Mining Fuzzy Association Patterns in Gene Expression Databases

Vincent S. Tseng, Yen-Hsu Chen, Chun-Hao Chen, and J. W. Shin

Abstract

In this paper, we propose two fuzzy data mining approaches for microarray analysis, namely Fuzzy Associative Gene Expression (FAGE) and Ripple Effective Gene Expression Rule (REGER) algorithms. Both of them first transform microarray data into fuzzy items, and then use fuzzy operators and specially-designed data structures to discover the relationships among genes. Through the proposed algorithms, a novel pattern named *Ripple Pattern* is discovered that indicates the genes active at the same time with their linguistic terms being monotone increasing or decreasing. The experimental results show that the proposed algorithms are effective in discovering novel and useful rules from microarray data.

Keywords: Microarray, Gene Expression Analysis, Association Rule, Fuzzy Set, Ripple Pattern.

1. Introduction

Bioinformatics has become a more and more important research field since it provides powerful and effective ways for biologists to analyze and evaluate ideas through computer science methods [8]. Some domains such as multiple genes alignment [5], motif identification [4], microarray analysis [7], protein structure prediction [22] and pathway analysis [21] especially rely on it for discovering useful information. Recently, microarray data analysis received extensive attentions due to its wide applications. In the past, if biologists want to know certain gene expression, the only way to get the result is through experiments in the wet laboratory. However, the processes of the experiments are time-consuming, especially when a number of genes are involved. The microarray technology that can get lots of gene expressions at the same time thus becomes a popular way for gene expression analysis. How to analyze the relationships among genes from the microarray data has become an important issue.

Corresponding Author: Vincent S. Tseng is with the Department of Computer Science and Information Engineering, National Cheng Kung University, No.1, Ta-Hsueh Road, Tainan 701, Taiwan.
E-mail: tsengsm@mail.ncku.edu.tw

Early researches on microarray analysis were primarily based on statistical methods [8]. As mentioned above, they are time-consuming when the dataset is very huge. Thus, data mining technologies that can find information efficiently are used to solve this problem. Many approaches based on clustering [15, 18], classification [3], association rules [6, 7] were proposed to analyze microarray data. In [6, 7], the raw gene expression values need to be transformed into category values like "up-regulated" or "down-regulated" before the association rule mining method can be performed. However, microarray data were numerical data, thus how to design a suitable algorithm to deal with this data is an important issue.

Recently, fuzzy set theory [26] has been used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning [14]. Several fuzzy learning algorithms for inducing rules from given datasets have been proposed and applied in various domains like manufacturing, engineering, diagnosis, economics, etc [11][14][19][25]. Hong *et al.* [13] proposed a fuzzy mining approach to find fuzzy interesting itemsets and fuzzy association rules from quantitative data. The mining results obtained could be smooth due to the characteristics of fuzzy membership.

In this paper, we propose two novel methods that combine fuzzy concept and A*priori-like* approach for mining fuzzy association patterns in microarray datasets. First, we explore the issue of discovering fuzzy association rules from the microarray data called *Fuzzy Associative Gene Expression (FAGE)* algorithm. It first transforms the numerical gene expression data into fuzzy items, and then use fuzzy operators to find the association rules among them. Secondly, we extend *FAGE* to propose the *Ripple Effective Gene Expression Rule (REGER)* algorithm for finding *Ripple Patterns* that are hidden in the microarray data. *Ripple Pattern* means a potentially chained reaction among genes. For example, a ripple pattern like "WSC4:L→SOK1:SH→HSP12:H" means that the genes WSC4, SOK1, and HSP12 are active at the same time with their expressions as *Low* (*L*), *Slightly-High* (*SH*) and *High* (*H*), respectively, in monotone manner. Since the *Ripple Patterns* discovered by the *REGER* algorithm are represented by linguistic rules, they will be friendlier for biologists to analyze than in form of quantitative representation. Through experimental evaluation on real microarray data, the

proposed algorithms are shown to be very effective in discovering novel and useful relations among genes from gene expression data.

Remaining of this paper is organized as follows: In Section 2, we describe some related work. The proposed algorithms are given in details in Section 3. Section 4 gives the experimental results, and the concluding remarks are made in Section 5.

