Identifying Pathway Modules from High Throughput Genetic Interaction Data using Simulated Annealing

Jeremy Colebrook-Soucie, Quinn Collins

December 12, 2016

1 Abstract

2 Introduction

In this paper, we explore an alternative method of detecting patterns from high throughput genetic interaction data. This work has been done as an extension of the genecentric package and the method defined in Leiserson *et al.* The method that we explore simulated annealing. We also touch upon subsequent verification techniques.

High throughput genetic interaction data can be represented as a weighted, undirected graph. Each dd

The patterns that we attempted to detect are called BPMs, standing for between pathway model. These are defined by two sets of genes. Each set of genes should have predominantly positive edges within itself and predominantly negative edges with the other set.

High throughput genetic interaction

The recent rise of high throughput genetic interaction data What problem are we trying to solve? What is it an extension of problem domain talk about genecentric

3 Data

4 Methods

4.1 Simulated Annealing

Simulated annealing is a hueristic that is used for solving optimization problems with a very large sample space. Its function is analogous to the physical process of annealing, where a metal is heated and then slowly cooled in order to remove imperfections. When the metal is hot, the smith can safely make large changes to it. However, as it colder, he must work it more slowly into its final shape.

Let us define the following terminology.

Definition 1 Objective Function *The objective function is a function from an element in the sample space to the real numbers. Simulated annealing attempts to minimize this function.*

Definition 2 Permutation *In the context of simulated annealing, a permutation is how one gets from one valid point in the sample space to another. In general, such permutations are simple, random changes in a solution.*

Definition 3 Temperature *The temperature of simulated annealing defines how willing it is to accept worse solutions. When temperature is high, simulated annealing has a large probability of accepting a solution that is worse than the previous candidate. However, when the temperature is low, simulated annealing will accept worse solutions with lower and lower probabilties.*

Simulated annealing works by exploring the sample space, with the end goal of minimizing the objective function of the final solution. It begins with a very high temperature, which exponentially decays over a fixed number of total iterations. At every iteration, it generates a new solution based on the previous solution. It then evaluates this new solution and then decides whether or not to accept it based on how it compares to the previous solution and the current temperature.

Thus, in order to apply simulated annealing to the problem domain of extracting BPMs from a genetic interaction network, we must define the sample space, and how to both permuate and evaluate a point in the sample space.

4.2 Sample Space

We defined the sample space as a partition of the nodes in the genetic interation network. Each partition represents a single BPM. Each BPM is in turn divided into two modules.

This is a notable divergence from the work of Leiserson *et al.* In this work, genes are allowed to be placed in multiple BPM's at once. We opted not to include this feature because we believed it would have greatly excessively expanded the sample space and would have exacerabated some on the problems inherent to our approach. An unfortunate of this is that it made comparing our work to previous work a little bit harder.

4.3 Objective Function

We defined our objective to be the sum of a BPM objective function times a scaling factor across all BPMs.

Define this BPM function as:

$$BpmObj(x) = \sum_{a \in M_x} \sum_{b \in M_x} e_{a,b} + \sum_{a \in N_x} \sum_{b \in N_x} e_{a,b} - \sum_{a \in M_x} \sum_{b \in N_x} e_{a,b} \tag{1}$$

M_x and N_x are the two submodules defining BPM x. $e_{a,b}$ is the weight of the edge between a and b. If an edge does not exist, then it is 0.

Define the scaling factor as:

$$scale(i) = \frac{e^{-\frac{(i-\mu)^2}{2*\sigma^2}}}{\sqrt{2*\pi}*\sigma}$$

We experimented with multiple σ and μ values, but eventually settled on $\mu = 30$ and $\sigma = 5$.

Finally, define our objective function as:

$$obj(Bpms) = \sum_{x \in Bpms} BpmObj(x) * scale(|M_x| + |N_X|)$$

We chose the normal distribution as the scaling factor because it is an easily understandable and modifiable bell curve. Such a scaling factor was necessary because all the other objective functions that we tried led to the creation of many modules with a size less than 3 or greater than 25, which were subsequently discarded when pruning results.

We completely recognize that using a scaling factor may have caused some problems. We will address these in the discussion section.

4.4 Permutation Behavior

The procedure for permuting a set of BPMs is as follows.

Begin by selecting a random BPM. If that BPM has a module that has a size greater than 25, perform a split. That is, split both modules in have and create two new BPMs created from these two splits. Remove the original BPM.

Otherwise, randomly pick one of the two modules and remove a random gene from that module. If removing this gene empties the BPM (as in, both modules are empty), delete the BPM. Next, pick another random BPM and a random module from within it. Add this gene to that module.

5 Convergence

How we converge How genecentric converges Quinns pretty graph Wow we're bad probably Is this ia good metric? Still need to decide

6 Comparison to Genecentric

It is not entirely clear how compare genecentric to simulated annealing. This is because genecentric allows for genes to be in mulitple modules, obviously results in more, higher quality modules than simulated annealing. Ultimately, we decided to lean on percent of BPMs enriched and percent of BPMs enriched for same function to standardize these measurements.

This line of thought led us to the conclusion that we would need a random group, where BPMs were randomly generated, to compare our results to. Each random group contained about 22 BPMs where each module contained 15 random genes. The average over many random trials would form a baseline. In the end, we took the average over FIXME FIXME FIXME.

Additionally, to correct for the apparent lack of convergence in the simulated annealing method, we took the average results over 55 trials.

In order to compare these different methods, we examined several different benchmarks. Modules found is the number of modules found between size 3 and 25. Modules enriched is the number of modules enriched for at least one GO term at consider $p \leq 0.05$. BPMs Enriched for Same Function is the number of BPMs where both modules had at least one shared GO term.

Method	Modules Found	Modules Enriched	BPMs Enriched for Same Function
Genecentric	66	62 (94%)	27 (41%)
Simulated Anneal Average	45.6	27.1 (59.7%)	1.4 (0.031%)
Random Partition Average	44.0	26.47 (64.7%)	1.5 (0.037%)

This data is particuarly telling. As anticipated, genecentric performed very well. Consequently, our method certainly did not, performing at the same level as completely random selection. This will be explored further in the discussion section.

7 Discussion

Postmordem about why SA bad in this domain