Unravelling the Yeast Cell Cycle Using the TriGen Algorithm

David Gutiérrez-Avilés, Cristina Rubio-Escudero, and José C. Riquelme

Dpto. Lenguajes y Sistemas Informáticos, Universidad de Sevilla
dgutierrez@alum.us.es, {crubioescudero,riquelme}@us.es

Abstract. Analyzing microarray data represents 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 present the results of applying the TriGen algorithm, a genetic algorithm that finds triclusters that take into account the experimental conditions and the time points, to the yeast cell cycle problem, where the goal is to identify all genes whose expression levels are regulated by the cell cycle.

Keywords: microarrays, temporary data, genetic algorithms, yeast.

1 Introduction

The use of high throughput processing techniques has revolutionized the technological research and has exponentially increased the amount of data available [5]. 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 [12]. 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 [8] 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 [6,10]. 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 [1]. Therefore, the use of specific tools for data analysis in which genes are evaluated under certain conditions considering the time factor becomes necessary. The TriGen algorithm 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. We present the results of applying the TriGen algorithm to the yeast cell cycle problem [11], where the objective is to create a comprehensive catalog of yeast genes whose transcript levels vary periodically within the cell cycle. The rest of the paper is structured as follows. Section 2 describes the TriGen algorithm in detail, Section 3 shows the results with synthetic data and with the yeast data. Section 4 summarizes the conclusions reached and proposals for future work.

2 Methodology

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.

2.1 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.

2.2 Definition of Tricluster

We define a tricluster as a subset of time instants T, a subset of genes G and a subset of conditions C extracted from the input data (described above in section 2.1) which provide a behavior pattern of expression levels of each gene g contained in G under each experimental condition c contained in C and on each time point t contained in T. In comparison of predecessor technologies, a tricluster is, as we said before, a subset of genes, conditions and time points while a cluster is a subset of genes and a bicluster is a subset of genes and conditions. 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 2.3 (Evaluation operator).

2.3 TriGen Algorithm Description

TriGen is based on a genetic algorithm. The evolutionary process is composed for an initialization method in which the initial population will be created with chromosomes or candidate solutions and several operators: 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 and finally mutation, which performs punctual changes to individuals to ensure genetic variability of future generations, i.e., exploring new spaces of solutions (See Figure 1).

We discuss in detail each of these methods and operators.

```
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
```

Fig. 1. TriGen algorithm

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 "Genetic Operators" below. This genetic material is composed by a set of chromosomes, they are a subset of time instants T, a subset of genes G and a subset of conditions C extracted from the input data. Each of them is composed by a number of genes, they are the components of the tricluster (they correspond to the components of the input data).

Generation of Initial Population: This method receives a 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 Operators

Selection. A tournament selection mechanism, in which groups of individuals are randomly created sorted from lowest to highest according to the fitness function, and then a random selection from the three groups of the individuals required according to an input parameter is made.

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.

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.

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 [3], 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:

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

in the first member of equation, the numerator is:

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

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 equation, $Weights$, corresponds to:

$$Weights = |G| * w_g + |C| * w_c + |T| * w_t$$

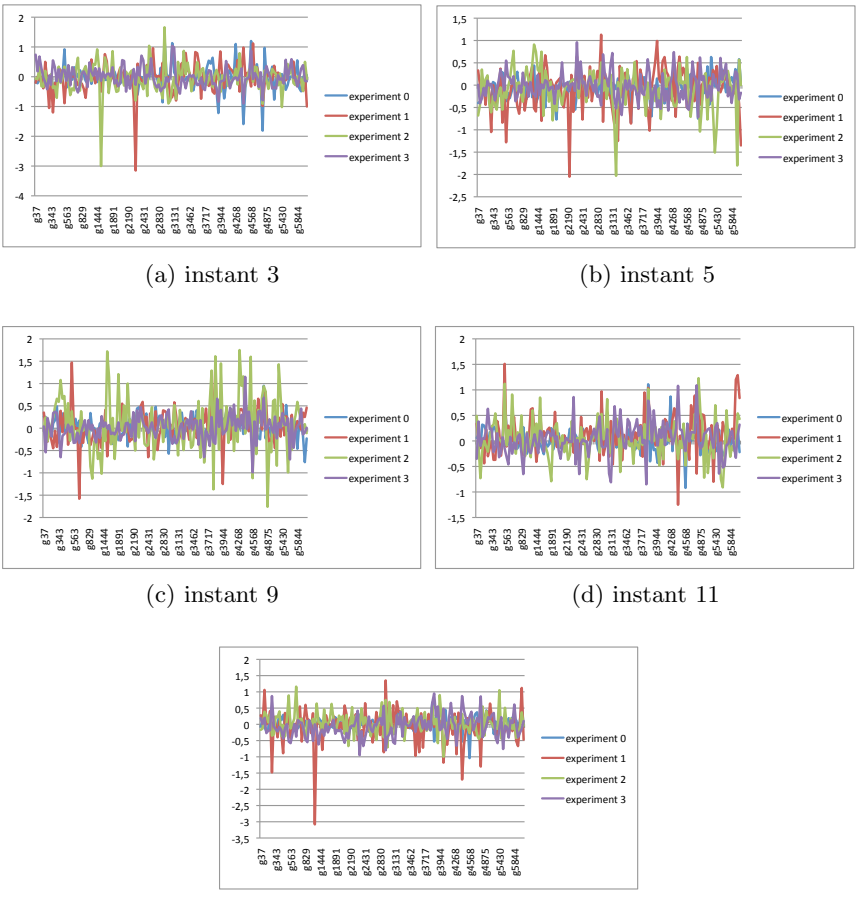
where w_g, w_c and w_t are the weights of the genes, conditions and times for the solution tricluster respectively and $|G|$, $|C|$ and $|T|$ correspond again to the number of genes, times and conditions in the tricluster under evaluation. When increasing the value of one of these weights, we favor the TriGen algorithm finding triclusters with a greater number of components on that term.

