Function-oriented edge probability assignment for signaling networks

CIS 6930 project report

Haitham Gabr

Abstract

A major question in the field of probabilistic biological networks is: how do we get the correct values for edge probabilities. Most available methods are ad hoc, relying on aggregation of available data about the edge, like gene expression correlation, evidence of interaction and confidence in the experiments producing this interaction. We aim at developing a method for assigning edge probability values for a given signaling network such that the network utility is maximal. Research shows that gene expression correlation is tightly related to signaling [1:5], where a signaling pathway is more probable to exist between to genes if their expression correlation is high. Following from this, we develop a method that assigns edge probability values such that signal reachability between receptor and reporter nodes is closest to their gene expression correlation. We present experimental results on H. sapiens signaling networks from KEGG.

1. Problem statement

Given

- A directed network $G = (V, E)$.
- Set of source and target nodes $S, T \subset V$.
- Function $C(s, t)$ returns a normalized correlation value of expression between $s \in S$, $t \in T$.

Find

- Function $P(e \in E)$, such that $\|C - R(P)\|$ is minimum, where:
  - C is a vector of $C(s, t)$ values $\forall\ s \in S$ & $t \in T$,
  - $R(P)$ is a vector of reachability probability $\forall\ s \in S$ & $t \in T$, which is calculated based on P [6].

2. Method

Let $n = |E|$. Our method considers the n-dimensional search space of all possible probability assignments. It aims at converging to a point where ‖C – R‖ becomes minimal. We proceed in two phases. The first phase is a genetic algorithm which yields a starting point that is at least as good as a random one. The second phase is a hill climbing phase where we locally optimize the given starting point. Next we describe both phases in detail.

2.1 Phase I: genetic algorithm

Like any genetic algorithm, ours is defined by two main attributes:

- Genetic representation: an *n*-length vector *P* of probabilities for all edges in any given order.
- Fitness function:

$$1 - \frac{\| C - R(P) \|}{|S| \times |T|}$$

This phase starts by generating 50 random *P* vectors. It then proceeds in the following steps:

1. Crossover: We select two vectors P_1 and P_2 at random, where each vector has a selection chance weighted by its fitness. We cross them over to produce a new *P'* as follows. For $1 < i < n$, we select either P_{1i} or P_{2i} to be P'_i. If $R(P_1)$ and $R(P_2)$ are both larger than *C*, we select the smaller of them, trying to bring the resulting $R(P')$ closer to *C*. Inversely, if $R(P_1)$ and $R(P_2)$ are both smaller than *C*, we select the larger of them. If one is larger than *C* and the other is smaller than *C*, we select one of them based on a coin toss weighted by their fitness. We repeat this process 50 times until we have a total of a 100 *P* vectors.
2. Mutation: For every *P* vector, for $1 < i < n$, we perform a Bernoulli trial with a success probability of 0.01. If the trial is successful, we reset the value of P_i to a new value selected at random between 0 and 1.
3. Selection: We select the top 10 *P* vectors according to their fitness. Then we select another 40 *P* vectors at random, where each vector has a selection chance weighted by its fitness. The total selected now is 50 *P* vectors.
4. We repeat steps 1 through 3 for 100 times. At the end, we select the *P* vector with the highest fitness as input for the next phase.

2.2 Phase II: hill climbing

Here we seek to optimize the *P* vector obtained from the genetic phase. We do this by updating one of its elements at a time, until no further improvement can be done.

1. For $1 < i < n$
    i. We set P_i as an unknown.
    ii. We keep all values of other elements in P as they stand now.
    iii. We compute $R(P)$ in terms of P_i.
    iv. We solve for P_i:

$$\frac{d}{dp} \parallel \mathrm{C} - \mathrm{R(P)} \parallel \; = \; 0.$$

    v. We assign this value as the new value of P_i.
2. We repeat step 1 until all the values in P do not change.

We produce the result from step 2 as the output of the method.

3. Results

We did our experiments using some H. sapiens signaling networks from KEGG. The gene expression data that we used is randomly synthesized. However, this method is projected to be experimented using real gene expression data from different leukemia subtypes. The following table lists the tested networks and their attributes.

Network	# Nodes	# Edges	# Sources	# Targets
Wnt	56	65	13	13
ErbB	46	55	10	17
MAPK	117	165	18	19

3.1 Preliminary performance evaluation

Using the random synthetic gene expression values, we were able to test the performance of our method. Processing of ErbB and Wnt each takes less than a 100 seconds. However, when trying to process MAPK, the first genetic round of phase I didn't finish after 10 minutes. When running without phase I, also the first hill climbing round from phase II didn't finish after 10 minutes.

3.2 Preliminary observations

Looking at the probability values obtained by running our method on ErbB network using synthetic random gene expression data, we made some observations. First, we observe that there exist multiple solutions all with the same quality. This means that we will need to decide on a strategy for choosing one of several solutions when there is a tie. We also observed that the quality of these solutions is high (> 96% fitness). We tried removing phase I from the

method and observed that the quality of the result does not change. This is expected given the fact that there exist many solutions, so even when the starting point is random, we can reach to a high-quality local optimum that is close to it.

4. Conclusion and future work

We presented a method for assigning edge probability in signaling networks, based on maximizing the network utility. We achieve this by finding edge probability values that bring signal reachability as close as possible to gene expression correlation between receptor and reporter genes.

The next step is to prepare real gene expression correlation values from real datasets of leukemia subtypes. Running this method in these different phenotypes and observing the results can uncover some interesting conclusions about differences in signaling between these subtypes. Also we plan to compare this method with a leading method [7] that computes edge probability values using logistic regression of gene co-expression, available evidence, and small world characteristics.

References

[1] Li K-C (2002) Genome-wide coexpression dynamics: Theory and application. Proceedings of the National Academy of Sciences 99: 16875–16880.

[2] Horvath S, Dong J (2008) Geometric Interpretation of Gene Coexpression Network Analysis. PLoS Comput Biol 4: e1000117.

[3] D'A¨ oˆhaeseleer P, Liang S, Somogyi R (2000) Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics 16: 707–726.

[4] Novak BA, Jain AN (2006) Pathway recognition and augmentation by computational analysis of microarray expression data. Bioinformatics 22: 233–241.

[5] Allocco DJ, Kohane IS, Butte AJ (2004) Quantifying the relationship between co-expression, co-regulation and gene function. BMC bioinformatics 5: 18.

[6] H. Gabr, A. Todor, H. Zandi, A. Dobra, and T. Kahveci. PReach: Reachability in Probabilistic Signaling Networks. ACM-BCB, 2013.

[7] R. Sharan, S. Suthram, et al. Conserved patterns of protein interaction in multiple species. PNAS, 2002.