Journal of Bioinformatics and Computational Biology
© Imperial College Press

DISTANCE-WISE PATHWAY DISCOVERY FROM PROTEIN-PROTEIN INTERACTION NETWORKS WEIGHTED BY SEMANTIC SIMILARITY

SLAVKA JAROMERSKA

Department of Computer Science, Baylor University,
One Bear Place #97356, Waco, Texas 76798, USA
slavka_jaromerska@baylor.edu

PETR PRAUS

Department of Computer Science, Baylor University,
One Bear Place #97356, Waco, Texas 76798, USA
petr_praus@baylor.edu

YOUNG-RAE CHO*

Bioinformatics Program, Department of Computer Science, Baylor University,
One Bear Place #97356, Waco, Texas 76798, USA
young-rae_cho@baylor.edu

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Reconstruction of signaling pathways is crucial for understanding cellular mechanisms. A pathway is represented as a path of a signaling cascade involving a series of proteins to perform a particular function. Since a protein pair involved in signaling and response have a strong interaction, putative pathways can be detected from protein-protein interaction (PPI) networks. However, predicting directed pathways from the undirected genome-wide PPI networks has been challenging. We present a novel computational algorithm to efficiently predict signaling pathways from PPI networks given a starting protein and an ending protein. Our approach integrates topological analysis of PPI networks and semantic analysis of PPIs using Gene Ontology data. An advanced semantic similarity measure is used for weighting each interacting protein pair. Our distance-wise algorithm iteratively selects an adjacent protein from a PPI network to build a pathway based on a distance condition. On each iteration, the strength of a hypothetical path passing through a candidate edge is estimated by a local heuristic. We evaluate the performance by comparing the resultant paths to known signaling pathways on yeast and worm. The results show that our approach has higher accuracy and efficiency than previous methods.

Keywords: Protein-protein interaction networks; Signaling pathways; Gene Ontology; Semantic similarity.

*Corresponding author

1. Introduction

Understanding signal transduction processes is a central step to elucidate functional mechanisms of cellular molecules and agents that surround them. Over the past decade, systematic approaches using high-throughput experimental techniques[1,2] have attempted to reconstruct signaling pathways. A signaling pathway is defined as a linear path of the signaling cascade involving a series of genes to perform a particular function. It generally starts with a membrane-bound receptor gene, contains a series of genes which cause signal transduction, and ends with a transcription factor gene. Since a protein pair involved in signaling and response typically have a strong interaction, signaling pathways can be detected from protein-protein interaction (PPI) networks given starting and ending proteins. Various high-throughput methods have recently generated PPI data on the scale of entire genome[3]. Subsequently, signaling pathways have been predicted computationally from the genome-wide PPI networks. A general idea of the computational approaches is to assign each PPI an edge weight and search for the strongest path which is considered as a putative signaling pathway.

Previously, Scott *et al.*[4] used the idea of color-coding to assign each vertex in a PPI network a random color between 1 and k and search for paths with distinct colors instead of searching for the strongest paths. The complexity of the dynamic programming algorithm is thereby reduced. However, a path fails to be discovered if any two of its vertices receive the same color, so many trials of random colorings are required to ensure that all desired paths are considered. The running time of the color-coding algorithm is exponential in k and linear in n, number of nodes in the graph, and the storage requirement is exponential in k and linear in n. Since k also limits the possible discovered path length, the algorithm is exponential in maximum path length. Moreover, because this method uses summation of edge weights to compute path strength, it is biased towards longer paths.

Gitter *et al.*[5] defined the Maximum Orientation (MEO) problem as searching for edge orientation of the undirected PPI network, which would maximize the sum of strengths of all satisfied paths from given sources to targets. However, because MEO is a typical NP-hard problem, they suggested several approximation algorithms. Their first suggestion is random orientation of edges. Second, they used known approximation algorithms by reducing the problem to MAX-k-CSP and MIN-k-SAT. Finally, they proposed adding local search to further optimize the results. The local search iteratively finds an edge whose orientation can be flipped that benefits the optimization the most. However, each additional constraint is a trade-off between runtime and approximation accuracy. We found that the runtime of this method for any paths longer than 5 becomes problematic, nevertheless the memory requirement was a more serious and limiting issue.

In this paper, we present a novel computational approach to discover signaling pathways from the genome-wide PPI networks integrated with Gene Ontology (GO) annotation data. A key assumption in this approach is that we can place confidence values on interactions between different proteins. Recent research[6] has suggested that PPIs can be validated by ontological analysis of interacting proteins. We thus quantify the confidence of PPIs as their weights by semantic similarity[7], a function that returns a numeric value reflecting closeness in meaning between two ontological terms to which the proteins are annotated. Gene Ontology (GO) annotation data[8] are used to measure semantic similarity because GO is currently the most complete repository of biological ontologies and annotations over various model organisms.

