Triclustering on Temporary Microarray Data using the TriGen Algorithm

D. Gutiérrez-Avilés, C. Rubio-Escudero and J. C. Riquelme
Departamento Lenguajes y Sistemas Informáticos
Universidad de Sevilla
Sevilla, España
Email: davgutavi@alum.us.es, crubioescudero@us.es and riquelme@us.es

***Abstract*—The analysis of microarray data is a computational challenge due to the characteristics of these data. Clustering techniques are widely applied to create groups of genes that exhibit a similar behavior under the conditions tested. Biclustering emerges as an improvement of classical clustering since it relaxes the constraints for grouping allowing genes to be evaluated only under a subset of the conditions and not under all of them. However, this technique is not appropriate for the analysis of temporal microarray data in which the genes are evaluated under certain conditions at several time points. In this paper, we propose the TriGen algorithm, which finds triclusters that take into account the experimental conditions and the time points, using evolutionary computation, in particular genetic algorithms, enabling the evaluation of the gene's behavior under subsets of conditions and of time points.**

***Keywords*-triclustering, microarrays, temporary data, genetic algorithms**

I. Introduction

The use of high throughput processing techniques has revolutionized the technological research and has exponentially increased the amount of data available [1] . Particularly, microarrays have revolutionized biological research by its ability to monitor changes in RNA concentration in thousands of genes simultaneously [2]. A common practice when analyzing gene expression data is to apply clustering techniques, creating groups of genes that exhibit similar expression patterns. These clusters are interesting because it is considered that genes with similar behavior patterns can be involved in similar regulatory processes [3]. Although in theory there is a big step from correlation to functional similarity of genes, several articles indicate that this relation exists [4]. Traditional clustering algorithms work on the whole space of data dimensions examining each gene in the dataset under all conditions tested. Biclustering techniques [5] go a step further by relaxing the conditions and by allowing assessment only under a subset of the conditions of the experiment, and it has proved to be successful finding gene patterns [6], [7]. However, clustering and biclustering are insufficient when analyzing data from microarray experiments where attention is payed on how the time affects gene's behavior. There is a lot of interest in this type of time series experiment because they allow an in-depth analysis of molecular processes in which the time evolution is important, for example, cell cycles, development at the molecular level or evolution of diseases [8]. Therefore is necessary to develop specific tools for data analysis in which genes are evaluated under certain conditions considering the time factor. In this context we present the TriGen algorithm, which goes a step further than clustering and biclustering techniques in the creation of groups of pattern similarity for genes. TriGen works on a three-dimensional space, thus taking into account the time factor, and allowing the evaluation of the behavior of genes only under certain conditions and only under certain time points. TriGen applies an evolutionary technique, genetic algorithms, to find solutions that we refer to as triclusters. Other works related with this approach are in [9] and [10]. The rest of the paper is structured as follows: Section II describes the algorithm in detail, Section III shows the results using both synthetic and real data. Section IV summarizes the conclusions reached and proposals for future work.

II. Methodology

We describe the implementation of the TriGen algorithm. In this section we explain the inputs and outputs of the algorithm and we provide a detailed description of the evolutionary process and all the operators implied.

A. Input data

The input data is obtained from temporal microarray experiments. Each of these microarrays reveals the expression level under specific experimental conditions and at an instant of time. Therefore, the input data consists of T number of microarrays, as many as time points to be analyzed. Each value of a microarray for an specific time t represents the level of gene expression of a gene g under a specific experimental condition c.

B. Definition of Tricluster

We define a tricluster as a subset of time points T, a subset of genes G and a subset of conditions C extracted from the input data. In this particular work, each tricluster contains the expression values of the these three sets and a fitness value that indicates the tricluster's quality. The fitness function will be described in detail in Section II-C6. Qualitatively, a tricluster will provide information on behavior pattern of a

```
Input:  Temporary microarray data
Output:  Tricluster Solution Set

Begin TriGen algorithm
  Repeat for each Tricluster solution
    Generate Initial Population
    Evaluate population
    Repeat for Number of Generations
      Select Population
      Cross Population
      Mutate Population
      Evaluate Population
    End Repeat
    Select Best One
    Include Best One in Solution Set
  End Repeat
End TRIGEN algorithm
```

Figure 1: TriGen algorithm

subset of genes under certain conditions and at certain time points.

