MVGA: Method for Simultaneous Identification of Multiple Driver Cancer Pathways

Bowen Deng 邓博文 1000010186

摘要

癌症治疗的一个重要挑战是从大量非关键性的基因中提取出那些对于癌症传播有推进作用的关键性基因。我们在关键癌信号通路层面而非单个基因层面来研究。在本文中，我们提出了多值遗传算法以解决一个对信号通路打分函数的最大化问题。该打分是基于Vandin提出的De novo打分法的多条信号通路推广。我们还提出了自动选择算法参数的方法。我们在模拟数据上测试了多值遗传算法和自动选择参数的方法来说明其功效。文中提出的算法非常容易推广并且可以使用到De novo打分法以外的各种类似方法。

Abstract

A major challenge for cancer treatment is to identify driver mutation genes that promote cancer proliferation, and to extract them from huge amount of passenger genes. We consider the problem on pathway level rather than single gene level. In our study, we propose an algorithm called Multiple Value Genetic Algorithm to maximize an innovative scoring method for selected gene pathways. The scoring is the multiple pathway version for the

De novo scoring proposed by Vandin et al.. We also suggest an automatic decision of parameters in the algorithm. We tested MVGA and automatic parameter selection with simulation data to show its efficiency. Our method is flexible and could be applied to various scoring strategy similar to the De novo scoring.

Contents

1 INTRODUCTION

Genetic mutations are long known to be related with cancer. Yet, only a small fraction of whole genomes are significant. The advent of high-throughput technologies enables scientists to obtain huge amount of genetic mutation data easily. Yet, analyzing those high dimensional data remains a challenge.

At first, people focused on single gene mutations that might cause cancer. The chronic myeloid leukemia (CML) was cured by designing drugs for specific gene considered to be cancer-related.

However, the complexity of cancer proliferation is later realized: Cancer does not attack certain genes, instead, they attack gene networks called pathways[1]. Therefore, we have to study cancer related gene mutation on a pathway level rather than on genomic level. Identifying gene pathway is by no means easy, taking into account the huge amount of possible combinations of genes. Even the collection of three genes combinations has a size of 10^9, making it impossible to calculate within tolerable time. To cope with the huge computational cost, people imposed two assumptions that make the problem well-posed. The first assumption is high coverage: most patients have at least on mutation in one pathway[2]. Otherwise, the pathway detected is not significant. Another assumption is high exclusivity: most patients have no more than one mutations in one pathway.

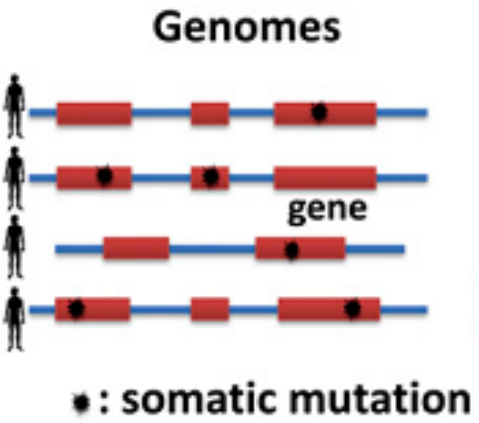

Figure 1: High Coverage: almost every patients have mutation in one of the genes in the pathway. High exclusivity: most patients have no more than one mutations in the mutated driver pathway.

We can view tumorigenesis as an annually accumulated Darwinian evolutionary process. As shown in figure 1, it is expensive and redundant for the cancer to attack two genes in the same network. Consequently, the cancer would prefer to attack genes on different pathways. This assumption is supported by numerous examples of mutually exclusive driver mutations including: EGFR, KRAS in lung cancer; TP53, MDM2 in glioblastoma; KRAS, PTEN in endometrial and skin cancers.