2. Related Work

Data mining is most commonly used in attempts to induce association rules from transaction data [1][2]. The goal is to discover important associations among items such that the presence of some items in a transaction will imply the presence of some other items. To achieve this purpose, Agrawal *et al.* proposed several mining algorithms based on the concept of large itemsets to find association rules in transaction data [2]. They divided the mining process into two phases. In the first phase, if the number of an itemset appearing in the transactions was larger than a pre-defined threshold value (called minimum support), the itemset was considered a large itemset. In the second phase, association rules were induced from the large itemsets found in the first phase. All possible association combinations for each large itemset were formed, and those with calculated confidence values larger than a predefined threshold (called minimum confidence) were output as association rules.

Chen *et al.* proposed a mining algorithm to find transcription factors essential to gene expression [6]. They first defined each type of tissues as a set of transactions. Each transaction thus consists of transcription factors and the target genes. After transformation, *Apriori* mining algorithm [2] was then used to mine association rules to obtain transcription factors associated with gene expressions. In [7], Creighton *et al.* proposed an *Apriori-like* algorithm to mine association rules among gene expressions. Their approach is composed of two phases: quantitative values discretization and association-rule generation. In quantitative values discretization phase, the approach used experimental results from laboratory to specify up-regulated and down-regulated interval, and then transformed each gene expression value in microarray into up-regulated or down-regulated state. In association-rule generation, *Apriori* mining algorithm [2] was then used to mine association rules. The rule format was "$G_a \uparrow \rightarrow G_b \uparrow$", which means if G_a is up-regulated, then G_b is also up-regulated. In [16], Kotala *et al.* proposed a mining algorithm that used Peano Count Tree (P-tree) to discover association rules from microarray data. The rule format was "$\{G_1, ..., G_n\} \rightarrow G_m$", which means that a given confidence level of the expression genes, G_1, ..., G_n, will result in the expression of G_m gene. Other mining approaches like clustering [9][24], which groups genes into its similar group and gives a view of gene family, and classification [3], which used to predict the gene family.

As to fuzzy data mining, Hong *et al.* proposed several fuzzy mining algorithms to mine linguistic association rules from quantitative data [12][13][17]. They transformed each quantitative item into a fuzzy set and used fuzzy operations to find fuzzy rules. Their proposed algorithm focused on transaction data. Thus, in this paper we propose the fuzzy association rules mining algorithm to mine rules and discover *Ripple Patterns* from microarray data.

3. Proposed Methods

In this session, the two proposed algorithms are described. The first algorithm, namely *Fuzzy Association Gene Expression* (*FAGE*), is proposed to mine fuzzy rules from microarray data by applying Hong's approach [12]. The *Ripple Effective Gene Expression Rule* (*REGER*) algorithm, which is then extended from the *FAGE* algorithm, is proposed to mine *Ripple Patterns*. Both algorithms are stated in details in the following.

A. *The Fuzzy Associative Gene Expression (FAGE) Algorithm*

The main idea of *FAGE* algorithm is to first transform microarray data into fuzzy items, and then use fuzzy operators to find the relationships among them as [12]. The detail of the *FAGE* algorithm is shown in Figure 1.

Input: A microarray data set M with n experiments, each of which has w genes; a set of k membership functions for data value; a predefined minimum support α; a predefined minimum confidence λ.

Output: A set of fuzzy association rules among genes with confidence values.

1. For $i \leftarrow 1$ to n do
2. For $j \leftarrow 1$ to w do
3. Transform quantitative value $v_j^{(i)}$ into fuzzy items using MS

$$\left(\frac{f_{j1}^{(i)}}{R_{j1}} + \frac{f_{ji2}^{(i)}}{R_{j2}} + ... + \frac{f_{jk}^{(i)}}{R_{jk}} \right)$$

4. Count every fuzzy item R_{jl}

$$count_{jl} = \sum_{i=1}^{n} f_{jl}^{(i)}$$

5. Prune fuzzy items whose *count* is smaller than α
6. Collect fuzzy items to form L_1
7. Set $r = 1$
8. $C_{r+1} \leftarrow L_r$ join L_r
9. For each $(r + 1)$-itemset I in C_{r+1}:

10. Calculate the fuzzy value of I in each experiment i as:

$$f_I^{(i)} = f_{I_1}^{(i)} \Lambda\ f_{I_2}^{(i)} \Lambda\ ... \Lambda\ f_{I_{r+1}}^{(i)}$$