3 Results

We show the results obtained applying the TriGen algorithm both to real and to synthetic data.

3.1 Results Using Synthetic Data

Synthetic data are widely used not only for testing the performance of microarray analyzing techniques [7] but also in more general data mining publications [9]. 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.



(a) instant 3

(b) instant 5

(c) instant 9

(d) instant 11

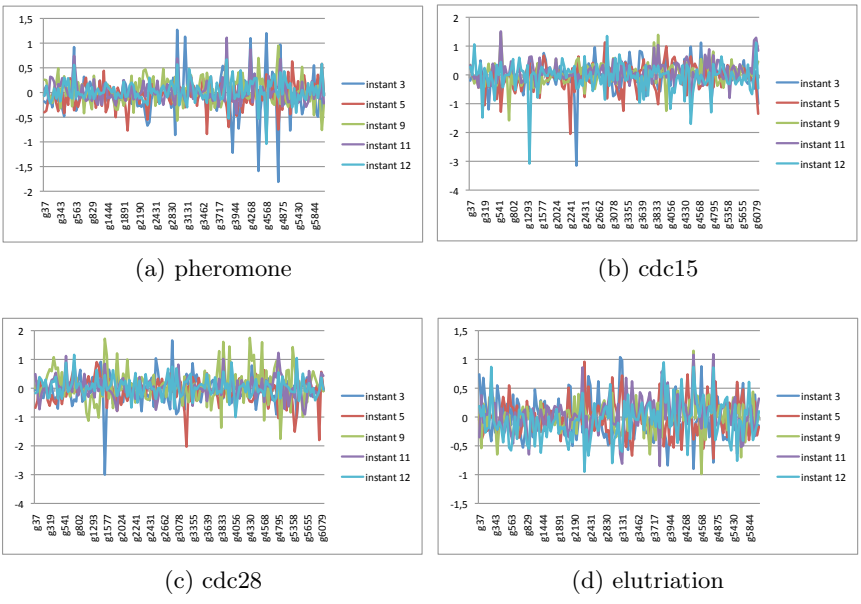
(e) instant 12

Fig. 2. Gene expression values under four experiments at instant 3 (a), 5 (b), 9 (c), 11 (d) and 12 (e)

3.2 Results Using Real Data

We have applied the TriGen algorithm to the yeast (Saccharomyces Cerevisiae) cell cycle problem [11]. The yeast cell cycle analysis project's goal is to identify all genes whose mRNA levels are regulated by the cell cycle. By applying TriGen to this dataset, we aim to find a pattern on this cell cycle.

The data is available in http://genome-www.stanford.edu/cellcycle/. In this experiment, 6179 genes are analyzed under 6 conditions, termed cln3, clb2, pheromone, cdc15, cdc28 and elutriation [11]. Samples were taken at 2 time points for cln3, 2 for clb2, 18 for pheromone, 24 for cdc15, 17 for cdc28 and 14 for elutriation. To apply the TriGen algorithm we did not take into account the conditions with only 2 time points, since they are not relevant for a time course



