CTGR-Span: Efficient Mining of Cross-Timepoint Gene Regulation Sequential Patterns from Microarray Datasets

Chun-Pei Cheng[1], Yi-Lin Tsai[1] and Vincent S. Tseng[1, 2*]
[1] *Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan*
[2] *Institute of Medical Informatics, National Cheng Kung University, Tainan, Taiwan*
* *Correspondence: tsengsm@mail.ncku.edu.tw*

***Abstract*—Sequential pattern mining techniques have been widely used in different topics of interest, such as mining customer purchasing sequences from a transactional database. Notably, observation of gene expressions to discover gene regulations during biological or clinical progression via microarray approaches has become the dominant trend. By converting microarray datasets into the format of transactional databases, sequential patterns implying gene regulations could be identified. However, there exists no effective method in current studies that can handle such kind of dataset as every transaction may contain too many items/genes and the resultant patterns are very susceptible to item order. We propose a new method called *CTGR-Span* (Cross-Timepoint Gene Regulation Sequential Patterns) to efficiently mine *CTGR-SPs* (cross-timepoint gene regulation sequential patterns). The proposed method was experimented with two publicly available human time course microarray datasets and it outperformed traditional methods over 2,000 times in terms of the execution efficiency. Furthermore, via a Gene Ontology enrichment analysis, the resultant patterns are more meaningful biologically compared to previous literature reports. Hence, it could provide biologists more insights into the mechanisms of novel gene regulations in certain disease progressions.**

Keywords—Cross timepoint; Gene regulation; Long transaction; Sequential pattern; Time course microarray

I. INTRODUCTION

Microarray has been widely conducted in the biomedical field for identifying differentially expressed genes in human diseases [1, 2]. An increasing number of data performed by various experimental designs has led to the development of diverse data mining approaches. Studies on time course issue have become more and more important as the illness event, such as cancer formation, requires a period of time in which aberrant alterations in gene expression may contribute to an interruption of basal condition, facilitating the cell death. Blocking a key component in gene regulations may reduce the severity of illnesses or injuries. In these regards, a consecutive monitoring of massive gene expressions is helpful to unveil the hidden layer of gene regulations during the progression of clinical or biological manifestations.

In the data mining field, the potential gene regulations occurred in a period of time could be identified by mining sequential patterns from the time course microarray datasets. The sequential patterns refer to a set of singleton frequent items/genes that is followed by another item/gene in the time-stamp ordered transaction/microarray profiling set. Based on different computational designs, several parental algorithms, such as *AprioriAll* [3] and *PrefixSpan* [4], have been successfully developed to discover the sequential patterns from a transactional database. The *apriori*-like *GSP* [5] and pattern-growth-based *Prefix-growth* [6] are designed allowing for the incorporation of constraints such as the gap size among the sequence elements. Besides, any subpatterns of the parental sequential pattern satisfy the user-defined constraints, i.e., *downward closure property*. Certain redundant shorter sequential patterns could be dominated by the longer ones when both patterns have the same occurrence times over all transactions in the database. Some newer algorithms of mining *closed sequential pattern*, such as *CloSpan* [7], are subsequently proposed to address this issue. However, to our knowledge, all of these traditional strategies are not suitable for the widely used microarray data since a whole-genome gene expression microarray is typically composed of tens of thousands genes/probes, such as over 20,000 probes/genes in human microarray platforms. A set of significant genes which are differentially expressed on a microarray platform could be regarded as one transaction. The traditional methods could not handle such kind of long transactional databases since there are too many significant genes/items contained in every transaction after converting the microarrays into the format of transactional database [8]. In this paper, we propose a novel algorithm called *CTGR-Span (Cross-Timepoint Gene Regulation Sequential Patterns)* to solve the issue mentioned above by mining *CTGR-SPs* (cross-timepoint gene regulation sequential patterns).

II. MATERIALS AND METHODS

In this section, we explain the detailed procedures of the proposed *CTGR-Span* for discovering *CTGR-SPs* from time course large-scale microarray datasets. Basically, it consists of three important parts including an introduction to the input microarray datasets, data transformation, and the detailed mining processes of *CTGR-Span* with several critical bio-logical-designed arguments for this research.

