Hindawi Publishing Corporation
Computational and Mathematical Methods in Medicine
Volume 2015, Article ID 353146, 11 pages
http://dx.doi.org/10.1155/2015/353146

Research Article

Finding Top-k Covering Irreducible Contrast Sequence Rules for Disease Diagnosis

Yuhai Zhao,[1] Yuan Li,[1] Ying Yin,[1] and Gang Sheng[2]

[1]*College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China*
[2]*Software Center, Northeastern University, Shenyang, Liaoning 110004, China*

Correspondence should be addressed to Yuhai Zhao; zhaoyuhai@ise.neu.edu.cn

Received 1 October 2014; Accepted 20 January 2015

Academic Editor: Lev Klebanov

Diagnostic genes are usually used to distinguish different disease phenotypes. Most existing methods for diagnostic genes finding are based on either the individual or combinatorial discriminative power of gene(s). However, they both ignore the common expression trends among genes. In this paper, we devise a novel sequence rule, namely, top-k irreducible covering contrast sequence rules (Top$k$IRs for short), which helps to build a sample classifier of high accuracy. Furthermore, we propose an algorithm called MineTop$k$IRs to efficiently discover Top$k$IRs. Extensive experiments conducted on synthetic and real datasets show that MineTop$k$IRs is significantly faster than the previous methods and is of a higher classification accuracy. Additionally, many diagnostic genes discovered provide a new insight into disease diagnosis.

1. Introduction

It has been proved that many diseases are closely related with genes [1–3]. In bioinformatics, such genes are called *diagnostic genes*. Capturing these genes is an important task, which helps in diagnosis, prediction, and treatment of diseases [4].

According to biological theory, only a small number of genes are directly related with a certain disease [5]. Biologists always want to exploit fewer genes to provide higher disease prediction accuracy. In practice, how to pick out these diagnostic genes to distinguish different disease phenotypes from a massive amount of gene expression data is often an intractable problem.

Many studies have shown that contrast rules are very promising for this problem. Contrast rules refer to the rules that frequently appear in one class but rarely in other classes, denoted as $X \rightarrow C$, where X represents the diagnostic genes and C represents a certain disease. Most of such methods can be divided into two categories, that is, single discrimination based [6] and combinatorial discrimination based [7]. The former evaluates every gene according to their individual discriminative power to the target classes and then selects top-ranked genes. The latter often models the problem as a subset search problem and focuses on the combinatorial discriminative power of a set of genes. However, neither of the two exploits the relationship among genes such that some important diagnostic genes may be missed.

In this paper, we tackle the problem by utilizing the order relationship among genes. Below is a real example for an immediate comprehension to our basic idea.

Example 1. Figure 1 consists of two subfigures. In the top subfigure, 4 genes are expressed over 25 samples. Samples 1~16 are cancerous (labeled as "C") and samples 17~25 are normal (labeled as "N"). In the bottom subfigure, another set of 3 genes is expressed over the same set of samples. The existing singleton or combination discriminability-based methods cannot distinguish the two phenotypes. Since most genes are of similar average expression values in the two phenotypes, they will not be selected by the singleton approach. Moreover, all genes are expressed in both phenotypes. Thus, the combination approach based on the cooccurrence of genes will not select them either. Both of the methods ignore the hidden interrelation among genes. In the top subfigure, the gene order over the samples of cancerous phenotype "C"

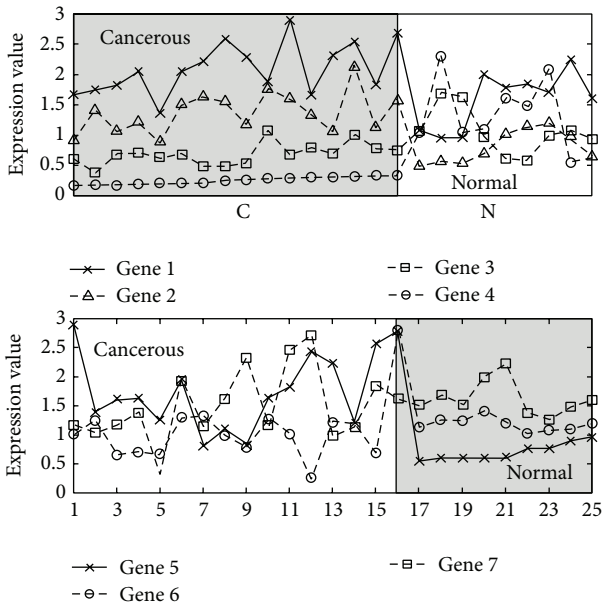


Figure 1: A real example from the prostate cancer dataset.

is always $gene_4 \prec gene_3 \prec gene_2 \prec gene_1$. Such order is disturbed in normal phenotype "N". In the bottom subfigure, the gene order in normal phenotype "N" is $gene_5 \prec gene_6 \prec gene_7$, while in cancerous phenotype "C" such order does not exist. Based on the ordered expression values, the disease phenotypes (the two shadowed "blocks") are well identified.

Example 1 indicates that *contrast sequence rules* may be a promising solution to the mentioned problem. Another advantage of incorporating the sequence rule into diagnostic gene finding is that we may obtain higher disease prediction accuracy by fewer genes. This is intuitively because the order contains both individual information and combinatorial information. In [8], we proposed a contrast sequence rules mining algorithm, namely, NRMINER, and showed its effectiveness and efficiency. However, there are still some issues demanding a further consideration.

Given n genes, there is up to 2^n subsets of genes. Moreover, each subset of i genes corresponds to $i!$ permutations. Thus, the number of contrast sequence rules is at least $\sum_{i=1}^{n}(C_n^i \cdot i!) \gg n!$ in theory. On one hand, massive rules pose a crucial challenge for biologists to interpret and validate the results. On the other hand, this may take too much time such that the proposed method is not practically feasible. In practice, we often need only a small set of representative contrast sequence rules instead of all the rules. This is also the so-called *top-k problem* in database and data mining communities. Accordingly, the goal of this paper is to discover *top-k covering irreducible contrast sequence rules* (*TopkIRs* for short) from a given gene expression dataset.

Compared with the existing methods, our contributions in this paper are claimed as follows.

(1) We propose the concept of top-k covering irreducible contrast sequence rule, which greatly reduces the burden for biologist to interpret and validate the results and practically enables an efficient diagnostic gene finding method.

(2) We devise the criteria of ranking irreducible contrast sequence rules. Based on the criteria, we can pick out shorter and fewer but more representative rules to build classifier with higher classification accuracy.

(3) We develop a novel algorithm called MineTop$k$IRs to directly discover top-k covering irreducible contrast sequence rules without postprocess. As we know, few works address this problem in the context of sequence mining.

The rest of this paper is organized as follows. In Section 2, we introduce some preliminaries and give our problem definition. Section 3 introduces the criteria of ranking rules. Section 4 details the MineTop$k$IRs algorithm. Section 5 includes the experimental results and analysis. Finally, Section 6 concludes this paper.

2. Preliminary

In this section, we first introduce some basic concepts useful for further discussion and then formalize the problem to be addressed in this paper.

2.1. Basic Concepts. A microarray dataset D is an $m \times n$ matrix, with m samples $S = \{s_1, s_2, \ldots, s_m\}$ and n genes $G = \{g_1, g_2, \ldots, g_n\}$. A real value d_{ij} in D represents the expression value of gene g_j on sample s_i. An example microarray dataset of 7 genes and 6 samples is shown in Table 1, where the last column lists the class label for each sample.

As mentioned, we want to tackle the problem from the gene order perspective. Accordingly, we propose the EWave model, a sequence model to represent the gene expression data. Next are some necessary concepts.