We propose a distance-wise algorithm to predict pathways from weighted PPI networks given a starting protein (a source), an ending protein (a target) and the maximum path length. Recall that searching for the strongest (or longest) path under such conditions is an NP-complete problem and its efficient (however, suboptimal) solutions must introduce at least some level of approximation. Our approach aims at not only collecting all proteins involved in a particular signaling cascade, but also ordering them to form a directed signaling pathway. The algorithm iteratively adds an edge into an already discovered path based on a distance condition. On each iteration, the algorithm selects the set of candidate edges and estimates the path strength of a hypothetical path passing through each candidate edge using a local heuristic. The combination of the distance condition and our local heuristic approach achieves superior results of pathway discovery in terms of space and time complexity, even for long pathways, than previous methods.

2. Method

2.1. *Pathway detection algorithm*

The path finding problem is to find the strongest simple paths between a source and a target in an undirected weighted graph. The strength of a path p is defined as $S(p) = \Pi_{e \in p} w(e)$ where $w(e)$ is the weight of an edge e in p. Here, we state that $w(e)$ must be in the range of $[0, 1]$ where 1 means we are the most confident of the interaction. It is important to note that the path strength function forces longer paths to have smaller values than shorter paths basically giving preference to shorter paths. Since real pathways are predictably short because biological responses are usually controlled shortly, this is a reasonable model for pathway prediction.

Our algorithm searches for a path while traveling from a source to a target on a PPI network. The most determining idea behind the proposed algorithm is the restriction of search space only for the paths of given maximum length or shorter. By setting an upper bound of total path length, we are simultaneously restricting search space of the next candidate nodes. Any node participating in the discovered paths must hold on the following distance condition.

(**Distance Condition**) For any node v, the sum of minimum distances from the source to v and from v to the target must be lower than or equal to the given maximum path length.

To efficiently meet the distance condition, we use breadth-first search (BFS) to

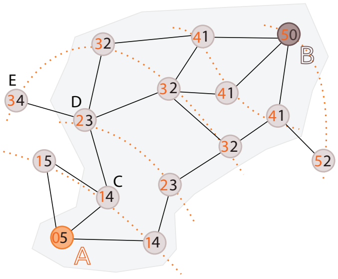


Fig. 1. Distance labeling by breadth-first search (BFS) from source A to target B. The first number in node labels refers distance to source, and the second to target. Dotted lines show the waves of BFS from A. Grey area indicates the restricted search space by maximum path length of 5.

label minimum distances for each node with respect to the source and target pair (see Figure 1 for details). Once all necessary labels are computed, the query on satisfiability is just constant-time summation of distances to the source and target.

2.1.1. *Use of distance bounding condition*

Our algorithm, starting from the source, iteratively selects an edge towards the target and consequently builds a path. At each node, an edge is chosen based on a local heuristic and strategy (see the next two subsections), the edge is added to the path, and the search continues at the recently added edge towards the target. However, searching for all possible edges is not necessary. Only the edges which meet the distance condition are chosen – we call these candidate edges. Here, we define the extended distance condition which is more restrictive. As a consequence of this definition, "source minimum distance labels" on each node are not necessary anymore. For all paths to the same target, we thus need only a single BFS run.

(**Extended Distance Condition**) The sum of minimum distance to target and distance already traveled by the algorithm (the number of edges currently added for each path) must be lower than or equal to the given maximum path length.

In Figure 1, suppose we are currently at D, we are looking for a path of maximum length 5 from A to B, and we have come through node C. This setup means that we have already traveled two hops. We will need another hop for the edge we are about to choose, thus the remaining possible number of hops is 2. Therefore, we may consider only the edges whose connecting nodes have the "target minimum distance label" lower than or equal to 2. Due to this reasoning, the edge connecting to the node E will be excluded from the set of candidate edges.

2.1.2. *Local heuristic for candidate edge ordering*

At the point of the algorithm when we are about to add a new edge to the already discovered path segment, we need to make a choice from candidate edges – the set of edges holding on the extended distance condition minus the set of edges that make cycles (see a following subsection on how to modify the set of candidate edges to achieve simple paths). First, we order the candidate edges by a local heuristic. Second, we select one of them, but not necessarily the best edge, e.g. in case we want to introduce randomness. A naive approach to the local heuristic would use the edge weights that are already given. However, remember our path strength scheme. The path strength is given by the product of all edge weights. Such a simple greedy strategy, i.e. repeated selection of the edge having the highest weight, behaves as computing the path strength by rather a sum of weights and definitely does not give any preference to shorter paths. We thus need a more sophisticated approach for edge selection.

