J Math Model Algor (2009) 8:175–202
DOI 10.1007/s10852-009-9102-8

Multi-Objective Biclustering: When Non-dominated Solutions are not Enough

Guilherme Palermo Coelho · Fabrício Olivetti de França · Fernando J. Von Zuben

Received: 15 October 2008 / Accepted: 19 December 2008 / Published online: 28 February 2009

Abstract The biclustering technique was developed to avoid some of the drawbacks presented by standard clustering techniques, such as their impossibility of finding correlating data under a subset of features, and, consequently, to allow the extraction of more accurate information from datasets. Given that biclustering requires the optimization of at least two conflicting objectives (residue and volume) and that multiple independent solutions are desirable as the outcome, a few multi-objective evolutionary algorithms for biclustering were proposed in the literature. However, these algorithms only focus their search in the generation of a global set of non-dominated biclusters, which may be insufficient for most of the problems as the coverage of the dataset can be compromised. In order to overcome such problem, a multi-objective artificial immune system capable of performing a multipopulation search, named MOM-aiNet, was proposed. In this work, the MOM-aiNet algorithm will be described in detail, and an extensive set of experimental comparisons will be performed, with the obtained results of MOM-aiNet being confronted with those produced by the popular CC algorithm, by another immune-inspired approach

---

G. P. Coelho (✉) · F. O. de França · F. J. Von Zuben
Laboratory of Bioinformatics and Bioinspired Computing (LBiC),
Department of Computer Engineering and Industrial Automation (DCA),
School of Electrical and Computer Engineering (FEEC),
University of Campinas (Unicamp),
Av. Albert Einstein – 400 – Building G2 – Room LE 14G,
13083–852, P. O. Box 6101, Campinas, São Paulo, Brazil
e-mail: gcoelho@dca.fee.unicamp.br

F. O. de França
e-mail: olivetti@dca.fee.unicamp.br

F. J. Von Zuben
e-mail: vonzuben@dca.fee.unicamp.br

for biclustering (BIC-aiNet), and by the multi-objective approach for biclustering proposed by Mitra & Banka.

Keywords Biclustering · Multi-objective optimization · Multipopulation search · Artificial immune systems

1 Introduction

Traditional clustering techniques [1] have become very popular in the data mining community nowadays, due to the increasing amount of information acquired in a wide range of fields, from business to science and technology. However, such techniques present two significant drawbacks: first, they are not capable of finding correlating data under only a subset of attributes (partial matching); and second, they cannot assign a given sample to more than one cluster [2]. These limitations have a large impact in some applications, such as the analysis of gene expression data [3], extraction of information in collaborative filtering databases [4] and text-mining [5], to name a few. In real-world applications, data samples are generally correlated with several distinct subsets of samples, and the correlation presented within each one of these subsets of samples can be supported by distinct subsets of attributes.

In order to reveal these multi-faceted correlations and to allow deeper inferences from the available data, the *biclustering* technique was proposed [6–8]. Considering that the datasets are structured as data matrices of objects (rows) and their corresponding attributes (columns), the biclustering technique is capable of finding several subsets of rows and columns from the data matrix, so that each subset will be composed of objects (rows) that share some similarities specifically on the selected attributes (columns). In this approach, a single object may take part in multiple biclusters, in association with a distinct subset of attributes at each one of them, thus allowing the extraction of additional information from the dataset.

The problem of finding several biclusters in a dataset can be seen as a combinatorial optimization problem, in which two criteria must be maximized: (*i*) the 'volume' of the bicluster, which is generally given by the number of rows times the number of columns; and (*ii*) the degree of similarity among the elements of the bicluster. The extracted biclusters should present a high similarity among elements in order to represent highly correlated samples, and should also present a high volume in order to be functional and to allow a deeper analysis of the dataset.

Therefore, the bicluster extraction problem is inherently a multi-objective optimization problem [9], as at least two criteria must be optimized. However, only a few works in the literature adopt multi-objective optimization concepts to biclustering. Mitra & Banka [10] proposed a multi-objective evolutionary algorithm that combines the NSGA–II (*Non-dominated Sorting Genetic Algorithm*—[11]) with a local search based on the original algorithm of Cheng & Church [8], and this algorithm was applied to gene expression problems [10, 12]. Divina & Aguilar-Ruiz [13] proposed the SMOB algorithm (*Sequential Multi-Objective Biclustering*), which sequentially calls a multi-objective evolutionary algorithm capable of generating a bicluster with maximum volume that attends a lower bound of similarity among its components. The SMOB algorithm was also applied to gene expression problems in [13] and [14], but in the latter work a different metric of similarity among the elements of the bicluster was also proposed. A third proposal in the literature that applies multi-objective optimization concepts to biclustering is the work of Maulik et al. [15], which is also based on the NSGA–II algorithm and employs fuzzy concepts in the codification of the individuals and the fuzzy k-medoids clustering algorithm [16] to perform the simultaneous clustering of both genes and conditions in gene expression problems.

One common aspect between the proposals of Mitra & Banka [10], Divina & Aguilar-Ruiz [13] and Maulik et al. [15] is that, like all the applications of multi-objective optimization, the search is aimed at the generation of a global set of non-dominated solutions [9]. However, in the biclustering problem, a good trade-off between volume and similarity of components (or any other criteria that is measured individually in each bicluster) is not the only desirable aspect of a good set of solutions. It is also very important that the final set of biclusters provides a good *coverage* of the dataset or, in other words, exhibits the maximum number of possible correlations between the samples in the dataset, even if some of these correlations are sub-optimal. A larger coverage of the dataset is very important to several applications that employ biclusters, as the higher the number of correlations that are highlighted, the greater the amount of information that can be inferred from the dataset.

