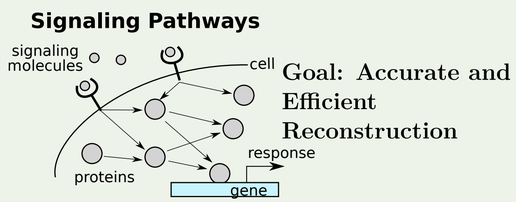
Reconstructing Signaling Pathways with DAGs

Jiarong Li[1], Tunc Basar Kose[2], Ibrahim Youssef[2] and Anna Ritz[2]

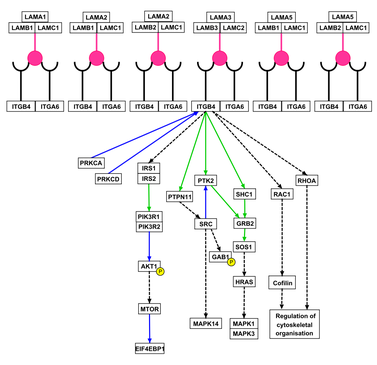
[1]Mathematics & [2]Biology Departments, Reed College

Motivation

- Signaling pathway reconstruction is useful.
- Available methods suffer from low recall or stalling new information addition.

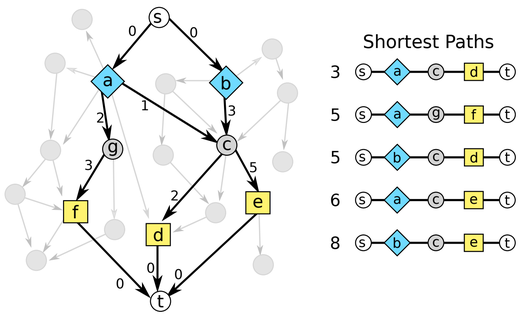


Example NetPath Pathway (Alpha6Beta4 Integrin)[1]



Previous Work

PathLinker uses a k-shortest paths method to connect receptors and transcription factors; however, the resulting network reuses a lot of edges and nodes as k increases.[2]



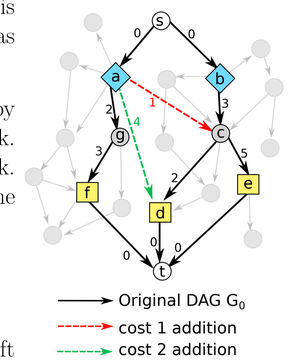
Proposed Solution: Make the addition of new information a requirement at every iteration.

Method and Algorithms

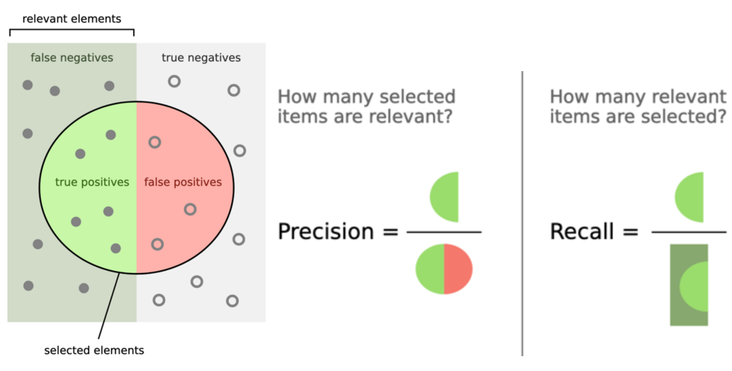
Directed Acyclic Graphs (DAGs) are graphs without cycles that ensure that every node is on a path from a super-source **s** to a super-sink **t** and allow efficiency for using algorithms such as Dijkstra's or topologically sorting the nodes.