To evaluate a candidate edge e, we use the overall path strength of a path leading through the edge e. We know the strength of the already discovered path segment and the weight of the next edge e. We also know the minimum distance l_r we yet have to travel after selecting e. What we do not know are the edge weights on the undiscovered portion of the path. For this remaining part of the potential pathway, we use a weight estimate $\hat{w_r}$. We compute $\hat{w_r}$ in two different ways. One is the average of edge weights in the search area. This computation requires two BFS runs but the result is a constant value reusable for the entire pathway discovery. The other is the average of edge weights on the already determined path (including $w(e)$). We will refer to them in the rest of the paper as **Average** and **Experience** heuristics, respectively. The path strength estimate $\hat{S_e}$ of a possible path going though an edge e is computed as

$$\hat{S_e} = S_d \cdot w(e) \cdot \hat{w_r}^{l_r} \tag{1}$$

where S_d is the strength of the already discovered path as part of the potential pathway, and l_r is the minimum remaining path length after adding e.

2.1.3. *Strategies for edge selection*

Once we have the candidate edges evaluated and ordered, we need to select one of them. Selecting the edge with the highest path strength estimate (as computed in the previous section) does not have to yield the best results. In this way, first, we would be able to return only a single path between a source and a target. Second, one very strong edge might lead the algorithm astray. To avoid this phenomenon, we introduce probabilistic selection.

We propose several different strategies to select an edge from ordered candidates. **Top Random Maximum** takes a fixed portion of top edges and selects one edge according to a discretized Gaussian probability distribution, where the highest ranking edge receives the highest probability of being selected. The fixed percentage is a parameter value. **Smooth Random Maximum** also takes a portion of top edges, but instead of using a percentage of their ranks, it uses a percentage of the estimated path strength value range. Also, a value from that range is selected by the probability distribution, rather then an edge ranking, so the edge winner becomes the edge with the closest value.

While both previous two strategies need the parameter of a fixed percentage, **Variable Smooth Random Maximum** uses a variable percentage which are changed during different stages of discovery. At the beginning of pathway discovery, we are more likely to make an incorrect decision because the vast majority of the path strength is guessed. Thus, at the beginning of the path, we need more randomization in order to explore more possibilities. As the target approaches, our path estimates are becoming more precise and we can rely on them more. Therefore, the ratio of top edges that are to be included in the randomization process is proportional to the remaining path length and inversely proportional to the traveled path length. Since our edge selection has a randomization mechanism, we can now run the whole algorithm multiple times and discover several possibly strong paths. We can also combine different strategies and heuristics to improve the results.

2.1.4. *Cycle Avoidance*

Note that the proposed algorithm enables us to perform distance-wise search without explicitly selecting the search area (the gray area on Figure 1. Moreover, apart from keeping the path only in the search area, this mechanism also prevents the path from unnecessary walking in spirals. In other words, the output path will never be longer than the maximum path argument and a simple path (i.e., a path with no cycles) will always be found if exists. Here a cycle avoidance mechanism comes in handy. When selecting candidate edges, all edges that would lead to completing a cycle (edges with the target node already on the discovered path) are omitted. In this approach, we could encounter a situation where no candidate edges are available. In this case, the last edge from the discovered path is dropped and handled as an edge that completes a cycle (thus omitted from the set of candidate edges). It indicates that we will backtrack by one step since we reach an impasse on the previously chosen edge. This mechanism guarantees that a simple path of given length or shorter will be discovered if exists.

2.1.5. *Time and space complexity*

The random nature of the algorithm makes it hard to put meaningful upper bounds on time complexity. Let n be the number of nodes in the PPI network, m number of edges and l the maximum path length between the source and target. In the worst, however non-realistic, case that the PPI graph is a clique, all edge weights are equal to 1 and we are searching for a very long path ($l = m$), the algorithm could have time complexity as high as $O(n \cdot m)$. For an optimistic case (no backtracking occurs) with a typical PPI network, time complexity is about $O(l \cdot b)$, where b is the maximum degree of the PPI graph, if we reasonably implement cycle detection. A large number of nodes in the typical PPI networks have sufficient degrees and thus the backtracking is quite rare. Sorting edges adds $log(b)$ for each selected edge, but it doesn't change the asymptotic bound $O(l \cdot (b + log(b)) = O(l \cdot b)$. BFS search (once or twice) adds extra $O(m + n)$, nevertheless that computation can be reused for several runs of the algorithm according to the selected heuristic and strategy. The space complexity is driven by necessity to store distance labels. Although some additional memory will be needed for each step of the algorithm (at most b), the total bound is about $2n + b$.

