Possibilistic Approach to Biclustering: An Application to Oligonucleotide Microarray Data Analysis

Maurizio Filippone[1], Francesco Masulli[1], Stefano Rovetta[1],
Sushmita Mitra[2,3], and Haider Banka[2]

[1] DISI, Dept. Computer and Information Sciences, University of Genova and CNISM,
16146 Genova, ITALY {filippone, masulli, rovetta}@disi.unige.it
[2] Center for Soft Computing: A National Facility, Indian Statistical Institute,
Kolkata 700108, INDIA {hbanka_r, sushmita}@isical.ac.in
[3] Machine Intelligence Unit, Indian Statistical Institute,
Kolkata 700108, INDIA

Abstract. The important research objective of identifying genes with similar behavior with respect to different conditions has recently been tackled with biclustering techniques. In this paper we introduce a new approach to the biclustering problem using the Possibilistic Clustering paradigm. The proposed Possibilistic Biclustering algorithm finds one bicluster at a time, assigning a membership to the bicluster for each gene and for each condition. The biclustering problem, in which one would maximize the size of the bicluster and minimizing the residual, is faced as the optimization of a proper functional. We applied the algorithm to the Yeast database, obtaining fast convergence and good quality solutions. We discuss the effects of parameter tuning and the sensitivity of the method to parameter values. Comparisons with other methods from the literature are also presented.

1 Introduction

1.1 The biclustering problem

In the last few years the analysis of genomic data from DNA microarray has attracted the attention of many researchers since the results can give a valuable information on the biological relevance of genes and correlations between them [1].

An important research objective consists in identifying genes with similar behavior with respect to different conditions. Recently this problem has been tackled with a class of techniques called *biclustering* [2–5].

Let x_{ij} be the expression level of the i-th gene in the j-th condition. A *bicluster* is defined as a subset of the $m \times n$ data matrix X. A bicluster [2–5] is a pair $(\mathbf{g}, \mathbf{c})$, where $\mathbf{g} \subset \{1, \ldots, m\}$ is a subset of genes and $\mathbf{c} \subset \{1, \ldots, n\}$ is a subset of conditions. We are interested in largest biclusters from DNA microarray data that do not exceed an assigned homogeneity constraint [2] as they can supply relevant biological information.

The size (or volume) n of a bicluster is usually defined as the number of cells in the gene expression matrix X belonging to it, that is the product of the cardinalities $n_g = |\mathbf{g}|$ and $n_c = |\mathbf{c}|$:

$$n = n_g \cdot n_c \tag{1}$$

Let

$$d_{ij}^2 = \frac{\left(x_{ij} + x_{IJ} - x_{iJ} - x_{Ij}\right)^2}{n} \tag{2}$$

where the elements x_{IJ}, x_{iJ} and x_{Ij} are respectively the bicluster mean, the row mean and the column mean of X for the selected genes and conditions:

$$x_{IJ} = \frac{1}{n} \sum_{i \in \mathbf{g}} \sum_{j \in \mathbf{c}} x_{ij} \tag{3}$$

$$x_{iJ} = \frac{1}{n_c} \sum_{j \in \mathbf{c}} x_{ij} \tag{4}$$

$$x_{Ij} = \frac{1}{n_g} \sum_{i \in \mathbf{g}} x_{ij} \tag{5}$$

We can define now G as the mean square residual, a quantity that measures the bicluster homogeneity [2]:

$$G = \sum_{i \in \mathbf{g}} \sum_{j \in \mathbf{c}} d_{ij}^2 \tag{6}$$

The residual quantifies the difference between the actual value of an element x_{ij} and its expected value as predicted from the corresponding row mean, column mean, and bicluster mean.

To the aim of finding large biclusters we must perform an optimization that maximizes the bicluster cardinality n and at the same time minimizes the residual G, that is reported to be an NP-complete task [6]. The high complexity of this problem has motivated researchers to apply various approximation techniques to generate near optimal solutions. In the present work we take the approach to combine the criteria in a single objective function.

1.2 Overview of previous works

A survey on biclustering is given in [1] where a categorization of the different heuristic approaches is shown, such as iterative row and column clustering, divide and conquer strategy, greedy search, exhaustive biclustering enumeration, distribution parameter identification and others.

