Boosting Multifactor Dimensionality Reduction Using Pre-evaluation

Yingfu Hong, Sangbum Lee, and Sejong Oh

The detection of gene–gene interactions during genetic studies of common human diseases is important, and the technique of multifactor dimensionality reduction (MDR) has been widely applied to this end. However, this technique is not free from the "curse of dimensionality" — that is, it works well for two- or three-way interactions but requires a long execution time and extensive computing resources to detect, for example, a 10-way interaction. Here, we propose a boosting method to reduce MDR execution time. With the use of pre-evaluation measurements, gene sets with low levels of interaction can be removed prior to the application of MDR. Thus, the problem space is decreased and considerable time can be saved in the execution of MDR.

Keywords: Gene interactions, multifactor dimensionality reduction, cross validation, genotype, pre-evaluation, MDR.

I. Introduction

Gene–gene interactions occur when gene expression is influenced by the expression of one or more other genes. Gene expression is also related to human diseases; in general, a single gene is not responsible for a specific disease, rather multiple genes participate in the occurrence of a disease. Therefore, the ability to detect gene interactions with respect to specific phenotypes is an important research goal. Researchers have tried to develop computational methods to detect gene–gene interactions, including those based on logistic regression models, information theory, multifactor dimensionality reduction (MDR) [1], and machine learning approaches [2].

The MDR method was developed by Ritchie and others [3] and is based in part on the combinatorial partitioning method [4]. MDR has been used to detect gene–gene interactions in case–control studies. In general, a dataset for MDR analysis is composed of a phenotype (case or control) column and multiple columns of single nucleotide polymorphisms (SNPs). The values of the SNP columns are defined as 0, 1, and 2 to express three genotypes — AA, Aa, and aa, respectively. Within a dataset, we want to find the best k SNPs from n SNPs that will explain the phenotype well.

The MDR algorithm first divides the dataset into nine training sets and one test set and calculates the case (control) ratios for each multilocus genotype. Next, the high-risk multilocus genotypes are identified, and an optimal model for producing maximum training accuracy is chosen. MDR performs a cross-validation test to find the model that will maximize cross-validation consistency. The main advantage of MDR is that it facilitates the simultaneous detection and characterization of multiple genetic loci associated with a discrete clinical end point. Furthermore, it is a nonparametric

Manuscript received July 26, 2014; revised June 15, 2015; accepted July 24, 2015.

The present research was conducted by the research fund of Dankook University in 2013.

Yingfu Hong (yeongbok112@gmail.com) and Sejong Oh (corresponding author, sejongoh@dankook.ac.kr) are with the Department of Nanobiomedical Science, Dankook University, Cheonan, Rep. of Korea.

Sangbum Lee (sblee@dankook.ac.kr) is with the Department of Computer Science, Dankook University, Cheonan, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.16.0114.0040

method that overcomes the limitations of traditional parametric methods, and it assumes there is no particular genetic model [3]. The main disadvantage of MDR is that it can be computationally intensive, especially when more than 10 polymorphisms (a set of SNPs) need to be evaluated, and its predictive ability is low when the dimensionality of the best model is relatively high and the sample is relatively small [4].

Various follow-up studies have been undertaken to improve the original MDR method. Li and others suggested weighted risk score-based MDR [5]. Bulinski studied the impact of the choice of penalty function, and his study is used as a basis for MDR methods [6]. Lou and others proposed a generalized MDR (GMDR) framework that is based on the score of a generalized linear model, of which the original MDR method is a special case [7]. GMDR allows adjustment for covariates. Bush and others suggested improving the ability of MDR by replacing a classification error with a different measure to score the quality of the model [8]. Multivariate GMDR was also introduced by Xu and others [9]. Chen and others proposed a unified GMDR [10]. They used principal components as genetic background controls. Fisher and others reported a cell-based metric that improves the detection of gene–gene interactions with MDR [11]. Lee and others surveyed recent MDR methods, and compared their characteristics [12].

Although many advanced MDR models have been proposed, the primary disadvantage of MDR — the problem of dimensionality — has still not been overcome.

Let us suppose a dataset has 1,000 SNPs. If we want to investigate 10-way interactions between the SNPs, MDR should calculate ${}^{1,000}C_{10} \approx 2.6 \times 10^{23}$ combinations, and this would be almost impossible to calculate in the allowable time. If we reduced the problem space from 1,000 to 100, then the number of combinations to be calculated would become ${}^{100}C_{10} \approx 1.7 \times 10^{13}$ cases, but this would also require a long computation time.

Some studies have been carried out to try to reduce the computation time. Sinnott-Armstrong and others suggested a rapid MDR method [13], [14] that handles continuous data by modifying MDR's constructive induction algorithm to use a *t*-test. Yang and others proposed an ensemble approach to filter out irrelevant SNPs and improve MDR execution time [15]; it runs multiple filter algorithms and then merges the results. Even though these approaches tried to reduce computation time, they focused on pre-filtering (see Fig. 1) and did not reduce the number of SNP combinations. Since total computation time mainly depends on the number of SNP combinations, we need to find a way to decrease it.