Definition 2. Given an expression matrix D of a sample set, $S = \{s_1, s_2, \ldots, s_m\}$, and a gene set, $G = \{g_1, g_2, \ldots, g_n\}$, if for a grouping threshold δ, $\delta \geq 0$, and some sample $s_i \in S$, there exists a subset, G', of genes holding both (1) and (2), we say G' is an equivalent dimension group, or an EDG in short, of the sample s_i:

$$\max_{g_j, g_{j'} \in G'} \left| d_{ij} - d_{ij'} \right| < \delta \times \min_{g_j \in G'} d_{ij}, \tag{1}$$

$$\forall g_i, g_j \in G', \quad \min_{g_{j'} \in G'} \left| d_{ij} - d_{ij'} \right| < \min_{g_{j''} \in (G - G')} \left| d_{ij} - d_{ij''} \right|. \tag{2}$$

Specifically, we call a gene satisfying (1) but excluded from an EDG by (2) a "breakpoint." The method of creating EDGs is detailed in [8]. It is worthy to note that no order is considered in an EDG, where the expression values have no significant differences.

An EWave model can be used to represent the sequences of EDGs. Figure 2 shows the EWave model corresponding to the running example in Table 1, where $\delta = 0.5$. In each row i of

Table 1: An illustrative expression matrix with sample labels.

Sample	g_1	g_2	g_3	g_4	g_5	g_6	g_7	tag
s_1	2.2	1.3	0.8	2.38	1.44	0.3	0.48	C_1
s_2	3.3	1.25	2.54	6.3	2.3	0.62	1.4	C_1
s_3	1.26	6.6	3.1	5.4	5.62	0.94	1.72	C_1
s_4	4.3	0.34	7.2	7.1	1.9	2.1	2.66	C_2
s_5	2.78	0.62	5.1	1.86	1.74	1.34	2.92	C_2
s_6	1.1	2.85	2.1	0.48	2.4	0.52	2.33	C_2

an EWave model, all genes are increasingly ordered according to their expression values on sample s_i, and the pointer pointing from g_a to g_b indicates an EDG starting at g_a and ending at g_b. We omit the pointers pointing to a gene itself.

Different from the other traditional sequence-like data, since the overlap among different EDGs is allowed in the EWave model, a gene in an EDG can also belong to several other EDGs at the same time. Given a sample s_i and a gene g_a, the sequence of EDGs of s_i is denoted as $\$_i$. Then, we call the index of the first EDG in $\$_i$ containing g_a the head position of g_a with respect to s_i and the index of the last EDG in $\$_i$ containing g_a the tail position of g_a with respect to s_i, denoted as $H_i(g_a)$ and $T_i(g_a)$, respectively.

Example 3. For s_3 in Figure 2, the head position of g_1, $H_3(g_1)$, is 2, and the last position of g_1, $T_3(g_1)$, is 3.

Definition 4. Let \$ be a sequence of EDGs in an EWave model, where the order of genes is $g_{i_1} g_{i_2}, \ldots, g_{i_n}$. Then, we call a gene sequence $\mathcal{G} = h_{j_1} h_{j_2}, \ldots, h_{j_t}$ is contained by \$, denoted as $h \sqsubseteq \$$, if there exist the integers $1 \le k_1 \le k_2 \le \cdots \le k_t \le n$ such that $h_{j_1} = g_{i_{k1}}, h_{j_2} = g_{i_{k2}}, \ldots, h_{j_t} = g_{i_{kt}}$. Further, we refer to the gene sequence $\mathcal{G}$, where any pair of genes are not in the same EDG, as a *significant chain*.

Example 5. In Figure 2, $g_6 g_7 g_5$ is a significant chain of $\$_1$, but $g_6 g_2 g_5$ is not, since g_2 and g_5 coexist in the same EDG.

As mentioned above, we aim to capture the difference among different sample phenotypes from a sequence point of view. Thus, the benefit of EWave model has two aspects. On one hand, not only the gene expression data are very noisy, but also sometimes the gene expression values are very close. If we only consider the significant chain, the difference between genes is large enough so that the difficulty to determine the order among genes is overcome. On the other hand, the high dimension of gene expression data is largely reduced at the same time. Next, we introduce some concepts related with the contrast sequence rule under the EWave model.

Definition 6. Let D be EWave modeled gene expression data. Then, for a given sequence rule γ, denoted as $X \to C$, where X is a significant chain and C is a given class label, the support of γ is defined as the number of the sequences of EDGs in D containing XC, denoted as $\text{supp}(\gamma)$ and the sample support set of γ denoted as $R(\gamma)$. The confidence of γ is defined as the ratio of the number of the sequences of EDGs containing XC to that of the sequence of EDGs containing X, denoted as $\text{conf}(\gamma) = \text{supp}(XC)/\text{supp}(X)$.

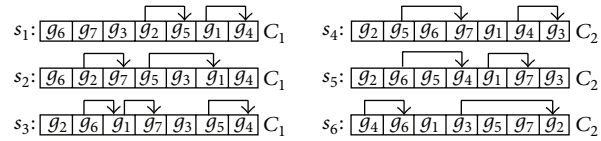


Figure 2: The EWave model of data in Table 1.

Example 7. In Figure 2, let γ be the rule $g_7\, g_4 \to C_1$. Then, $R(\gamma) = \{s_1, s_2, s_3\}$ and $\text{supp}(\gamma) = 3$. Further, since $\text{supp}(g_7 g_4) = 4$, $\text{conf}(\gamma) = 3/4 = 75\%$.

Definition 8. Let D be an EWave modeled gene expression dataset and C a specified class label. $\text{RG} = \{X_i \to C \mid \exists s \in D, X_i \sqsubseteq \$\}$ is a *rule group* with antecedent support set R and consequent C, iff (1) $\forall X_i \to C \in \text{RG}$, $R(X_i) = R$ and (2) $\forall R(X_i) = R$, $X_i \to C \in \text{RG}$.

Example 9. In Figure 2, $R(g_6 g_7 g_3 g_4) = R(g_6 g_7 g_4) = R(g_6 g_3 g_4) = R(g_7 g_3 g_4) = R(g_3 g_4) = \{s_1, s_2, s_3\}$. Thus, they make up a rule group $\text{RG} = \{g_6 g_7 g_3 g_4 \to C_1,\ g_6 g_7 g_4 \to C_1,\ g_6 g_3 g_4 \to C_1,\ g_7 g_3 g_4 \to C_1,\ g_3 g_4 \to C_1\}$ with antecedent support set $\{s_1, s_2, s_3\}$ and a specified class label C_1.

In this paper, we want to use the contrast sequence rules to distinguish the sample phenotypes. However, the number of contrast sequence rules in the dataset is prohibitively large, and most of them are redundant. Discovering all contrast sequence rules is inefficient and trivial. Thus, we propose the concept of irreducible contrast sequence rule, which is more concise and representative.

Definition 10. Let D be an EWave modeled gene expression dataset. A sequence rule γ in the form of $X \to C$ is called a contrast sequence rule if $\text{supp}(\gamma)$ and $\text{conf}(\gamma)$ are no less than the minimum support threshold α and the confidence threshold β, respectively, where X is a sequence and C is a class label.

Example 11. Suppose $\alpha = 3$ and $\beta = 90\%$. Then, γ: $g_6 g_7 g_3 g_4 \to C_1$ in Figure 2 is a contrast sequence rule since $\text{supp}(\gamma) = 3 \ge \alpha$ and $\text{conf}(\gamma) = 100\% > \beta$.