A. Input microarray datasets

As stated above in the section of the introduction, simultaneous reading multiple gene expressions through a microarray approach for identifying potential gene regulations is indispensable to the most notorious diseases, such as cancer development, inflammation and chronic hepatitis. Two publicly available gene expression microarrays with time course are curated from the GEO database including the accession number GSE6377 [1] and GSE11342 [2]. The former was performed to detect 8,793 leukocyte transcriptional changes at about 10 timepoints from 11 patients with ventilator-associated pneumonia, and the latter was attempted to monitor 22,283 gene expressions induced by the Peg-interferon alfa-2b plus ribavirin in peripheral blood monocytes of 20 hepatitis C patients during the first 10 weeks of treatment. Although

978-1-4673-2560-8/12/$31.00 ©2012 IEEE

several differentially expressed genes have been successfully identified from both approaches, the key modulators of gene regulations have yet been discovered during the period of the observations. Hence, both datasets are individually considered as the input datasets for mining *CTGR-SPs* in this study.

B. Convert microarray datasets into transactional databases

Since the non-numeric data are valid for mining sequential patterns in a transactional database, the probe/gene expression values involved in microarray need to be discretized into singleton items within every transaction. For each patient, the deviations of their gene expression values at each timepoint relative to the first timepoint will be divided by the first timepoint for calculating the fold changes (*FCs*) of the genes. If the FC of a gene (*G*) exceeds a *user-set positive threshold*, the expressed gene will be defined as an item G_{+}, whereas below the *-threshold* is G_{-}. After the converting process, suppose a patient has a sequence $<(G_{1+}G_{2-}G_{3-})_1(G_{1-})_2(G_{1-}G_{2+})_3>$ containing converted significant gene items from 3 timepoints. The expressed G_1, repressed G_2 and repressed G_3 occur at the first timepoint. They are presented in the same parenthesis (transaction).

However, the gene items within a transaction will be affected by different *thresholds*. How to set a suitable value is a big challenge. In this study, the *thresholds* for GSE6377 and GSE11342 datasets are set as 1.03 and 1.5, respectively, according to the same criteria used in the original articles [1, 2].

C. CTGR-Span: Cross-timepoint gene regulation sequential pattern

In this section, we introduce a new pattern-growth-based *CTGR-Span*. Firstly, the basic idea of the *CTGR-Span* and its mining processes will be illustrated using an example. Then, to further improve the effectiveness in the domain of biology, several extra bio-logical-designed parameters will be presented in a later paragraph.

1) Kernel procedure

We propose a new framework to find *CTGR-SPs*. This framework overcomes an issue that the transactions contain too many items/significant genes. The following example guides you how to trace this new framework performed on a transactional database. A set *S* of sequences containing the transactions of 4 patients is shown in Table I. Suppose the *minimum support*, *minSupp*, is set as 50%. If the items appear in at least two different sequences, they will be regarded as the frequent items to generate *CTGR-SPs* through a *prefix-projection*-based manner [4] by 3 steps.

a) Step 1: Find length-1 CTGR-SPs

After scanning the *S*, the frequent items of *length-1* $<G_{1+}>$, $<G_{2-}>$ and $<G_{3+}>$ can be successfully identified as they appear over one half of the sequences. Therefore, these 3 frequent items are regarded as the *lengh-1 CTGR-SPs*.

b) Step 2: Divide search space

Each item in the set of *length-1 CTGR-SPs* is individually considered as a *prefix* to generate its projected database and find the *postfixs* in which they are also frequent in *S*.

c) Step 3: Find postfixs of CTGR-SPs

For each *prefix*, the subsets of sequential patterns can be explored in a depth-first search approach from the corresponding projected database.