In order to deal with the coverage issue, Coelho et al. [17] have proposed the MOM-aiNet algorithm (*Multi-Objective Multipopulation Artificial Immune Network*), which is an immune-inspired technique [18] that explores the concept of multiple populations to generate a set of biclusters that provide a high coverage of the dataset. As in [17] the authors have just introduced the MOM-aiNet algorithm and provided only preliminary experiments, in this work the algorithm will be fully presented, together with a few improvements, a more extensive set of experiments will be performed, and the results will be compared with those obtained by three proposals from the literature: the original CC algorithm, proposed by Cheng & Church [8]; BIC-aiNet (*Artificial Immune Network for Biclustering*—[19]), which is a single-objective immune-inspired algorithm; and the proposal of Mitra & Banka [10].

This paper is organized as follows. Sections 2, 3 and 4 present some general conceptual aspects of biclustering, multi-objective optimization and artificial immune systems, respectively. In Section 5 the three algorithms adopted in this work for comparison with MOM-aiNet will be briefly presented. The rationale of the exploration of multiple subpopulations to increase coverage and an illustration of the limitations of returning only the non-dominated set of biclusters in multi-objective biclustering will be given in Section 6, and the MOM-aiNet algorithm itself will be described in details in Section 7. In Section 8 the experimental methodology employed will be presented, and the obtained results will be given. Finally, the concluding remarks of the paper and further steps of the research will be presented in Section 9.

2 Biclustering

The term *biclustering* (also *co-clustering* or *two-mode clustering*) is referred to the process of finding subsets of rows and columns of a given data matrix [6–8], which may represent different kinds of numerical data, such as objects (rows) and their attributes (columns). Two examples of biclusters of a data matrix can be found in Fig. 1.

$$\underset{(a)}{\begin{bmatrix} 3 & 1 & 2 & 1 & 5 \\ 4 & 1 & 5 & 1 & 5 \\ 5 & 1 & 5 & 2 & 5 \\ 4 & 2 & 3 & 2 & 6 \\ 5 & 3 & 4 & 3 & 5 \end{bmatrix}} \qquad \underset{(b)}{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}} \qquad \underset{(c)}{\begin{bmatrix} 4 & 5 & 5 \\ 5 & 5 & 5 \\ 5 & 4 & 5 \end{bmatrix}}$$

Fig. 1 Two biclusters (**b** and **c**) extracted from the original matrix (**a**). The bicluster (**b**) was created with rows $\{1, 2\}$ and columns $\{2, 4\}$, and the bicluster (**c**) was created with rows $\{2, 3, 5\}$ and columns $\{1, 3, 5\}$. The similarity criterion here is the constant value of the elements

The biclustering approach covers a wide scope of different applications such as dimensionality reduction [20], information retrieval and text mining [5, 19, 21], electoral data analysis [6], collaborative filtering [22–24], and biological data analysis [10, 12–15, 20, 25].

Cheng & Church [8] were responsible for the popularization of the biclustering paradigm with their CC algorithm, which was the first biclustering algorithm applied to gene expression problems. However, several distinct approaches were also proposed in the literature, so that different taxonomies were suggested for these techniques [26]. The biclustering task can be classified into several categories, according to (*i*) the way the bicluster quality is measured; (*ii*) how the set of biclusters are built; and (*iii*) which structure of bicluster is adopted.

The classification based on the quality measure of a biclustering algorithm is related to the concept of similarity between the elements of the matrix. For instance, some algorithms search for constant value biclusters, some for constant columns or rows, and others for *coherency* in the values of the elements. In Fig. 2 some of the quality measures of biclustering algorithms are illustrated. In practical applications, the obtained biclusters will not follow the quality measure without some *deviation*, which can be interpreted as an error (*residue*) to be minimized at the same time that the volume of the biclusters is maximized.

The way that the set of biclusters is generated is associated with the amount of biclusters that is returned in each run of a given algorithm. Some methods find only one bicluster per run, while others are capable of simultaneously finding several biclusters. Besides, there are also nondeterministic and deterministic algorithms,

$$\underset{(a)}{\begin{bmatrix} 3 & 1 & 2 & 1 & 5 \\ 4 & 1 & 5 & 1 & 5 \\ 1 & 1 & 3 & 2 & 5 \\ 4 & 2 & 3 & 2 & 6 \\ 5 & 3 & 4 & 3 & 7 \end{bmatrix}} \qquad \underset{(b)}{\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}} \qquad \underset{(c)}{\begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{bmatrix}} \qquad \underset{(d)}{\begin{bmatrix} 1 & 5 \\ 1 & 5 \\ 1 & 5 \end{bmatrix}} \qquad \underset{(e)}{\begin{bmatrix} 3 & 2 & 1 & 5 \\ 4 & 3 & 2 & 6 \\ 5 & 4 & 3 & 7 \end{bmatrix}}$$

Fig. 2 A didactic example of four biclusters (**b**, **c**, **d** and **e**), each one obeying a specific optimization criterion, extracted from the original matrix (**a**). The bicluster (**b**) was created with rows $\{1, 2\}$ and columns $\{2, 4\}$, and is an example of a constant bicluster. The bicluster (**c**) was created with rows $\{1, 4, 5\}$ and columns $\{2, 4\}$, and is an example of a bicluster with constant rows. The bicluster (**d**) was created with rows $\{1, 2, 3\}$ and columns $\{2, 5\}$, and is an example of a bicluster with constant columns. The bicluster (**e**) was created with rows $\{1, 4, 5\}$ and columns $\{1, 3, 4, 5\}$, and is an example of a bicluster with coherent values being the non-deterministic ones capable of finding different solutions for the same problem at each execution, while the deterministic ones produce always the same solution.

