A single source k-shortest paths algorithm to infer regulatory pathways in a gene network

Ekaterina Starostina

30.03.2013

Gene interaction network

- **Node** - gene or its corresponding protein
- **Edge** - protein–protein interaction (PPI) or a transcription factor (TF)–DNA binding
- **Regulatory pathway** - chain of interacting genes within a network. A regulatory pathway begins with a causal gene and ends at a target gene
- **Goal** - identify the potential regulatory pathways passing through the given gene in the gene network

Existing approach

- **Random walk.** A random walk typically starts from the given gene, walks through several nodes, and terminates according to some pre-defined parameters (such as length and edge weights).
- **k shortest paths algorithm** Executes O(n) times Dijkstra algorithm to generate candidate paths for each of the k shortest paths. Time complexity - $O(kn(m + nlogn))$, where n is the number of nodes and m is the number of edges.

Problem definition

Let $G = (V, E)$ denote a gene network, which is a weighted directed graph, where

- V is the set of nodes (genes or proteins),
- E is the set of directed edges (interactions)
- n = |V|, m = |E|.

Each edge $e \in E$ denoted by (g_a, g_b), $g_a, g_b \in V$, represents the direction of interaction from g_a to g_b.
$w(e) \in [0, 1]$ is the weight of an edge e, and the weight represents the confidence level of the interaction.

Sub problems

- UnknownCausal
  infer possible causal genes if the given gene is a target gene
- UnknownTarget
  infer possible target genes if the given gene is a causal gene
- CandidateCausal
  infer the true causal gene in a given set of candidate causal genes, if the given gene is a target gene

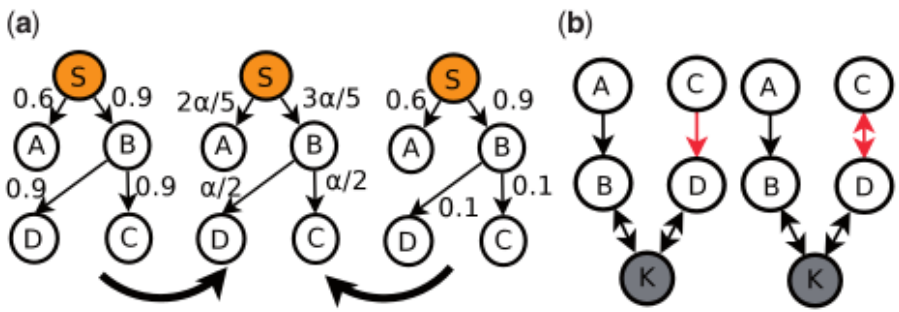
Random walk model

- P is $n \times n$ transition matrix constructed by normalizing the adjacent matrix A of the graph G, where
  - $A_{ij} = w((gi, gj))$
  - $P_{ij} = \alpha A_{ij} / \sum_k A_{ik},\ \alpha \in (0, 1)$ is called 'damping factor'.
- If the current node is g_i, the walk would terminate at g_i with probability $1 - \alpha$ and go to another node g_j with probability P_{ij}
- A random walk starts from a source node (causal gene) and once the walk reaches a sink node(target gene), the walk immediately terminates.

Problems of random walk model

(a) Normalization of edge weights is lossy

(b) The walks repeatedly go through the bidirectional edge



Single-source k-shortest paths

First we convert the weight of an edge into a distance as follows:

$$d(e) = -\log(w(e)) + c \tag{1}$$

, where d(∗) denotes the distance of an edge or a path.
Then, the importance value of a gene g_a is defined as

$$Imp(g_a) = \sum_{i=1}^{k} 1/d(P_i) \tag{2}$$

,where P1 ,P2 ,...,Pk are k-shortest paths from the given gene to g_a
With this transformation, it is obvious that genes having shorter distances to the given gene of interest will be assigned greater importance values.

Sub problems

The inference problem is now a single-source k-shortest paths problem, defined for subroblems as it follows:

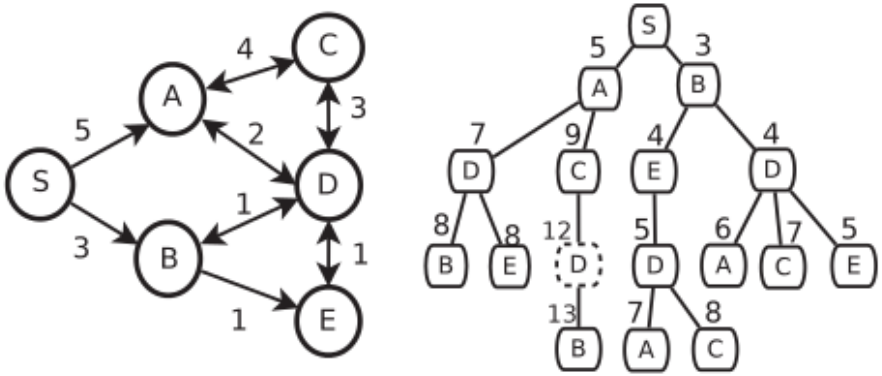
- UnknownTarget - starting node is the given causal gene
- UnknownCasual - starting node is the given target gene, the direction of all edges should be reversed first
- CandidateCasual - starting node is the given target gene, the direction of all edges should be reversed first and we select the candidate causal gene with the highest importance value as our predicted causal gene

Definitions

Def. 1 Given all top k-shortest paths from the starting node to each other node, a **pseudo tree** stores all paths in a tree structure. If $k = 1$, the pseudo tree is equivalent to the shortest path tree; as $k > 1$, all 2nd to k-th shortest paths are iteratively merged into the pseudo tree by sharing the longest common prefix path.

Def. 2 A **tree-path** is a path from the root to another node in a pseudo tree.

