Simultaneously Identifying Pathways via Multiple Value Genetic Algorithm

Bowen Deng

Dept. of Prob. and Stat.

Organization

Literature Review

Candidate Criteria

Genetic Algorithm

Multiple Value Genetic Algorithm

Parameter Selection

- Selection of t
- Selection of range

Further Work

Optimization Problem

Objective:

$$\max O(M_1, \cdots, M_t) = \sum_{i=1,\cdots,t} W(M_i)$$
$$s.t. |M_i| \in [k_{\min}, k_{\max}]$$
$$M_i \cap M_j = \emptyset$$

Binary Linear Programming

$$\max O(M_1, \cdots, M_t) = \sum_{\rho=1}^{t} \sum_{i=1}^{m} (2C_i(M_\rho) - \sum_{j=1}^{n} I_{M_\rho}(j) A_{ij})$$

$$s.t. \sum_{j=1}^{n} I_{M_\rho}(j) A_{ij} \geqslant C_i(M_\rho)$$

$$\sum_{\rho=1}^{t} I_{M_\rho}(j) \leqslant 1$$

$$k_{\min} \leqslant \sum_{j=1}^{n} I_{M_\rho}(j) \leqslant k_{\max}$$

MDendrix and IterDendrix

To find t pathways, one method is to solve the ILP directly. (MDendrix)
Another iterative approach is to solve the ILP with $t = 1$, delete the identified genes, and run iteratively. (iterDendrix)
By theory, the time cost of MDendrix and iterDendrix is comparable with small t.

$$\begin{aligned} TC(M) &\geqslant TC(\text{iter once}) \\ &= \frac{\sum_{\rho=1}^{t} TC(\text{iter once})}{t} \\ &= \frac{TC(\text{iterDendrix})}{t} \\ &\geqslant \frac{TC(\text{mDendrix})}{t} \end{aligned}$$

The authors used CPLEX v12.3 for implementation. We use lpSolve package in R. Set simulation data with $m = 200$, $n = 1000$, $I = 10$, $t = 3$. $k_{\min} = 8$, $k_{\max} = 12$.
MDendrix:

Time Cost	184.32	
Result	Score	Score of Standard
$1 \sim 10\,761\,774$	30	32
$11 \sim 20\,752\,973$	35	40
$21 \sim 30\,34\,109$	32	36

IterDendrix:

Time Cost	7.06	
Result	Score	Score of Standard
$1 \sim 10\,214\,774$	30	32
$21 \sim 30\,34\,109$	32	36
$11 \sim 20\,652\,752$	35	40

Candidate Criteria

For mutation matrix A, p takes value in m patients, and g takes value in n genes. M is a set of genes.
We borrow the criteria from RME, an alternative approach for driver pathway identification.
Coverage Score:

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

Exclusivity Score:

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

Denote $I_M(p)$ the occurrence indicator of mutations of patient p in M.

$$\begin{aligned} S(M) &= C(M) + E(M) \\ &= \frac{\#\{I_M(p) > 0\}}{m} + \frac{\#\{I_M(p) = 1\}}{\#\{I_M(p) > 0\}} \\ W(M) &= 2\#\{I_M(p) > 0\} - \sum_p I_M(p) \end{aligned} \tag{1}$$

The objective is to find $\hat{M}$ as the maximizer of $S(M)$. We could also restrict $|M| = k$.

Simultaneous Detection with New Criteria

We combine this new criteria with mDendrix to detect a mutually exclusive set of genes $M = \{M_1, \cdots, M_t\}$ which maximizes:

$$S(M) = \sum_{\rho=1}^{t} S(M_\rho)$$

$k_{\min} \leqslant |M_\rho| \leqslant k_{\max}$.

Computation

The objective is to maximize $S(M)$. GA (Genetic Algorithm) is one of the top choices:

- The problem is no longer an BLP (Binary 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's time cost is tractable.
- GA is flexible for further integrated approach and variable scoring settings.

However, we should generalize the GA because it only works for binary case.

Toy Example of GA

We would like to maxmize $f(x) = e^{-x^2}, x \in [-5, 5]$.