The third possible classification of biclustering techniques, proposed by Madeira & Oliveira [26], is related to the structure of the biclusters adopted by each algorithm, which can be of different types: (*i*) those that cannot present overlap in neither columns nor rows of the matrix, which are known as exclusive columns and/or rows; (*ii*) arbitrarily positioned and possibly overlapping biclusters; and (*iii*) overlapping biclusters with hierarchical structure.

In this paper, the focus will be on possibly overlapping sets of arbitrarily positioned biclusters that follow a given coherency metric (more specifically, the metric proposed by Cheng & Church [8]), equivalent to bicluster (e) in Fig. 2. All the algorithms employed in this work are nondeterministic and capable of proposing multiple biclusters in a single run.

In this work, the coherence metric among the elements of a bicluster is the mean-squared residue, introduced by Cheng & Church [8]. This metric consists in the calculation of the additive coherence inside a bicluster by considering that each row (or column) of the bicluster presents a profile identical to (or very similar to) the one exhibited by other rows (or columns), except for a constant bias. Thus, each element of a perfectly coherent bicluster can be expressed by:

$$a_{ij} = \mu + \alpha_i + \beta_j, \tag{1}$$

where μ is the base value of the bicluster, α_i is the additive adjustment of line $i \in I$, β_j is the additive adjustment of column $j \in J$, I is the set of rows in the bicluster and J is the set of columns in the bicluster.

As the additive adjustments α_i and β_j are usually unknown, it is possible to rewrite Eq. 1 so that the theoretical element a_{ij} of the bicluster can be obtained from the mean value of elements in row i (a_{iJ}), the mean value of elements in column j (a_{Ij}) and the mean value of the elements in the bicluster (a_{IJ}). These mean values are given by:

$$a_{iJ} = \mu + \alpha_i + \overline{\beta}, \tag{2}$$

$$a_{Ij} = \mu + \overline{\alpha} + \beta_j, \tag{3}$$

$$a_{IJ} = \mu + \overline{\alpha} + \overline{\beta}, \tag{4}$$

where $\overline{\alpha}$ is the mean value of the row additive adjustment and $\overline{\beta}$ is the mean value of the column additive adjustment.

Isolating α_i, β_j and μ in Eqs. 2, 3 and 4 above, we obtain:

$$\alpha_i = a_{iJ} - \mu - \overline{\beta}, \tag{5}$$

$$\beta_j = a_{Ij} - \mu - \overline{\alpha}, \tag{6}$$

$$\mu = a_{IJ} - \overline{\alpha} - \overline{\beta}. \tag{7}$$

Substituting Eq. 7 in Eqs. 5 and 6:

$$\alpha_i = a_{iJ} - a_{IJ} + \overline{\alpha}, \tag{8}$$

$$\beta_j = a_{Ij} - a_{IJ} + \overline{\beta}. \tag{9}$$

And, finally, substituting Eqs. 7, 8 and 9 in Eq. 1 leads to:

$$a_{ij} = a_{Ij} + a_{iJ} - a_{IJ}. \tag{10}$$

Therefore, finding a coherent bicluster is basically the same thing as finding a bicluster that minimizes the error between the theoretical value given in Eq. 10 and the real value of an element of the matrix. So the mean-squared residue $H(I, J)$ of a bicluster becomes:

$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I} \sum_{j \in J} (a_{ij} - a_{Ij} - a_{iJ} + a_{IJ})^2, \tag{11}$$

where $|I|$ is the total number of rows of the bicluster, $|J|$ is the total number of columns of the bicluster, a_{ij} is the value in row i and column j, a_{Ij} is the mean value of column j, a_{iJ} is the mean value of row i, and a_{IJ} is the mean value considering all the elements of the bicluster.

Other important aspect of biclustering is the 'volume' of the biclusters, which is generally denoted in the literature by $|I| \times |J|$. In order to be functional and to allow a deeper analysis of the dataset, it is usually required that a bicluster presents a large volume (large number of rows and columns).

Therefore, the process of extracting biclusters from a given dataset can be basically seen as a multi-objective optimization problem, where the mean-squared residue should be minimized, in order to increase the degree of coherency among the elements of the bicluster, and the volume of such biclusters should be maximized, so that they can be functional and allow a deeper analysis of the dataset. It is important to notice that the minimization of the mean-squared residue and the maximization of the volume are two conflicting objectives, since larger biclusters tend to present higher residues.

3 Multi-objective Optimization

In single-objective optimization problems, the solution corresponds to one or more feasible points that lead to an extreme value of the objective function being optimized, i.e., if you have a minimization (maximization) problem, its solution will be the set of feasible points that lead to the lowest (highest) value of the objective function.