In the microarray analysis framework, the pioneering work by Cheng and Church [2] employs a set of greedy algorithms to find one or more biclusters in gene expression data, based on a mean squared residue as a measure of similarity. One bicluster is identified at a time iteratively. The masking of null values of the discovered biclusters are replaced by large random numbers that helps to find new biclusters at each iteration. Nodes are deleted and added and also the inclusion of inverted data is taken into consideration when finding biclusters. The masking procedure [7] results in a phenomenon of *random interference*, affecting the subsequent discovery of large-sized biclusters. A two-phase probabilistic algorithm termed Flexible Overlapped Clusters (FLOC) has been proposed by Yang et al. [7] to simultaneously discover a set of possibly overlapping biclusters. Initial biclusters are chosen randomly from the original data matrix.

Iteratively genes and/or conditions are added and/or deleted in order to achieve the best potential residue reduction. Bipartite graphs are also employed in [8], with a bicluster being defined as a subset of genes that jointly respond across a subset of conditions. The objective is to identify the maximum-weighted subgraph. Here a gene is considered to be responding under a condition if its expression level changes significantly, under that condition over the connecting edge, with respect to its normal level. This involves an exhaustive enumeration, with a restriction on the number of genes that can appear in the bicluster.

Other methods have been successfully employed in the Deterministic Biclustering with Frequent pattern mining algorithm (DBF) [9] to generate a set of good quality biclusters. Here concepts from the Data Mining practice are exploited. The changing tendency between two conditions is modeled as an item, with the genes corresponding to transactions. A frequent item-set with the supporting genes forms a bicluster. In the second phase, these are iteratively refined by adding more genes and/or conditions.

Genetic algorithms (GAs) have been employed by Mitra et al. [10] with local search strategy for identifying overlapped biclusters in gene expression data. In [11], a simulated annealing based biclustering algorithm has been proposed to provide improved performance over that of [2], escaping from local minima by means of a probabilistic acceptance of temporary worsening in fitness scores.

1.3 Outline of the paper

In this paper we introduce a new approach to the biclustering problem using the possibilistic clustering paradigm [12]. The proposed Possibilistic Biclustering algorithm (PBC) finds one bicluster at a time, assigning a membership to the bicluster for each gene and for each condition. The membership model is of the fuzzy possibilistic type [12].

The paper is organized as follows: in section 2 the possibilistic paradigm is illustrated; section 3 presents the possibilistic approach to biclustering, and section 4 reports on experimental results. Section 5 is devoted to conclusions.

2 Possibilistic Clustering Paradigm

The central clustering paradigm is implemented in several algorithms including C-Means [13], Self Organizing Map [14] Fuzzy C-Means [15], Deterministic Annealing [16], Alternating Cluster Estimation [17], and many others. Often, central clustering algorithms impose a *probabilistic constraint*, according to which the sum of the membership values of a point in all the clusters must be equal to one. This competitive constraint allows the unsupervised learning algorithms to find the barycenter of fuzzy clusters, but the obtained evaluations of membership to clusters are not interpretable as a *degree of typicality*, and moreover can give sensibility to outliers, as isolated outliers can hold high membership values to some clusters, thus distorting the position of centroids.

The possibilistic approach to clustering proposed by Keller and Krishnapuram [12], [18] assumes that the membership function of a data point in a *fuzzy* set (or cluster) is absolute, i.e. it is an evaluation of a degree of typicality not depending on the membership values of the same point in other clusters.

Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_r\}$ be a set of unlabeled data points, $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_s\}$ a set of cluster centers (or prototypes) and $U = [u_{pq}]$ the *fuzzy membership matrix*. In the Possibilistic C-Means (PCM) Algorithms the constraints on the elements of U are relaxed to:

$$u_{pq} \in [0, 1] \quad \forall p, q; \tag{7}$$

$$0 < \sum_{q=1}^{r} u_{pq} < r \quad \forall p; \tag{8}$$

$$\bigvee_{p} u_{pq} > 0 \quad \forall q. \tag{9}$$

Roughly speaking, these requirements simply imply that cluster cannot be empty and each pattern must be assigned to at least one cluster. This turns a standard fuzzy clustering procedure into a mode seeking algorithm [12].