11. Calculate the count of I in all the subsequences as:

$$count_I = \sum_{i=1}^{n} f_I^{(i)}$$

12. Prune fuzzy items whose *count* is smaller than α

13. If L_{r+1} = null, then do the next step; otherwise, set $r = r + 1$ and repeat Lines 8 to 12.

14. Construct the association rules for each large q-itemset I with items (I_1, I_2, $\cdots$, I_q), $q \geq 2$ as follows:

15. Form each possible association rule as follows:

$$I_1 \wedge ... \wedge I_{h-1} \wedge I_{h+1} \wedge ... \wedge I_q \rightarrow I_h,\ h = 1 \text{ to } q$$

16. Calculate the confidence values of all possible rules as follows:

$$\frac{\sum_{i=1}^{n} f_I^{(i)}}{\sum_{i=1}^{n} (f_{I_1}^{(i)} \Lambda\ ... \Lambda\ f_{I_{h-1}}^{(i)},\ f_{I_{h+1}}^{(i)} \Lambda\ ... \Lambda f_{I_q}^{(i)})}$$

17. Output the association rules with confidence values larger than or equal to the predefined confidence threshold λ.

Figure 1. The *FAGE* Algorithm.

From Fig. 1, the algorithm is divided into two phases, the frequent itemsets generating phase (Line 1-13) and rules generating phase (Line 14-17). In frequent itemsets generating phase, lines 1 to 3 are to transform gene expression into fuzzy values. Lines 4 to 13 are then used to generate all frequent itemsets. In rules generating phase, line 15 is for forming candidate fuzzy rules. Line 16 is then used to calculate confidence values of all rules. Finally, the fuzzy rules with confidence values large then or equal to the predefined minimum confidence are output as results.

B. *The Ripple Effective Gene Expression Rule (REGER) Algorithm*

The *REGER* algorithm is used to mine *Ripple Patterns*. The detail of the *REGER* algorithm is shown in Figure 2.

Input: A microarray data set M with n experiments, each of which has w genes; a set of k membership functions MS for data value; a predefined minimum support α; a predefined minimum confidence λ; a minimum rule length l.
Output: The Ripple Patterns.

1. For $i \leftarrow 1$ to n do
2. For $j \leftarrow 1$ to w do
3. Transform quantitative value $v_j^{(i)}$ into fuzzy items using MS

$$\left(\frac{f_{j1}^{(i)}}{R_{j1}} + \frac{f_{ji2}^{(i)}}{R_{j2}} + ... + \frac{f_{jk}^{(i)}}{R_{jk}} \right)$$

4. Count every fuzzy item R_{jl}

$$count_{jl} = \sum_{i=1}^{n} f_{jl}^{(i)}$$

5. Prune fuzzy items whose *count* is smaller than α

Figure 2. The *FAGE* Algorithm.

From Figure 2, there are three phases, namely rule generating phase (lines 1-10), tree building phase (lines 11-28) and *Ripple Patterns* generating phase (lines 19-22), for the *REGER* algorithm. In rule generating phase, the fuzzy association rules are generated from large 2-itemset using the same way in *FAGE*. In line 10, the constraint means the linguistic terms of genes must be monotone increasingly or decreasingly. In tree building phase, the specified tree structure is then used to store fuzzy rules. It uses two hash trees, T_i and T_d, namely the increasing and decreasing hash tree, respectively, to store the rules generated from the first phase. From the tree structure, we can find all *Ripple Patterns* by reusing the increasing and decreasing hash tree instead of scanning the fuzzy rule set again and again. For example, assume the rules (only monotone increasing) generated from the first phase are shown in Table 1, the increasing hash tree is then built as shown in Figure 3.

Table 1. Eight fuzzy rules

RULE$_I$			
Rule$_1$	G_1:L→G_2:SL	*Rule*$_5$	G_2:SL→G_3:SH
Rule$_2$	G_1:L→G_3:SH	*Rule*$_6$	G_1:L→G_5:SH
Rule$_3$	G_2:SL →G_4:H	*Rule*$_7$	G_2:SL →G_5:SH
Rule$_4$	G_5:SH →G_6:H	*Rule*$_8$	G_5:SH →G_9:H

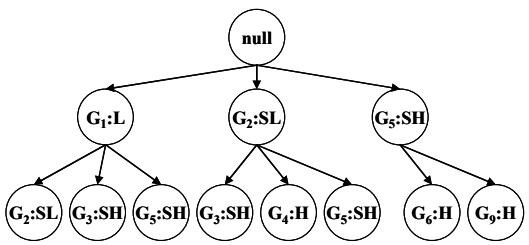

Figure 3. The increasing hash tree.