For a multi-objective optimization problem, this concept of solutions corresponding to extreme values of the objective functions cannot be directly applied, since the objectives involved can be (and generally are) in conflict. What generally happens in such cases is that there is a set of final solutions that correspond to different trade-offs between the objectives, i. e., each of these solutions indicates, for a given value of one of the objectives, what is the best value that can be obtained for the remaining ones. Therefore, an infinite set of solutions can exist for a given problem, which corresponds to the infinite number of existing trade-offs among the objectives. As will be seen, this set of solutions for a given multi-objective problem is known as the *Pareto front*.

Without loss of generality, the concepts and definitions that will be presented in this section will consider the definition of a multi-objective optimization problem (MOP) given in Formulation (12), where all objectives should be minimized.

Therefore, in this work a constrained M-objective ($M \geq 1$) minimization problem will be defined by:

$$\begin{array}{llll} \text{Minimize} & (f_1(\vec{x}),\ f_2(\vec{x}), \ldots .\, f_M(\vec{x})), & & \\ \text{Subject to} & g_j(\vec{x}) \geq 0, & j = 1, 2, \ldots, J, & \\ & h_k(\vec{x}) = 0, & k = 1, 2, \ldots, K, & \\ & x_i^L \leq x_i \leq x_i^U, & i = 1, 2, \ldots, n, & \\ & x_i \in \Re & i = 1, 2, \ldots, n. & \end{array} \tag{12}$$

where n is the number of variables (dimension of the problem), J is the number of inequality constraints, K is the number of equality constraints, x_i^L is the lower bound of variable i and x_i^U is the upper bound of variable i. This formulation corresponds to a real-valued optimization problem. However, the concepts that will be presented here can be directly extended to a multi-objective problem involving discrete variables.

For the problem given in Formulation 12, n-dimensional real-valued vectors that satisfy all constraints are called *feasible* solutions. The set of solutions for a multi-objective optimization problem contains only *non-dominated individuals*, and is known as the *Pareto front* of the problem. These concepts of *dominance* and *Pareto front* will be defined below.

A given vector $\vec{u} = (u_1, \ldots, u_M)$ is said to dominate a vector $\vec{v} = (v_1, \ldots, v_M)$ (denoted by $\vec{u} \preceq \vec{v}$) if and only if all M components of $\vec{u}$ are better than or equal to the corresponding components of $\vec{v}$, and there is at least one component of $\vec{u}$ that is strictly better than the corresponding component of $\vec{v}$.

Therefore, given two feasible solutions $\vec{x}$ and $\vec{y}$ of the problem in Formulation 12, $\vec{x}$ is said to dominate $\vec{y}$ ($\vec{x} \preceq \vec{y}$) if and only if $\forall m \in \{1, \ldots, M\}$, $f_m(\vec{x}) \leq f_m(\vec{y})$ and $\exists i \in \{1, \ldots, M\} : f_i(\vec{x}) < f_i(\vec{y})$.

As mentioned before, the notion of a solution in multi-objective optimization problems is associated with the concept of *Pareto Front*. This notion of optimality was originally introduced by Edgeworth [27] and later generalized by Pareto [28], and it states that a given solution $\vec{x}^*$ belongs to the *Pareto front* (or is *Pareto optimal*) if there is no other feasible solution $\vec{x}$ capable of reducing the value of one objective without simultaneously increasing at least one of the others.

Therefore, the *Pareto optimal set* (the solution of the multi-objective problem) is given by the set of solutions that are not dominated by any other feasible solution in the domain of the problem, and the corresponding *Pareto front* is the set obtained by the application of the objective functions to each solution in the *Pareto optimal set*. Examples of *Pareto fronts* can be found in Fig. 3.

Given these two main concepts, it is important to notice that, differently from a single-objective problem where the main goal is to obtain the global optimum (or the multiple optimal solutions, if the problem is multimodal), to solve a multi-objective problem, two goals should be pursued: (*i*) find a finite set of solutions in the optimal Pareto front; and (*ii*) guarantee that these solutions present a high degree of diversity, in order to properly cover all the Pareto front. This proper coverage is generally associated with a uniform sampling of the Pareto front, and is necessary to give the user a set of trade-offs as complete as possible, without previously assigning relative weights to any objective of the problem.

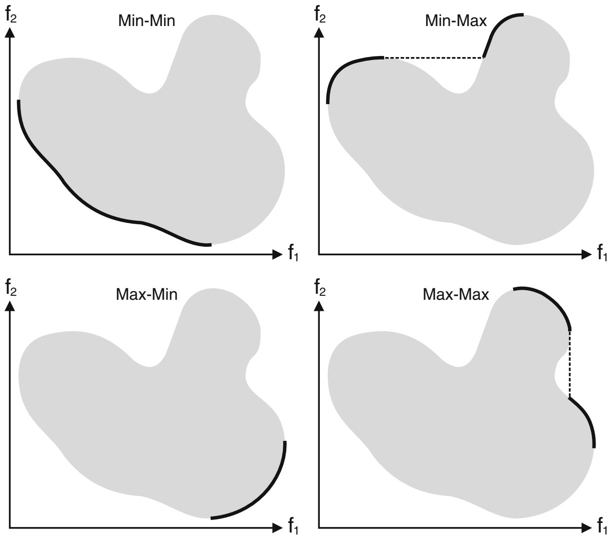


Fig. 3 Examples of Pareto fronts (*solid lines*) for two-objective optimization problems, where objectives f_1 and f_2 are being both minimized (Min–Min), both maximized (Max–Max), f_1 minimized and f_2 maximized (Min-Max) and f_1 maximized and f_2 minimized (Max–Min). The *gray area* corresponds to the feasible region of the problem, and the *dashed lines* (in the Min–Max and Max–Max cases) were drawn to emphasize that the concave regions are NOT in the Pareto front

It can be said that these two goals are, in some sense, orthogonal, as the satisfaction of one of them does not necessarily mean the satisfaction of the other. Therefore, multi-objective optimization algorithms must incorporate implicit and/or explicit methods to emphasize both the convergence of the optimal Pareto front and the diversity of the final set of solutions [9].