In this paper, we propose a new method to reduce the execution time of MDR. Our proposed method suggests ways to diminish combinations of SNPs, whereas previous approaches tried to reduce individual SNPs. With our method, the target combinations of SNPs are pre-evaluated, and only a small number of combinations that are highly relevant to the phenotype are recommended (see bold-outlined box in Fig. 1(b)). MDR then calculates only the recommended combinations of SNPs to save execution time.

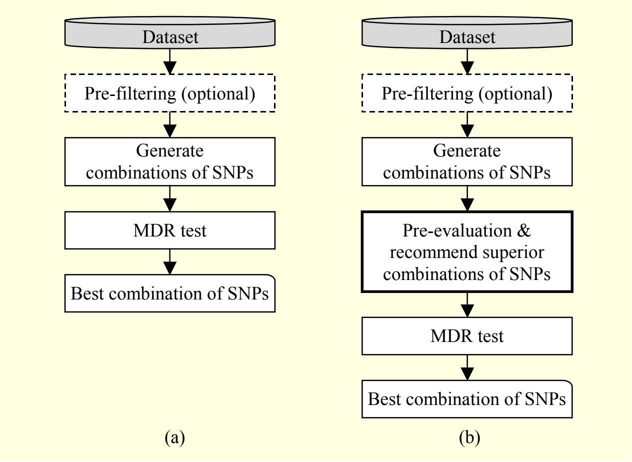

Fig. 1. Current vs. proposed methods for boosting MDR: (a) normal MDR test and (b) boosted MDR test.

Figure 1 summarizes the goal of this study. To find the most appropriate pre-evaluation method for the SNP combinations, we tested two entropy measures, since the philosophy of entropy is similar to the evaluation of contingency tables in the MDR process. We also developed new pre-evaluation methods based on a contingency table. The details of these pre-evaluation methods are described in Section II. The difference between our pre-evaluation method and well-known strategies for selecting genetic features [16], such as forward selection or backward elimination, is that our method searches throughout the entire problem space, whereas a genetic selection strategy searches over only part of such a space. Therefore, a genetic selection strategy is less likely to find an optimal solution.

The remainder of this paper is structured as follows. Section II summarizes the basic information theory that we used in developing our proposed method and describes how the proposed method reduces the number of SNP combinations. It also introduces the benchmark datasets and several MDR methods for comparison with our proposed method. Section III describes and discusses the results of these experiments, and Section IV presents our conclusions.

II. Materials and Methods

1. Basic Information Theory

In information theory, entropy is widely used to evaluate variables (features) in a dataset. The intuitive meaning of

http://dx.doi.org/10.4218/etrij.16.0114.0040

entropy is that it is a measure of the "uncertainty" of a variable (an outcome). A high level of entropy expresses a high uncertainty. The entropy of variable X is denoted by

$$H(X) = -\sum_{x\in X} p(x)\log p(x). \qquad (1)$$

Mutual information (MI) or information gain is used to evaluate the correlation between two variables X and Y, and the MI measures the shared information between X and Y. In other words, it measures how much the knowledge of one of the two variables reduces the uncertainty about the other. Therefore, a high level of MI indicates a strong relationship between the two variables. In a case (control) study, if an SNP has a high level of MI with a phenotype, then we can declare that the SNP has a strong relationship with the phenotype. MI can be defined according to the following equations:

$$I(X;Y) = H(X) - H(X\,|\,Y) = H(Y) - H(Y\,|\,X), \qquad (2)$$

$$I(X;Y) = I(Y;X). \qquad (3)$$

The shortcoming of finding MI is that it is difficult to understand the meaning of its absolute value. We cannot just perform a relative comparison between values of MI, because we don't know its maximum and minimum values. Therefore, normalized MI, which is called symmetrical uncertainty (SU), is widely used instead of MI. SU is defined as

$$\mathrm{SU}(X,Y) = 2\frac{I(X;Y)}{H(X)+H(Y)}. \qquad (4)$$

Ghiselli [17] suggested a correlation measure, called "merit function," between Z (a set of variables), and C (a target variable) as follows:

$$r_{zc} = \frac{k\overline{r_{zc}}}{\sqrt{k+k(k-1)\overline{r_{IJ}}}}, \qquad (5)$$

where $\overline{r_{zc}}$ is the average correlation between a variable in Z and the target variable C; k is the number of variables in Z; and $\overline{r_{IJ}}$ is the average intercorrelation between the variables in Z.

Hall [18] suggested a modified merit function based on (5). Let us suppose $Z = \{S_1, S_2, \ldots, S_k\}$, where S_i is an SNP and P is a phenotype. Then, the new merit function is defined as

$$M_{ZP} = \frac{\sum_{i=1}^{k}\mathrm{SU}(S_i,P)}{\sqrt{k+z\sum_{i,j\in k}\mathrm{SU}(S_i,S_j)}}. \qquad (6)$$

The approach to detecting gene–gene interactions can be redefined by asking the question "Which set of variables (features, SNPs) is most highly correlated with a given phenotype?" If we use the proper evaluation function as a measure, then we can sort out highly correlated SNPs.

2. Motivation and Pre-evaluation Function

To determine the relationship between the MDR result and the entropy-related functions, we tested the D1 dataset (Table 1), which contains 20 SNPs. The MDR found the highly correlated SNPs {1, 2, 4, 6, 8, 15} (see Table 6). We tested for entropy; MI between each SNP and phenotype; and SU between each SNP and phenotype.