Definition 12. For any given contrast sequence rule γ: $X \to C$ of $\text{conf}(\gamma) = \beta$, we call it an irreducible contrast sequence

rule if any of $X' \rightarrow C$ ($X' \sqsubseteq X$) has $\text{conf}(X' \rightarrow C) < \beta$. In other words, any subrule of a contrast sequence rule γ should not be a contrast sequence rule.

Example 13. γ: $g_6g_7g_3g_4 \rightarrow C_1$ in Figure 2 is not an irreducible contrast sequence rule since there exists a subrule of γ, say γ': $g_6g_7g_4 \rightarrow C_1$, such that $\text{conf}(\gamma') \geq \text{conf}(\gamma) = 100\%$.

Definition 14. Given D, an EWave modeled gene expression dataset, the top-k covering irreducible contrast sequence rules for a sample s_i is the set of rules $\{\gamma_{s_i,j}\}$ $(1 \leq j \leq k)$, where the antecedent of $\gamma_{s_i,j}$ is contained by s_i, $\forall x, y \in k$, $R(\gamma_{s_i,x}) \neq R(\gamma_{s_i,y})$ and there exists no rule γ', $\gamma' \notin \{\gamma_{s_i,j}\}$ such that γ' can substitute any rule in $\{\gamma_{s_i,j}\}$ based on the rule priority. For brevity, we will use the abbreviation Top$k$IRs to refer to top-k covering irreducible contrast sequence rules for each sample.

Example 15. Suppose $k = 2$. Then, for sample s_1 in Figure 2, the top-k covering irreducible contrast sequence rules is the set of rules $\{\gamma_{s_1,1}$: $g_7g_5 \rightarrow C_1$, $\gamma_{s_1,2}$: $g_3g_4 \rightarrow C_1\}$. This is because (1) $s_1 \in R(\gamma_{s_1,1})$ and $s_1 \in R(\gamma_{s_1,2})$, (2) both $\gamma_{s_1,1}$ and $\gamma_{s_1,2}$ are irreducible contrast sequence rules, and (3) there is no other rule γ' which can substitute $\gamma_{s_1,1}$ or $\gamma_{s_1,2}$ due to $\text{conf}(\gamma_{s_1,1}) = \text{conf}(\gamma_{s_1,2}) = 100\%$.

2.2. Problem Description. Given (1) a gene expression dataset D where each sample is attached with a unique class label, (2) the equivalent threshold δ, (3) the minimum support threshold α, and (4) the confidence threshold β, the problem is to efficiently discover the set of top-k covering irreducible contrast sequence rules for each sample.

3. Criteria of Ranking Rules

In this section, we introduce the criteria of ranking rules. In order to evaluate the (dis)similarity between sequences, we propose the concept of projection distance which is more suitable for EWave modeled gene expression data. The reason is that projection distance takes into account not only the difference on the same position between two sequences but also the displacement between the two items.

Assume o_i is a gene sequence and \$ is the gene sequence corresponding to sample s, the *projection* of o_i on \$, denoted as $o_i \mid s$, refers to the sequence of all elements in o_i, permuted according to their relative orders in \$. Further, if a pair of items in o_i, denoted as (x, y), has the reversal relative order in $o_i \mid s$, we call it *a reverse pair*. Then, for an item x, if it is at the kth locus in o_i and at the jth locus in $o_i \mid s$, we call $|k - j|$ the *displacement* of x between o_i and $o_i \mid s$, denoted as $\text{dist}_x(o_i, s)$.

Definition 16. Given a gene sequence o_i and the sequence \$ corresponding to sample s, the *projection distance* between o_i and $o_i \mid s$ is defined by the following formula:

$$\text{PD}(o_i, o_i \mid s) = \sum_{\substack{x,y \in o_i \\ x \neq y}} \phi(x, y) \left[\text{dist}_x(o_i, s) + \text{dist}_y(o_i, s)\right], \tag{3}$$

where $\phi(x, y)$ is a Boolean function expressed as $\phi(x, y) = 1$, if (x, y) is a reversal pair; otherwise, $\phi(x, y) = 0$.

Now, we adopt a similarity function defined based on the concept of projection distance (or simply PD) to identify the (dis)similarity between a sequence and its projection on sample s. The similarity function is formally defined as follows.

Definition 17. Given a gene sequence o_i and the gene sequence S corresponding to sample s, the *PD similarity* between o_i and $o_i \mid s$, denoted as $\text{Sim}_{\text{PD}}(o_i, o_i \mid s)$, is defined as

$$\text{Sim}_{\text{PD}}(o_i, o_i \mid s) = 1 - \frac{\text{PD}(o_i, o_i \mid s)}{\sum_{j=1}^{o_i} (|o_i| + 1 - j) * (|o_i| - j)}, \tag{4}$$

where $|o_i|$ is the length of gene sequence o_i.

From (4), we can find that the smaller the projection distance between two sequences, the more the similarity of the sequences. If $\text{PD}(O_1, O_2) = 0$, $\text{Sim}_{\text{PD}}(o_1, o_2) = 1$, which means the two sequences are totally the same. Next, we introduce the criteria of ranking rules with two cases.

Definition 18. The priority within the same rule group: given two rules γ_1: $X_1 \rightarrow C$, γ_2: $X_2 \rightarrow C$, and $R(\gamma_1) = R(\gamma_2)$, we say γ_1 is prior to γ_2 if

$$\frac{\sum_{s' \in (S - R(\gamma_1))} \text{Sim}_{\text{PD}}(X_1, X_1 \mid s')}{|S - R(\gamma_1)|} < \frac{\sum_{s' \in (S - R(\gamma_2))} \text{Sim}_{\text{PD}}(X_2, X_2 \mid s')}{|S - R(\gamma_2)|}. \tag{5}$$

From (5), we can conclude that the more the antecedent of the rule is different from the gene sequence in the nonsupport set, the higher the priority the rule has.

Example 19. In Figure 2, the support sets of rules $\langle g_3g_4\rangle \rightarrow C_1$ and $\langle g_7g_5\rangle \rightarrow C_1$ are both $\{s_1, s_2, s_3\}$, but based on (5), $(0 + 0 + 0) < (0 + 0 + 1/3)$, so $\langle g_3g_4\rangle \rightarrow C_1$ is more prior than $\langle g_7g_5\rangle \rightarrow C_1$.

Definition 20. The priority between rule groups: given two rules γ_1: $X_1 \rightarrow C$, γ_2: $X_2 \rightarrow C$, and $R(\gamma_1) \neq R(\gamma_2)$, we say γ_1 is prior to γ_2, if and only if one of the following three conditions satisfied: (1) $\text{conf}(\gamma_1) > \text{conf}(\gamma_2)$; (2) $\text{conf}(\gamma_1) = \text{conf}(\gamma_2)$ and $\text{supp}(\gamma_1) > \text{supp}(\gamma_2)$; (3) $\text{conf}(\gamma_1) = \text{conf}(\gamma_2)$, $\text{supp}(\gamma_1) = \text{supp}(\gamma_2)$ and γ_1 is discovered before γ_2.

Example 21. In Figure 2, assume γ_1: $\langle g_3g_4\rangle \rightarrow C_1$, γ_2: $\langle g_6g_5g_4\rangle \rightarrow C_1$, and γ_3: $\langle g_7g_4\rangle \rightarrow C_1$. Because $\text{conf}(\gamma_1) = \text{conf}(\gamma_2) = 100\%$ and $\text{supp}(\gamma_1) = 3$, which is higher than that of $\text{supp}(\gamma_2) = 2$, γ_1 is more prior than γ_2. Also, $\text{conf}(\gamma_3) = 75\%$ and $\text{conf}(\gamma_2) > \text{conf}(\gamma_3)$, so γ_2 is more prior than γ_3.