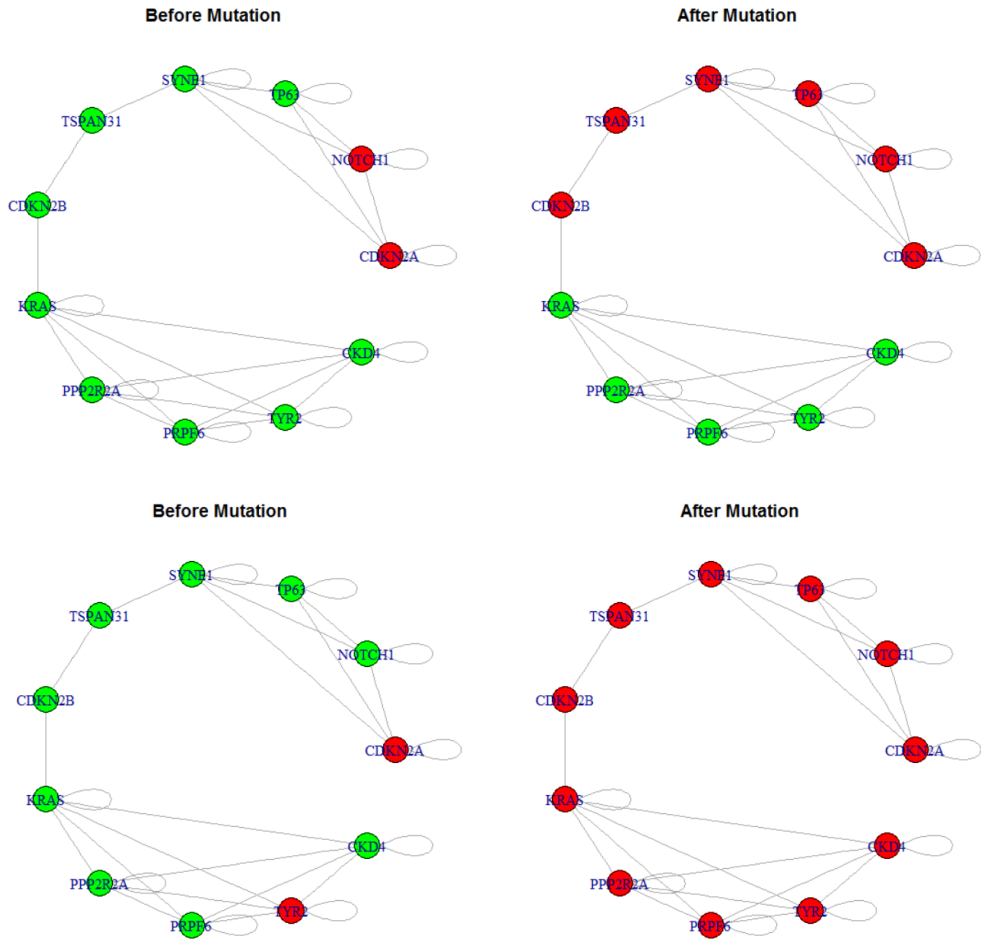

Figure 2: If the cancer gene has ability to attack two genes in the network, attacking genes on different cliques is preferred to attacking genes on the same clique: If genes on different cliques are attacked, both cliques would be infected; if only genes on certain clique are attacked, only that clique will be infected.

Under the two assumptions, the problem then became solvable. There are studies based on prior knowledge about pathways[3][4]. However, the prior knowledge is limited compared with the huge amount of possible gene links. It would be inappropriate to use those biased prior knowledge. Miller et al. (2011)[5] proposed a method that identifies mutational patterns without prior knowledge. In contrast, Ciriello, G. et al. (2012)[6] proposed MEMo to detect modules obeying mutually exclusive. Vandin et al. (2012)[7] proposed an innovative scoring strategy, called Maximum Weight Submatrix Problem (MWSP) for a possible pathway and find the maximizer of the score, which is the basis of our work. They proposed, in their work, a greedy algorithm and an MCMC algorithm for maximizing the objective score. In spite of the nice theoretical results in [7], both methods are flawed. Greedy algorithms work well under Gene Independence Model, which is only reasonable for some types of somatic mutations (e.g., SNP) but not others (e.g., CNV)[7]. On the other hand, the MCMC might trap in a local maxima and is inexact.