Table 1 summarizes the results in which the 20 SNPs are sorted according to the value of each of the three evaluation functions. From these results, we can observe the following: (a) significant SNPs from MDR are highly ranked (see shaded cells in Table 1); (b) MI and SU show almost the same evaluation power; (c) MI and SU correlate better with the MDR results than does the entropy measure; and (d) in some cases, entropy performs better than MI and SU (see SNP 2).

Table 1 gives us some insight into how MDR can be boosted by reducing the problem space. When we make combinations of SNPs for the MDR test, we may skip the combinations where the SNPs have a low ranking in the entropy-related evaluation. Entropy calculation is simple and fast, so we can test every target combination of SNPs using the entropy-

Table 1. Relationship between MDR result and entropy-related functions in D1 dataset.

SNP	Entropy	SNP	MI	SNP	SU (*C*)
6	0.937714	1	0.01040	1	0.01235
8	0.988021	6	0.00758	6	0.00929
1	0.991954	7	0.00370	8	0.00432
4	0.997404	8	0.00363	7	0.00432
14	1.004446	3	0.00320	3	0.00364
18	1.015421	15	0.00221	15	0.00255
2	1.018505	9	0.00217	9	0.00246
7	1.022245	16	0.00163	16	0.00188
12	1.024120	4	0.00133	4	0.00157
5	1.030241	11	0.00132	11	0.00154
11	1.030738	20	0.00127	20	0.00147
13	1.032181	5	0.00123	5	0.00143
17	1.032564	2	0.00081	2	0.00095
10	1.037466	10	0.00075	10	0.00087
20	1.039621	19	0.00055	19	0.00063
15	1.043112	14	0.00048	14	0.00056
16	1.046232	12	0.00028	12	0.00032
19	1.046428	17	0.00023	17	0.00027
3	1.061025	18	0.00008	18	0.00010
9	1.073202	13	0.00008	13	0.00009

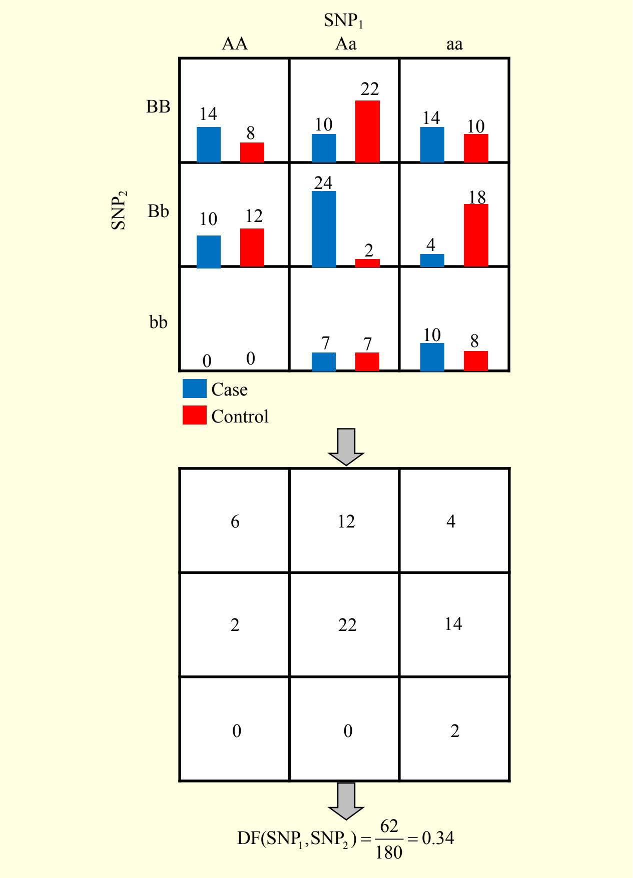

Fig. 2. Example of how DF function is calculated.

related function, and we can remove the low-ranked combinations.

Now, we introduce three pre-evaluation functions. Using MI, we composed the first pre-evaluation function (PMI) as follows:

$$\mathrm{PMI}_Z = \sum_{i \in k} I(S_i, P), \tag{7}$$

where $I(S_i, P)$ denotes the MI between SNP S_i and phenotype P. Equation (7) implies that a high value of MI is important to have a significant combination of SNPs. The merit-based pre-evaluation function, PMERIT, is the same as (6).

We developed a new pre-evaluation function based on a contingency table (PCT), which can be made using the same process as in the third step of the MDR process, as shown in (8),

$$\mathrm{PCT}_Z = \sum_{i,j \in k} \mathrm{DF}(S_i, S_j), \tag{8}$$

where DF is defined to be

$$\mathrm{DF}(S_i, S_j) = \frac{\text{total \# of differences in contingency table}}{\text{total \# of cases and controls}}. \tag{9}$$

Figure 2 is an example of how to calculate the DF function. If the evaluation value obtained using PCT is high, then we may expect that a given pair (SNP_1, SNP_2) has a high likelihood of interaction because the pair explains the phenotype well. PCT summates the DF value of every pair of (S_i, S_j).