(a) pheromone (b) cdc15

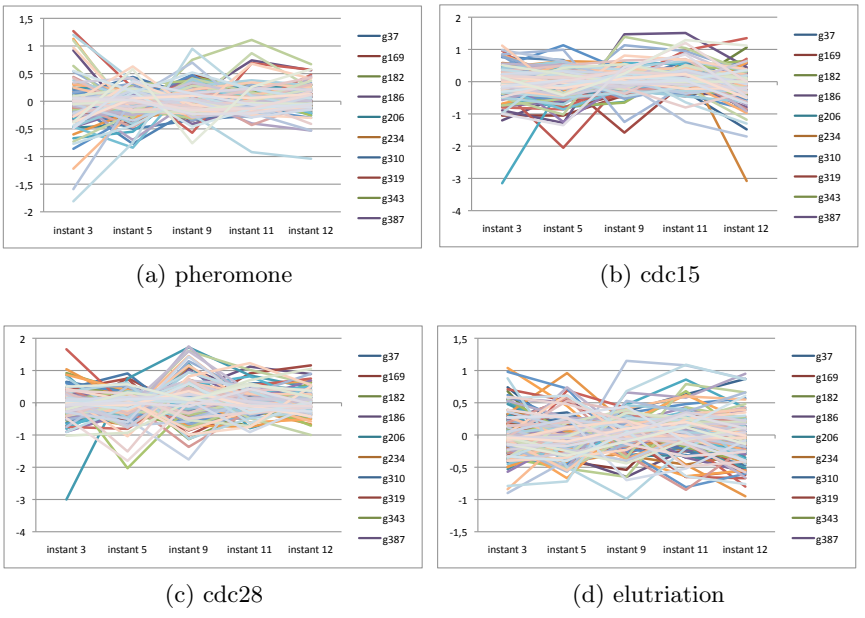
(c) cdc28 (d) elutriation

Fig. 3. Gene expression values under five instants at pheromone (a) cdc15 (b) cdc28 (c) and elutriation (d) experiments

experiment, so we used the first 14th time points of the pheromone, cdc15, cdc28 and elutriation experiment. Therefore our dataset contains 14 time points, 6179 genes and 4 conditions. The algorithm has been executed to extract 10 solutions, i.e. 10 triclusters with the following parameters: 100 generations, 50 members in the population, 50% for selection probability and 70% for mutation. The weights applied have been $w_g = 0.0$, $w_c = 100.0$ y $w_t = 0.0002$, thus we favored the condition dimension and penalized gene dimension to get a reduced subset of genes on solution triclusters.

For legibility reasons we focuss in one of the solutions, a tricluster gathering 142 genes under the 4 experiments, pheromone (experiment 0), cdc15 (experiment 1), cdc28 (experiment 2) and elutriation (experiment 3), and 5 time points, instants 3, 5, 9, 11 and 12.

We show three groups of graphics related to this solution: In Figure 2 we present the outline of the gene expression values (Y axis) for each solution gene point (X axis) comparing the pheromone (experiment 0), cdc15 (experiment 1), cdc28 (experiment 2) and elutriation (experiment 3) experiments setting time points to instants 3 (a), 5 (b), 9 (c), 11 (d) and , 12 (e). In Figure 3 we present the outline of Gene expression values (Y axis) for each solution gene point (X axis) comparing 3, 5, 9, 11 and 12 time points setting the experiments to pheromone (a) cdc15 (b) cdc28 (c) and elutriation (d). Finally in Figure 4 we present the outline of Gene expression values (Y axis) for each time point (X axis) comparing each solution gene setting the experiments to pheromone (a) cdc15 (b) cdc28 (c) and elutriation (d).



(a) pheromone (b) cdc15

(c) cdc28 (d) elutriation

Fig. 4. Gene expression for five instants under gene solution set at pheromone (a) cdc15 (b) cdc28 (c) and elutriation (d) experiments

We see that the algorithm has been capable to group together genes with very similar gene expression values (comprised in the [-2,2] interval) for the three dimensions visited. Therefore, TriGen has shown its ability to mine groups of co-expressed genes taking into account the time dimension.

4 Conclusions and Future Work

We have presented the results obtained by applying the tricluster algorithm TriGen to the yeast cell cycle problem. TriGen represents an step further than clustering and biclustering in the analysis of temporal microarray data since it 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 [3]. 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 for other types of data, such as image analysis.

References

1. Bar-Joseph, Z.: Analyzing time series gene expression data. Bioinformatics 20(16), 2493 (2004)
2. Brown, P., Botstein, D.: Exploring the new world of the genome with dna microarrays. Nature Genet. 21(suppl.), 33–37 (1999)
3. Cheng, Y., Church, G.: 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, p. 93 (2000)
4. D'haeseleer, P., Liang, S., Somogyi, R.: Genetic network inference: from coexpression clustering to reverse engineering. Bioinformatics 16(8), 707–726 (2000)
5. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press (1998)
6. Eisen, M., Spellman, P., Brown, P., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences 95(25), 14863 (1998)
7. Hakamada, K., Okamoto, M., Hanai, T.: Novel technique for preprocessing high dimensional time-course data from dna microarray: mathematical model-based clustering. Bioinformatics 22(7), 843 (2006)
8. Hartigan, J.A.: Direct clustering of a data matrix. Journal of the American Statistical Association 67(337), 123–129 (1972)
9. Pargas, R., Harrold, M., Peck, R.: Test-data generation using genetic algorithms. Software Testing Verification and Reliability 9(4), 263–282 (1999)
10. Pontes, B., Divina, F., Giráldez, R., Aguilar-Ruiz, J.: Improved biclustering on expression data through overlapping control. International Journal of Intelligent Computing and Cybernetics 3(2), 293–309 (2010)
11. Spellman, P., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Molecular Biology of the Cell 9(3), 3273–3297 (1998)
12. Tan, M., Smith, E., Broach, J., Floudas, C.: Microarray data mining: A novel optimization-based approach to uncover biologically coherent structures. BMC Bioinformatics (in press)