In [18], the objective function contains two terms, the first one is the objective function of the CM [13], while the second is a penalty (regularization) term considering the entropy of clusters as well as their overall membership values:

$$J_m(U, Y) = \sum_{p=1}^{s} \sum_{q=1}^{r} u_{pq} E_{pq} + \sum_{p=1}^{s} \frac{1}{\beta_p} \sum_{q=1}^{r} (u_{pq} \log u_{pq} - u_{pq}), \tag{10}$$

where $E_{pq} = \|\mathbf{x}_q - \mathbf{y}_p\|^2$ is the squared Euclidean distance, and the parameter β_p (that we can term *scale*) depends on the average size of the p-th cluster, and must be assigned before the clustering procedure. Thanks to the regularizing term, points with a high degree of typicality have high u_{pq} values, and points not very representative have low u_{pq} values in all the clusters. Note that if we take $\beta_p \to \infty \quad \forall p$ (i.e., the second term of $J_m(U, Y)$ is omitted), we obtain a trivial solution of the minimization of the remaining cost function (i.e., $u_{pq} = 0 \quad \forall p, q$), as no probabilistic constraint is assumed.

The pair (U, Y) minimizes J_m, under the constraints 7-9 only if [18]:

$$u_{pq} = e^{-E_{pq}/\beta_p} \quad \forall p, q, \tag{11}$$

and

$$\mathbf{y}_p = \frac{\sum_{q=1}^{r} \mathbf{x}_q u_{pq}}{\sum_{q=1}^{r} u_{pq}} \quad \forall p. \tag{12}$$

Those conditions for minimizing the cost function $J_m(U, Y)$. Eq.s 11 and 12 can be interpreted as formulas for recalculating the membership functions and the cluster centers (Picard iteration technique), as shown, e.g., in [19].

A good initialization of centroids must be performed before applying PCM (using, e.g., Fuzzy C-Means [12], [18], or Capture Effect Neural Network [19]). The PCM works as a refinement algorithm, allowing us to interpret the membership to clusters as cluster typicality degree, moreover PCM shows a high outliers rejection capability as it makes their membership very low.

Note that the lack of probabilistic constraints makes the PCM approach equivalent to a set of s independent estimation problems [20]:

$$(u_{pq}, \mathbf{y}) = \arg \bigwedge_{u_{pq},\mathbf{y}} \left[\sum_{q=1}^{r} u_{pq} E_{pq} + \frac{1}{\beta_p} \sum_{q=1}^{r} (u_{pq} \log u_{pq} - u_{pq}) \right] \quad \forall p, \tag{13}$$

that can be solved independently one at a time through a Picard iteration of eq. 11 and eq. 12.

3 The possibilistic approach to biclustering

In this section we generalize the concept of biclustering in a fuzzy set theoretical approach. For each bicluster we assign two vectors of membership, one for the rows and one other for the columns, denoting them respectively **a** and **b**. In a crisp set framework row i and column j can either belong to the bicluster ($a_i = 1$ and $b_j = 1$) or not ($a_i = 0$ or $b_j = 0$). An element x_{ij} of X belongs to the bicluster if both $a_i = 1$ and $b_j = 1$, i.e., its membership u_{ij} to the bicluster is:

$$u_{ij} = \text{and}(a_i, b_j) \tag{14}$$

The cardinality of the bicluster is then defined as:

$$n = \sum_i \sum_j u_{ij} \tag{15}$$

A fuzzy formulation of the problem can help to better model the bicluster and also to improve the optimization process. In a fuzzy setting we allow membership u_{ij}, a_i and b_j to belong in the interval $[0, 1]$. The membership u_{ij} of a point to the bicluster can be obtained by an integration of row and column membership, for example by:

$$u_{ij} = a_i b_j \qquad \text{(product)} \tag{16}$$

or

$$u_{ij} = \frac{a_i + b_j}{2} \qquad \text{(average)} \tag{17}$$

The fuzzy cardinality of the bicluster is defined as the sum of the memberships u_{ij} for all i and j as in eq. 15. We can generalize eqs. 3 to 6 as follows:

$$d_{ij}^2 = \frac{\left(x_{ij} + x_{IJ} - x_{iJ} - x_{Ij}\right)^2}{n} \tag{18}$$

where:

$$x_{IJ} = \frac{\sum_i \sum_j u_{ij} x_{ij}}{\sum_i \sum_j u_{ij}} \tag{19}$$

$$x_{iJ} = \frac{\sum_j u_{ij} x_{ij}}{\sum_j u_{ij}} \tag{20}$$

$$x_{Ij} = \frac{\sum_i u_{ij} x_{ij}}{\sum_i u_{ij}} \tag{21}$$

$$G = \sum_i \sum_j u_{ij} d_{ij}^2 \tag{22}$$

Then we can tackle the problem of maximizing the bicluster cardinality n and minimizing the residual G using the fuzzy possibilistic paradigm. To this aim we make the following assumptions:

- we treat one bicluster at a time;
- the fuzzy memberships a_i and b_j are interpreted as typicality degrees of gene i and condition j with respect to the bicluster;
- we compute the membership u_{ij} using eq. 17.