Using the pre-evaluation functions PMI, PMERIT, and PCT, we evaluate every combination of SNPs in a given dataset, and we recommend the combinations of SNPs that have a high ranking. The MDR test is then repeated for the recommended combinations. Our testing process is as follows:

Testing process

```
// input: dataset $D = \{S_1, S_2, \ldots, S_n\}$,
//     phenotype $P$, number of selected SNP $k$
//     proportion $m$
// output: subset of $D$ which has best values of MDR test

Generate $m$ combinations (Z) from $D$ which have $k$ elements

// pre-evaluation
FOR $i = 1$ TO $n$
  PResult$_i$ $\leftarrow$ $PE(Z_i)$
END FOR

Sort PResult by decreasing order
$r \leftarrow {}_nC_k \times m$
Choose best $r$ indexes of PResult and store into $q1, q2, \ldots, qr$

// MDR test after pre-evaluation
FOR $j = 1$ TO $r$
  MResult$_j$ $\leftarrow$ $MDR(Z_{qj}, P)$
END FOR

Sort MResult by increasing order
Choose first index of MResult and store into $s$

RETURN $Z_s$
```

In the testing process (see box), "proportion m" indicates the recommendation ratio of the pre-evaluation results. If $m = 0.05$, then 5% of the SNP combinations with a high rank in the pre-evaluation results are recommended for the MDR test that follows.

3. Datasets and Testing Environment

We implemented the proposed method in R (http://www.r-project.org). For the MDR test, the "*mdr*" function in the MDR package was used; "*mdr*" implements Ritchie's method [3]. We tested the proposed method on an HP Linux server that has an Intel Xeon X5670 CPU (6 Cores, 2.93 GHz) and 18 gigabytes of RAM. To test the multi-core PCT algorithm, a "parallel" package was used.

Table 2 summarizes the five benchmark datasets we used in our experiments. These datasets included different numbers of

SNPs and samples. Dataset D5 contained synthetic values; only the first two SNPs were real values. To measure the quality of the SNP combinations, we used a support vector machine (SVM) [19] classifier because SVM performs well in binary-class (case, control) problems. The classification accuracies of the given SNP combinations were then compared. If a combination of SNPs produced high classification accuracy, then it meant that it had a strong relationship with the phenotype and perhaps the SNPs in the combination would have a strong interaction. The SVM code was chosen from the e1071 R package.

Table 2. Summary of datasets.

ID	Name of dataset	No. of cases	No. of controls	No. of SNPs	No. of cases
D1	mdr_sample_data[1]	200	200	20	200
D2	mdr1[2]	125	125	25	125
D3	mdr2[2]	554	446	50	554
D4	simSNP[3]	558	442	10	558
D5	20.400.1[4]	200	200	1,000	200

[1] http://www.multifactordimensionalityreduction.org

[2] MDR R package

[3] mrmr R package

[4] http://sydney.edu.au/engineering/it/~yangpy/software/EnsembleFilter/20.400.1.txt

III. Results and Discussion

Table 3 summarizes the fit analysis of the proposed method. If *mdr* produced a combination $Z = \{S_1, S_2, \dots, S_k\}$, then we could expect that Z would be ranked near the top in our pre-evaluation list. We then calculated the proportion of the ranking of Z from an ^nC_k size pre-evaluation list. As can be seen, every case showed that Z ranked low in the pre-evaluation list. This meant that highly ranked combinations of SNPs during the pre-evaluation were also highly ranked in the MDR test. Therefore, we could reduce the test cases for *mdr* by adopting highly ranked combinations of SNPs from the pre-evaluation. The PCT measure in particular showed higher efficiency than did

Table 3. Location and proportion that *mdr* result is found in pre-evaluation.

Dataset		Number of selected SNPs			
		$k = 2$	$k = 3$	$k = 4$	$k = 5$
D1	No. of combination	190	1,140	4,845	15,504
	PMI	7 (3.6%)	68 (5.9%)	50 (1%)	1,621 (10.4%)
	PMERIT	7 (3.6%)	68 (5.9%)	50 (1%)	1,621 (10.4%)
	PCT	1 (0.5%)	1 (0.08%)	23 (0.47%)	37 (0.23%)
D2	No. of combination	300	2,300	12,650	53,130
	PMI	74 (25%)	783 (34%)	5,726 (45%)	30,947 (58%)
	PMERIT	74 (25%)	78 (34%)	5,726 (45%)	30,947 (58%)
	PCT	1 (0.33%)	5 (0.34%)	1 (0.008%)	334 (0.6%)
D3	No. of combination	1,225	19,600	230,300	2,118,760
	PMI	74 (24%)	3,560 (18%)	53,926 (23.4%)	636,459 (30%)
	PMERIT	74 (24%)	3,560 (18%)	53,926 (23.4%)	636,459 (30%)
	PCT	1 (0.08%)	31 (0.15%)	355 (0.15%)	4,998 (0.23%)
D4	No. of combination	45	120	210	252
	PMI	1 (2.2%)	6 (5%)	23 (10.9%)	55 (21.8%)
	PMERIT	1 (2.2%)	6 (5%)	23 (10.9%)	55 (21.8%)
	PCT	1 (2.2%)	1 (0.83%)	1 (0.47%)	27 (10.8%)
D5	No. of combination	499,500			
	PMI	1 (0.02%)			
	PMERIT	1 (0.02%)			
	PCT	1 (0.02%)			

k : number of SNPs that we want to find in the MDR test

No. of combination: number of combinations to test

Proportion value of each cell = (ranking of Z) / ^nC_k, where n is the total number of SNPs of a given dataset

Table 4. Comparison of execution time between single-core and multicore algorithms (unit: seconds).

k	Single core (core = 1)	Multicore (core = 6)
2	1.0	0.4
3	1.1	0.5
4	2.0	0.9
5	5.5	2.2
10	156.8	49.0

PMI or PMERIT. Therefore, PCT will be discussed as the proposed pre-evaluation measure in the rest of this paper.