To circumvent those limitations, Zhao,J., Zhang,S., Wu,L.Y. and Zhang,X.S. (2012)[8] proposed a binary linear programming (BLP) problem which is exact, and a genetic algorithm which could be generalized easily. Similarly, Leiserson,M.D., Blokh,D., Sharan,R. and Raphael,B.J. (2013)[9] identifies multiple mutually exclusive pathways simultaneously by generalizing the original problem and applying BLP for solving it. The problem of all those BLP is that they lack flexibility and is hard to generalize. Our work is based on the work of [9], but includes a strategy for automatically selecting the number of pathways. It also proposed a Multiple Value Genetic Algorithm (MVGA) to avoid local trapping, while the algorithm is also easy to generalize. The MVGA borrowed idea from Greedy Algorithm [10], and works for not only this specific problem, but also several important optimization problems.

2 MATERIALS AND METHODS

2.1 A brief introduction

Two aspects of driver mutation pathway in cancer genes were emphasized as the key to identify pathways. According to [7], we can quantify:

- High Coverage: many patients have at least one mutation in the pathway

- High Exclusivity: most patients have no more than one mutation in this pathway.

with an innovative scoring strategy called Maximum Weight Submatrix Problem (MWSP).
Given the mutation data represented by a binary mutation matrix A with m rows and n columns, the maximum weight submatrix problem is defined as finding a submatrix M of size $m \times k$ in A by maximizing the scoring function:

$$W(M) = |\Gamma(M)| - \omega(M) = 2|\Gamma(M)| - \sum_{g \in M} |\Gamma(g)|,$$

where $\Gamma(g)$ represents the set of patients in which gene g is mutated. Thus, the first term $|\Gamma(M)| = |\cup_{g \in M} \Gamma(g)|$ is the coverage score.
And $\omega(M) \triangleq \sum_{g \in M} |\Gamma(g)| - |\Gamma(M)|$ measures the coverage overlap of M.
We want to solve the following NP-hard problem:
Maximum Weight Submatrix Problem (MWSP): Given an $m \times n$ mutation matrix A and an integer $k > 0$, find the $m \times k$ column submatrix $\hat{M}$ of A that maximizes W(M).
Leiserson,M.D., Blokh,D., Sharan,R. and Raphael,B.J. (2013)[9] formulated this problem as an integer linear program (ILP), while the MWSP became generalized to Multiple Maximum Weight Submatrices Problem.

The drawback of their method is that it does not generalize to other scoring strategy easily. And they did not give a strategy to select parameters such as the lower boundary for gene number in one pathway $k_{\min}$, upper boundary for pathway gene number $k_{\max}$ (without such a constraint, we might get too large a set of genes in one pathway).

2.2 Alternative Scoring Methods

Although considering the problem as an ILP runs efficiently, it cannot be applied to other scoring criteria. For example, Miller et al. (2011)[5] proposed an alternative approach for driver pathway is proposed.

Coverage Score:

$$C(M) = \frac{\#(\exists g \in M \text{ mutates in } p)}{m}$$

Exclusivity Score:

$$E(M) = \frac{\#(\text{exactly one } g \in M \text{ mutates in } p)}{\#(\exists g \in M \text{ mutates in } p)}$$

and the rest is similar.

Consider

$$\max F(x) = 2\sum_{i=1}^{m} w_i s(\sum_{j=1}^{n} x_j a_{ij})$$

$$s.t. \begin{cases} \sum_{j=1}^{n} x_j = k, \\ x_i \in \{0,1\}, i = 1, \cdots, n. \end{cases}$$

where x_j is the indicator whether column j of A falls into the submatrix M, y_i is the indicator whether the entries of row i of M are not all zeros.

The original De Novo scoring method in [7] could be equivalently represented as the above optimization problem with $s(x) = 1 - |x - 1|$. However, we can adjust the penalty function: $p_1(x) = \sqrt{x}$ for more emphasis on coverage, $p_2(x) = x^2$ for more emphasis on exclusivity, $p_3(x) = I(x > 0)$

for same penalty on non-one row sums, $p_4(x) = 0$ for homogeneous penalty (no penalty on overlapping). We can also set asymmetric, e.g. $s_1(x) = I(x \leqslant 1)$ or $s_2(x) = I(x \geqslant 1)$, according to our need and emphasis.