4. The MineTop$k$IRs Algorithm

In this section, we present our algorithm, called MineTop-kIRs, to solve the problem given in Problem Statement. First,

Table 2: The Head-Tail matrix of gene expression data.

Sample	g_1	g_2	g_3	g_4	g_5	g_6	g_7	tag
s_1	5.5	4.4	3.3	5.5	4.4	1.1	2.2	C_1
s_2	3.3	2.2	3.3	4.4	3.3	1.1	2.2	C_1
s_3	2.3	1.1	4.4	5.5	5.5	2.2	3.3	C_1
s_4	3.3	1.1	4.4	4.4	2.2	2.2	2.2	C_2
s_5	3.3	1.1	4.4	4.4	2.2	2.2	2.2	C_2
s_6	2.2	3.3	3.3	1.1	3.3	1.1	3.3	C_2

we give a naive method to construct classifier based on contrast sequence rules.

Step 1. Discover all the frequent sequence patterns with a low minimum support threshold.

Step 2. Combine each sequence pattern with a class label to generate a sequence rule. Then, pick out the contrast sequence rule with highest confidence for each sample in the dataset.

Obviously, this naive two-step mining method generates too many rules in Step 1, which takes too long time. Moreover, selecting only one rule for each sample is often not enough. Instead, our algorithm is one-pass process, which is much more efficient. Further, each sample is guaranteed to be covered by top-k irreducible contrast sequence rules. In what follows, we detail the proposed MineTop$k$IRs algorithm.

4.1. Head-Tail Matrix. The Head-Tail matrix M is a useful structure to accelerate the detection whether a sequence is a significant chain corresponding to some sample template sequence $\$_i$, which is a necessary condition of the antecedent of a contrast sequence rule. Table 2 gives the Head-Tail matrix corresponding to the model shown in Figure 2, where each row represents a considered sample, and each column represents a remained gene. Every entry (i, j) in the matrix M records a two-dimensional vector (x, y), where x denotes the head position of the gene g_j in $\$_i$, that is, $H_i(g_j)$, and y denotes the tail position of the gene g_j in $\$_i$, that is, $T_i(g_j)$. For example, in Figure 2, $H_3(g_1) = 2$ and $T_3(g_1) = 3$, so the entry at row 3 and column 1 of Table 2 records $(2, 3)$.

An efficient way to decide whether a sequence X is a significant chain with respect to $\$_i$ is that we only consider any neighboring pair of genes such as g_a and g_b; if $T_i(g_a) < H_i(g_b)$ is always true, we say that X must be a significant chain for $\$_i$, which is the sequence of EDGs of sample s_i. *Note*: While computing the support of a gene sequence, we use the Head-Tail matrix with $\delta > 0$, which makes the order between genes in the sequence significant enough. However, when computing the projection distance of a gene sequence for some $\$_i$, we use the Head-Tail matrix with $\delta = 0$, which makes the displacement of a reverse pair easily determined.

4.2. The Mining Algorithm. The search space of enumerating all gene sequences is prohitably large. Thus, a suitable traversal framework with some effective pruning strategies is necessary.

In this paper, we adopt a breadth-first traversal framework. As we know, most sequence pattern mining methods such as BIDE [9] and FEAT [10] adopt a depth-first traversal. The goodness is that exploiting the antimonotonicity of support, the depth-first traversal can directly prune searching space based on the current sequence without generating candidate set. However, depth-first traversal is not suitable to solve the problem raised in this paper. The reason is that (1) the confidence of irreducible contrast sequence rule is not antimonotonic, which requires us to detect whether all subrules of the current rule satisfy the conditions defined in Definition 12 that is the confidence of all subrules below β. For example, suppose the length of current sequence rule is l, we need to detect all the subrules, which shows the computation is very large. (2) Under the premise of not establishing access rules index, it is possible to repeatedly access many rules. The abovementioned two cases are very time-consuming. On the contrary, the breadth-first traversal can be a good solution to the problem mentioned above. We only need to detect whether all the $(l - 1)$-size subrules meet the conditions. Further these subrules can be obtained by directly accessing the current rule candidate set which is more efficient.

Formally, the algorithm is shown in Algorithm 1. There are four input parameters of the algorithm, the original dataset D, equivalent threshold δ, the minimum support α, and confidence threshold β. Because of solving the problem in gene sequence perspective, the algorithm will first transform D into the EWave model D and then construct the Head-Tail matrix which can accelerate the calculation of rule support. At the same time, the top-k covering irreducible contrast sequences rules for each sample s_i with consequent C, denoted as $\zeta_{s_i} = [\gamma_{s_i1}, \ldots, \gamma_{s_ik}]$, will be initialized. Also, we put all the 1-size rules that consist of single gene into rule candidate set *Candi_R*. Then the function *breathfirst_search* is called to perform the breath-first traversal to find out the top-k rules for each sample.

The function *breathfirst_search* takes in four parameters: the rule candidate set *Candi_R*, minimum support α, confidence threshold β, and the size of rule l. When the algorithm executes to the l level, it generates all the $(l + 1)$-size rules based on the rules in *Candi_R* (line 2). For each $(l + 1)$-size rule, the algorithm is based on three pruning rules (lines 4, 6, and 11) to detect whether it will be put into *Candi_R* for further extension (line 8) or used to update the top-k covering rules for samples in its support set (line 13) or just be pruned. It is worth noting that the confidence of all the rules in *Candi_R* must be below β because once the

Input: a $m \times n$ gene expression dataset D; the required number of rules k; the equivalent threshold δ; the support threshold α; the confidence threshold β
Output: All top-k covering irreducible contrast sequence rules $\zeta_{s_i} = [\gamma_{s_i1}, \ldots, \gamma_{s_ik}]$ for each sample s_i with class label C
(1) Convert dataset D into the EWave model D, w.r.t. δ;
(2) Construct Head-Tail matrix;
(3) Initiate a list of k rules with both support and confidence values of 0, $\zeta_{s_i} = [\gamma_{s_i1}, \ldots, \gamma_{s_ik}]$ for each sample s_i with class label C;
(4) Initiate the rule candidate set *Candi_R* with all 1-*size* sequence rules;
(5) Call *breathfirst_search*(*candi_R*, α, β, 1);
(6) Return ζ_{s_i} for every s_i with class label C;
Function: *breathfirst_search*(*candi_R*, α, β, l)
(1) while *candi_R* $\neq \phi$ do
(2) foreach $l+1$-*size rule* γ' *generated based on the* l-*size rules in candi_R* do
(3) if $\forall l$-*size subrule of* γ' *exists in candi_R* then
(4) if $supp(\gamma') > \alpha$ then *Pruning rule* 1;
(5) if $conf(\gamma') < \beta$ then
(6) *Pruning rule* 2;
(7) add γ' into *candi_R*;
(8) else
(9) Check the kth covering rule γ_{s_ik} for each sample $s_j \in R(\gamma')$ to find the lowest confidence *minconf* and the corresponding support *sup*;
(10) if $(conf(\gamma') > minconf) \vee (conf(\gamma') = minconf \wedge supp(\gamma') \geq sup)$ then
(11) *Pruning rule* 3;
(12) Update $\zeta_{s_i} = [\gamma_{s_i1}, \ldots, \gamma_{s_ik}]$ for each sample $s_j \in R(\gamma')$ with γ' based on Definitions 18 and 20;
(13) end
(14) end
(15) Delete all the l-*size* rules in *candi_R*;
(16) l++;
(17) end

Algorithm 1: The MineTop$k$IRs Algorithm.

confidence of a rule exceeds β, all the super rules of it cannot be irreducible contrast sequence rules. After the end of each loop, the algorithm deletes the whole l-size rules in *Candi_R* (line 17). The algorithm ends when *Candi_R* = ϕ (line 1).