An issue for the proposed method (PCT) is to determine the proportion m in the testing process (see box). A small m allows for a fast calculation of MDR, but it introduces the possibility that the proposed method will produce different results from those of the original MDR. Table 3 shows that $m = 0.02$ or $r < 50$ is enough to produce the correct result. This means that 98% of the combinations can be skipped in the MDR process, thus saving processing time.

We implemented the PCT algorithm in both single-core and multicore environments. Table 4 compares the execution time between single-core and multicore algorithms for the D1 dataset. A multicore algorithm was more efficient than a single-core algorithm when the dimension k was high. For the remainder of the test, we used the PCT measure implemented by the multicore algorithm. Execution time in our test was measured in seconds.

Table 5 and Fig. 3 present the effects of the proposed method. The total execution time of the proposed method consists of a pre-evaluation time plus the *mdr* execution time after pre-evaluation. Pre-evaluation requires extra computation time compared with *mdr* alone, but it can save an enormous amount of time in the execution of *mdr*. Overall, the proposed method reduces the *mdr* processing time. The higher the dimension of the SNP combination, the more time is saved by the proposed method. Figure 3 shows that the proportion of the pre-evaluation segment compared with the total processing time of the proposed method is very small. Therefore, the pre-evaluation is a quick and efficient way to boost MDR.

Tables 6 to 10 show detailed comparisons between the original MDR method and the respective proposed method. In the cases of the higher dimensions, the original MDR method requires an extremely long execution time, and these cases cannot be tested. The parenthesized execution time (see Table 8 and Table 10) was estimated using proportional expression. The recommendation ratio $m = 0.02$ was used for testing the proposed method. If the number of combinations found was less than 50, then the proposed method recommended 50 combinations of SNPs. The results show that the proposed method dramatically reduced the execution time of the original MDR method, whereas the selection lists were almost the same; in two cases, the proposed method and the original MDR resulted in different selection lists.

Table 5. Analysis of execution time on D1 dataset (unit: seconds).

k	*mdr* only	Pre-evaluation	*mdr* after*
2	1.1	0.4	0.3
3	16.1	0.5	0.5
4	21.2	0.9	4.3
5	2206.8	2.2	47.9

*'*mdr* after' refers to the execution time of *mdr* when it tests only the combinations recommended after the pre-evaluation.

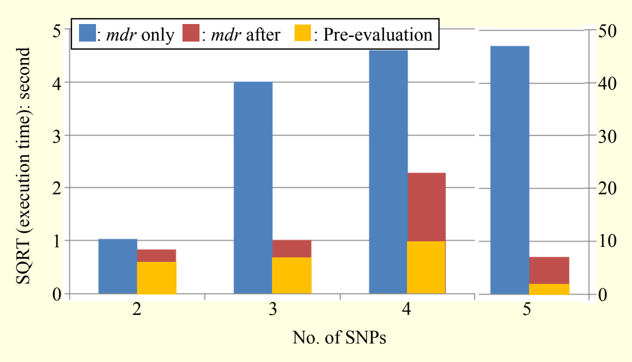

Fig. 3. Effect of proposed method (on execution time).

In the case of $k = 5$ in Table 7, the original MDR produced {11, 16, 19, 24, 25} and the proposed method produced {4, 9, 20, 22, 24}. The classification test determined that the proposed method was more accurate than the original MDR. Some internal error might explain why the results of the MDR and proposed method were much different in the case of $k = 4$. In the case of $k = 2$ in Table 10, the original MDR repeatedly produced irregular results when we repeated the test several more times; the proposed method produced {1, 2}. Yang and others [15] point out that only SNP_1 and SNP_2 are interactive; the other SNPs are generated randomly. Therefore, the proposed method led to the correct result. Because the combinations located by the original MDR in the proposed method were below 1% or were found in the 50th element from the first in the pre-evaluation list of the proposed method, 99% of the combinations found using the MDR test could be excluded.