Finally, *Ripple Patterns* generating phase uses Find_Pattern procedure to find *Ripple Patterns* as show in Figure 4.

```
1.  Find_Pattern(prefix_list, postset)
2.  For each item in postset
3.    prefix_list←prefix_list+item
4.    new_postset←post_set ∩ get_children(item)
5.    If( new_postset is empty )
6.      If( length(prefix_list) > l )
7.      Output rule
8.      remove_last_item(prefix_list)
9.    Else
10.     Find_Pattern(prefix_list,new_postset)
11. End For each
12. End procedure
```

Figure 4. Find_Pattern procedure.

From Fig. 4, the primary action is using the recursive procedure, Find_Pattern, to find *Ripple Pattern*s. Continued from previous example, when *l* is set as 2, the Find_Pattern procedure generates two *Ripple Pattern*s, namely "G_1:L → G2:SL → G_3:SH" and "G_1:L → G_2:SL → G_5:SH".

4. Experimental Evaluation

A series of experiments were conducted to evaluate the performance of the proposed methods and to verify the accuracy of the discovered rules. The simulations were implemented in C++ at a personal computer with AMD K8N 2800+ and 1GB memory. The yeast dataset used in the experiments is the same as [7]. The dataset contains gene expression profiles for 6316 genes corresponding to 300 diverse mutation and chemical treatments in yeast, which yields to a skewed dataset. We tested four membership functions, namely *Low* (*L*), *SlightlyLow* (*SL*), *SlightlyHigh* (*SH*), and *High* (*H*), which represent different levels of gene expressions in the experiments as shown in Figure 5. In [7], two crisp intervals, up-regulated and down-regulated interval, were used to transform gene expressions into up-regulated or down-regulated state. In this way, only about 300 genes of 6316 genes are used to mine association rules. However, in our experimental evaluation, more than 2000 genes were used to analyze the yeast dataset. Hence, more interesting rules can be discovered by our methods.

A. *Experimental Results on the FAGE Algorithm*

The first experiment was made to show the difference of the number of rules between the *FAGE* algorithm and the crisp mining algorithm in [7]. The minimum support and minimum confidence are set as 10% and 80%, respectively. The experimental results were shown in Figure 6.

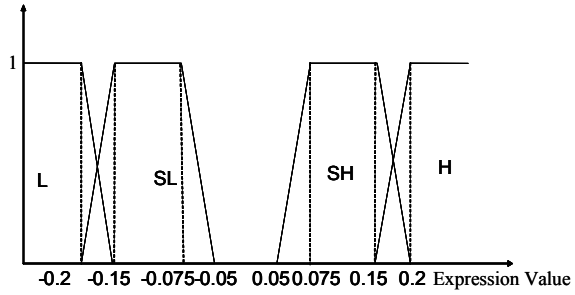

Figure 5. The trapezoid membership functions.

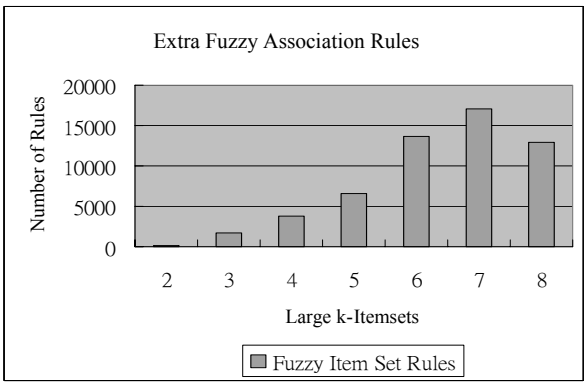

Figure 6. The difference of number of rules between the *FAGE* algorithm and the crisp mining algorithm [7].

From Figure 6, it is easily seen that the difference of number of association rules between the proposed approach and the crisp mining algorithm [7] increases along with the increasing in large itemsets, except for the case of large 8-itemset. For example, over 5,000 rules are generated by the proposed approach than the crisp mining algorithm [7] from large 5-itemsets. Totally, more than 40,000 rules can be derived using the *FAGE* algorithm. Among the discovered fuzzy rules, we list two verified fuzzy rules as shown in Table 2.