4.2.1. Pruning Strategies. We next illustrate the pruning techniques that are used in MineTop$k$IRs. With the help of these pruning rules, we can find out the top-k covering irreducible contrast sequence rules for each sample efficiently.

Pruning Rule 1. Let γ: $X \rightarrow C$ be the current considered sequence rule; if there exists a sequence rule γ': $X' \rightarrow C$, $X' \sqsubseteq X$, and $\text{conf}(\gamma') > \beta$, the rule itself and all its super rules can be pruned.

Proof. Based on the definition of irreducible contrast sequence rule, if a sequence rule γ: $X \rightarrow C$ is irreducible contrast sequence rule, it requires that $\forall \gamma'$: $X' \rightarrow C$ ($X' \sqsubseteq X$), $\text{conf}(\gamma') < \beta$. Thus, if any of its subrules γ': $X' \rightarrow C$ do not satisfy this condition, the sequence rule γ: $X \rightarrow C$ cannot be an irreducible contrast sequence rule. Similarly, none of its super rules can be irreducible contrast sequence rules. □

Specific to our algorithm, we store each rule whose confidence and all its subrules' confidence are below β in *Candi_R* for further extension. When deciding whether a newly generated l-size rule is to be pruned, we only need to test if all of its $(l-1)$-size subrules are in *Candi_R*. If not, we can safely prune this sequence rule and all its super rules.

Pruning Rule 2. Let γ: $X \rightarrow C$ be the current considered sequence rule and α the minimum support threshold. If $\text{supp}(X \rightarrow C) < \alpha$, then the current rule γ and all its super rules are pruned.

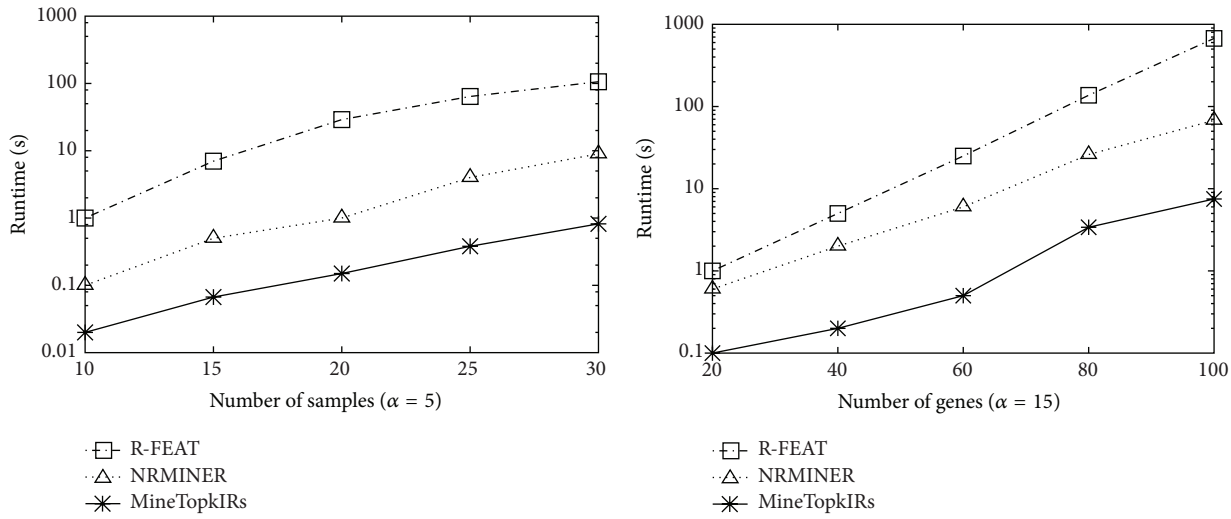
Proof. It is immediately derived from the a priori property of sequence [11] and Definition 12. □

In MineTop$k$IRs, we can use the constraint of top-k to prune rules. Combined with Definition 20, we compute minconf and sup, the critical point of Top$k$IRs thresholds for the samples in $R(\gamma)$, where minconf is the minimum confidence value of the discovered Top$k$IRs of all the samples in $R(\gamma)$ and sup is the corresponding support. Assume the top-k covering irreducible contrast sequence rules of each sample s_i are ranked according to the priority between rule groups such that $\gamma_{s_i1} \prec \gamma_{s_i2} \prec \cdots \prec \gamma_{s_ik}$:

$$\begin{aligned} \text{minconf} &= \min_{s_i \in R(\gamma)} \left\{ \text{conf}\left(\gamma_{s_ik}\right) \right\}, \\ \text{sup} &= \text{sup}\left(\gamma_{s_xk}\right), \quad \text{where conf}\left(\gamma_{s_xk}\right) = \text{minconf}. \end{aligned} \tag{6}$$

Table 3: The information of gene expression data.

Dataset	# sample	# gene	C_1	C_2	$C_1 : C_2$
Leukemia	38	5000	ALL	AML	27 : 11
DLBCL	77	7129	DLBCL	FL	58 : 19
HBC	22	3326	BRCA	Sporadic	15 : 7
PC	25	6500	Cancer	BPH	16 : 9



(a) Varying #*sample* ($\alpha = 5$)

(b) Varying #*gene* ($\alpha = 15$)

Figure 3: Scalability.

Pruning Rule 3. Given the current considered sequence rule γ: $X \rightarrow C$ and $\text{conf}(\gamma) \geq \beta$, minconf and sup computed according to (6), if the rule is less prior based on the priority between rule groups (Definition 20) than $\gamma_{s_x k}(\text{conf}(\gamma_{s_x k}) = \text{minconf}, \text{sup} = \text{sup}(\gamma_{s_x k}))$, then the rule γ and all its super rules cannot become a rule in the top-k covering irreducible contrast sequence rules list of any sample and can be safely pruned.

If the current sequence rule γ: $X \rightarrow C$ cannot be pruned by Pruning Rule 3, there are two situations. On one hand, $\forall s_i \in R(\gamma)$ when there are no rules in $\{\gamma_{s_i 1}, \ldots, \gamma_{s_i k}\}$ that have the same sample support set as that of γ, we only need to detect if γ is more prior than $\gamma_{s_i k}$, if so, we substitute $\gamma_{s_i k}$ for γ. On the other hand, because in this paper we want to find out top-k rules for each sample with different sample support sets, $\forall s_i \in R(\gamma)$ when there is some rule in $\{\gamma_{s_i 1}, \ldots, \gamma_{s_i k}\}$ that has the same sample support set as that of γ, we need to find out that if γ is more prior than the rule has the sample support set with γ based on the priority within the same rule group (Definition 18), if so, we replace this rule with γ which can guarantee the current rules in $\{\gamma_{s_i 1}, \ldots, \gamma_{s_i k}\}$ have the highest priority.

In addition, another optimization method is utilized in Pruning rule 3. If we find all Top$k$IRs have 100% confidence and the lowest support value of k rules is larger than α, we dynamically increase the user-specified support threshold.

5. Performance Studies

In this section, we will look at both the efficiency of our algorithm in discovering Top$k$IRS and the usefulness of the discovered rules. All our experiments were performed on a HP PC with 2.33 GHz Intel Core 2 CPU, 2 GB RAM, and a 160 GB hard disk running Windows XP. Algorithms were coded in Standard C.

Datasets. Four real gene expression datasets for experimental studies: Leukemia [1], DLBCL Tumor [2], Hereditary Breast Cancer (HBC) [3], and Prostate Cancer (PC) [12]. Table 3 shows the characteristics of the four datasets: the number of samples (#sample), the number of genes (#gene), and the label of class i (C_i). The number of samples in every class is shown in the last column. Moreover, we generate the synthetic datasets by using a specialized dataset generator [8].