In Tables 6 to 10, SVM accuracy shows the degree of interaction among given SNP combinations. Although MDR is designed to detect interacting genes (SNPs), we don't know what number of genes to choose. SVM accuracy provides

Table 6. Comparison between proposed method and original MDR method on D1 dataset.

	k	2	3	4	5
	No. of combinations	190	1,140	4,845	15,504
mdr only	Selected SNPs	1, 8	1, 6, 8	1, 2, 6, 8	1, 4, 6, 8, 15
	Execution time (s)	1.1	16.1	21.2	2,206.8
mdr with proposed method	Selected SNPs	1, 8	1, 6, 8	1, 2, 6, 8	1, 4, 6, 8, 15
	Execution time (s)	0.7	1.3	5.2	50.0
	Found location (%)	0.5	0.08	0.47	0.23
SVM accuracy		0.58	0.87	0.84	0.78

SNPs = single-nucleotide polymorphisms

Table 7. Comparison between proposed method and original MDR method on D2 dataset.

	k	2	3	4	5
	No. of combinations	300	2,300	12,650	53,130
mdr only	Selected SNPs	4, 9	4, 6, 9	4, 9, 17, 20	11, 16, 19, 24, 25
	Execution time (s)	3.9	45.1	630.1	7,698.5
mdr with proposed method	Selected SNPs	4, 9	4, 6, 9	4, 9, 17, 20	4, 9, 20, 22, 24
	Execution time (s)	0.7	1.6	12.7	157.0
	Found location (%)	0.33	0.34	0.008	0.6
SVM accuracy		0.66	0.67	0.63	0.51^* 0.67^+

* accuracy for *mdr* only

+ accuracy for *mdr* with proposed method

Table 8. Comparison between proposed method and original MDR method on D3 dataset.

	k	2	3	4	5
	No. of combinations	1,225	19,600	230,300	2,118,760
mdr only	Selected SNPs	4, 9	4, 9, 10	4, 7, 9, 22	N/A
	Execution time (s)	6.4	261.5	9,741.7	(320,700)
mdr with proposed method	Selected SNPs	4, 9	4, 9, 10	4, 7, 9, 22	4, 9, 10, 22, 31
	Execution time (s)	0.92	1.3	221.9	6,414.1
	Found location (%)	0.08	0.15	0.15	0.23
SVM accuracy		0.69	0.65	0.58	0.60

some insight. In Table 6, SVM accuracy is highest when $k = 3$. It means that SNP {1, 6, 8} is the strongest interaction group compared with all the other combinations of SNPs. The higher SVM accuracy means that the SNPs selected by MDR are important.

Figure 4 compares the execution time between the proposed

Table 9. Comparison between proposed method and original MDR method on D4 dataset.

	k	2	3	4	5
	No. of combination	45	120	210	252
mdr only	Selected SNPs	1, 2	1, 2, 8	1, 2, 7, 8	1, 2, 7, 9, 10
	Execution time (s)	0.3	2.0	10.0	37.3
mdr with proposed	Selected SNPs	1, 2	1, 2, 8	1, 2, 7, 8	1, 2, 7, 9, 10
	Execution time (s)	0.7	1.1	2.3	8.6
	Found location (%)	2.2	0.83	0.47	10.7
SVM accuracy		0.77	0.72	0.65	0.59

Table 10. Comparison between proposed method and original MDR method on D5 dataset.

	k	2
	No. of combination	499,500
mdr only	Selected SNPs	N/A
	Execution time (s)	(103,800)
mdr with proposed	Selected SNPs	1, 2
	Execution time (s)	2,076.6
	Found location (%)	0.02
SVM accuracy		0.7325

Table 11. Analysis of execution time on very high dimensional datasets.

Dataset	# of SNPs	Execution time (s)	Proposed	Armstrong	Yang
BD1	10,000	Filtering	14	N/A	775
		MDR	19	N/A	1,553
		Total	**33**	434	**2,323**
BD2	50,000	Filtering	72		3,341
		MDR	16		557
		Total	**88**	**9,000**	**3,898**
BD3	100,000	Filtering	161		3,500
		MDR	17		616
		Total	**177**	**34,980**	**4,116**
BD4	150,000	Filtering	312		11,937
		MDR	21		1,617
		Total	**333**	**93,600**	**13,554**
BD5	200,000	Filtering	405		14,340
		MDR	19		1,500
		Total	**424**	**169,200**	**15,840**

In the case of Amstrong, supported tool just gives total execution time.