2.2. *Edge weighting method*

We compute edge weights of PPI networks by semantic similarity measurement. Various semantic similarity measures have been introduced to quantify functional similarity between proteins. Previous studies[6,9] showed the semantic similarity metrics based on information contents such as Resnik's and Jiang's methods[10], node-based methods such as simUI[11], and integrative methods such as simGIC[12] have relatively high accuracy. Resnik's method measures the semantic similarity using the information content of the most specific common ontology term. Jiang's method measures the differences of information contents between the most specific common ontology term and the two terms of interest. Node-based methods explore the overlap of two sets of ontology terms having the annotation of two proteins of interest, respectively. simUI is the normalized version of this method by the union of the two sets. simGIC is a typical integrative method of simUI with information contents.

In this study, we use another type of integrative methods, called simICND[13], which has a great performance on assessing functional consistency of interacting protein pairs. This is the normalized version of Resnik's method by the distance of the information content between two ontology terms (used in Jiang's method).

$$sim_{ICND}(C_1, C_2) = \frac{-\log P(C_0)}{1 - \log P(C_1) - \log P(C_2) + 2 \cdot \log P(C_0)}, \tag{2}$$

where C_1 and C_2 are the ontology terms having the annotation of interacting proteins, respectively, and C_0 represents the most specific common ontology term of C_1 and C_2. In order to have the similarity of an interacting protein pair, we need to aggregate the simICND scores between pairwise combinations of two ontology term sets having the annotation of the interacting proteins, respectively. It has been examined that the best-match average (BMA) approach[14] which takes the average of all pairwise best-matches has the best performance on aggregating term-to-term semantic similarities. We thus apply the BMA score of simICND of each PPI to the edge weight so as to build a weight PPI network.

3. Experimental Results

3.1. *Data source and experimental setting*

We tested our approach using the genome-wide PPI data set of *S. cerevisiae* from BioGRID[15]. To quantitatively evaluate the performance of our pathway discovery approach, we compare the predicted results to well-studied signaling pathways in *S. cerevisiae* such as MAPK signaling pathways. This known pathway information was extracted from KEGG[16]. Since our approach takes the given source, target and maximum path length as input parameters, we selected the proteins having transmembrane signaling receptor activities as sources and the proteins having nucleic acid binding transcription factor activities as targets. We then made all distinct combinations of each source protein and target protein, and measured the maximum length between them.

Since our approach and other previous methods generate a possibly large set of paths, we order the predicted paths by their strength and select the particular number of top ranked paths. We use **absolute ordering**, where we just order all paths decreasingly by their strength. We also deploy **path-wise ordering**, where we order paths of each source-target pair separately in a decreasing manner. We then merge them into a single path list one by one from each source-target list.

To evaluate prediction results, we used two metrics – **recall** and **precision** for nodes, edges and oriented edges, respectively. For node recall and node precision, we compare the set of proteins in predicted pathways with that in known signaling pathways. Similarly, for edge recall and edge precision, the set of undirected edges (i.e., protein pairs connected with each other) in predicted pathways is compared to that in known signaling pathways. Finally, oriented edge recall and oriented edge precision compare between two sets of directed edges in predicted pathways and known signaling pathways. Note that, in a pathway, all edges are oriented from the source towards the target. All these metrics will be computed for increasing sets of the best pathways predicted (for every 10 pathways up to 100).

Achieving high precision or recall can be done by sacrificing the other metric, thus we also present the **Receiver Operating Characteristic (ROC)** that provides a combined view on the prediction. Because of a very large number of negative examples for edges or oriented edges, we present the ROC curve for nodes only.

3.2. *Assessment of semantic similarity measures*

First, we assessed the effect of semantic similarity measures on the proposed approach. We used the PPI data set of *S. cerevisiae*, which includes 5,590 distinct proteins and 92,906 interactions. To weight each PPI, we implemented five different semantic similarities – Resnik's method, Jiang's method, simUI, simGIC, and simICND. We then used the five weighted PPI networks for pathway discovery. Since path strength is computed by the product of all edge weights on the path, each edge weight should be bounded by 0 and 1. Resnik's method and simICND, however, generate the scores between 0 and ∞. We thus apply the linear transformation of Resnik's and simICND scores. We statistically found the upper and lower bounds of the semantic similarity scores and projected them into the range between 0 and 1. All outlier values greater than the upper bound were assigned 1.