Table I. Example of transactional database

Patient IDs	Sequences
1	$<(G_{1+})_1(G_{2-}G_{3+})_2(G_{3+})_3>$
2	$<(G_{1+}G_{4-})_1(G_{3+})_2(G_{2-}G_{3+})_4(G_{5+})_5>$
3	$<(G_{8-})_1(G_{1+}G_{2-})_2(G_{2-}G_{3+})_3>$
4	$<(G_{7+})_1(G_{1+}G_{3+}G_{6-})_2(G_{2-}G_{3+})_3>$

For the current example, firstly, the *length-1 CTGR-SPs* are regarded as *prefixs* shown in the left-most column of Table II. Secondly, only the subsequences prefixed with the first occurrence of the *prefixs* and started from the next transaction will be presented in the projected databases. As an example, the *prefix* $<G_{1+}>$ contained in the sequence $<(G_{1+}G_{4-})_1(G_{3+})_2(G_{2-}G_{3+})_4(G_{5+})_5>$ of patient 2 (Table I), only the subsequence $<(G_{3+})_2(G_{2-}G_{3+})_4(G_{5+})_5>$ will be listed in the projected database for mining longer *CTGR-SPs*. According to the same principle, the sequences in *S* containing $<G_{1+}>$ are projected to form the $<G_{1+}>$-projected database, which consists of 4 *candidate postfixs*. Finally, by scanning $<G_{1+}>$-projected database once, the *length-2 CTGR-SPs* having *prefix* $<G_{1+}>$ can be then identified including $<(G_{1+})(G_{2-})>$: 4 ($<(G_{1+})(G_{2-})>$ appears 4 times) and $<(G_{1+})(G_{3+})>$: 4. Repeat the 3 steps, the *CTGR-SPs* longer than *length-2* can be further generated from the current *length-2 CTGR-SPs*. After constructing their respective projected databases, the $<(G_{1+})(G_{2-})>$-projected database consists of 2 *candidate postfixs*: $<(G_{3+})_3>$ and $<(G_{5+})_5>$. However, both $<(G_{3+})>$ and $<(G_{5+})>$ appear only once over the sequences involved in the $<(G_{1+})(G_{2-})>$-projected database that is below the *minSupp* (50%). Hence, the further processes for mining the $<(G_{1+})(G_{2-})>$-projected database will be terminated. On the other hand, recursive mining patterns from the $<(G_{1+})(G_{3+})>$-projected database, which contains 2 *candidate postfixs* including $<(G_{3+})_3>$ and $<(G_{2-}G_{3+})_4(G_{5+})_5>$, returns one eligible *postfix* $<G_{3+}>$ to form a *length-3 CTGR-SPs* $<(G_{1+})(G_{3+})(G_{3+})>$. According to the same criteria, we can find the remaining *CTGR-SPs* prefixed with $<G_{2-}>$ or $<G_{3+}>$ by constructing their corresponding projected databases. All resultant *CTGR-SPs* are presented in Table II.

2) Bio-logical parameter design

Although the *CTGR-Span* can efficiently mine the sequential patterns implying gene regulations across different timepoints, how to enrich the identified patterns more meaningful biologically is a very important issue. Therefore, in addition to the inherent design *minSupp*, we attempt to add 3 parameters *minimum timepoint support* (*minTSupp*), *sliding*

Table II. Example of pattern-growth-based *CTGR-Span*

Prefixs	Projected databases	CTGR-SPs
G_{1+}	$<(G_{2-}G_{3+})_2(G_{3+})_3>$ $<(G_{3+})_2(G_{2-}G_{3+})_4(G_{5+})_5>$ $<(G_{2-}G_{3+})_3>$ $<(G_{2-}G_{3+})_3>$	$<(G_{1+})(G_{2-})>$ $<(G_{1+})(G_{3+})>$ $<(G_{1+})(G_{3+})(G_{3+})>$
G_{2-}	$<(G_{3+})_3>$ $<(G_{5+})_5>$ $<(G_{2-}G_{3+})_3>$ <>	$<(G_{2-})(G_{3+})>$
G_{3+}	$<(G_{3+})_3>$ $<(G_{2-}G_{3+})_4(G_{5+})_5>$ <> $<(G_{2-}G_{3+})_3>$	$<(G_{3+})(G_{3+})>$ $<(G_{3+})(G_{2-})>$

window size (*SWS*) and *maximum time constraint* (*maxTC*) to the *CTGR-Span* based on some biological properties.