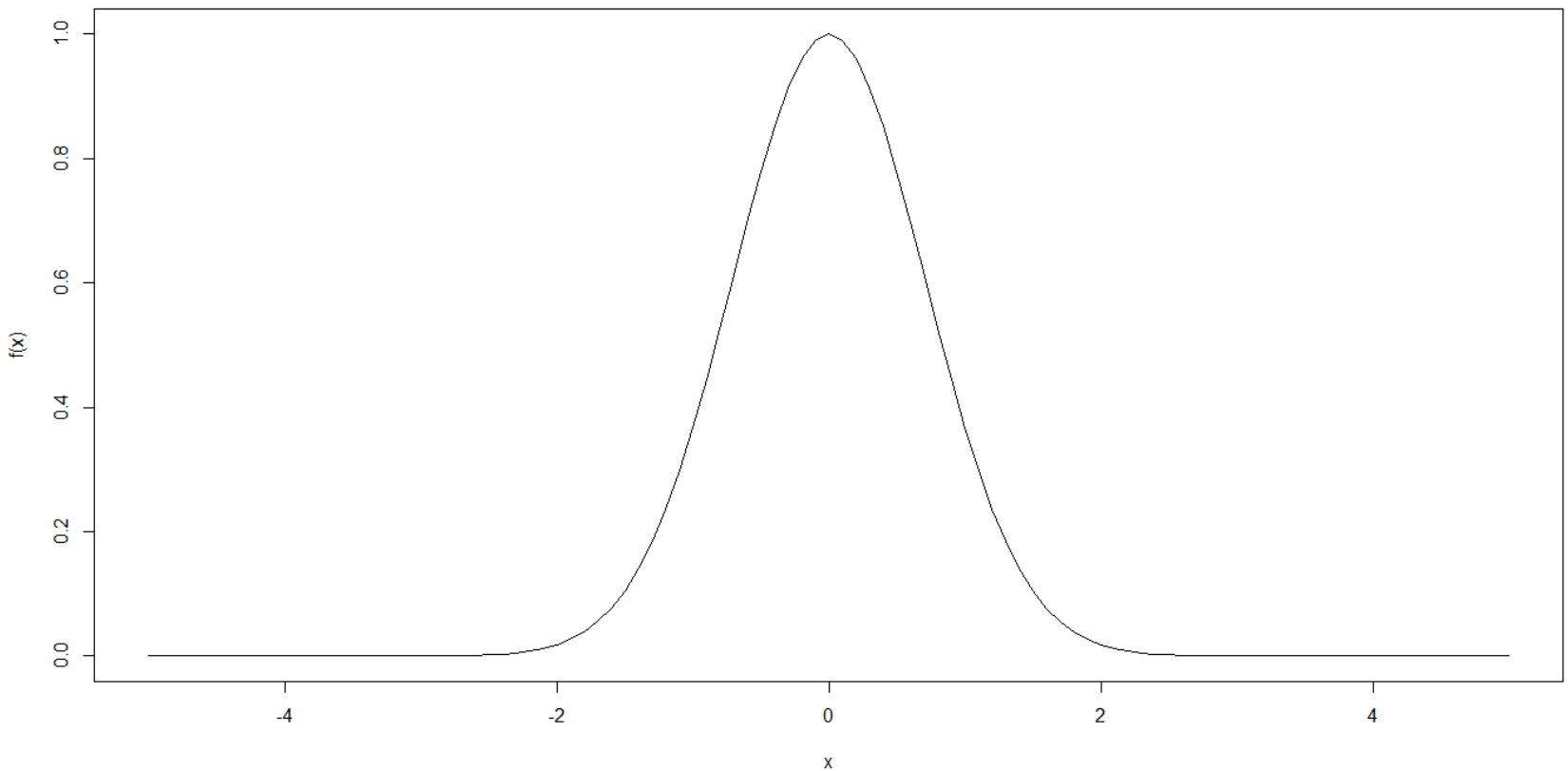

Figure: Objective Function

Initialize a Population

We first select a parent generation of size $P = 10$ from $[-5, 5]$.
2.42 2.33 2.04 2.00 -0.71 1.16 -1.16 1.75 3.66 4.20

Crossproduct

We select two of the parents to produce a child.
The probability of x being selected is proportional to $f(x)$, therefore, an elite is more likely to be chosen than a normal person candidate.
The crossproduct is flexible with specific problems, here we could use $cp(x, y) = \frac{x+y}{2}$.
We generate $P = 10$ children.
-0.938 -0.938 0.224 -0.938 0.224 0.811 0.224 -0.938 -0.938 -0.938

Mutation

With probability 0.1, a child will mutate if it increases its score.

Iteration

We pool the parents and the children and get 20 candidates.
We remain the 10 top scored ones.
0.224 -0.715 0.811 -0.938 -1.161 1.163 1.750 2.005 2.047 2.338 2.427 3.664 4.209
We use them to produce the next generation iteratively until convergence.
-7.24e-09 5.27e-09 2.85e-10 -9.82e-10 -5.02e-09 -3.48e-10 -7.14e-09 4.05e-09 2.78e-09 6.54e-09

Analysis

To improve the efficiency of Genetic Algorithm, i.e. the convergence rate, we should use large population size P, higher mutation rate (though a high mutation rate might increase the computation cost).
To avoid unnecessary computation, we set up a reasonable, rather than a perfect ending condition.