The predicted pathways by our approach using five different semantic similari-

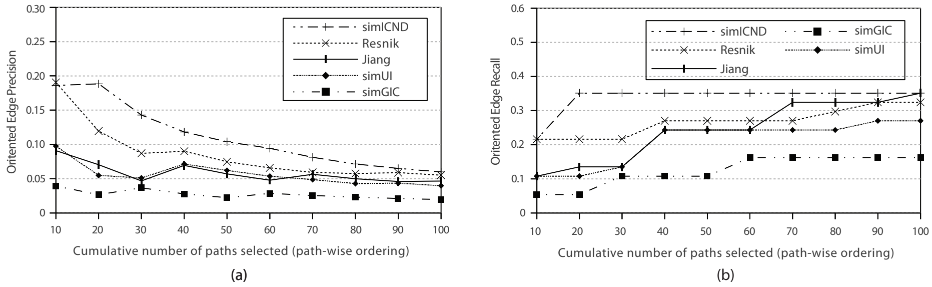


Fig. 2. Performance comparison of five semantic similarity methods. (a) Precision and (b) Recall were measured for oriented edges on up to 100 predicted paths in path-wise order using yeast PPI data weighted by five different semantic similarities.

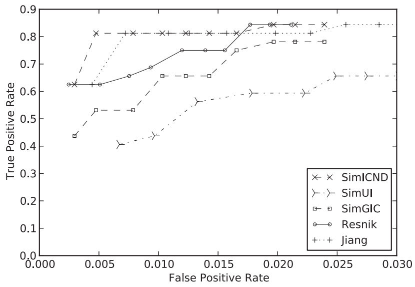


Fig. 3. Evaluation of five semantic similarity methods by ROC curves which were plotted for nodes on predicted paths in absolute order using yeast PPI data weighted by five semantic similarities.

ties were compared to MAPK signaling pathways of *S. cerevisiae* for four different functions: pheromone response, high osmolarity, filamentous growth and cell wall integrity. For both absolute and path-wise ordering, simICND reaches the best precision and best recall for nodes, edges and oriented edges likewise. Figure 2 (a) and (b) show oriented edge precision and recall, respectively, by path-wise ordering. The results demonstrate that the annotation-based semantic similarity methods, such as Resnik's, Jiang's and simICND, perform better than the other types, and simICND outperforms the other two in the same category. It confirms the advantage of combining two orthogonal approaches: Resnik's method to measure commonality between interacting proteins and Jiang's method to measure their difference. The same result is confirmed by the ROC curves in Figure 3. We will therefore use simICND to weight PPIs for all following experiments.

3.3. *Validation of pathway discovery – Local heuristics comparison*

To compare and evaluate local heuristics for candidate edge ordering, we used a particular strategy setting for edge selection (more-or-less randomly, but empirically),

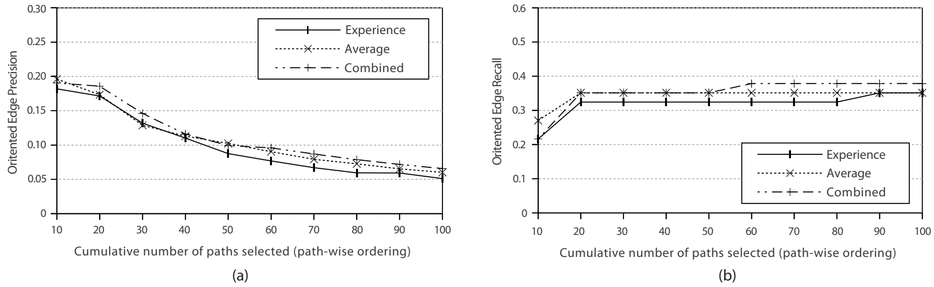


Fig. 4. Performance comparison of local heuristics. (a) Precision and (b) Recall were measured for oriented edges on up to 100 paths in path-wise order. Three different local heuristics for candidate edge ordering were compared.

and tested three different settings of local heuristics – Average heuristic, Experience heuristic and combination of both. For each combination of local heuristics settings and source-target pairs, the algorithm was launched 10 times to obtain a larger set of output paths. Figure 4 (a) and (b) show oriented edge precision and recall, respectively, of predicted pathways in path-wise order. The difference of results between two metrics was quite subtle. However, it can be observed that the combined heuristic might be considered as better than the other two, especially for recall while improving precision.

If we investigate the heuristics closely, we know that the Experience heuristic uses the average of edge weights on discovered paths as a future edge weight estimate. Since we try to select strong edges, the estimate is likely to be high as opposed to the average of all edges in the search area of Average heuristic that will probably be lower. From the definition of path strength, a higher estimate will tolerate discovery of longer paths whereas a lower estimate will force shorter paths. By intuition, the combined heuristic should provide a better result because it offers variety in length of pathways discovered, i.e., it has strength to predict both short and long pathways. We will thus use the combined heuristic for all future experiments.