5.1. Efficiency of MineTopkIRs. In term of efficiency, we compare MineTop$k$IRs with R-FEAT and NRMINER [8]. On one hand, R-FEAT is changed from the sequence generator mining algorithms FEAT [10]. Briefly, we apply FEAT on a given dataset, when a generator X is found, we decide whether $X \rightarrow C$ could be a result by checking all rules $X' \rightarrow C$, where $X' \sqsubseteq X$, satisfying the conditions based on Definition 14. On the other hand, the NRMINER algorithm adopts the template driven method to find out all the interesting nonredundant contrast sequence rules, which are necessary for checking whether the conditions in Definition 14 are satisfied. We should point out that the rules discovered by MineTop$k$IRs are a subset of the above two existing methods.

In Figure 3, we study how the running time varies with #*sample* and #*gene* by increasing #*sample* from 10 to 30 while

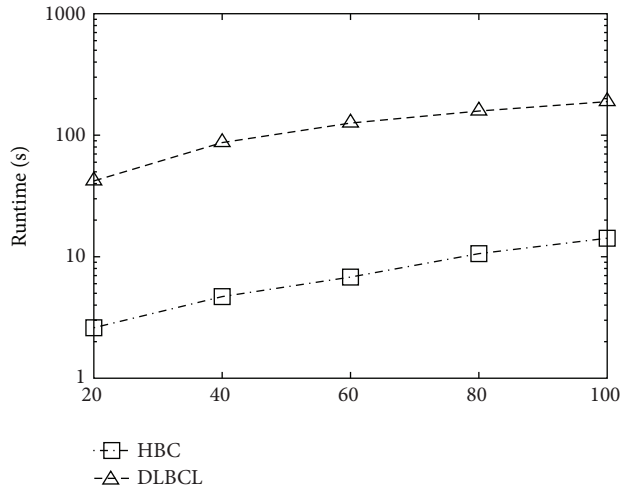


Figure 4: Varying k.

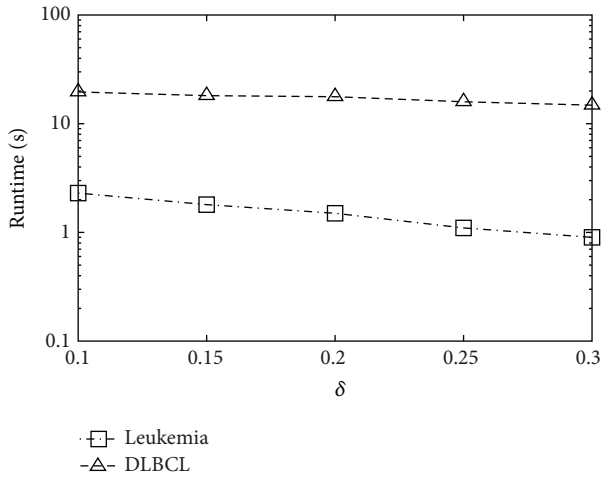


Figure 5: Varying δ.

fixing *#gene* to 100 and then increasing *#gene* from 20 to 100 while fixing *#sample* to 30, where the synthetic datasets are utilized. Figures 3(a) and 3(b) show that the running time becomes longer with *#sample* and *#gene* increasing. This is because the searching space also becomes larger. However, the MineTop$k$IRs is always much faster than the other two methods; the reason is that our algorithm can directly discover the results in one step. However, the other two are two-step mining methods, which need to first discover a bigger result set and then conduct the postprocessing. Further, with the searching space increasing, the number of rules after first step mining grows exponentially. Thus, it is very time-consuming.

Figure 4 shows the effect of varying k towards runtime. We observe similar tendencies on all datasets. It is quite reasonable that MineTop$k$IRs is monotonously increasing with k. Also, as shown in Figure 5, MineTop$k$IRs is monotonously decreasing with δ. Figures 6 and 7 show the effect of varying minimum support threshold α and the minimum confidence threshold β on four real gene expression datasets. Figures 6(a)–6(d) show the running time varying with the minimum support threshold α, where the other two parameters β and δ are set to 0.8 and 0. Note that the y-axes in Figures 6 and 7 are in logarithmic scale. We run MineTop$k$IRs by setting $k = 10$. In Figure 7, β changes from 70% to 90% while $\delta = 0$ and α is fixed in every dataset. As seen from Figure 6, running time decreases with the increasing of α. This is because the increasing of α prunes more useless rules. We also find out that MineTop$k$IRs is usually one order of magnitude faster than the other two algorithms, especially at low minimum support. The reason MineTop$k$IRs outperforms the other two algorithms is that R-FEAT and NRMINER discover a large number of rules at lower minimum support while the number of rules discovered by MineTop$k$IRs is bounded. Besides, MineTop$k$IRs can use Pruning strategy 1 to prune the search space; however, R-FEAT and NRMINER do not meet this property. Figure 7 shows that the running time of both NRMINER and R-FEAT does not change significantly as β is increasing, which is because the pruning strategies of these methods are mainly based on support threshold α. However, the running time a little increases with the increasing β. This is because with the increasing of β, the rules whose confidence below β will also increase; thus the pruning ability decreases a little. Despite so, the MineTop$k$IRs is still faster than the other two algorithms based on the above reasons in Figure 6.

5.2. Effectiveness of MineTopkIRs. In terms of the effectiveness of MineTop$k$IRs, the classification accuracy and the complexity are used as the performance standard for evaluation. Moreover, the biological significance of the discovered genes is also discussed.

5.2.1. Accuracy and Complexity. We build a classifier called Top$k$IR classifier based on the rules that MineTop$k$IRs discovered. The Top$k$IR classifier is composed of k subclassifiers, denoted as $\mathrm{IR}_1, \ldots, \mathrm{IR}_k$. Each IR_1 classifier is built based on all the top-j rules for each sample in the dataset. We call IR_1 the main classifier and $\mathrm{IR}_2, \ldots, \mathrm{IR}_k$ are backup classifiers. We use every subclassifier in order until the test sample is successfully classified. Besides both main and backup classifiers we set a default class which is set as the majority class of the training data. If a test sample cannot be classified by the k classifiers, we put it into the default class.

When building each subclassifier, the score function in (7) [13] is adopted, where $r \in \mathfrak{R}(C, s)$ represents the rules matching the test sample s in class C, and $r \in \mathfrak{R}(C)$ represents all the rules in class C. To which class a test sample should be assigned is decided by a matched rule of the highest score:

$$\mathrm{Score}\,(s \in C) = \frac{\sum_{r \in \mathfrak{R}(C,s)} \mathrm{supp}\,(r)\,\mathrm{conf}\,(r)}{\sum_{r \in \mathfrak{R}(C)} \mathrm{supp}\,(r)\,\mathrm{conf}\,(r)}. \quad (7)$$

In the experiments, we adopt 10-fold cross validation to test the average classification accuracy of Top$k$IR classifier and compare it with NR [8] and CBA and IRG [14] classifiers. The results in Table 4 show that Top$k$IR classifier performs much better than CBA and IRG classifiers. Compared with CBA which is built with the Top-1 covering irreducible