C. TriGen Algorithm Description

TriGen is based on a genetic algorithm. The evolutionary process is composed of several operators (See Figure 1):

- Initialization: in which the initial population will be created with chromosomes or candidate solutions.
- Evaluation: which measures the quality of each chromosome or individual of the population.
- Selection: which serves to decide which individuals will survive to the next generation.
- Crossover: creates the necessary connections between pairs of individuals to share new genetic material.
- Mutation, which performs specific changes to individuals to ensure genetic variability of future generations, i.e. exploring new spaces of solutions.

We discuss in detail each of these operators.

1) Codification of Individuals: Each member of the population represents a tricluster which is a potential solution. It has genetic material that will be manipulated by the genetic operators described in Sections II-C4 and Section II-C5. This genetic material is composed by a set of chromosomes, they are a subset of time points T, a subset of genes G and a subset of conditions C extracted from the input data. Each of these subsets has a number of genes, they correspond to the coordinates of the input data. Figure 2 we can see a chart with a correspondence between genetic representation and tricluster.

2) Generation of Initial Population: This operator receives as parameter the number of individuals desired for the initial population.To compose each individual, we choose randomly a subset of timing, genes and conditions of the input data.This process is repeated as many times as specified by the input parameter described above.

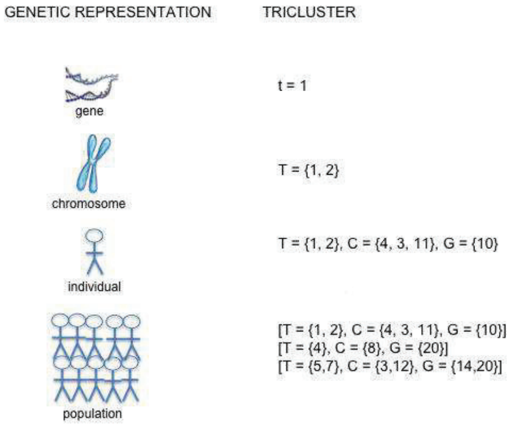

GENETIC REPRESENTATION
TRICLUSTER
gene
t = 1
chromosome
T = {1, 2}
individual
T = {1, 2}, C = {4, 3, 11}, G = {10}
population
[T = {1, 2}, C = {4, 3, 11}, G = {10}]
[T = {4}, C = {8}, G = {20}]
[T = {5,7}, C = {3,12}, G = {14,20}]

Figure 2: Genetic algorithm representation

3) Selection: A tournament selection mechanism, in which three groups of individuals are randomly created, then they are sorted from lowest to highest according to the fitness function.The method selects randomly a number of individuals according to an input parameter from these three groups.

4) Crossover: This operator completes the population in the next generation P_t generating two new individuals (children) combining the genetic material from two existing ones (parents). For each point of the two parents get two children so the number of crossings is determined by number of individuals who are required to complete the population. This is a one-point cross that determine a random point cross for the times, genes and conditions and mixing each of the parts to obtain two child by crossing.

5) Mutation: This operator selects, based on a mutation probability input parameter, a number of individuals who suffer a random out of six: add a time component, a gene component or a condition component , or remove a time component, gene component or condition component.

6) Evaluation: Since triclustering emerges as an improvement of biclustering to analyze microarray data taking into account the temporal dimension, we have adapted the classical biclustering fitness function, Mean Squared Residue (MSR), presented by Cheng and Church in [11], to the three dimensional space. MSR compares the similarity of each value in the bicluster to the mean values of all genes under the same condition, the mean of the gene under the other conditions included in the bicluster, and the mean of all values in the bicluster. In the case of triclustering, we will assess the similarity of each value not only related to genes and conditions, but also including the temporary plane, i.e., we asses how a gene g behaves under all conditions C at the time points T, how a condition c affects all genes G in time T, and the time factor t in relation to genes G and conditions C, as well as the mean value of all the tricluster. This is formalized as follows:

$$r_{GCT} = \frac{\displaystyle\sum_{g\epsilon G, c\epsilon C, t\epsilon T} r_{gct}^2}{|G| * |C| * |T|} - Weights$$

in the first member of subtraction, the numerator is:

$$\begin{aligned} \rlap{r}g_{ct} \quad = \quad & V_{gct} + M_{GC}(t) + M_{GT}(c) + M_{CT}(g) - \\ & M_G(c,t) - M_C(g,t) - M_T(g,c) - M_{GCT} \end{aligned}$$

where V_{gct} is the tricluster value being evaluated, $M_{GC}(t)$ is the mean of the genes under conditions at a point in time t, $M_{GT}(c)$ is the mean of the genes over time under a condition c, $M_{CT}(g)$ is the mean of a gene g in time under the conditions, $M_G(c,t)$ is the mean of the genes under one condition and a time point, $M_C(g,t)$ is the mean of the values of a gene at a time point under conditions, $M_T(g,c)$ is the mean of a gene under a condition at all time points and M_{GCT} is the mean value of all points of tricluster.
The denominator factor is:

$$|G| * |C| * |T|$$

where $|G|$, $|C|$ and $|T|$ are, respectively, the number of genes, times and conditions in the tricluster under evaluation. And the second member of subtraction, $Weights$, corresponds to:

$$Weights = |g| * w_g + |c| * w_c + |t| * w_t$$

where w_g, w_c and w_t are the weights of the genes, conditions and times for the solution tricluster respectively. When increasing the value of one of these weights, we favor the TriGen algorithm finding triclusters with a greater number of components on that term.

III. Results

We show the results obtained applying the TriGen algorithm both to real and to synthetic data. Synthetic data are widely used not only for testing the performance of microarray analyzing techniques [12] but also in more general data mining publications [13].

A. Results using Synthetic Data

The set of synthetic data has been generated using a software application developed for such purpose. For this particular work, we have simulated data from 5 different time points and 10 conditions using microarrays containing 1000 genes. Each gene is assigned a random value which is contained in the rank, respectively for each condition, $[1, 15]$, $[7, 35]$, $[60, 75]$, $[0, 25)$, $[30, 100]$, $[71, 135]$, $[160, 375]$, $[5, 30]$, $[25, 40]$ y $[10, 30]$. In such data set, we have allocated a tricluster with all its values fixed to 1. The size of the tricluster is $time = 5$, $genes = 8$ and $conditions = 8$. TriGen was able to successfully find a solution containing the aforementioned tricluster. The execution was made with the following parameters: 100 generations and 500 members in the population. The selection parameter is 70% and the mutation probability is 5%. The weight values have been adjusted to $w_g = 0.01$, $w_c = 0.55$ y $w_t = 0.35$, in order to favor the number of conditions and time points, since the genes show high dimensionality in relation to conditions and time.

B. Results using Real Data

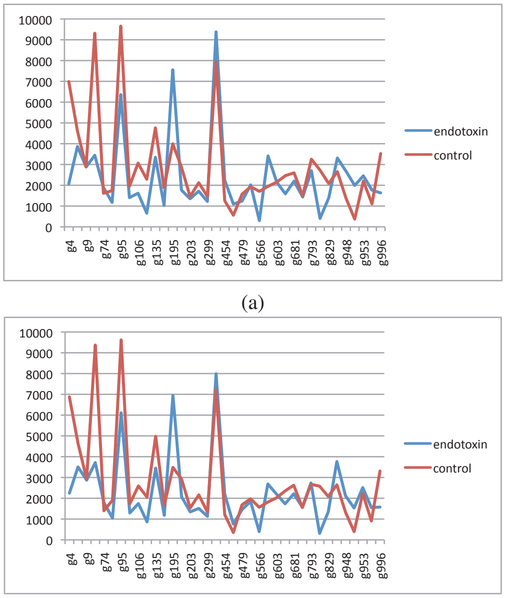
The problem under study deals with inflammation and the host response to injury. Understanding the inflammation process is critical because the body uses inflammation to protect itself from infection or injury (e.g., crushes, massive bleeding, or a serious burn). The host response to trauma and burns is a collection of biological and pathological processes that depends critically upon the regulation of the human immuno-inflammatory response [14].