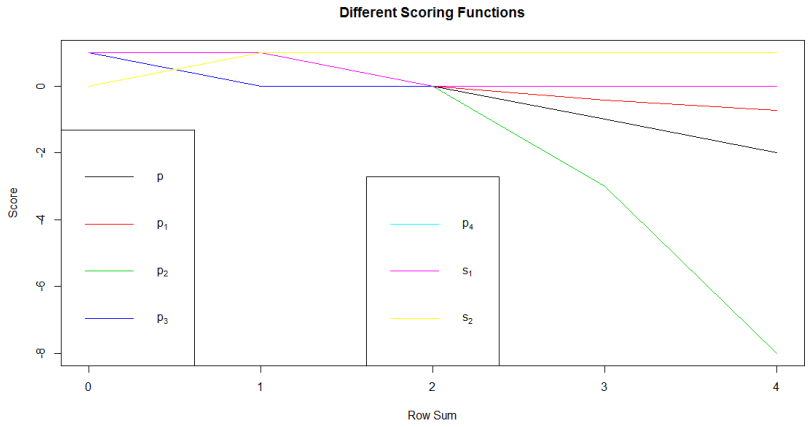

Figure 3: Different scoring strategy

Therefore, we need method that could be applied to various scoring methods.

2.3 Multiple Value Genetic Algorithm: a flexible method

Genetic algorithm works well on generality, as in [8]. Inspired by their work, MVGA generalize the Genetic Algorithm for multiple value case rather than binary value scenario.

Multiple Maximum Weight Submatrices Problem. Given an $m \times n$ mutation matrix A and an integer $t > 0$, find a collection $M = \{M_1, M_2, \cdots, M_t\}$ (mutually exclusive) of $m \times k$ column submatrices that maximizes $W'(M) = \sum_{\rho=1}^{t} W(M_\rho)$.

$W'(M)$ is uniquely determined by $M = \{M_1, M_2, \cdots, M_t\}$, and we can represent M with an indicator vector x of length n. x_i is ρ if the i-th gene is

in M_ρ, $x_i = 0$ if i-th gene is not in M.

Therefore, the problem could be viewed as

$$\max f(\vec{x}), \vec{x} \text{ consists of } 0, 1, \cdots, t$$

However, we also have constraint that $k_{\min} \leqslant \#\{x_i = \rho\} \leqslant k_{\max}, \forall \rho \in \{1, \cdots, t\}$, denote the space of $\vec{x}$ satisfying the inequality constraint $S(k_{\min}, k_{\max})$, simplified as S. This constraint could be a huge problem: if we set $k_{\min} = k_{\max}$, the space of feasible $\vec{x}$ is extremely small. For generality, we want to apply Genetic Algorithm for this problem, and therefore, we need to devise a crossproduct process satisfying:

$$\forall x, y \in S(k_{\min}, k_{\max}), c(x, y) \in S(k_{\min}, k_{\max})$$

where $c(x, y)$ is the product of x and y.

We call this property closure, the operator $c : S \times S \to S$ closure operator. A easy but feasible closure operator would be $c(x, y) = x$, but it is terrible for genetic algorithm: the population never evolves, it just permutes over time. Consequently, we need to make $c(x, y)$ as different from its parent x, y as possible. We can represent it with the following integer linear programming problem.

Denote father: $F = (a_1, \cdots, a_n)$, mother: $M = (b_1, \cdots, b_n)$, $a_i, b_i \in \{0, 1, \cdots, t\}$.

$a_i = \rho$ means i-th gene is in the ρ-th set, if $\rho = 0$, i-th gene is not selected.

The motivation is to find a feasible solution corresponding to child $c_1, \cdots, c_n$ under constraint:

$$\sum_{i=1}^{n} [(1 - x_i)\mathrm{I}(a_i = \rho) + x_i \mathrm{I}(b_i = \rho)] \in [k_{\min}, k_{\max}], \forall \rho \in \{1, \cdots, t\} (1)$$

where $x_i = 0$ represents $c_i = a_i$, and $x_i = 1$ represents $c_i = b_i$.