Table 2. The verified fuzzy association rules

	$RULE_I$
1	SNO1:H,SNZ1:H→SPO21:SH, CTF13:H
2	GRX1:SH->HSP12:H

From Table 2, in antecedent part of the first rule, SNO1 and SNZ1 are stationary-phase induced genes that appear to be involved in the cellular response to nutrient limitation and growth arrest [20]. In consequent part, SPO21 is related to formation of the prosper membrane as described in MIPS (http://mips.gsf.de/). The three genes are thus related genes. The second rule indicates a disclosed truth, because GRX1 and HSP12 are all stress related genes and GRX1 is the sub-process of HSP12 [23]. However, an interesting result left for further study is why the gene GRX1 has lower expression (SlightlyHigh).

More experiments were then made to show the relationships between the number of rules and minimum support values under different minimum confidence

values. The results are as shown in Figure 7.

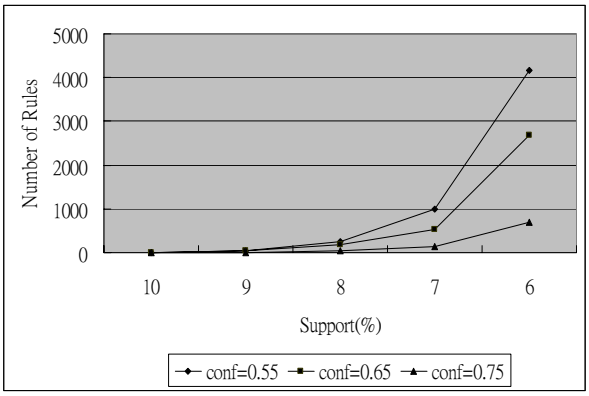

Figure 7. Relationships between number of rules and minimum support values.

From Fig. 7, it is easily seen that the number of association rules decreases along with minimum support values increased. Also, the curve for numbers of association rules with larger minimum confidence values was more smooth than those with smaller minimum confidence values, meaning that the minimum support value has significant impact on the number of association rules derived from small minimum confidence values. We can also see that there is a sharp rise in the number of rules when minimum support exceeds 8%. It thus gives the indication that we may find novel rules and avoid confusion when minimum support is set around 8%.

B. *Experimental Results on The REGER Algorithm*

In order to show the effectiveness of the *REGER* algorithm, the experiment was made to find *Ripple Patterns* when the minimum support and minimum confidence are set as 10% and 55%, respectively. The results were shown in Table 3.

Table 3. The rules derived from the *REGER* algorithm

$Rule_i$	*Ripple Patterns*
$Rule_1$	WSC4:L→SOK1:SH→HSP12:H
$Rule_2$	WSC4:L→SOK1:SH→ARO9:H
$Rule_3$	WSC4:L→SOK1:SH→NCA3:H
$Rule_4$	WSC4:L→SOK1:SH→SNZ1:H
$Rule_5$	WSC4:L→SOK1:SH→NCE103:H
$Rule_6$	WSC4:L→SOK1:SH→YOR338:H
$Rule_7$	YPL267W:SL→RIM101:SH→CTF13:H
$Rule_8$	YPL267W:SL→RIM101:SH→SNO1:H
$Rule_9$	YPL267W:SL→RIM101:SH→SNZ1:H
$Rule_{10}$	YPL267W:SL→RIM101:SH→NCE103:H

In Table 3, we could find an interesting phenomenon as described below. From Rules 1 to 6, we can easily see that they all contain WSC4 and SOK1 genes, which then imply to different genes. In this case, we may consider WSC4/SOK1 as mediator genes in some pathways or biological processes. From Rules 7 to 10, we can also consider RIM101 as mediator genes for linking gene YPL267W to other genes.

After the validated process, we divided the discovered rules into three classes, namely well-known rules, partially-known rules and unknown rules, as shown in Table 4.

Table 4. The classes of rules

Class	*Rule numbers*
Well-known rules	1,4,5
Partially-known rules	2,3,6
Unknown rules	7,8,9,10

For well-known rules, WSC4 and SOK1 are found to be the signal and communication related genes in Rule 1 from MIPS. The relationship between WSC4 and HSP12 can be found in Gene Ontology that describes that both of them are heat shock functions, and SOK1 and HSP12 are in the same protein family. Thus, Rule 1 is a validated rule. In [10], it was described that if WSC4 is down-regulated, then SNZ1 is usually up-regulated. Hence, Rule 4 is also a validated rule. Rule 5 can also be validated by Gene Ontology.