The data has been acquired from an experiment about inflammation and host response to injury carried out with microarrays. In this experiment, blood samples from 8 volunteers are analyzed, 4 treated with a toxin that simulates an inflammatory process and 4 with a placebo. Samples were taken at 6 time points throughout 24 hours, obtaining a total of 48 microarrays. We work with a set of 2155 genes selected as relevant for the problem [15] considering 2 conditions: endotoxin and placebo. The algorithm has been executed to extract 10 solutions, i.e. 10 triclusters with the following parameters: 100 generations, 400 members in the population, 60% for selection and mutation probability 20%. The weights applied have been $w_g = 0.01$, $w_c = 0.55$ y $w_t = 0.55$.

Compared to the previous experiment we have changed w_t, members in the population, selection and mutation parameters.The principal reason which this values have changed is for the input data characteristics, now input data has more number of genes and only two conditions therefore we have to favor recombination of chromosomes in evolutionary process. Specifically, the effect over solutions that we achieve with parameter change is as follow:

- w_t increasing: we favor triclusters with greater number of time points.
- Members in the population decreasing: we favor solutions which have built by crossover and mutation operators.
- Selection decreasing: we get that a greater number of triclusters have crated by crossover and mutation operators.
- Mutation increasing: we get a greater variability of chromosomes in the solutions.

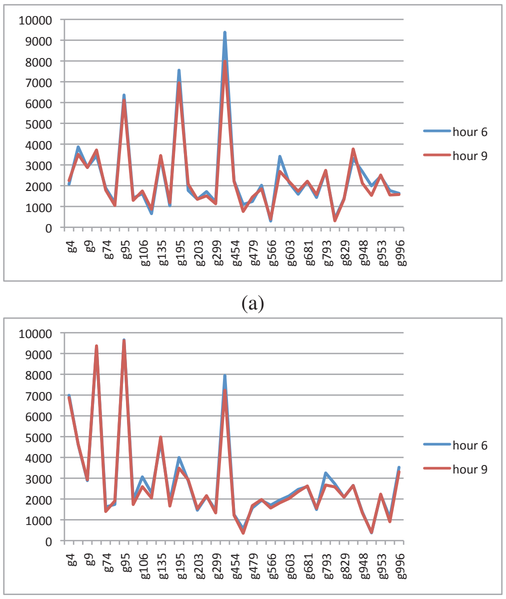
For legibility reasons we focus in one of the solutions, a tricluster gathering 37 genes under the 2 conditions, endotoxin and placebo (control), and 2 time points, hours 6 and 9.



(a)

(b)

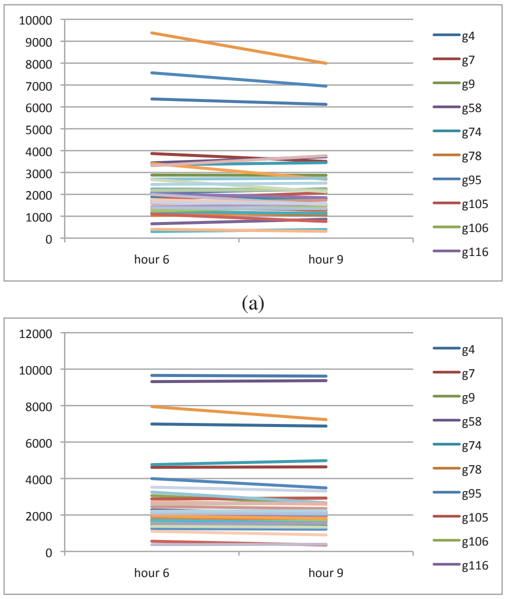
Figure 3: Gene expression values under two conditions at hours 6 (a) and 9 (b).



(a)

(b)

Figure 4: Gene expression values under two hours at endotoxin (a) and control (b) conditions.



(a)

(b)

Figure 5: Gene expression for 6 and 9 hours under gene solution set at endotoxin (a) and control (b) conditions.

To view this solution, we show three groups of graphics: In Figure 3 we present the outline of gene expression values (Y axis) for each solution gene point (X axis) comparing the endotoxin and control experiment setting time points to 6 (a) and 9 (b) hours. In Figure 4 we present the outline of gene expression values (Y axis) for each solution gene point (X axis) comparing 6 and 9 time points setting the experiments to endotoxin (a) and control (b). Finally in Figure 5 we present the outline of gene expression values (Y axis) for each time point (X axis) comparing each solution gene setting the experiments to endotoxin (a) and control (b). These figures show that TriGen is able to mine coherent clusters across any combination of the gene-sample-time dimensions.