MVGA

We could generalize the GA algorithm to multiple value case.
Most steps are similar, but the crossproduction is much harder.
Let $k_{\min} = k_{\max} = 2$,
Father:
11223300
Mother:
22331100
Their child should inherit two 1s, two 2s and two 3s. But the child should be as distinct from its parents as possible.

Crossproduct

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:

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

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).

Discussion

At first glance, 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.

Selection of t

We set up a high score threshold s_0, a gene pathway M is good if $S(M) > s_0$.
Release the constraint on the length of pathways found (although we set bounds wide enough for the sake of computation in real application), and we use MVGA to identify t pathways M_ρ^i which maximize $\sum_{\rho=1}^{t} S(M_\rho)$ for $i = 1, 2, \cdots, \ell$.
We define RGP (rate of good pathway) as:

$$\text{RGP} = \frac{\#\{W(M_\rho^i) > s_0\}}{\ell t}$$

The RGP would decrease after the best t, and we could select the decreasing point as the estimation of t.

Identifying the decreasing point of RGP

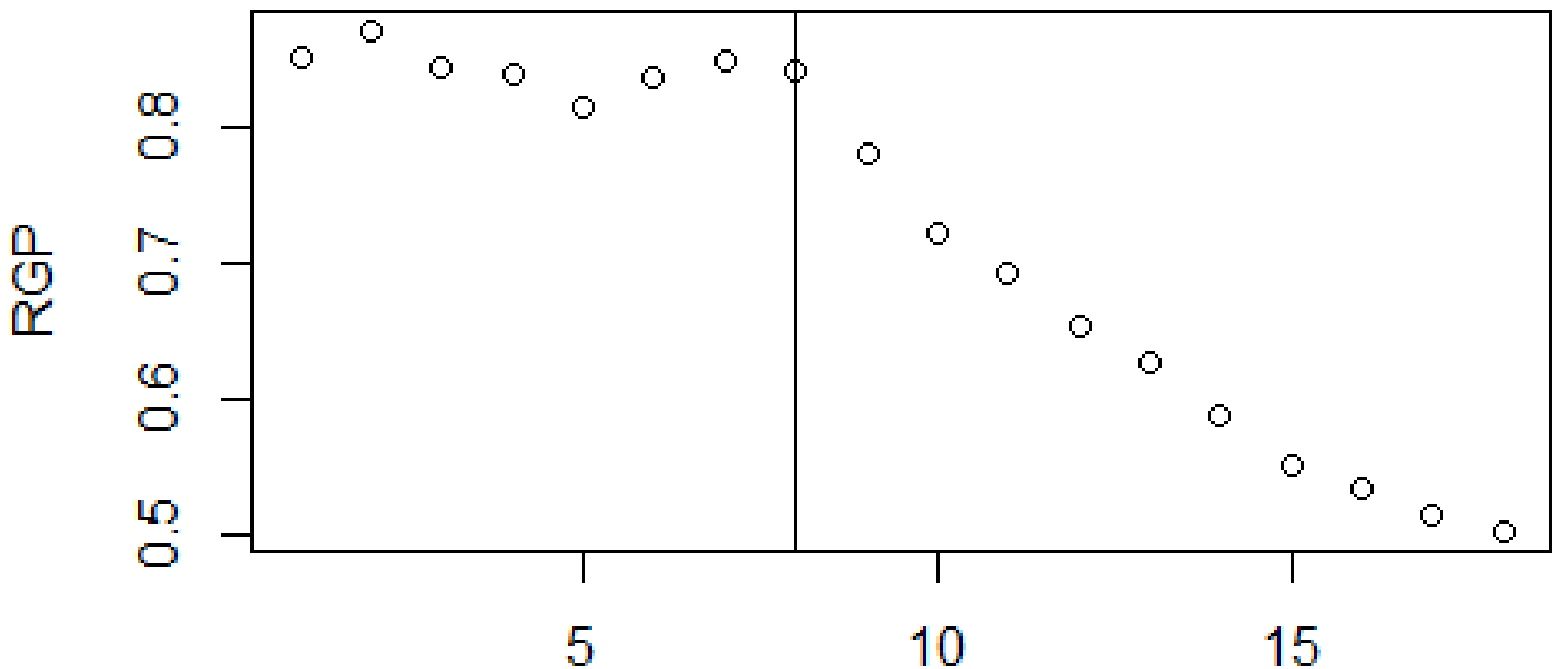

Figure: RGP in respect with t

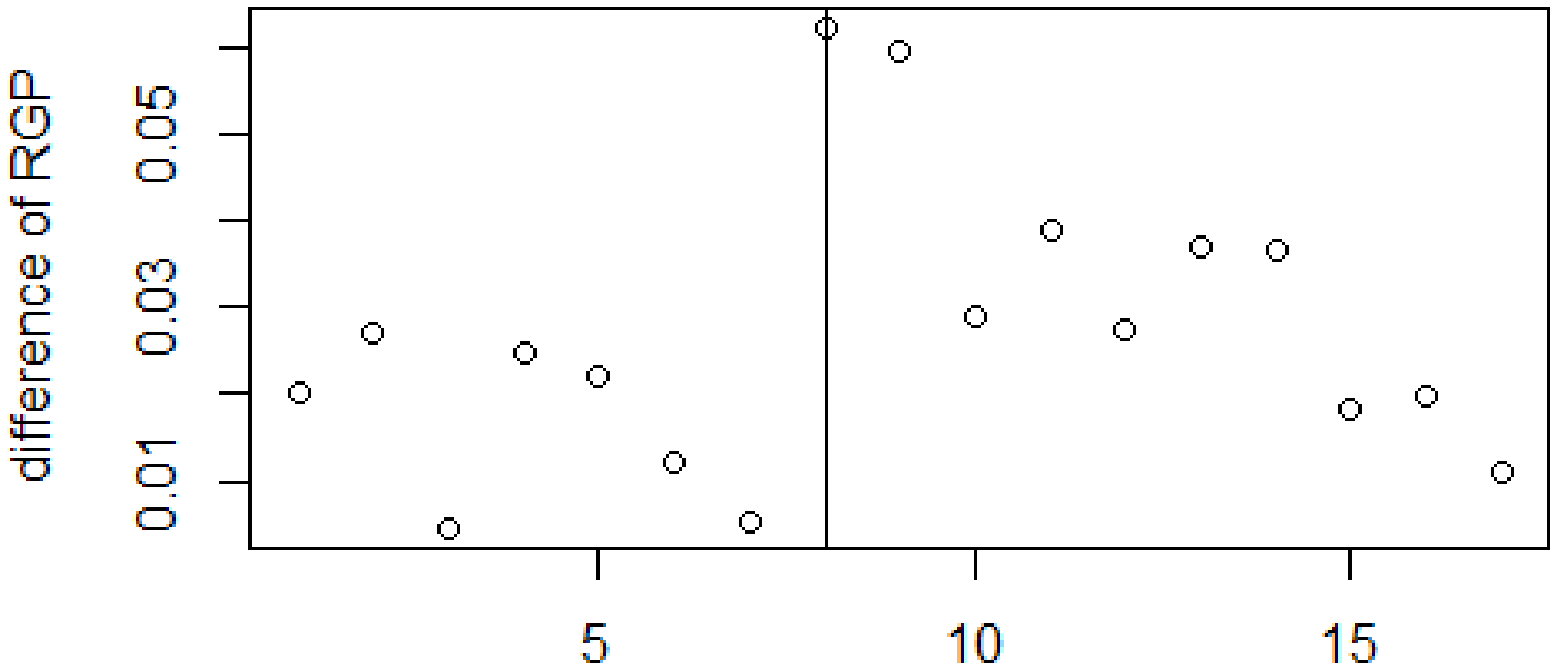