4 Artificial Immune Systems

The Artificial Immune System (AIS) paradigm was originated from attempts to model and apply immunological principles to problem solving in a wide range of areas such as optimization, data mining, computer security and robotics [18]. The algorithms developed based on this paradigm present three advantages over other population-based strategies: (*i*) they are inherently able to maintain population diversity (as modules with some resemblance with niching and fitness sharing are intrinsic parts of them); (*ii*) the size of the population at each generation is automatically defined according to the demand of the application; and (*iii*) local optimal solutions are simultaneously preserved once located.

The natural immune system, which is the inspiration for AISs, can be considered as one of the most important components of superior living organisms. Its permanent cycle of recognition and combat against pathogens (infectious foreign elements) has the goal of keeping the organism healthy. During this recognition cycle, the molecular patterns expressed in those invading pathogens (or *antigens*) are responsible for triggering the immune response when properly recognized by the immune cells.

The immune-inspired algorithms for biclustering that will be discussed in this work are mainly based on the reproduction of the *Clonal Selection* and *Affinity Maturation* principles [29], together with the *Immune Network Theory* [30].

The clonal selection theory is generally adopted to explain how the immune system responds to the attack of antigens. According to this theory, when an antigen invades the organism, some specific immune cells start proliferating to face the infection. This immune response results in some cells capable of secreting antibodies at high rates, and in other ones that become memory cells. During the proliferation of these cells, some individuals suffer a controlled mutation that, together with the selective pressure of the antigen, may allow the selection of immune cells that presents higher affinity with the infectious antigen (higher recognition and, thus, response capability). These processes of recognition, reproduction and maturation (mutation plus selection) are continuously repeated throughout life and they ensure that our organisms become increasingly better at recognizing the antigens.

Although the clonal selection theory explains how the immune system responds to antigens, it does not explain well how the immune system reacts to itself. A well-known proposal that tries to explain the autonomous dynamics of the immune system is the immune network theory, proposed by Jerne [30]. According to this theory, the immune cells are not only capable of recognizing foreign antigens, but they also contain some endogenous antigens that allow them to be recognized by other immune cells. When an immune cell is recognized by another immune cell, the recognizing cell is stimulated and the recognized cell is suppressed. The meanings of stimulation and suppression in AIS will depend on how the theory is interpreted in each algorithm. In the particular case of this work, both immune-inspired algorithms consider only suppression, which corresponds to the elimination of some of the self-recognizing cells. Although Jerne's theory proposes an explanation to the immune system reactions to itself, it has been discredited in the natural immune system community as no empirical evidences of such theory was found. However, the immune network theory is computationally interesting as it provides mechanisms to stimulate diversity among solutions in population-based algorithms, and so it still inspires the development of new approaches nowadays.

The concepts of clonal selection and immune network theories have been by the first time explored together to produce an immune-inspired algorithm for multi-modal optimization in [31].

5 Algorithms for Biclustering

In this section, the three algorithms that will be employed here, in the comparisons with MOM-aiNet, will be presented and described in details. The choice of these algorithms were made according to their characteristics and similarities with the algorithm proposed in this work: CC was chosen as it is one of the most popular biclustering algorithms nowadays; the algorithm of Mitra & Banka [10] is one of the few multi-objective proposals for biclustering in the literature; and the BIC-aiNet was selected because, as MOM-aiNet, it is an immune-inspired approach based on the original aiNet algorithm, proposed by de Castro & Von Zuben [32]. These algorithms will be described in the following subsections.

5.1 CC – Cheng & Church Algorithm

The CC algorithm [8] was the first biclustering algorithm applied to gene expression problems, and it is based on the concept of δ-*biclusters*. A δ-bicluster is basically a bicluster that presents a mean-squared residue (see Eq. 11) smaller than a predefined maximum acceptable value δ. Therefore, the goal of CC is to obtain K δ-biclusters for a given dataset.

CC is a constructive heuristic that starts with a single bicluster, representing the whole dataset, and iteratively removes rows and columns of this bicluster until the residue is equal to or less than δ. After that, it starts to insert rows and columns (that are not in the bicluster yet) sequentially, until the insertion of any other row or column increases the residue to a value above δ. After the first bicluster is constructed, the rows and columns already present in this bicluster are replaced by random values in the original dataset, and the whole process is restarted until a predefined number K of biclusters is created.

As CC masks the previously generated biclusters in the original dataset, it may seem that this algorithm is not capable of finding overlapping biclusters. But this is not completely true, as the row/column insertion is made considering the original values of the dataset. However, although CC is capable of generating biclusters with a certain degree of overlap, highly overlapping biclusters are not likely to be found.

5.2 The Algorithm of Mitra & Banka

The algorithm of Mitra & Banka [10] is basically an adaptation of the NSGA-II algorithm [11] to generate a set of biclusters, with the addition of a local search based on the node insertion and deletion mechanisms proposed by Cheng & Church [8] in the CC technique.