a) minTSupp

The average lengths of the two input dataset transactions are presented at the left-most N tick in Figure 1. According to the properties of the cellular physiology, several housekeeping or maintenance genes are constitutively expressed to maintain the basic cellular functions [9]. Suppose they could continuously express over all timepoints and might not be susceptible to the cellular responses. Hence, the items involved in the transactions can be firstly filtered by the literature-reported housekeeping (HK) genes. Then, we assume that the same items constitutively appear in most timepoints, these HK-like items may not associate with the cellular responses. In this regard, the items can be eliminated by a proposed parameter, *minTSupp*. The average lengths of transactions as the functions of varying *minTSupp* are shown in Figure 1. The items will be eliminated if they appear over all timepoints when the *minTSupp* is 100%. More and more HK-like items can be eliminated when decreasing the *minTSupp*. Taken together, the HK-like items contained in the converted transactions will be preliminarily filtered by the two ways.

b) SWS

Since the response time of transcriptional regulation between the pairs of endonuclear genes are not identical, detecting the gene expression values at multiple different timepoints with a fixed interval may lose some important signals, leading to an inadequate observation. On the other hand, an entire process of gene regulation may take place across more than one timepoint. In these regards, although the probe/gene expression values are detected at different timepoints, these values could be regarded as the simultaneous events in which they take place at the same timepoint. Therefore, referring to a previous work [5], the *sliding window size* (*SWS*) can be utilized to address this issue.

c) maxTC

In cell and molecular biology, since cells must react quickly to resist adverse environmental changes, massive short-term gene regulations involved in intracellular signaling pathways are dominant within the cells. If a gene regulation takes place in a long time interval, it may not exist. In this regard, a parameter *maxTC* is designed to tackle this issue. A analogous concept has also been utilized to avoid mining useless purchase patterns for sales promotion when the gap between two adjacent transactions in the pattern is too big [10].

III. EXPERIMENTAL RESULTS

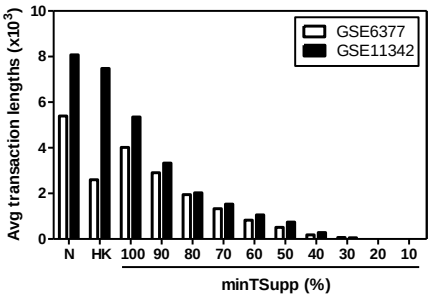

Figure 1. Average transaction lengths

1) Performance comparisons

Figure 2 showed the numbers and running times of mining sequential patterns from GSE6377 (Figure 2A and 2B) and GSE11342 (Figure 2C and 2D) using the *GSP*, *PrefixSpan* and *CTGR-Span* methods without considering *SWS* and *maxTC*. Notably, the *PrefixSpan* took over 100 hours (67,156,094 patterns) when the *minSupp* was 90% in GSE11342. In contrast, our proposed *CTGR-Span* was below 1 hour (5,340 concise patterns). For the both datasets, the numbers of traditional sequential patterns and the execution times were exponentially increased (over 500 hours) when the *minSupp* was 90%. These comparisons pointed out the strengths of *CTGR-Span* in the efficiency and effectiveness.

2) Optimal parameter tuning

Based on the biological traits, we introduced 3 parameters including the *minTSupp*, *SWS* and *maxTC* to the *CTGR-Span*. However, two questions might arise as to whether these designs are profitable for mining gene regulations and how to set these parameter values for most biologists.

In GSE6377, McDunn *et al.* have proven 85 genes involved in the inflammatory response that as the ventilator associated pneumonia (VAP) patients recovered from critical illness complicated by acute infection, the general trajectory (riboleukogram) converged, consistent with an immune attractor [1]. For the other GSE11342, Taylor *et al.* identified 85 immune response-related genes, which were altered over 6 timepoints from the blood monocytes of hepatitis C patients treated with the Peg-interferon alfa-2b plus ribavirin [2].