Figure: difference of RGP in respect with t

Based on these observations, we select the t that maximizes $\frac{|x_{t+1}-x_t|}{|x_t-x_{t-1}|}$ as the final estimation.

Selection of $k_{\min}$ and $k_{\max}$

Once t has been selected, we now need to estimate $k_{\min}$ and $k_{\max}$.
For the sake of computational cost, we constrain that
$\ell(M_\rho) \in [gk_{\min}, gk_{\max}]$.
Assume

$$\Pr(\ell(M_\rho) = x) = \begin{cases} \frac{q}{k_{\min} - gk_{\min}} & x \in [gk_{\min}, k_{min}) \\ \frac{1-2q}{k_{\max} - k_{\min} + 1} & x \in [k_{\min}, k_{\max}] \\ \frac{q}{gk_{\max} - k_{\max}} & x \in (k_{\max}, gk_{\max}] \end{cases}$$

where $gk_{\min} < k_{\min} \leqslant k_{\max} < gk_{\max}$.
We run s results $x_{11}, x_{12}, \cdots, x_{st}$. Denote $l(x_{ij}) = l_{ij}$.
The likelihood of $k_{\min}$ and $k_{\max}$ is

$$L(k_{\min}, k_{\max}) = \prod_{1 \leqslant i \leqslant s, 1 \leqslant j \leqslant t} \Pr(\ell(M) = l_{ij})$$

We use M.L.E. to estimate $\hat{k}_{\min}$ and $\hat{k}_{\max}$.

$$(\hat{k}_{\min}, \hat{k}_{\max}) = \arg\max \ln L(k_{min}, k_{\max})$$

Further Study Directions

- Find alternative crossproduct method for MVGA
- Apply both simulation data and biological data to test the new method
- Mathematically proof of the stability of parameter selection of $k_{\min}$ and $k_{\max}$