This technique employs a binary encoding, where each individual in the population is a binary vector with size equal to the number of rows plus the number of columns of the dataset. If a given position presents value 1, it indicates that the associated row or column is present in the bicluster.

The local search of Mitra & Banka's proposal basically calculates the contribution of each row and column of the bicluster ('nodes') to its residue, and removes those nodes that present a contribution higher than a predefined threshold. After that, a fine tuning is made, where new nodes with low contribution to the residue are inserted, in order to increase the volume of the bicluster. This node removal and insertion steps are greedy procedures, similar to those proposed by Cheng & Church [8].

One distinct aspect of the algorithm of Mitra & Banka is the way the mean-squared residue is optimized. In this approach, the algorithm maximizes the mean-squared residue as long as it is smaller than the upper bound δ.

The algorithm starts with the random initialization of the population and the application of a local search to this initial population. Then, the main loop is repeated for a predefined number of iterations, in which the population is ranked according to the dominance criterion (see Section 3), a *crowding tournament* selection is performed, the selected individuals suffer crossover and mutation, the offspring and parent population are mixed together and the best individuals are selected to remain in the next generation.

This dominance-based ranking procedure basically consists in dividing the individuals into categories, being set to the first category those that are not dominated by any other individual in the population, to the second those that are dominated only by the individuals in the first category, and so on.

The *crowding tournament* selection is based both on the rank of each individual and on their *crowding distance*, which evaluates the spread degree of each solution [11]. When two individuals are compared, first their rank is evaluated and, if both of them are on the same rank (which means that both present the same quality with respect to dominance), the crowding distance criterion is considered, and the individual that is farther from its neighbors (in the objective space) is chosen.

5.3 BIC-aiNet—Artificial Immune Network for Biclustering

The BIC-aiNet algorithm (*Artificial Immune Network for Biclustering* [19]), is an extension of the aiNet algorithm proposed by de Castro & Von Zuben [32] to solve traditional clustering problems. Both algorithms are based on the Clonal Selection and Immune Network Theories (see Section 4), so they are basically constituted by sequences of cloning, mutation, selection and suppression steps.

In BIC-aiNet, the individuals (biclusters) in the population are coded in two ordered integer vectors: one of them represents a list of the index of the rows of the original data matrix that form the bicluster, and the other represents a list of the index of columns in the bicluster.

The algorithm begins by generating a random population of biclusters, formed by just one row and one column. After the initialization, the algorithm enters its main loop, where the population is cloned and the generated clones suffer a process of *mutation*. The mutation operator consists in the application of one of three different possible actions, chosen randomly with equal probability:

- **Insert one row:** randomly inserts one row that is not included in the bicluster yet.
- **Insert one column:** randomly inserts one column that is not included in the bicluster yet.
- **Remove one row or one column:** randomly chooses one element of the bicluster to be removed.

After the cloning and mutation steps, the *affinity* (or *fitness*, using an evolutionary computation notation) of each mutated clone is calculated, and the fittest individual is selected to replace the original cell (if it presents a higher quality).

As BIC-aiNet does not deal with an explicit set of antigens, the affinity measure of each cell corresponds to an evaluation of the quality of each individual within the context of the problem being solved (which is also a valid approach in artificial immune systems, as can be seen in [18]). This affinity metric is calculated by Eq. 13, which tries to balance the residue and the volume of the bicluster:

$$Aff_k = \frac{H(I_k, J_k)}{\delta} + \frac{w_r \times \delta}{|I_k|} + \frac{w_c \times \delta}{|J_k|}, \tag{13}$$

where $H(I_k, J_k)$ is the mean-squared residue of bicluster k (see Eq. 11), I_k and J_k are the sets of rows and columns of bicluster k, δ is the residue threshold, w_r and w_c are, respectively, the degree of relevance of the number of rows and columns of a bicluster and $|I_k|$ and $|J_k|$ are the number of rows and columns of bicluster k.

As the BIC-aiNet algorithm is a single-objective optimization algorithm, it tries to minimize the mean-squared residue and maximize the volume of the biclusters with the minimization of a single criteria: the affinity measure given in Eq. 13. Therefore, the algorithm tries to control the maximum acceptable residue with parameter δ, and stimulates the generation of biclusters with large numbers of rows and columns with the second and third terms of Eq. 13.

As the BIC-aiNet algorithm is also based on the immune network theory, the individuals in the population are also capable of recognizing each other and, if the affinity among them is high, this indicates the existence of self-reactive cells that must be suppressed. This suppression step is performed periodically (not in every iteration), and causes a fluctuation in the population size, helping the algorithm to maintain diversity and to work with just the most meaningful biclusters.

The suppression mechanism of BIC-aiNet is rather simple: the affinity between two cells is given by the degree of overlapping between the two individuals (the intersection of lines and rows in both biclusters). If the overlapping between biclusters A and B exceeds a given threshold, the fittest bicluster survives and the other one is eliminated from the population. This suppression procedure is followed by an insertion step that generates new cells in the same way as the initial population is created, but giving preference to pairs of rows and columns that do not belong to any bicluster yet.

It is important to highlight that these two last procedures were conceived to give to the algorithm the capability of self-adapting the population size with a proper number of biclusters (for the given suppression threshold) to suitably represent the dataset.