Moreover,

$$\sum_{a_i \text{ or } b_i > 0} x_i \in [\frac{s - c}{2}, \frac{s + c}{2}].$$

where $s = \#\{a_i \text{ or } b_i > 0\}$.

We should minimize c (ILP). The motivation is to find a feasible child $c_1, \cdots, c_n$ (satisfying (1)), which deviates most from its parents (the overlapping of the child and the mother $\sum x_i$ is closest to $\frac{s}{2}$, where s be the number of genes that mutates in either parent (elements that are nonzero in either parent).

2.4 Selection of pathway number

Apart from proposing an algorithm, we also provide a method for automatically selecting t - the number of pathway. The idea is based on permutation test:

when appropriate parameter t chosen, the scores of the selected pathways are significantly larger than the scores of perturbed pathways selected (the pathways selected when that original mutation matrix became shuffled).

If we shuffle the matrix by columns (for each gene, randomly choose patients that have mutation while preserving the total number of mutation), the resulting matrix is structurally similar to the original data matrix.

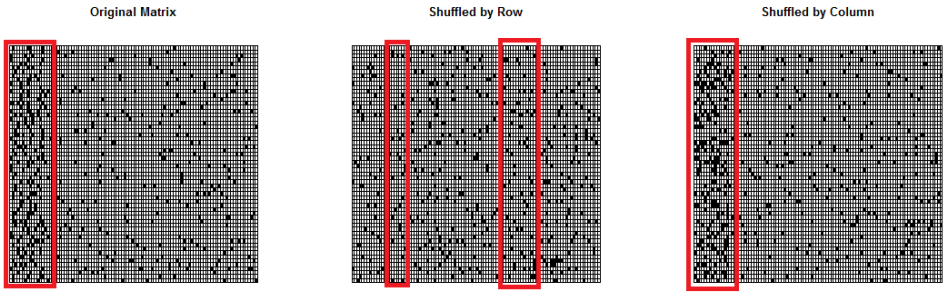

Figure 4: The comparison of different shuffling strategies

As a result, the final pathway selected will be close to the original path-

way, and the score is almost the same (guaranteed by Law of Large Number). However, if we shuffle the mutation matrix by row, the selected pathways would be very random and thus have low scores.

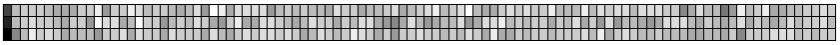

Figure 5: Score Pattern with exact parameters: $t = 3$, $4 \leqslant k \leqslant 6$, the first column represents score of pathways in original data matrix, the others represent the scores of shuffled matrices

2.5 Selection of Range for k

Besides the selection of t, the number of pathways for detection, we also proposed method to set $k_{\min}$ and $k_{\max}$ that restrict the number of genes in each pathway. The idea is to expand the range $[k_{\min}, k_{\max}]$ if it is too tight, squeeze it when it is loose. The flow chart of the algorithm is shown in the figure.