We observe in Figure 4(a) and Figure 4(b) that the lines exhibit an extremely similar behavior, so the TriGen algorithm has been capable to extract an existing pattern in gene expressions levels, time points and experiments. We see that the same situation occurs for the rest of figures (3(a), 3(b), 5(a) and 5(b)). Therefore, we can conclude that the presented algorithm TriGen has been capable to perform the task it has been created for.

IV. Conclusions and Future Work

We have presented the tricluster algorithm TriGen, which represents an step further than clustering and biclustering in the analysis of temporal microarray data. TriGen groups genes which exhibit a similar behavior under a subset of conditions and under a subset of time points. It is genetic based algorithm, with an evaluation function developed as the natural 3D extension from the classic function evaluation for biclustering proposed by Cheng y Church in [11]. The results show that the algorithm is capable to mine triclusters of genes based on their expression levels. TriGen is still in an early development stage, so there is still a lot of work to do, not only for the algorithm, such as a deeper study of the evaluation function or parallelization of the algorithm to make it faster, but also for the validation phase or the application of this algorithm to other types of data, such as image analysis.

REFERENCES

[1] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[2] P. Brown and D. Botstein, "Exploring the new world of the genome with dna microarrays," *Nature Genet.*, vol. 21, no. Suppl., pp. 33–37, 1999.

[3] M. Tan, E. Smith, J. Broach, and C. Floudas, "Microarray data mining: A novel optimization-based approach to uncover biologically coherent structures," *BMC Bioinformatics*, vol. in press.

[4] P. D'haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.

[5] J. Hartigan, "Direct clustering of a data matrix," *Journal of the American Statistical Association*, vol. 67, no. 337, pp. 123–129, 1972.

[6] M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proceedings of the National Academy of Sciences*, vol. 95, no. 25, p. 14863, 1998.

[7] B. Pontes, F. Divina, R. Giráldez, and J. Aguilar-Ruiz, "Improved biclustering on expression data through overlapping control," *International Journal of Intelligent Computing and Cybernetics*, vol. 3, no. 2, pp. 293–309, 2010.

[8] Z. Bar-Joseph, "Analyzing time series gene expression data," *Bioinformatics*, vol. 20, no. 16, p. 2493, 2004.

[9] L. Zhao and M. J. Zaki, "tricluster: An effective algorithm for mining coherent clusters in 3d microarray data," *In Proc. of the 2005 ACM SIGMOD international conference on Management of data*, pp. 694–705, 2005.

[10] P. Mahanta, H. Ahmed, D. Bhattacharyya, and J. Kalita, "Triclustering in gene expression data analysis: A selected survey," *Emerging Trends and Applications in Computer Science (NCETACS)*, pp. 1–6, 2011.

[11] Y. Cheng and G. Church, "Biclustering of expression data." in *Proceedings/... International Conference on Intelligent Systems for Molecular Biology; ISMB. International Conference on Intelligent Systems for Molecular Biology*, vol. 8, 2000, p. 93.

[12] K. Hakamada, M. Okamoto, and T. Hanai, "Novel technique for preprocessing high dimensional time-course data from dna microarray: mathematical model-based clustering," *Bioinformatics*, vol. 22, no. 7, p. 843, 2006.

[13] R. Pargas, M. Harrold, and R. Peck, "Test-data generation using genetic algorithms," *Software Testing Verification and Reliability*, vol. 9, no. 4, pp. 263–282, 1999.

[14] S. E. Calvano, W. Xiao, D. R. Richards, R. M. Felciano, H. V. Baker, R. J. Cho, R. O. Chen, B. H. Brownstein, J. P. Cobb, S. K. Tschoeke, C. Miller-Graziano, L. L. Moldawer, M. N. Mindrinos, R. W. Davis, R. G. Tompkins, S. F. Lowry, and I. A. Large Scale Collab Res Program, "A network-based analysis of systemic inflammation in humans," *Nature*, vol. 437, 2005.

[15] C. Rubio-Escudero, R. Romero-Zaliz, I. Zwir, and C. del Val, "Optimization of multi-classifiers for computational biology: application to gene finding and expression," *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)*, vol. 125, no. 3, pp. 599–611, 2010.