Def. 3 **Path nodes** and **dummy nodes**. A tree-path to a path node is a top-k shortest path from the root to this path node, while a tree-path to a dummy node is not.



Theorems

T. 1 If a top k-shortest path from the root S to a node A contains a dummy node B, let the sub-path from the dummy node to A be PBA , then for each top k-shortest path from S to B, denoted by PSB , either PSB contains a node in PBA besides B or there exists a top k-shortest path from S to A using PSB as a prefix path.

T. 2 Given a graph G, the pseudo tree for the k-shortest paths contains at most $O(n^2k)$ nodes.

T. 3 The time complexity of Algorithm 1 is

$$(nklog(nk) + mk(h + k)) \tag{3}$$

Algorithm

Algorithm 1 k-shortest paths algorithm

Input: A weighted graph $G=(V,E)$, the starting node S, and k.

Output: A pseudo tree T representing all top k-shortest paths containing only path nodes, and arrays of distances Arr, where $Arr(N_i)[j]$ storing the distance of the $(j+1)$-th shortest path from S to N_i.

1. For each node $N_i \in V$, assign an array $Arr(N_i)$ that consists k values. All values are initialized to ∞.
2. count$(N_i) \leftarrow 0$ for each node $N_i \in V$
3. Put the root $<S,0>$ in T.
4. For all edges $e=(S,N_x) \in E$, put $<S,e,d(e)>$ in a priority-queue, pq, and count$(N_x) \leftarrow 1$. //pq is a min priority queue.
5. **while** pq is not empty **do**
6. $<N'_a,(N_a,N_b),dis> \leftarrow$ pop-min(pq)
7. concatenate $<(N_a,N_b),dis>$ to N'_a in T. //add a new path node N'_b and an edge (N'_a,N'_b) to T.
8. **for all** $e=(N_b,N_c) \in E$: $dis+d(e)<Arr(N_c)[k]$ and N_c is not in this tree-path $S \Rightarrow N'_a$ **do**
9. **if** count$(N_c)<k$ **then**
10. count$(N_c) \leftarrow$ count$(N_c)+1$
11. put $<N'_b,e,dis+d(e)>$ in pq
12. **else**
13. update the corresponding entry of $Arr(N_C)[k]$ in pq to $<N'_b,e,dis+d(e)>$.
14. $Arr(N_C)[k] \leftarrow dis+d(e)$ and sort $Arr(N_c)$ //only need to move $Arr(N_C)[k]$.

Diversity of a path

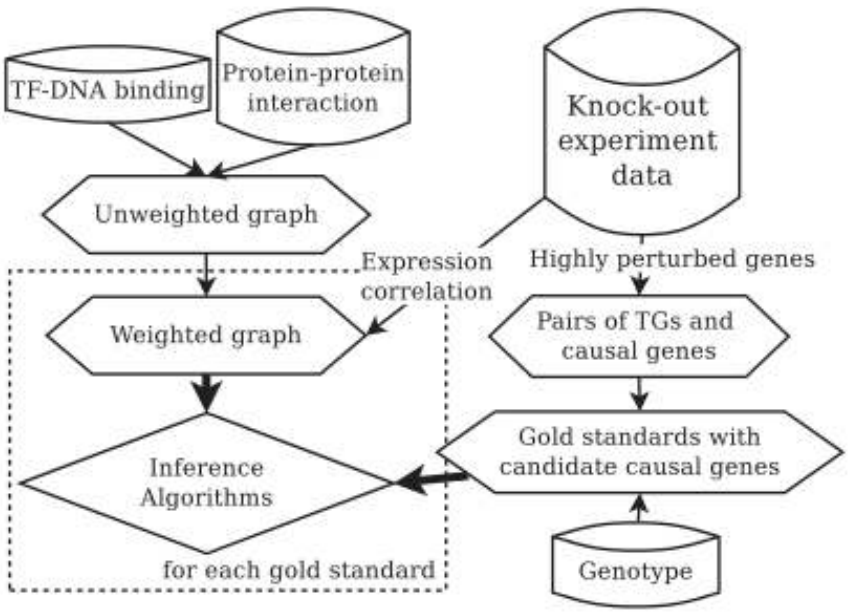
Diversity of a path is the number of edges in this path but not appearing in any already found paths divided by the total number of edges in the path.
If diverse paths P1 ,P2 ,...Pκ, κ<k, are already found, the diversity of a new candidate path Pnew is:

$$div(P_{new}) = \frac{|\{e|e \in P_{new}, \nexists P_i : e \in P_i, 1 \leq i \leq k\}|}{|\{e|e \in P_{new}\}|} \quad (4)$$

Single-source k-shortest diverse paths algorithm

1. Execute Algorithm 1
2. For each node, examine the diversities of k shortest paths in the increasing order of their distances
3. If the diverse paths are not enough ($<$ k) for some nodes, remove edges from the graph according to the probability
4. Repeat the procedure until k diverse paths are found for every node or less than m/n edges are removed in the last iteration

Overview of the inference framework



Testing aspects

- **CandidateCausial** The goal is to correctly pick the true causal gene from the ten candidate causal genes.
- **UnknownCasual and UnknownTarget** SaddleSam was used to evaluate the enrichment levels of the GO terms for the results of UnknownCausal and UnknownTarget of a gene set V.
- **Diversity, importance value and efficiency**
- **Diversity and enriched functions** The goal is to show that using different λ, the importance values generated by the k diverse paths algorithm can identify different potential regulatory pathways with different functions.

Conclusions

- Algorithm can identify pathways with higher potentiality than current methods based on the random walk model, and requiring the paths to be more diverse can further uncover other potential regulated functions.
- Heuristic algorithm can achieve a huge speedup than the previous single-pair shortest paths algorithm while the found paths are equivalent in our yeast gene network.