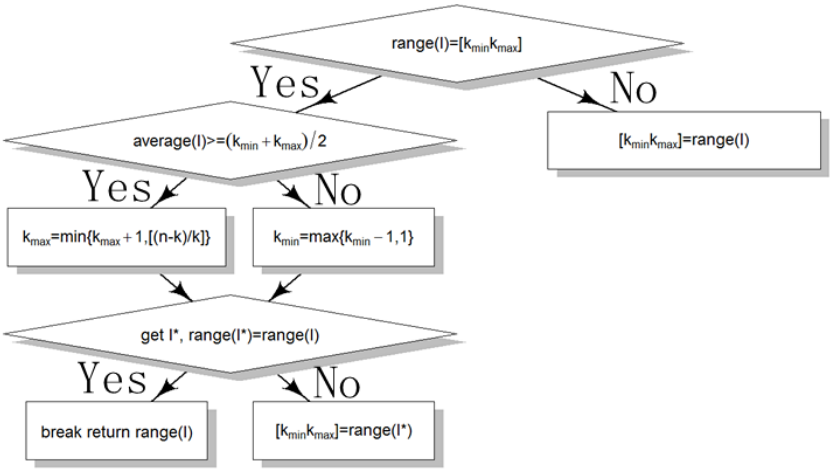

Figure 6: Algorithm for selection of $k_{\min}$ and $k_{\max}$

For each selected $k_{\min}$ and $k_{\max}$, we run the algorithm for 20 times, and we get a set of size of selected pathway. We then adjust the range according to the sizes of pathways selected. For example, if most of the size are much bigger than $k_{\min}$ and very close to $k_{\max}$, then the lower bound for k is too loose whereas the upper bound is too tight. We adjust by decreasing $k_{\min}$ and increasing $k_{\max}$. We have no quantitative result for this selection algorithm, but the idea of this method is widely applicable.

3 RESULTS

We first apply the MVGA algorithm to the simulated data, and next test the permutation test for selecting t, and finally use a toy example to validate MVGA.

3.1 Simulation study

We simulated mutation data starting with gene sets $M_1, \cdots, M_t$. Every set has $k_\rho \in [k_{\min}, k_{\max}]$ genes. For each patient, we randomly mutate x_ρ genes in M_ρ, $\rho = 1, \cdots, t$, while

$$\Pr(x_\rho = x) = \alpha, x = 0, 2, \cdots, k_\rho, \Pr(x_\rho = 1) = 1 - \alpha k_\rho$$

where α is a sufficiently small parameter. The smaller α is, the better the simulation data fits our method.

We use simulation data with $t = 2$, $k_1 = 5, k_2 = 4$, $m = n = 50$.

The identified pathways when setting $t = 2$, $k_{\min} = 2$, $k_{\max} = 6$ are $1, 2, 3, 5, 16, 6, 7, 8, 9, 20, 28$. The accuracy was around 72%, which is quite good. For more accuracy, we can also check the suboptimal solution obtained from the last generation of MVGA. (Genetic Algorithm generates a population of solutions, which is great for further analyze.)

3.2 Permutation Test

For the above simulated mutation data, we use permutation test to identify the appropriate t.

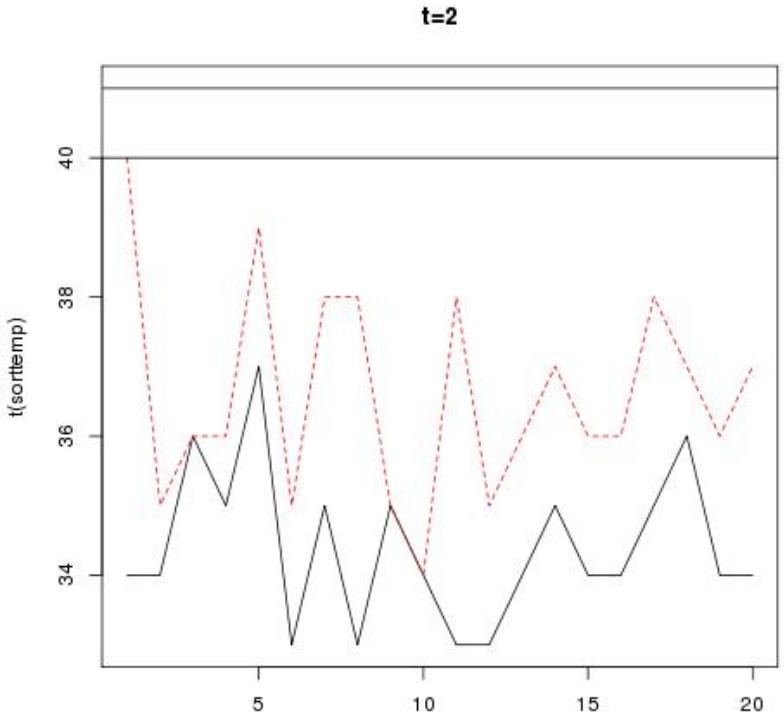

Figure 7: set $t = 2$, the scores of the original pathways is dominant over scores of pathways for shuffled matrix