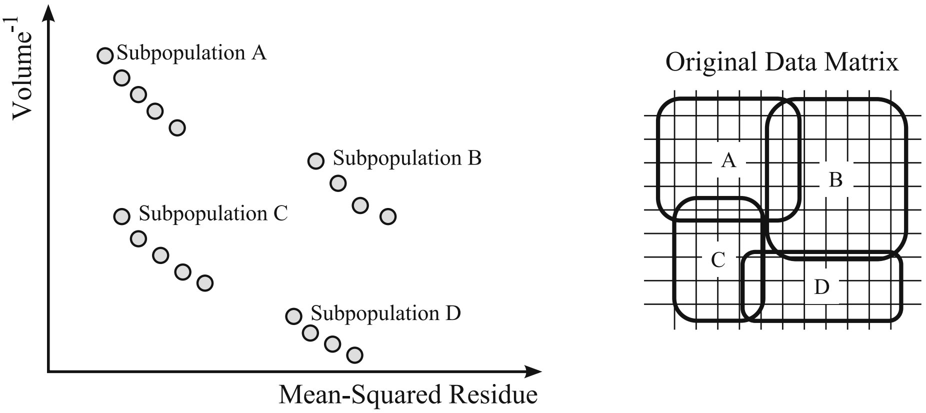
6 When Non-dominated Solutions are not Enough in Multi-Objective Biclustering

In this work, with the addition of the multi-objective optimization concepts, the MOM-aiNet algorithm will explore a multipopulation aspect inherent of the aiNet-based algorithms, by keeping in the population not only the best individual of each subpopulation, but several locally (within their subpopulations) non-dominated ones.

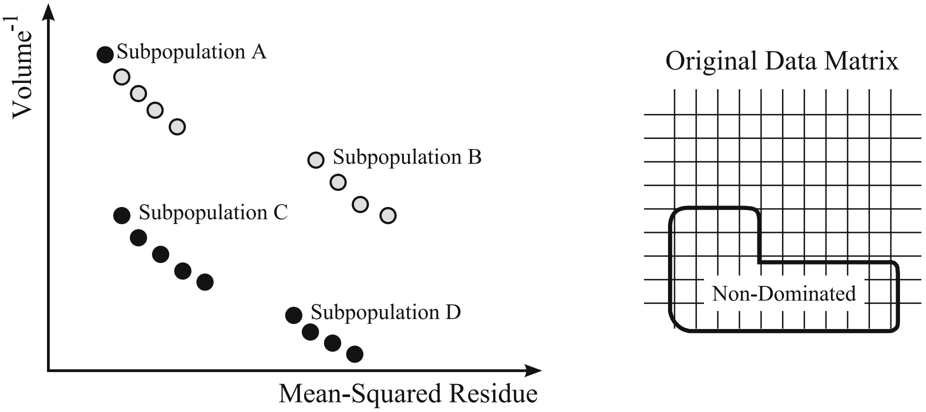
As each of these subpopulations will be stimulated to explore distinct regions of the search space (thus different regions of the data matrix or different correlations in the dataset), it will be possible to consider the individuals in each subpopulation as the non-dominated front found by the algorithm for the region of the dataset explored by such subpopulation (see Fig. 4a).

Also, in the end of its execution, the MOM-aiNet algorithm is configured to return not only the final set of non-dominated individuals, but all the non-dominated individuals within each sub-population. The reasons for this are twofold: (*i*) the coverage of the dataset provided by the full final set of "locally" non-dominated biclusters may be much higher than the one provided by the set constituted only by the "global" non-dominated set of solutions (see Fig. 4b); and (*ii*) sub-optimal biclusters may represent significant correlations of the dataset, which could be very useful in the post analysis. An example of this second motivation is given in Fig. 5.

In Fig. 5, it is possible to find examples of three biclusters (Fig. 5b, c and d) extracted from a data matrix (Fig. 5a). Calculating the volume and the mean-squared residue (Eq. 11) of these biclusters, we can see that bicluster (b) dominates bicluster (d). However, biclusters (b) and (c) are less informative than bicluster (d) as, in this example, they present constant rows and highlight much more trivial correlations of the data, while bicluster (d) represents a more complex correlation,



(a) Pictorial representation of the region of the data matrix covered by each subpopulation of the algorithm. In practice, the regions covered by the subpopulations may not be contiguous as in this example.



(b) Pictorial representation of the region of the data matrix covered by the global set of non-dominated individuals generated by the algorithm (blackdots). In practice, this region may not be contiguous as in this example.

Fig. 4 Pictorial representations of the coverage of the original data matrix by each subpopulation (**a**) and by the globally non-dominated individuals (**b**)

presenting a more elaborate structure, with distinct additive adjustment terms (see Section 2). Therefore, if only the non-dominated biclusters of these three examples were returned, bicluster (d) would have been eliminated and the final set of biclusters would correspond only to trivial correlations, what would certainly have a negative impact in the post analysis process. Also, if bicluster (d) is kept in the final results, a higher coverage of the original data matrix would also be accomplished.