For partially-known rules, Rule 2 and Rule 3 were only partially known for the relation between WSC4 and SOK1. Thus, they may be suggested to biologists as candidate interesting genes. Rule 6 could not be validated because YOR388 is unknown gene, and it is needed biological experiment to find its functions.

For unknown rules, since YPL267W has less information than other genes, Rule 7 to 10 can not be validated yet but these rules exhibit potential relations among genes for further studies.

5. Conclusions and Future Work

In this paper, we have proposed two algorithms for microarray analysis. Firstly, we propose the *FAGE* algorithm for mining fuzzy association rules from microarray data. Secondly, we extend the *FAGE* algorithm to propose the *REGER* algorithm to find *Ripple Patterns*. Both of them first transform numerical data into fuzzy items, and then used fuzzy operators to find the relationships among them.

The experimental results show that the *FAGE* algorithm can not only find more rules than the method in [7], but also demonstrate that the derived fuzzy association rules are useful in revealing unknown relations among genes. The experimental results also show that the *REGER* algorithm can provide a novel style to analyze gene expression data and mine useful and interesting patterns for biologists.

For future work, we will enhance the *REGER* algorithm to find the unknown mediator genes in pathways and extend it for analyzing time serial microarray data. Besides, it is another interesting issue concerning how to combine abundant bioinformatics resources on Web like the Gene Ontology or other

databases to build a system for automatically verifying discovered gene functions.

6. Acknowledgment

This research was partially supported by National Science Council, Taiwan, R.O.C., under grant number NSC 94-2213-E-006 -099.

7. References

[1] R. Agrawal, T. Imielinski and A. Swami, "Mining association rules between sets in large databases," *ACM SIGMOD Conf. on Management of Data*, pp 207-216, 1993.

[2] R. Agrawal and R. Srikant. "Fast algorithms for mining association rules," *Very Large Databases (VLDB) Conference*, pp 487-499, 1994.

[3] M. Bhasin and G. P. S. Raghava, "SVM based method for predicting HLA-DRB1*0401 binding peptides in an antigen sequence," *Bioinformatics* vol. 20, pp. 421 – 423, 2004.

[4] M. Bilgen, M. Karaca, A. Naci Onus, and Ayse Gu"l Ince, "A software program combining sequence motif searches with keywords for finding repeats containing DNA sequences," *Bioinformatics*, vol. 20, pp. 3379 – 3386, 2004.

[5] V. Brendel, L. Xing and W. Zhu, "Gene structure prediction from consensus spliced alignment of multiple ESTs matching the same genomic locus," *Bioinformatics*, vol. 20, pp. 1157-1169, 2004.

[6] R. Chen, Q. Jiang, H. Yuan and L. Gruenwald. "Mining association rules in analysis of transcription factors essential to gene expressions," *Atlantic Symp. on Computational Biology, and Genome Information Systems & Technology*. 2001.

[7] C. Creighton and S. Hanash. "Mining gene expression databases for association rules," *Bioinformatics*, vol. 19, no. 1, pp. 79-86, 2003.

[8] T. M. Chu, B. Weir and R. Wolfinger. "A systematic statistical linear modeling approach to oligonucleotide array experiments," *Mathematical Biosciences*, 2002.

[9] M. B. Eisen, P. T. Spellman, P. O. Brown and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl Acad. Sci.*, pp. 14863-14868, 1998.

[10] E. L. Braun, E. K. Fuge, P. A. Padilla, M. Werner-Washburne, "A stationary-phase gene in Saccharomyces cerevisiae is a member of a novel, highly conserved gene family", *J Bacteriol*, pp. 6865-6879, 1996.

[11] I. Graham and P. L. Jones, *Expert Systems – Knowledge, Uncertainty and Decision,* Chapman and Computing, Boston, pp.117-158, 1988.

[12] T. P. Hong, C. S. Kuo and S. C. Chi, "Trade-off between time complexity and number of rules for fuzzy mining from quantitative data," *Int'l Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 9, no. 5, pp. 587-604, 2001.

[13] T. P. Hong, C. S. Kuo and S. C. Chi, "Mining association rules from quantitative data", *Intelligent Data Analysis*, vol. 3, no. 5, pp. 363-376, 1999.

[14] A. Kandel, *Fuzzy Expert Systems*, CRC Press, Boca Raton, pp.8-19, 1992.