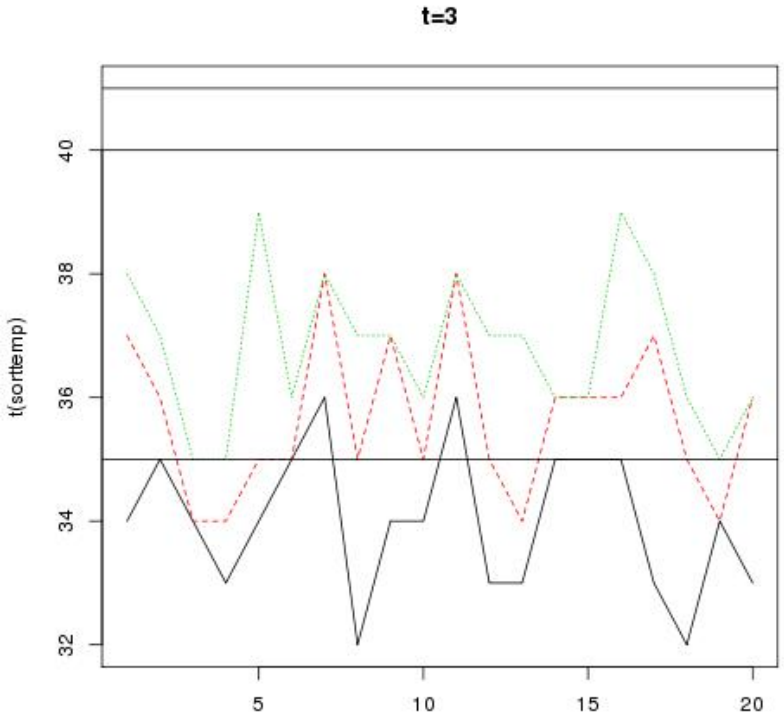

Figure 8: for $t = 3$, the scores became less significantly large over scores of shuffled pathways

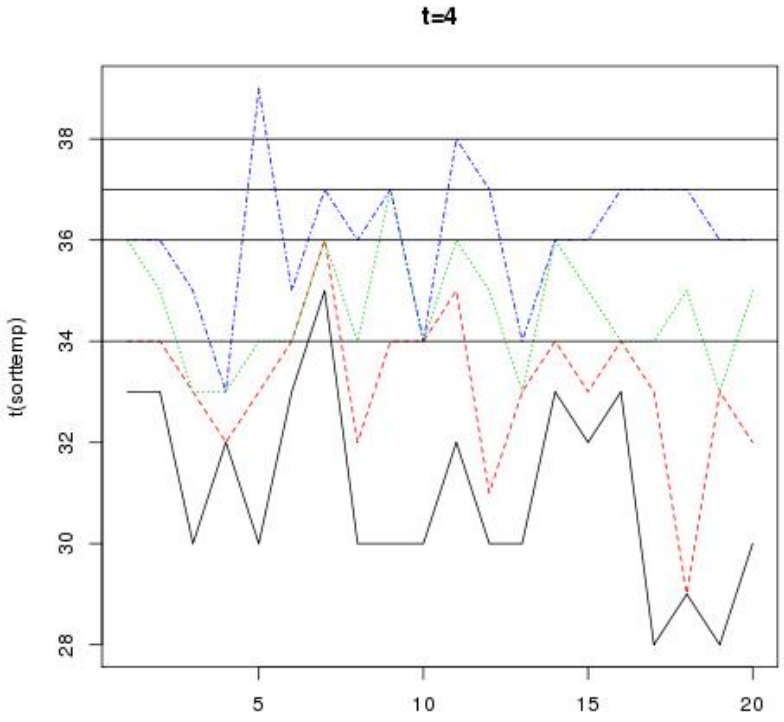

Figure 9: for $t = 4$, the scores became mixed over scores of shuffled pathways

As shown in the figures, the pathway scores for the original matrix is large over pathway scores for randomly shuffled mutation matrix only when the accurate t is chosen.