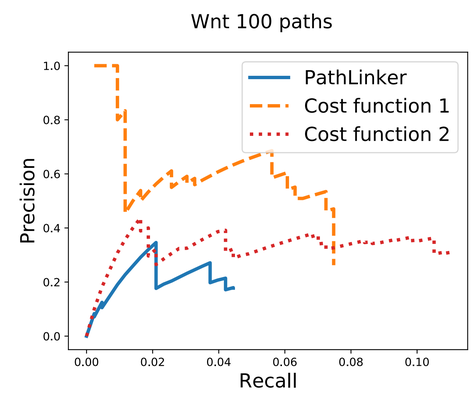
Starting with some ground-truth data, for example an annotated path, we extend the network by finding shortest paths between nodes already in the network that traverses outside the network. At every iteration, the shortest path that minimizes some cost function is added to the network. So far we have tested two main cost functions, with their differences illustrated in the figure on the right:

1. $c_1(G_j) = \sum_{(u,v)\in E_j} w_{uv}$. The sum of the cost of all edges in the network.
2. $c_2(G_j, s, t) = \sum_{p\in P_j(s,t)} \sum_{(u,v)\in p} w_{uv}$. The sum of the cost of all paths in the network.



In order to compare the performances, we used **Precision and Recall curves** [3]. The left figure below explains precision and recall while the one on the right shows the precision and recall curves of the results compared to PathLinker. In our case, true positives are interactions or nodes from the pathway of interest that are added to the network. False positives are interactions or nodes added to the network that are not in the pathway. The curves are created by calculating precision and recall after each step where a new path was added to the network.

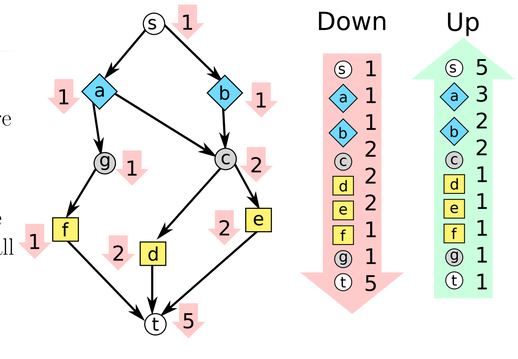




Computing the Cost of All Paths

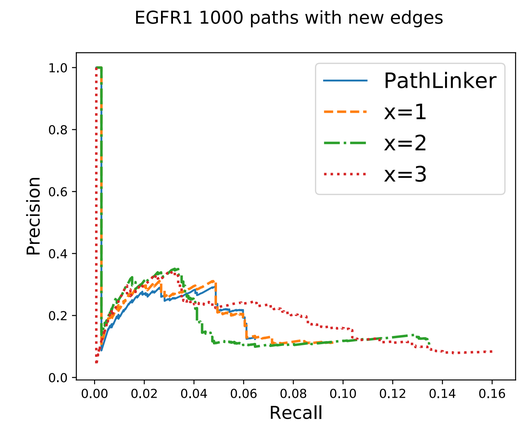
Enumerating all paths in a big network can be costly, instead we:

1. Count down from **s** how many different ways a node can be reached (figure to the right).
2. Count up from **t** how many different ways a node can be reached.
3. For each edge *(u, v)*, $down[u] \times up[v]$ gives the number of paths the edge appears in. Multiplied with edge weight gives the edges total cost across all paths
4. Sum those values for all edges for the total cost of all paths.



Combining PathLinker and DAG

In order to show that constant addition of new information as a requirement is an improving factor, we modified PathLinker to add paths only if they contain x number of new edges. We observed that for $x = 1, 2, 3$, the recall increases faster as x increases with all having precision comparable to PathLinker.



Ongoing Work

- **Developing a better cost function.**
  - Consistent with pathway structure.
- **Using a more comprehensive model**
  - Take into account feedback loops.
- **Expansion instead of reconstruction.**
  - Supply entire annotated pathway to find unannotated proteins and interactions.

References

[1] Alpha6Beta4 Integrin Pathway. NetPath

[2] Anna Ritz et al. Pathways on demand: automated reconstruction of human signaling networks. npj Systems Biology and Applications, 2:16002, 2016.

[3] By Walber - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=36926283

Contact Information

Email: aritz@reed.edu