Gene Ontology (GO) is useful for analyzing the biological characteristics for a gene list [11]. To test the enrichment of biological processes of our results relevant to the original article subjects, the genes involved in the longest (cross *maximal timepoints*) *CTGR-SPs* based on different parameter settings were separately uploaded to the DAVID [12]. For each gene list, we examined the p-values of the original experimental results-associated GO terms including the inflammatory response (GSE6377) and immune response (GSE11342) of all items categorized as "GOTERM_BP_FAT". In Figure 3, the *CTGR-SPs* derived genes were more meaningful biologically where the highest -log(p-value) occurred when the values of *minSupp*, *SWS* and *maxTC* were set as 70% to 75%, 1 standard deviation (SD) (Figure 3D and 3I) of a set of time intervals and 5 SD (Figure 3E and 3J), respectively. Figure 3B, 3C, 3G and 3H also

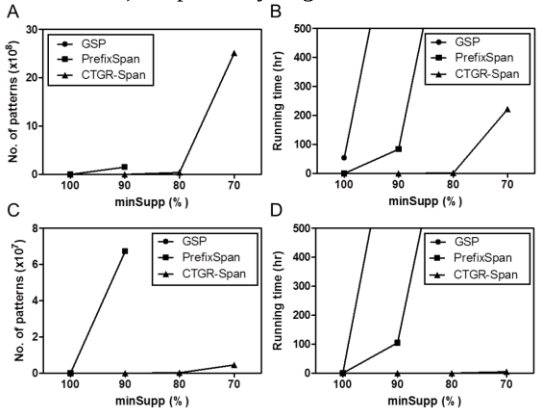

Figure 2. Performance comparisons

demonstrated that these phenomena were neither the rule number nor rule length dependent. These optimal parameter values could also be considered as the default settings to most biologists even if they have no any experiences before.

3) Evaluation with related GO terms

Although we had tuned the optimal values for each parameter, how to verify the *CTGR-SPs* more meaningful biologically is vital to the proposed method. In addition to examining the two major GO terms, we additionally referred to other highly related terms to further confirm if the *CTGR-SPs* are more strongly associated with the clinical characteristics. Figure 4A showed 11 left-to-right GO terms of GSE6377, Positive regulation of immune system process, Immune effector process, Positive regulation of immune response, Acute inflammatory response, Regulation of immune effector process, Positive regulation of immune effector process, Immune system development, Innate immune response, Inflammatory response, Immune response, and Negative regulation of immune system process. For the other GSE11342, 6 GO terms including Complement activation, Blood circulation, Lipid biosynthetic process, Blood coagulation, Immune response, and Oxygen transport were presented in Figure 4B. Overall, both examinations showed that the *CTGR-SPs* derived genes had higher correlations than the original article-provided genes.

IV. CONCLUDING REMARKS

The proposed *CTGR-Span* overcomes the flaws of the traditional sequential pattern mining methods. For the long-