All those requirements are fulfilled by minimizing the following functional J_{B} with respect to **a** and **b**:

$$J_{\mathrm{B}} = \sum_i \sum_j \left(\frac{a_i + b_j}{2} \right) d_{ij}^2 + \lambda \sum_i (a_i \ln(a_i) - a_i) + \mu \sum_j (b_j \ln(b_j) - b_j) \tag{23}$$

The parameters λ and μ control the size of the bicluster by penalizing to small values of the memberships. Their value can be estimated by simple statistics over the training set, and then hand-tuned to incorporate possible a-priori knowledge and to obtain the desired results.

Setting the derivatives of J_{B} with respect to the memberships a_i and b_j to zero:

$$\frac{\partial J}{\partial a_i} = \sum_j \frac{d_{ij}^2}{2} + \lambda \ln(a_i) = 0 \tag{24}$$

$$\frac{\partial J}{\partial b_j} = \sum_i \frac{d_{ij}^2}{2} + \mu \ln(b_j) = 0 \tag{25}$$

we obtain these solutions:

$$a_i = \exp\left(- \frac{\sum_j d_{ij}^2}{2\lambda} \right) \tag{26}$$

$$b_j = \exp\left(- \frac{\sum_i d_{ij}^2}{2\mu} \right) \tag{27}$$

As in the case of standard PCM those necessary conditions for the minimization of J_{B} together with the definition of d_{ij}^2 (eq. 18) can be used by an algorithm able to find a numerical solution for the optimization problem (Picard iteration). The algorithm, that we call Possibilistic Biclustering (PBC), is shown in table 1.

The parameter ε is a threshold controlling the convergence of the algorithm. The memberships initialization can be made randomly or using some a priori information about relevant genes and conditions. Moreover, the PBC algorithm can be used as a

Table 1. Possibilistic Biclustering (PBC) algorithm.

```
1. Initialize the memberships $\mathbf{a}$ and $\mathbf{b}$
2. Compute $d_{ij}^2$ $\forall i, j$ using eq. 18
3. Update $a_i$ $\forall i$ using eq. 26
4. Update $b_j$ $\forall j$ using eq. 27
5. if $||\mathbf{a}' - \mathbf{a}|| < \varepsilon$ and $||\mathbf{b}' - \mathbf{b}|| < \varepsilon$ then stop
6. else jump to step 2
```

refinement step for other algorithms using as initialization the results already obtained from them.

After convergence of the algorithm the memberships **a** and **b** can be defuzzified by comparing with a threshold (e.g. 0.5). In this way the results obtained with PBC can be compared with those of other techniques.

4 Results

4.1 Experimental validation

We applied our algorithm to the *Yeast* database which is a genomic database composed by 2884 genes and 17 conditions[4] [21] [22] [23]. We removed from the database all genes having missing expression levels for all the conditions, obtaining a set of 2879 genes.

We performed many runs varying the parameters λ and μ and considering a thresholding for the memberships **a** and **b** of 0.5 for the defuzzification. In figure 1 the effect of the choice of these two parameters on the size of the bicluster can be observed. Increasing them results in a larger bicluster.

In figure 1 each result corresponds to the average on 20 runs of the algorithm. Note that, even if the memberships are initialized randomly, starting from the same set of parameters, it is possible to achieve almost the same results. Thus PBC is slightly sensitive to initialization of memberships while strongly sensitive to parameters λ and μ. The parameter ε can be set considering the desired precision on the final memberships. Here it has been set to 10^{-2}.

In table 2 a set of obtained biclusters is shown with the achieved values of G. In particular it is very interesting the ability of PBC to find biclusters of a desired size just tuning the parameters λ and μ. A plot of a small and a large biclusters can be found in fig. 2.

The PBC algorithm has been written in C and R language [24], and run on a Pentium IV 1900 MHz personal computer with 512*Mbytes* of ram under a Linux operating system. The running time for each set of parameters was 7.5*s*, showing that the complexity of the algorithm depends only on the size of the data set.