3.4. *Validation of pathway discovery – Edge selection strategy comparison*

For edge selection strategy comparison, we tested five different settings – Top Random Maximum, Smooth Random Maximum, Variable Smooth Random Maximum, combination of all three strategies (Combined - All) and combination of two smooth strategies (Combined - Smooth). For Smooth Random Maximum, we used top 40% of the estimated path strength value range by the Gaussian probability distribution. For Top Random Maximum, we also used 40% of top ranked edges ordered by their estimated path strength and selected one by the discretized Gaussian probability distribution. Changing the percentage for the Smooth Random strategy does not affect the results greatly (except for extreme values) and 40% for the Top Random

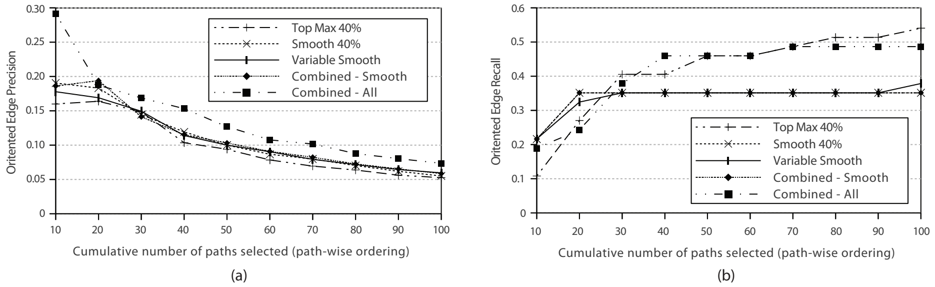


Fig. 5. Performance comparison of edge selection strategies. (a) Precision and (b) Recall were measured for oriented edges on up to 100 paths in path-wise order. Five different strategies for edge selection were compared.

strategy is based on the distribution of estimated path strength.

Figure 5 (a) and (b) show oriented edge precision and recall, respectively, of predicted paths by path-wise ordering. Results from this strategy comparison give more broad distinction between the best and worst settings than those from local heuristic comparison. Top Random Maximum by itself does not work well at all. Smooth strategies, each one separately and combined together, give very similar results. Combination of all strategies clearly outperforms the others in precision but is slightly loosing recall, especially on top 20 paths. Thus, for later experiments, we will use both strategies of Combined – Smooth (called Distance-wise Approach (1)) and Combined – All (called Distance-wise Approach (2)).

Why is precision of Top Random Maximum (40%) by itself the worst but significantly improves when cooperating? The key to this question is in the distribution of estimated path strength. Typically, among candidate edges, there are a few with very high path strength estimated, then the value drops off rapidly and majority of the candidate edges have values below 0.1. The Top Random Maximum strategy, even for a small percentage value, will include low value edges. This makes the resultant pathways more random and the output might contain less strong paths. On the other hand, semantic similarity methods do not always correctly quantify the relationship of signaling and response between two proteins, and even some edges in real signaling pathways have extremely low weights, e.g. $w(\text{MID2}, \text{RHO1}) = 0.05$ by simICND. In this case, larger randomness is essential in order to boost the algorithm forward. Combining different strategies is thus obviously efficient.

3.5. *Validation of pathway discovery – Method comparison*

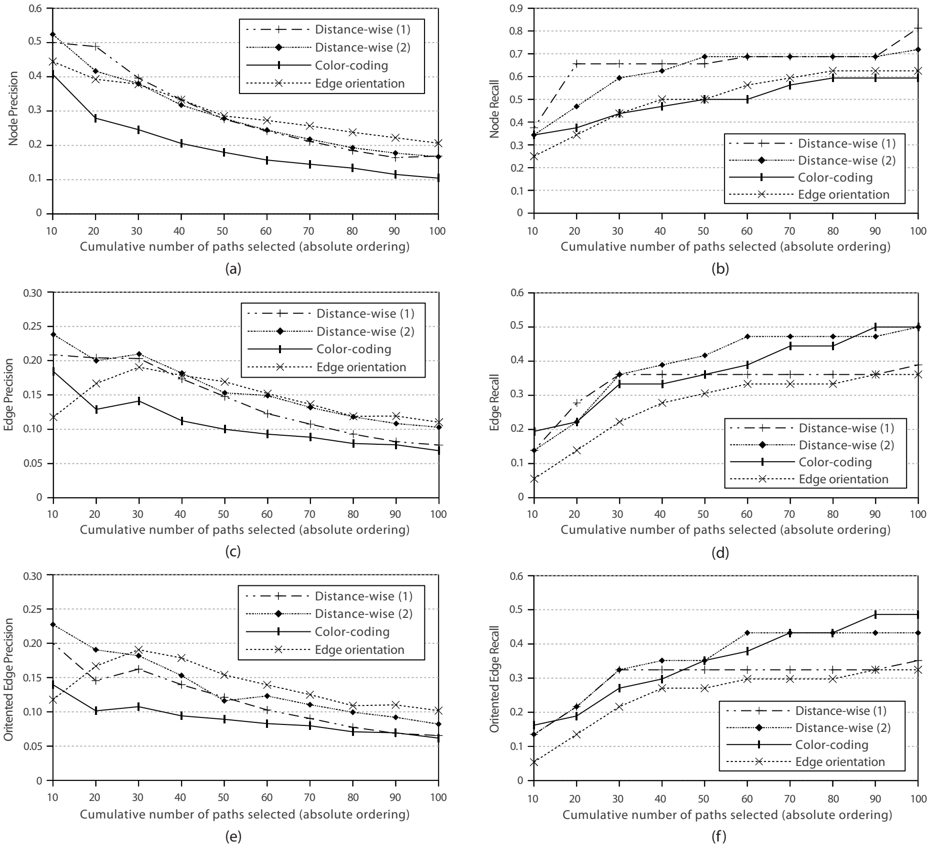
We compared the performance of our approach with that of two previous methods: the color-coding algorithm[4] and the edge orientation method[5]. The previous methods require confidence values for edges in PPI networks. In order to avoid bias to edge weighting and to make a fair comparison, we use identical semantic similarity values (i.e., simICND) of PPIs for all previous methods compared.