[15] L. Kaufrnan and P. J. Rousseeuw. "Finding groups in data: an introduction to cluster analysis." *New York: John Wiley & Sons*, 1990.

[16] P. Kotala, P. Zhou, S. Mudivarthy, W. Perrizo and E. Deckard. "Gene expression profiling of DNA Microarray data using Peano count trees (P-trees)," *Online Proc. of the First Virtual Conference on Genomics and Bioinformatics*, 2001.

[17] Y. C. Lee, T. P. Hong and W. Y. Lin, "Mining fuzzy association rules with multiple minimum supports using maximum constraints," *Lecture Notes in Computer Science*, vol. 3214, pp. 1283-1290, 2004.

[18] J. MacQueen. "Some methods for classification and anlysis of multivariate observations," *Proc. 5th Berkeley Symp. Math. Statist, Prob.*, pp. 281-297, 1967.

[19] E. H. Mamdani, "Applications of fuzzy algorithms for control of simple dynamic plants," *IEEE Proceedings*, pp.1585-1588, 1974.

[20] P. A. Padilla, "Nitrilase and Fhit homologs are encoded as fusion proteins," *Drosophila melanogaster and Caenorhabditis elegans,* pp. 8744-8749, 1998.

[21] R. Pandey, Raghavendra K. Guru, and David W. Mount "Pathway Miner: extracting gene association networks from molecular pathways for predicting the biological significance of gene expression microarray data," *Bioinformatics*, pp.2156-2158, 2004.

[22] I. Res, I. Mihalek, and O. Lichtarge, "An evolution based classifier for prediction of protein interfaces without using protein structures," *Bioinformatics*, vol. 21, pp. 2496 – 2501, 2005.

[23] S. Luikenhuis, G. Perrone, I. W. Dawes and C. M. Grant, "The Yeast Saccharomyces cerevisiae contains two Glutaredoxin genes that are required for protection agaist reactive Oxygen Species", *Molecular Biology of the Cell*, vol. 9, pp. 1081-1091, 1998.

[24] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander, and T. Golub, "Interpreting patterns of gene expression with self-organizing maps: methods and application

to hematopoietic differentiation," *Proc. Natl Acad. Sci.* USA, pp. 2907-2912, 1999.
[25] R.Weber, "Fuzzy-ID3: a class of methods for automatic knowledge acquisition," *The Second Int'l Conf. on Fuzzy Logic and Neural Networks*, pp. 265-268, 1992.
[26] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338-353, 1965.

Vincent S. Tseng received his B.S., M.S. and Ph.D. degree from National Chiao Tung University, Taiwan, R.O.C., in 1988, 1990 and 1997, respectively, all in computer science. During January 1998 and July 1999, he was an invited postdoctoral research fellow in Computer Science Division of University of California, Berkeley, U.S.A. Since August 1999, he has been the faculty at Department of Computer Science and Information Engineering at National Cheng Kung University, Taiwan. Since February 2004, he has also served as the director for Dept. Medical Informatics in National Cheng Kung University Hospital, Taiwan.

Dr. Tseng has a wide variety of research specialties covering data mining, Web technology and biomedical informatics. He has published a number of research papers in referred journals and international conferences. He is a member of the Association for Computing Machinery, IEEE and honorary member of Phi Tau Phi Society. He also serves as the co-chair of Special Interest Group on Data Mining of Berkeley Initiative in Soft Computing (BISC) in University of California, Berkeley, USA.

Yan-Shiu Chen received the B.S. degree from Tamkang University, Taiwan, R.O.C., in 2002 and the M.S. degree from the National Cheng Kung University, Taiwan, R.O.C., in 2005, both in Computer Science. He is currently a Ph.D. student in Electrical Engineering at the National Taiwan University. His research interests include data mining, bio-information and ad hoc network.

Chun-Hao Chen received his B.S. and M.S. degree from the I-Shou University, Taiwan, R.O.C., in 2002 and 2004, respectively, both in Information Management. He is currently a Ph.D. student in computer science and information engineering at the National Cheng Kung University. His research interests include data mining, genetic algorithms, time series and fuzzy theory.

Jyh-wei, Shin received his Ph.D. degree from National Taiwan University, Taiwan, R.O.C. in 1992. Since August 1994, he has been the faculty at Department of Parasitology, Medical College, National Cheng Kung University.

Dr. Shin has a wide variety of research covering parsitology and published a number of research papers in that field. Recently, he is working on the systems parasitology and cancer biology research.