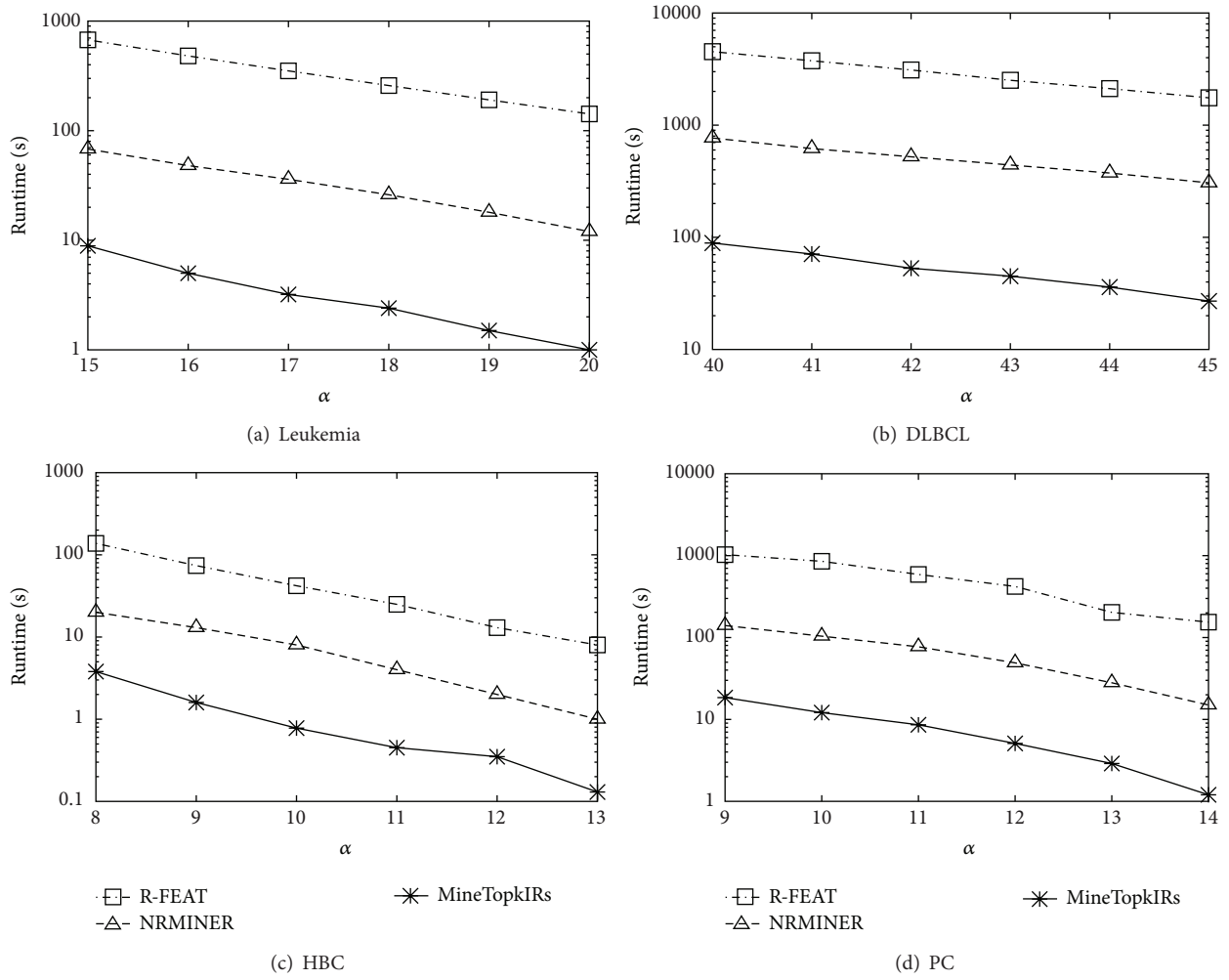


Figure 6: Varying α.

Table 4: The accuracy and complexity of classifiers.

Dataset	Acc. of Top$k$IR	Acc. of NR	Acc. of CBA	Acc. of IRG	AL. of Top$k$IR	AL. of Top$k$IR
Leukemia	96.84%	95.76%	91.18%	64.71%	3.42	4.80
DLBCL	94.97%	92.13%	82.16%	84.42%	3.16	4.95
HBC	92.68%	93.31%	85.61%	83.28%	2.45	4.25
PC	96.06%	91.28%	84.65%	88.24%	3.85	5.25

contrast sequence rules, Top$k$IR classifies much fewer test data using default class. IRG classifier is built based on the association rules, which illustrates that sequence rules can reflect data characteristics better. Top$k$IR classifier is more accurate than NR classifier on most dataset; however, it uses much fewer rules ($m * k$) to build classifier than NR. In our experiment, $k = 10$ and the rules used in NR classifier are usually more than ten thousand [8]. Furthermore, we discover that the average length (AL for short) of sequence rules used in Top$k$IR classifier is shorter than that of IRG. This result verifies that the MineTop$k$IRs could provide as high as diagnostic accuracy using as fewer as possible genes, which is very valuable for biologists to further follow up biological or clinical validation of selected genes [15].

5.2.2. Biological Significance. Different from the traditional methods, MineTop$k$IRs characterizes the pathogenesis of a disease from a sequence-like point of view, which incorporates the orders among genes and can be seen as the pathway of disease causing. In this part, by showing some interesting results from Leukemia dataset [1], we emphasize the fact that not only can MineTop$k$IRs find the genes revealed by the traditional methods, but also it can find some genes ignored by the traditional methods.

Table 5 lists the top-10 genes most frequently occurring in the discovered Top$k$IRs for the diagnosis of "AML" samples and "ALL" samples, where the genes with "$*$" mean they are also included in the benchmark, that results from eight statistics based gene ranking methods [16]. The two most