Both time and space complexity of the edge orientation method is extremely large, significantly increasing with the maximum path length. We were not able to run the edge orientation algorithm for the paths of length greater than 5 with effective 12GB memory dedicated to JVM (on 16GB up-to-date machine). This method ran out of heap space even for single random orientation to predict long pathways. Thus, to predict significantly short paths, it easily achieves high precision (less opportunity to make mistakes) but hardly gets the maximum recall (less opportunity to cover all). To represent the best results achieved by the edge orientation method, we limit the maximum path length by 5 across all MAPK signaling pathways. A significant drawback of the color-coding algorithm is that particular colorings of a graph can prevent certain paths to be found. To avoid this, we ran the color-coding algorithm 100 times, each time with different random graph coloring and combined all the discovered paths. Of course, the space of paths that can be found always compromises with running time of the algorithm.

Figure 6 depicts precision and recall of the strongest paths predicted by the color-coding algorithm, edge orientation method with maximum length 5 on the entire list of MAPK signaling pathways, and our distance-wise approach using both setups of combining all smooth strategies (distance-wise (1)) and combining all strategies (distance-wise (2)). For this comparison, we were forced to use absolute ordering, since the Edge Orientation algorithm presents more source-target pairs as viable results, selecting only paths from real pathways by path-wise ordering would give the Edge Orientation algorithm unfair advantage by discarding some of its results. When comparing to the color-coding algorithm, both strategy settings of our distance-wise approach have better precision and comparable recall for edges and oriented edges, whereas our method is always better for node precision and recall. In particular, when we focus on the top ranked paths, e.g. top 30 paths generated, our distance-wise approach achieves significantly better results on all evaluation metrics than the previous methods. Note that the generated paths in this test were listed by absolute ordering. That is why the oriented edge precision and recall plots of our approach in Figure 6 are slightly different from those in Figure 5. Overall, the proposed approach outperforms the competing computational methods of pathway discovery. We also present the ROC curve for this comparison in Figure 7 (for nodes only, using absolute ordering). Judging by the area under the curve (AUC) analysis, both setups of our distance-wise algorithm significantly outperform the previous methods.

4. Conclusion

We proposed a novel distance-wise computational approach of pathway discovery from weighted PPI networks. In order to find optimal settings, we took into consideration three factors: (1) semantic similarity measures for edge weights, (2) local heuristics for candidate edge ordering, and (3) strategies for edge selection. For each factor, we tested several different options. Among semantic similarity measures, we



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 6. Performance comparison of pathway prediction methods. (a) Node precision, (b) node recall, (c) edge precision, (d) edge recall, (e) oriented edge precision, and (f) oriented edge recall on up to 100 paths in absolute order generated by two different settings of our distances-wise algorithm ((1) Combined - All Smooth, (2) Combined - All), the color-coding algorithm and the edge orientation method. For the edge orientation method, we used maximum path length of 5 for all reference MAPK signaling pathways.

concluded that a combined measure, simICND, is superb beyond question. For local heuristics and edge selection strategies, we also argued that the combinations of proposed options yield the best results. We also compared the proposed algorithm to two previous competing methods: the color-coding algorithm and the edge orientation method. Not only that our approach is unbeatable in case of time and memory requirements whereas runtime of the color-coding algorithm is acceptable and that of the edge orientation method even cannot be fully run, our approach also has the best precision and recall when predicting known pathways, i.e. well-studied MAPK signaling pathways, of *S. cerevisiae.*