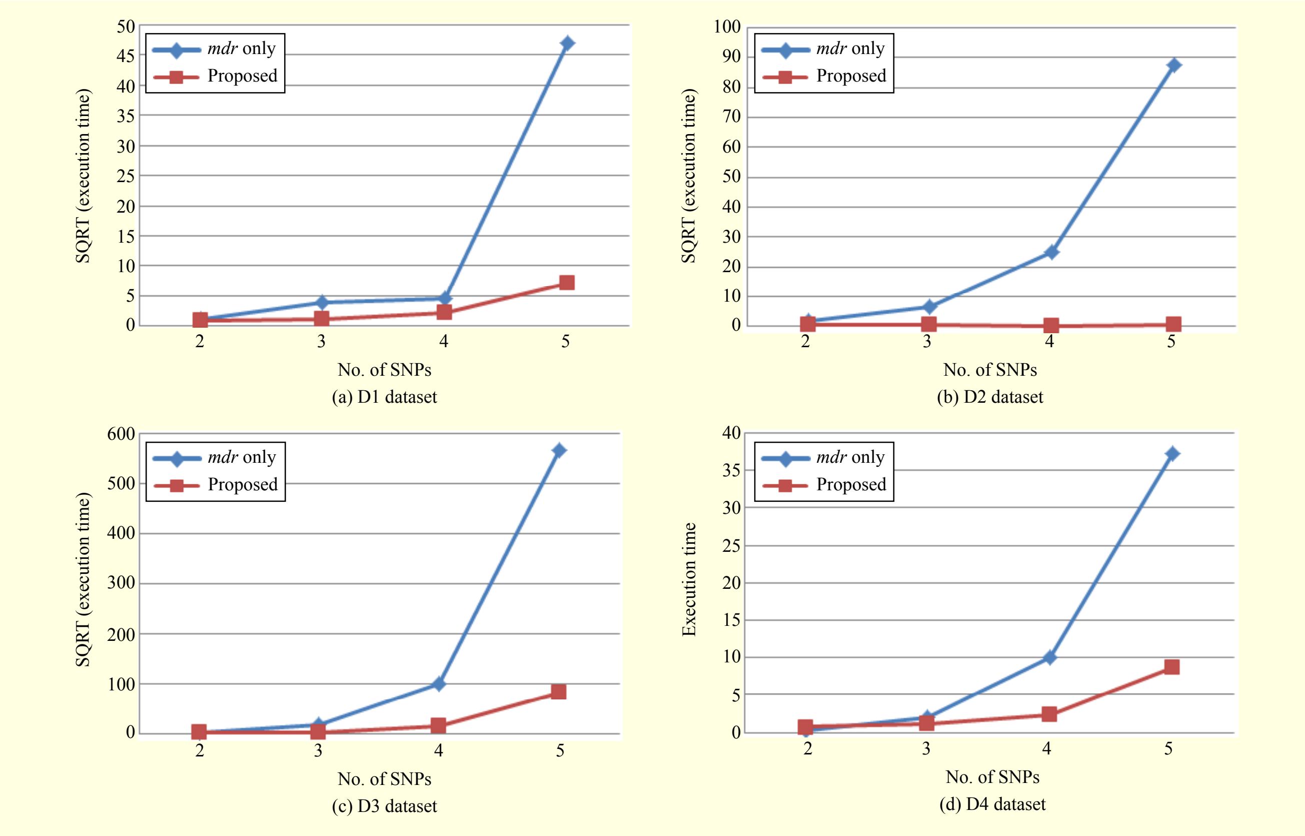

Fig. 4. Comparison of execution time between proposed method and original MDR.

method and the original MDR. The slope of the proposed method increased more slowly when compared with that of the original MDR.

As shown in Table 11, the proposed method works well on those datasets that have very high dimensions, with high efficiency when compared with other methods. We created high-dimensional datasets BD1–BD5 using the D1–D5 datasets. In the new datasets, the first two columns are interacting SNPs and the others are randomly selected non-interacting SNPs from D1–D5. We compared our proposed method with those of Sinnott-Armstrong [13]; Gui and others [14]; and Yang and others [15]. Three algorithms reduced the dimensions of the datasets into 100 columns and the MDR test was performed. The proposed method used MI as a filter. The results of the experiment showed that the proposed method was faster than the other two algorithms in filtering and MDR execution.

IV. Conclusion

In this paper, we have proposed a method for boosting MDR in which target combinations of SNPs are pre-evaluated. We developed the PCT function to exclude valueless combinations of SNPs, which was found to be more efficient than the PMI and PMERIT functions derived from well-known measures of information theory. In addition, the PCT function can work well with other types of MDR methods. Our experiments also showed that our proposed method effectively saves processing time when used along with the original MDR method.

In a practical case (control) study using MDR involving a large number of SNPs, we could apply various filter algorithms to pre-filter, such as ReliefF [20] and BEAM [21], but these filters do not guarantee an optimal solution. The proposed pre-evaluation is a good alternative. Even though the pre-evaluation step is faster than the whole MDR process, the time needed to execute MDR is also greatly increased when the dimension of the SNPs increases because it calculates all combinations of SNPs within the given dimensions. If we reduce the number of SNP combinations and get the same result for all combinations, then we may expect execution time to improve. This topic will require further study. The R code for the proposed method can be found at http://biosw.dankook.ac.kr/biosw/boostMDR/.

References

[1] A.A. Motsinger and M.D. Ritchie, “Multifactor Dimensionality Reduction: An Analysis Strategy for Modeling and Detecting

Gene–Gene Interactions in Human Genetics and Pharmacogenomics Studies," *Human Genomics*, vol. 2, no. 5, Mar. 2006, pp. 318–328.

[2] H.J. Cordell, "Detecting Gene-Gene Interactions that Underlie Human Diseases," *Nature Rev. Genetics*, vol. 10, no. 6, June 2009, pp. 392–404.

[3] M.D. Ritchie et al., "Multifactor-Dimensionality Reduction Reveals High-Order Interactions among Estrogen-Metabolism Genes in Sporadic Breast Cancer," *American J. Human Genetics*, vol. 69, no. 1, July 2001, pp. 138–147.

[4] M.R. Nelson et al., "A Combinatorial Partitioning Method to Identify Multilocus Genotypic Partitions that Predict Quantitative Trait Variation," *Genome Res*., vol. 11, no. 3, Mar. 2001, pp. 458–470.

[5] C.-F. Li et al., "Weighted Risk Score-Based Multifactor Dimensionality Reduction to Detect Gene-Gene Interactions in Nasopharyngeal Carcinoma," *Int. J. Molecular Sci*., vol. 15, no. 6, 2014, pp. 10724–10737.