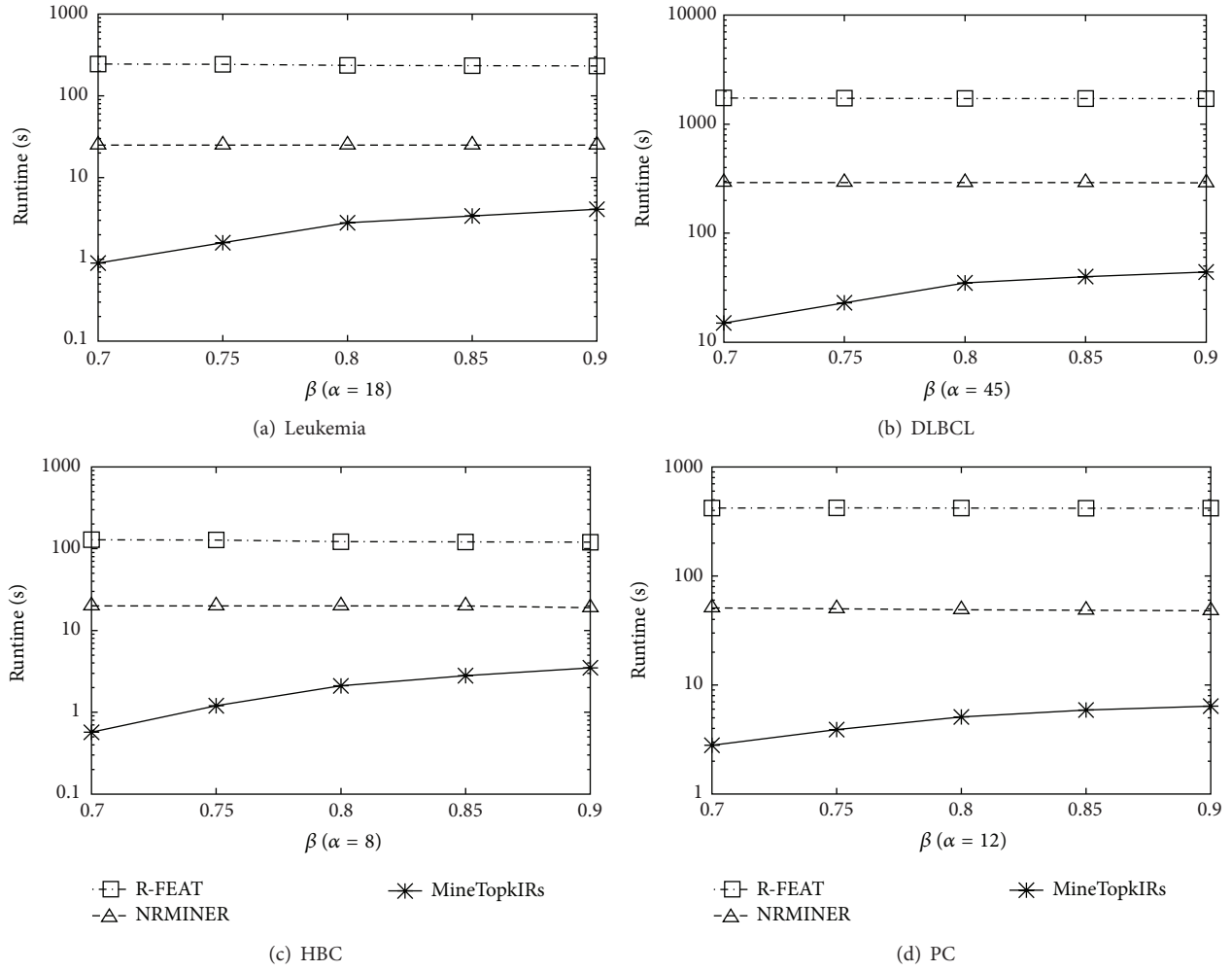


Figure 7: Varying β.

Table 5: Common frequent genes appear in Top$k$IRs.

Frequent gene in Top$k$IRs	Frequency (%)	Frequent gene in Top$k$IRs	Frequency (%)
TIMP2*	19.2	PTPRCAP*	7.9
ZFP36*	12.5	CCT5	5.6
MGST1*	12.4	CMYB	5.2
MYCL1	11	PSMA6*	4.9
LYZ1*	10.6	GIRX*	4.6

frequent genes appear in Table 5, which also appear in the benchmark. Gene TIMP2 is a member of the TIMP gene family, the proteins encoded by which are natural inhibitors of the matrix metalloproteinases. Reference [17] reveals that the transcription of TIMP2 in SHI-1 cells of AML is higher than other leukemic cells. Gene ZFP36 expression is upregulated in human T-lymphotropic virus 1- (HTLV-1-) infected cells. HTLV-1 is associated with adult T-cell leukemia/lymphoma [18].

In addition, for the genes without "$*$", though they are not in the benchmark, we still cannot ignore these genes. For example, the gene sequence $\langle RBL2\ DHPS\ CCT5\rangle$ including frequent gene CCT5 in Table 5 appears in most "ALL" sample but fewer occurs in "AML" samples. But, any of its subsequence does not have the ability of distinguishing samples which indicates that any gene in $\langle RBL2\ DHPS\ CCT5\rangle$ is irreducible and well reflects the synergy between the genes. Thus, these genes also have very important potential values for biologists to further explain.

6. Conclusion

In this paper, we study an important problem in bioinformatics, that is, discovering diagnostic gene patterns from gene

expression data. Unlike any previous work on this topic, we tackle the problem by exploiting the ordered expression trend of genes, which can better reflect the gene regulation pathway. In order to capture the more accurate diagnosis by using as few as possible rules, we propose the concept of top-k covering irreducible contrast sequence rules for each sample of gene expression data. Further, an efficient method called MineTop$k$IRs is developed to find all Top$k$IRs. Considering the real noisy scenario in gene expression data, we first use an EWave model, which, essentially different from the current models, characterizes gene expression data from a sequence-like perspective. Then, we can use MineTop$k$IRs to discover the bounded number of Top$k$IRs in one mining process, which can directly be used to build classifier. Extensive experiments conducted on both synthetic and real datasets show that MineTop$k$IRs is both effective and efficiency. It may offer a new point of view from diagnostic gene discovery to the biologists.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61272182, 61100028, 61073063, 61173030, and 61173029), 863 program (2012AA011004), 973 program (2011CB302200-G), National Science Fund for Distinguished Young Scholars (61025007), State Key Program of National Natural Science of China (61332014), New Century Excellent Talents (NCET-11-0085), China Postdoctoral Science Foundation (2012T50263 and 2011M500568), and Fundamental Research Funds for the Central Universities (N130504001).

References

[1] T. R. Golub, D. K. Slonim, P. Tamayo et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–527, 1999.

[2] M. A. Shipp, K. N. Ross, P. Tamayo et al., "Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning," *Nature Medicine*, vol. 8, no. 1, pp. 68–74, 2002.

[3] I. Hedenfalk, D. Duggan, Y. Chen et al., "Gene-expression profiles in hereditary breast cancer," *The New England Journal of Medicine*, vol. 344, no. 8, pp. 539–548, 2001.

[4] G. Wang, Y. Zhao, X. Zhao, B. Wang, and B. Qiao, "Efficiently mining local conserved clusters from gene expression data," *Neurocomputing*, vol. 73, no. 7–9, pp. 1425–1437, 2010.

[5] C. Tang, A. Zhang, and J. Pei, "Mining phenotypes and informative genes from gene expression data," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*, pp. 655–660, August 2003.

[6] X. Xu and A. Zhang, "Virtual gene: using correlations between genes to select informative genes on microarray datasets," in *Transactions on Computational Systems Biology II*, vol. 3680, pp. 138–152, Springer, 2005.

[7] J. H. Moore, F. W. Asselbergs, and S. M. Williams, "Bioinformatics challenges for genome-wide association studies," *Bioinformatics*, vol. 26, no. 4, Article ID btp713, pp. 445–455, 2010.

[8] Y. Zhao, G. Wang, Y. Li, and Z. Wang, "Finding novel diagnostic gene patterns based on interesting non-redundant contrast sequence rules," in *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM '11)*, pp. 972–981, Vancouver, Canada, December 2011.

[9] J. Wang and J. Han, "Bide: efficient mining of frequent closed sequences," in *Proceedings of the 20th International Conference on Data Engineering (ICDE '04)*, pp. 79–90, 2004.

[10] C. Gao, J. Wang, and Y. He, "Efficient mining of frequent sequence generators," in *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, pp. 1051–1052, Beijing, China, April 2008.

[11] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Proceedings of the IEEE 11th International Conference on Data Engineering (ICDE '95)*, pp. 3–14, March 1995.

[12] http://www.broadinstitute.org/cgi-bin/cancer/publications/view/75.

[13] R. Cai, A. K. H. Tung, Z. Zhang, and Z. Hao, "What is unequal among the equals? Ranking equivalent rules from gene expression data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 11, pp. 1735–1747, 2011.

[14] G. Cong, A. K. H. Tung, X. Xu, F. Pan, and J. Yang, "FARMER: finding interesting rule groups in microarray datasets," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '04)*, pp. 143–154, June 2004.

[15] M. Xiong, X. Fang, and J. Zhao, "Biomarker identification by feature wrappers," *Genome Research*, vol. 11, no. 11, pp. 1878–1887, 2001.

[16] Y. Su, T. M. Murali, V. Pavlovic, M. Schaffer, and S. Kasif, "RankGene: identification of diagnostic genes based on expression data," *Bioinformatics*, vol. 19, no. 12, pp. 1578–1579, 2003.

[17] https://ash.confex.com/ash/2009/webprogram/Paper22912.html.

[18] http://atlasgeneticsoncology.org/Genes/ZFP36L1ID42866ch14q22.