Among the three factors tested in this experiment, selecting a semantic similarity measure was the most sensitive to the pathway prediction accuracy. For calculating

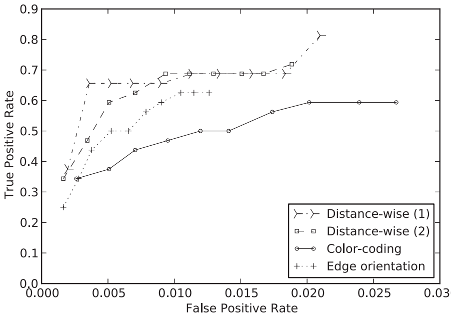


Fig. 7. Evaluation of three pathway prediction methods by ROC curves. The ROC curves were plotted for nodes on predicted paths in absolute order, generated by two different settings of our distances-wise algorithm ((1) Combined - All Smooth, (2) Combined - All), the color-coding algorithm and the edge orientation method.

semantic similarity scores as PPI weights, we use GO annotation data which already includes the specific information of cell signaling. Therefore, the results from the proposed approach might be over-optimistic. For example, to predict MAPK signaling pathways, its result can be biased towards the inclusion of proteins which are already annotated to MAPK in GO. It thus might lead to high recall. In contrast, predicting the pathways that are not annotated to any GO terms might result in lower accuracy.

Acknowledgments

This work was supported in part by funds from the Baylor University Research Committee and the Vice Provost for Research.

References

1. Papin JA, Hunter T, Palsson BO and Subramaniam S, Reconstruction of cellular signaling networks and analysis of their properties, *Nature Reviews: Molecular Cell Biology* **6**:99–111, 2005.
2. Hyduke DR and Palsson B, Towards genome-scale signalling-network reconstructions, *Nature Reviews: Genetics* **11**:297-307, 2010.
3. Yu H, *et al.*, High-quality binary protein interaction map of the yeast interactome network, *Science* **322**:104-110, 2008.
4. Scott J, Ideker T, Karp RM and Sharan R, Efficient algorithms for detecting signaling pathways in protein interaction networks, *Journal of Computational Biology* **13**(2):133-144, 2006.
5. Gitter A, Klein-Seetharaman J, Gupta A and Bar-Joseph Z, Discovering pathways by orienting edges in protein interaction networks, *Nucleic Acids Research* **39**(4):e22, 2011.
6. Jain S and Bader GD, An improved method for scoring protein-protein interactions using semantic similarity within the gene ontology, *BMC Bioinformatics* **11**:562, 2010.
7. Lord PW, Stevens RD, Brass A and Goble CA, Investigating semantic similarity mea-

sures across the Gene Ontology: the relationship between sequence and annotation, *Bioinformatics* **19**(10):1275-1283, 2003.

8. The Gene Ontology Consortium, The Gene Ontology: enhancements for 2011, *Nucleic Acids Research* **40**:D559-D564, 2012.
9. Guzzi PH, Mina M, Guerra C and Cannataro M, Semantic similarity analysis of protein data: assessment with biological features and issues, *Briefings in Bioinformatics* **13**(5):569-585, 2012.
10. Wang J, Zhou X, Zhu J, Zhou C and Guo Z, Revealing and avoiding bias in semantic similarity scores for protein pairs, *BMC Bioinformatics* **11**:290, 2010.
11. Guo X, Liu R, Shriver CD, Hu H and Liebman MN, Assessing semantic similarity measures for the characterization of human regulatory pathways, *Bioinformatics* **12**(8):967-973, 2006.
12. Pesquita C, Faria D, Bastos H, Ferreira AEN, Falcao AO and Couto FM, Metrics for GO based protein semantic similarity: a systematic evaluation, *BMC Bioinformatics* **9**(Suppl 5):54, 2008.
13. Cho, Y.-R., Mina, M., Lu, Y., Kwon, N. and Guzzi, P.H, M-Finder: Uncovering functionally associated proteins from interactome data integrated with GO annotation, *Proteome Science* **11**(Suppl 1), 2013.
14. Tao Y, Sam L, Li J, Friedman C and Lussier YA, Information theory applied to the sparse gene ontology annotation network to predict novel gene function, *Bioinformatics* **23**:i529-i538, 2007.
15. Stark C, *et al.*, The BioGRID interaction database: 2011 update, *Nucleic Acids Research* **39**:D698-D704, 2011.
16. Kanehisa M, Goto S, Sato Y, Furumichi M and Tanabe M, KEGG for integration and interpretation of large-scale molecular data sets, *Nucleic Acids Research* **40**:D109-D114, 2012.