[4] http://arep.med.harvard.edu/biclustering/yeast.matrix

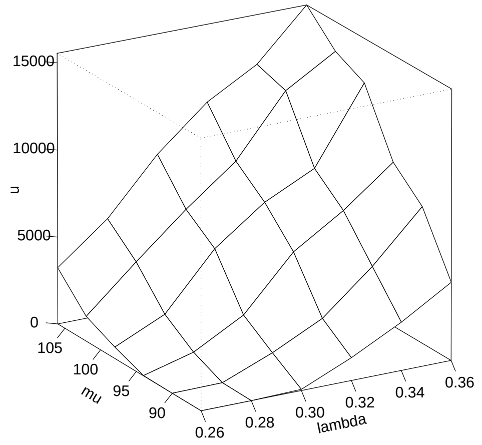

Fig. 1. Size of the biclusters vs. parameters λ and μ.

λ	μ	n_g	n_c	n	G
0.25	115	448	10	4480	56.07
0.19	200	457	16	7312	67.80
0.30	100	654	8	5232	82.20
0.32	100	840	9	7560	111.63
0.26	150	806	15	12090	130.79
0.31	120	989	13	12857	146.89
0.34	120	1177	13	15301	181.57
0.37	110	1309	13	17017	207.20
0.39	110	1422	13	18486	230.28
0.42	100	1500	13	19500	245.50
0.45	95	1622	12	19464	260.25
0.45	95	1629	13	21177	272.43
0.46	95	1681	13	21853	285.00
0.47	95	1737	13	22581	297.40
0.48	95	1797	13	23361	310.72

Table 2. Comparison of the biclusters obtained by our algorithms on yeast data. The G value, the number of genes n_g, the number of conditions n_c, the cardinality of the bicluster n are shown with respect to the parameters λ and μ.

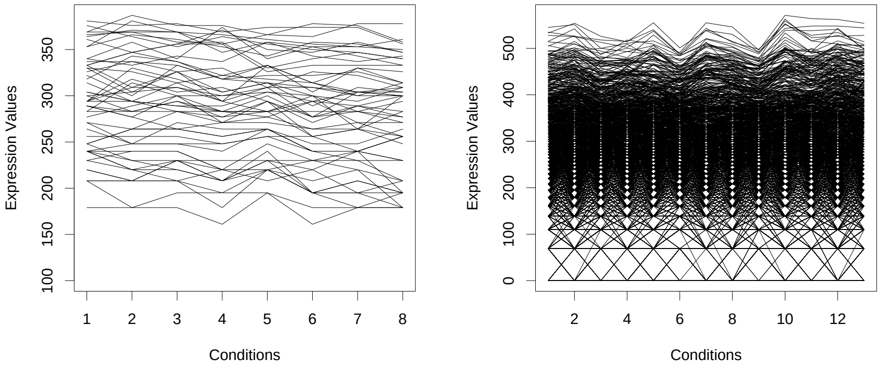

Fig. 2. Plot of a small and a large bicluster

4.2 Comparative study

Table 3 lists a comparison of results on *Yeast* data, involving performance of other, related biclustering algorithms with a $\delta = 300$ (δ is the maximum allowable residual for G). The deterministic DBF [9] discovers 100 biclusters, with half of these lying in the size range 2000 to 3000, and a maximum size of 4000. FLOC [7] uses a probabilistic approach to find biclusters of limited size, that is again dependent on the initial choice of random seeds. FLOC is able to locate large biclusters. However DBF generates a lower mean squared residue, which is indicative of increased similarity between genes in the biclusters. Both these methods report an improvement over the pioneering algorithm by Cheng et al. [2], considering mean squared residue as well as bicluster size.

Single-objective GA with local search has also been used [25], to generate considerably overlapped biclusters .

Method	avg. G	avg. n	avg. n_g	avg. n_c	Largest n
DBF [9]	115	1627	188	11	4000
FLOC [7]	188	1826	195	12.8	2000
Cheng-Church [2]	204	1577	167	12	4485
Single-objective GA [10]	52.9	571	191	5.13	1408
Multi-objective GA [10]	235	10302	1095	9.29	14828
Possibilistic Biclustering	297	22571	1736	13	22607

Table 3. Comparative study on *Yeast* data

The average results reported in table 3 concerning the Possibilistic Biclustering algorithm have been obtained involving 20 runs over the same set of parameters λ and μ.

The biclusters obtained where very similar, obtaining G close to $\delta = 300$ for all of them and the achieved bicluster size is on average very high. From table 3, we see that the Possibilistic Approach has better performances in finding large biclusters in comparison with others methods.

5 Conclusions

In this paper we proposed the PBC algorithm, a new approach to biclustering based on the possibilistic paradigm. The problem of minimizing the residual G and maximize the size n, has been tackled by optimizing a functional which takes into account these requirements. The proposed method allows to find one bicluster at a time of the desired size.

