity measures as tools for exploring the gene ontology. In *Proc. of the 8th Pacific Symposium on Biocomputing*, pages 601-612.

Lord, P. W., Stevens, R. D., Brass, A., and Goble, C. A. (2003b). Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. *Bioinformatics*, **19**(10), 1275-1283.

Mamano, N. and Hayes, W. B. (2017). SANA: simulated annealing far outperforms many other search algorithms for biological network alignment. *Bioinformatics*, **33**(14), 2156-2164.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448-453, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Saraph, V. and Milenković, T. (2014). MAGNA: maximizing accuracy in global network alignment. *Bioinformatics*, **30**(20), 2931-2940.

Smith, R. N., Aleksic, J., Butano, D., Carr, A., Contrino, S., Hu, F., Lyne, M., Lyne, R., Kalderimis, A., Rutherford, K., Stepan, R., Sullivan, J., Wakeling, M., Watkins, X., and Micklem, G. (2012). InterMine: a flexible data warehouse system for the integration and analysis of heterogeneous biological data. *Bioinformatics*, **28**(23), 3163-3165.

The Gene Ontology Consortium. (2019). The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Research*, (**47**)D1, D330–D338.