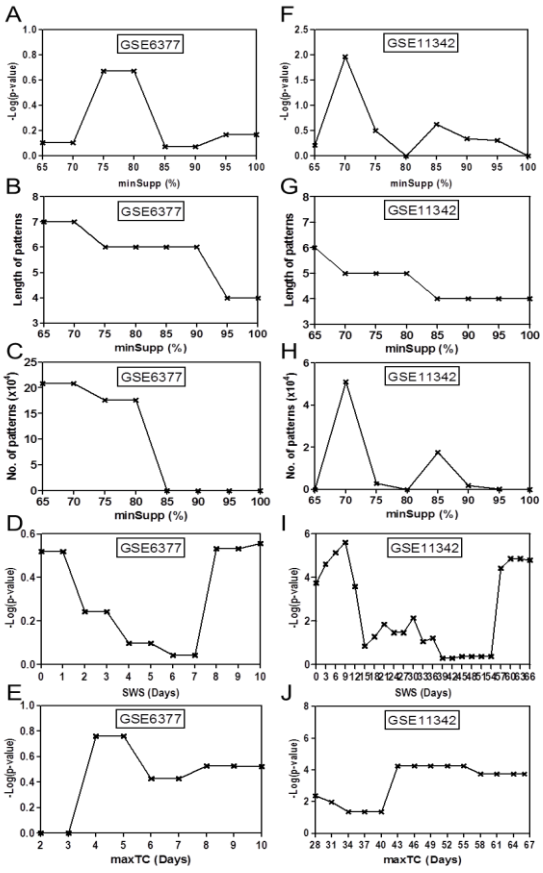

Figure 3. Examination of the longest *CTGR-SPs*

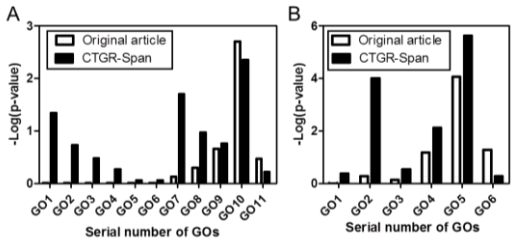

Figure 4. GO enrichment analysis

transactional databases, our method has several advantages: (i) more meaningful sequential patterns implying gene regulations in biology can be efficiently identified without sorting transactional items in advance, (ii) a lot of repeated redundant patterns can be avoided, (iii) although the GO enrichment analysis of our identified genes contained in the *CTGR-SPs* show high correlations with the original paper subjects, the gene regulations in *CTGR-SPs* will be evaluated with previous literature for further confirming their reliability in our future work. In this fashion, our evaluation results could provide more novel gene regulations during the disease course of patients.

ACKNOWLEDGMENT

This research was supported by the National Science Council of Taiwan (R.O.C.) under contract number "NSC 100-2627-B-006-020" and the National Cheng Kung University Top University Project of Ministry of Education.

REFERENCES

[1] J. E. McDunn, K. D. Husain, A. D. Polpitiya, A. Burykin, J. Ruan, Q. Li, W. Schierding, N. Lin, D. Dixon, W. Zhang, C. M. Coopersmith, W. M. Dunne, M. Colonna, B. K. Ghosh, and J. P. Cobb, "Plasticity of the systemic inflammatory response to acute infection during critical illness: development of the riboleukogram," *PLoS One*, vol. 3, p. e1564, 2008.

[2] M. W. Taylor, T. Tsukahara, J. N. McClintick, H. J. Edenberg, and P. Kwo, "Cyclic changes in gene expression induced by Peg-interferon alfa-2b plus ribavirin in peripheral blood monocytes (PBMC) of hepatitis C patients during the first 10 weeks of treatment," *J Transl Med*, vol. 6, p. 66, 2008.

[3] R. Agrawal and R. Srikant, "Mining Sequential Patterns," presented at the Proceedings of the Eleventh International Conference on Data Engineering, 1995.

[4] J. Pei, J. Han, B. Mortazavi-asl, H. Pinto, Q. Chen, U. Dayal, and M.-c. Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth," presented at the Proceedings of the 17th International Conference on Data Engineering, 2001.

[5] R. Srikant and R. Agrawal, "Mining Sequential Patterns: Generalizations and Performance Improvements," presented at the Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology, 1996.

[6] J. Pei, J. Han, and W. Wang, "Mining sequential patterns with constraints in large databases," presented at the Proceedings of the eleventh international conference on Information and knowledge management, McLean, Virginia, USA, 2002.

[7] X. Yan, J. Han, and R. Afshar, "CloSpan: Mining Closed Sequential Patterns in Large Datasets," in *In SDM*, 2003, pp. 166-177.

[8] M. Kim, H. Shin, T. Su Chung, J. G. Joung, and J. H. Kim, "Extracting regulatory modules from gene expression data by sequential pattern mining," *BMC Genomics*, vol. 12 Suppl 3, p. S5, Nov 30 2011.

[9] C. W. Chang, W. C. Cheng, C. R. Chen, W. Y. Shu, M. L. Tsai, C. L. Huang, and I. C. Hsu, "Identification of human housekeeping genes and tissue-selective genes by microarray meta-analysis," *PLoS One*, vol. 6, p. e22859, 2011.

[10] M.-Y. Lin, S.-C. Hsueh, and C.-W. Chang, "Mining Closed Sequential Patterns with Time Constraints," *J. Inf. Sci. Eng.*, pp. 33-46, 2008.

[11] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: tool for the unification of biology. The Gene Ontology Consortium," *Nat Genet*, vol. 25, pp. 25-9, May 2000.

[12] G. Dennis, Jr., B. T. Sherman, D. A. Hosack, J. Yang, W. Gao, H. C. Lane, and R. A. Lempicki, "DAVID: Database for Annotation, Visualization, and Integrated Discovery," *Genome Biol*, vol. 4, p. P3, 2003.