3.3 Toy Example

We then show an example of how MVGA could be applied to other application problems effectively.

We consider the hidden set detection problem:

Suppose there are $S_1, \cdots, S_t \subset \{1, \cdots, n\}$, we would like to detect those sets without prior knowledge. We could use a test set M and get its matching score with those hidden sets. For convenience, let $S(M) = n - \min_{1 \leqslant \rho \leqslant t} |M \triangle S_\rho|$. An iterative detection is to find maximizer $\hat{M} = \arg\max S(M)$, deleting $\hat{M}$, repeat. The simultaneous version would be $\max(M_1, \cdots, M_t) = \sum_{i=1}^{t} S(M_i)$. This problem is similar to our task, and we use this toy problem to test the methods.

We use MVGA to test whether it converges in this problem. Let $S_1 = \{1, \cdots, 5\}, S_2 = \{6, \cdots, 10\}$, $n = 20$. And we find M_1, M_2 simultaneously using MVGA.

Population	Iteration	Score	Time(s)
100	1	34	52
	10	37	135.41
	20	39	206.07
10	1	32	12
	10	34	10.19
	20	33	44.97
50	20	38	115.43

Table 1: The performance of MVGA on hidden set detection problem

4 DISCUSSION

Identifying mutated driver pathways in cancer genome is a major task in computational biology. In this report, we have studies the simultaneous identification of multiple driver mutated pathways via Multiple Value Genetic Algorithm (MVGA). We propose MVGA to solve the optimization

problem with variable a t-value vector, where t be the desired number of pathways selected. We further give suggestions on how to select k-the number of pathways identified, $k_{\min}$ and $k_{\max}$-the lower and upper bound for the gene number of each pathway.

On computational cost, we plug an ILP inside of a genetic algorithm, the computation is expensive. However, we only need to involve $|F \cup M| \leqslant 2tk_{\max} \ll n$ binary variables and one integer variable, making the cost tolerable.

About the reason for selecting Genetic Algorithm, we have the following reasons:

- The problem is not always an ILP (Integer Linear Programming) task.
- MCMC might trap in a local maxima.
- The return value of GA is a set of solutions, suboptimal solutions are obtained as bonus.
- GA is flexible for further integrated approach (such as involving expression data as in [8] and various scoring settings.

However, the major problem of MVGA is that although it is general for various problems, it is extremely slow in speed. This is also the reason why we did not include biological data in the report.

References

[1] Vogelstein B, Kinzler KW (2004) Cancer genes and the pathways they control. Nat Med 10: 789 – 799.

[2] Ciriello G, Cerami E, Sander C, Schultz N (2012)Mutual exclusivity analysis identifies oncogenic network modules. Genome Res 22: 398 - 406.

[3] Efroni, S. (2011)Detecting cancer gene networks characterized by recurrent genomic alterations in a population. PLoS ONE, 6, e14437.

[4] Boca, S.M. (2010)Patient-oriented gene set analysis for cancer mutation data. Genome Biol, 11, R112.

[5] Miller et al. (2011)Discovering functional modules by identifying recurrent and mutually exclusive mutational patterns in tumors. BMC Medical Genomics 4:34

[6] Ciriello, G. et al. (2012)Mutually exclusivity analysis identifies oncogenic network modules. Genome Res., 22, 398-406

[7] Vandin et al. (2012)*De novo* discovery of mutated driver pathways in cancer. *Genome Res*., 22, 375-385.

[8] Zhao,J., Zhang,S., Wu,L.Y. and Zhang,X.S. (2012)Efficient methods for identifying mutated driver pathways in cancer. Bioinformatics, 28, 2940 - 2947.

[9] Leiserson,M.D., Blokh,D., Sharan,R. and Raphael,B.J. (2013)Simultaneous identification of multiple driver pathways in cancer. PLoS Comput. Biol., 9, e1003054.

[10] Goldberg, D.E. (1989)Genetic Algorithms in Search Optimization and Machine Learning. Addison Wesley.