The results show the ability of the PBC algorithm to find biclusters with low residuals. The quality of the large biclusters obtained is better in comparison with other biclustering methods.

The method will be the subject of further study. In particular, several criteria for automatically selecting the parameters λ and μ can be proposed, and different ways to combine a_i and b_j into u_{ij} can be discussed. Moreover, a biological validation of the obtained results is under study.

Acknowledgment

Work funded by the Italian Ministry of Education, University and Research (2004 “Research Projects of Major National Interest”, code 2004062740).

References

1. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE Transactions on Computational Biology and Bioinformatics **1** (2004) 24–45
2. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, AAAI Press (2000) 93–103
3. Hartigan, J.A.: Direct clustering of a data matrix. Journal of American Statistical Association **67(337)** (1972) 123–129
4. Kung, S.Y., Mak, M.W., Tagkopoulos, I.: Multi-metric and multi-substructure biclustering analysis for gene expression data. Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conference (CSB’05) (2005)
5. Turner, H., Bailey, T., Krzanowski, W.: Improved biclustering of microarray data demonstrated through systematic performance tests. Computational Statistics and Data Analysis **48**(2) (2005) 235–254
6. Peeters, R.: The maximum edge biclique problem is NP-Complete. Discrete Applied Mathematics **131** (2003) 651–654
7. Yang, J., Wang, H., Wang, W., Yu, P.: Enhanced biclustering on expression data. In: Proceedings of the Third IEEE Symposium on BioInformatics and Bioengineering (BIBE’03). (2003) 1–7

8. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics **18** (2002) S136–S144
9. Zhang, Z., Teo, A., Ooi, B.C.a.: Mining deterministic biclusters in gene expression data. In: Proceedings of the Fourth IEEE Symposium on Bioinformatics and Bioengineering (BIBE'04). (2004) 283–292
10. Mitra, S., Banka, H.: Multi-objective evolutionary biclustering of gene expression data. to appear (2006)
11. Bryan, K., Cunningham, P., Bolshakova, N.: Biclustering of expression data using simulated annealing. In: 18th IEEE Symposium on Computer-Based Medical Systems (CBMS 2005). (2005) 383–388
12. Krishnapuram, R., Keller, J.M.: A possibilistic approach to clustering. Fuzzy Systems, IEEE Transactions on **1**(2) (1993) 98–110
13. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. Wiley (1973)
14. Kohonen, T.: Self-Organizing Maps. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2001)
15. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell, MA, USA (1981)
16. Rose, K., Gurewwitz, E., Fox, G.: A deterministic annealing approach to clustering. Pattern Recogn. Lett. **11**(9) (1990) 589–594
17. Runkler, T.A., Bezdek, J.C.: Alternating cluster estimation: a new tool for clustering and function approximation. Fuzzy Systems, IEEE Transactions on **7**(4) (1999) 377–393
18. Krishnapuram, R., Keller, J.M.: The possibilistic c-means algorithm: insights and recommendations. Fuzzy Systems, IEEE Transactions on **4**(3) (1996) 385–393
19. Masulli, F., Schenone, A.: A fuzzy clustering based segmentation system as support to diagnosis in medical imaging. Artificial Intelligence in Medicine **16**(2) (1999) 129–147
20. Nasraoui, O., Krishnapuram, R.: Crisp interpretations of fuzzy and possibilistic clustering algorithms. Volume 3., Aachen, Germany (1995) 1312–1318
21. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. Nature Genetics **22**(3) (1999)
22. Ball, C.A., Dolinski, K., Dwight, S.S., Harris, M.A., Tarver, L.I., Kasarskis, A., Scafe, C.R., Sherlock, G., Binkley, G., Jin, H., Kaloper, M., Orr, S.D., Schroeder, M., Weng, S., Zhu, Y., Botstein, D., Cherry, M.J.: Integrating functional genomic information into the saccharomyces genome database. Nucleic Acids Research **28**(1) (2000) 77–80
23. Aach, J., Rindone, W., Church, G.: Systematic management and analysis of yeast gene expression data (2000)
24. R Foundation for Statistical Computing Vienna, Austria: R: A language and environment for statistical computing. (2005)
25. Bleuler, S., Prelić, A., Zitzler, E.: An EA framework for biclustering of gene expression data. In: Congress on Evolutionary Computation (CEC-2004), Piscataway, NJ, IEEE (2004) 166–173