[6] A.V. Bulinski, "On Foundation of the Dimensionality Reduction Method for Explanatory Variables," *J. Math. Sci*., vol. 199, no. 2, May 2014, pp. 113–122.

[7] X.-Y. Lou et al., "A Generalized Combinatorial Approach for Detecting Gene-by-Gene and Gene-by-Environment Interactions with Application to Nicotine Dependence," *American J. Human Genetics*, vol. 80, no. 6, June 2007, pp. 1125–1137.

[8] W.S. Bush et al., "Alternative Contingency Table Measures Improve the Power and Detection of Multifactor Dimensionality Reduction," *BMC Bioinformatics*, vol. 9, May 2008, p. 238.

[9] H.-M. Xu et al., "Multivariate Dimensionality Reduction Approaches to Identify Gene-Gene and Gene-Environment Interactions Underlying Multiple Complex Traits," *PLoS One*, vol. 9, no. 9, Sept. 2014, pp. 10724–10737.

[10] G-B. Chen et al., "A Unified GMDR Method for Detecting Gene-Gene Interactions in Family and Unrelated Samples with Application to Nicotine Dependence," *Human Genetics*, vol. 133, no. 2, Feb. 2014, pp. 139–150.

[11] J.M. Fisher et al., "Cell-Based Metrics Improve the Detection of Gene-Gene Interactions Using Multifactor Dimensionality Reduction," in *Evol. Comput*., *Mach. Learning Data Mining Bioinformatics*, New York, USA: Springer, vol. 7833, 2013, pp. 200–211.

[12] S. Lee et al., "A Comparative Study on Multifactor Dimensionality Reduction Methods for Detecting Gene-Gene Interactions with the Survival Phenotype," *Biomed Res. Int*., vol. 2015, 2015, pp. 1–7.

[13] N.A. Sinnott-Armstrong, C.S. Greene, and J.H. Moore, "Fast Genome-Wide Epistasis Analysis Using Ant Colony Optimization for Multifactor Dimensionality Reduction Analysis on Graphics Processing Units," *Annual Conf. Genetic Evol. Comput*., Portland, OR, USA, July 7–11, 2010, pp. 215–216.

[14] J. Gui et al., "A Simple and Computationally Efficient Approach to Multifactor Dimensionality Reduction Analysis of Gene-Gene Interactions for Quantitative Traits," *PLoS One*, vol. 8, no. 6, June 2013, pp. 1–7.

[15] P. Yang et al., "Gene-Gene Interaction Filtering with Ensemble of Filters," *BMC Bioinformatics*, vol. 12, no. 1, Feb. 2011, pp. 1–10.

[16] V. Kumar and S. Minz "Feature Selection: A Literature Review," *Smart Comput. Rev*., vol. 4, no. 3, June 2014, pp. 211–229.

[17] E.E. Ghiselli, "*Theory of Psychological Measurement*," New York, USA: McGraw-Hill, 1964, pp. 11–19.

[18] M.A. Hall, *Correlation-Based Feature Selection for Machine Learning*, Ph.D. dissertation, Univ. of Waikato, Waikato, New Zealand, 1999.

[19] S. Amari and S. Wu, "Improving Support Vector Machine Classifiers by Modifying Kernel Functions," *Neural Netw*., vol. 12, no. 6, July 1999, pp. 783–789.

[20] M.A. Province and I.B. Borecki, "Gathering the Gold Dust: Methods for Assessing the Aggregate Impact of Small Effect Genes in Genomic Scans," *Pacific Sym. Biocomp*., Kohala Coast, HI, USA, vol. 13, Jan. 4–8, 2008, pp. 190–200.

[21] Y. Zhang and J.S. Liu, "Bayesian Inference of Epistatic Interactions in Case–Control Studies," *Nature Genetics*, vol. 39, no. 9, Aug. 2007, pp. 1167–1173.

Yingfu Hong received her BS degree in computer science from Yanbian University of Science and Technology, China, in 2013. She is currently an MS degree student at the Department of Nanobiomedical Science, Dankook University, Cheonan, Rep. of Korea. Her main research interests are machine learning algorithms and bioinformatics.

Sangbum Lee received his BS degree in mechanical engineering from Hanyang University, Seoul, Rep. of Korea, in 1983 and his MS and PhD degrees in computer science from Louisiana State University, Baton Rouge, USA, in 1989 and 1992, respectively. He worked as a senior researcher at ETRI from 1992 to 1993. Since October 1993, he has been with Dankook University, Cheonan, Rep. of Korea, as a professor and currently serves as a chairman of the Department of Computer Science. His research interests include software engineering, data mining, big data, and bioinformatics.

Sejong Oh received his BS, MS, and PhD degrees in computer science from Sogang University, Seoul, Rep. of Korea, in 1989, 1991, and 2001, respectively. From 2001 to 2003, he was a postdoctoral fellow with the Laboratory for Information Security Technology, George Mason University, VA, USA. Since 2003, he has worked at the Department of Computer Science, Dankook University, Cheonan, Rep. of Korea and is currently a professor with the Department of Nanobiomedical Science. His main research interests are bioinformatics, information systems, and information system security.