In the literature, some proposals like the SMOB algorithm [13] also maximize the row variance of the biclusters, in order to deal with this specific issue of generation of constant row biclusters. However, the inclusion of another criteria in the optimization task certainly increases the overall computational cost, and does not solve the coverage issue.

$$\begin{bmatrix} 1 & 7 & 1 & - & 1 & - & - & 7 & 7 & - \\ - & - & - & - & - & - & - & - & - & - \\ 1 & 8 & 1 & & 1 & & 10 & 8 & 8 & 12 \\ - & 9 & - & - & - & - & 13 & - & - & 15 \\ 1 & - & 1 & - & 2 & - & - & - & - & - \\ - & 10 & - & - & - & - & 15 & - & - & 16 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad \begin{bmatrix} 7 & 7 & 7 \\ 8 & 8 & 8 \end{bmatrix} \quad \begin{bmatrix} 8 & 10 & 12 \\ 9 & 13 & 15 \\ 10 & 15 & 16 \end{bmatrix}$$

(a) Original Data Matrix (b) (c) (d)

Fig. 5 Example of sub-optimal bicluster with significant meaning. Bicluster (**b**) was created with rows {1, 3, 5} and columns {1, 3, 5} of the original data matrix (**a**). Bicluster (**c**) was created with rows {1, 3} and columns {2, 8, 9} and bicluster (**d**) with rows {3, 4, 6} and columns {2, 7, 10}. The mean-squared residue and volume are, respectively, 0.05 and 9 for bicluster (**b**), 0.00 and 6 for bicluster (**c**) and 0.29 and 9 for bicluster (**d**). It is possible to see that bicluster (**d**), although dominated by bicluster (**b**), is meaningful. The non-relevant elements of the original data matrix (**a**) were omitted for the sake of simplicity

7 MOM-aiNet – A Multi-Objective Multipopulation Artificial Immune Network for Biclustering

As the BIC-aiNet algorithm, the MOM-aiNet (*Multi-Objective Multipopulation Artificial Immune Network*) is also based on de Castro & Von Zuben's aiNet. However, as mentioned before, MOM-aiNet introduces some multi-objective optimization aspects to the original algorithm, to address the multi-objective aspect of the biclustering problem, and promotes a better exploration of the multipopulation concept already present in aiNet, but not explored so far.

The aiNet algorithm evolves multiple cells (individuals) in parallel and, associated with each of these cells, there is a population of mutated clones. Each of these subpopulations are stimulated to converge to distinct promising regions of the search space (generally to the closest local optimum). However, at each iteration of aiNet, only the best individual of each subpopulation is maintained to the next generation. In MOM-aiNet, this multipopulation aspect is further explored, by keeping in the population not only the best individual of each subpopulation, but several locally non-dominated individuals (see Section 6).

Also, in the end of its execution, the MOM-aiNet algorithm returns not only the final set of non-dominated individuals, but all the non-dominated individuals within each sub-population. With this approach, the final set of solutions may provide a higher coverage of the data matrix and may also contain sub-optimal biclusters (according to the mean-squared residue and volume criteria) that still represent important correlations of the dataset that were not identified by the "optimal" set of solutions.

Another characteristic of the MOM-aiNet algorithm is that constraints can be easily added to the algorithm, as the comparison between individuals in the population is made according to the dominance criterion (see Section 3). In order to add a given constraint to the biclustering problem as, for example, an upper bound δ to the mean-squared residue (which was adopted in this work), it is only necessary to consider all the unfeasible solutions as dominated by every feasible solution of the problem. By doing that, the unfeasible solutions will be naturally eliminated from the algorithm's population and the generation of feasible solutions will be stimulated.

The aiNet paradigm have already been successfully adapted to constrained multi-objective optimization with the omni-aiNet algorithm [33]. But the approach presented in this paper is significantly different from the one in [33], as omni-aiNet, besides presenting additional attributes, is specialized in sampling the Pareto front uniformly and with high precision, while the MOM-aiNet algorithm makes a rougher approximation of the Pareto front. Besides, omni-aiNet (as most multi-objective optimization algorithms) generates a single set of non-dominated solutions, differently from MOM-aiNet.

The main modules of the proposed algorithm are depicted in Algorithm 1 and explained in the sequence.

Algorithm 1 The MOM-aiNet algorithm.

```
generate_initial_population();
while the maximum number of generations is not achieved do
    for each subpopulation $i$ do
        $clones$ = clone($subpopulation_i$);
        $clones$ = mutate($clones$);
        $subpopulation_i$ = select_nondominated($clones \cup subpopulation_i$);
    end for
    network_suppression();
    insert_new_individuals();
end while
```

The algorithm starts with the generation of n subpopulations of one bicluster each, generated by randomly choosing one row and one column of the dataset. If the dataset is sparse, the algorithm must choose only among the non-null values. Inside the main loop, for each subpopulation n_{clones} clones of all the individuals are generated. In the preliminary proposal of this algorithm in [17], half of the clones were copied from the bicluster with the smallest volume (generally the smallest mean-squared residue) in the subpopulation, and the other half from the bicluster with the highest volume. However, it was observed that cloning all the individuals in the subpopulation guides to a better coverage of the search space.

Each clone then suffers a mutation process, which consists of one of three possible actions, chosen randomly with the same probability (like the BIC-aiNet algorithm—see Section 5.3): insert a row, insert a column, remove a row or column. Each action randomly selects one element to be inserted/removed. After the mutation step is performed on each clone, all the non-dominated biclusters of this subpopulation, considering the mutated clones and original individuals, are selected to generate the new subpopulation for the next iteration. If the number of non-dominated elements exceed n_{clones}, a crowding-distance-based [11] suppression is performed in order to maintain a small and locally diverse subpopulation.

One important characteristic of the MOM-aiNet algorithm is the incremental nature of the mutation step, where only one row or one column can be inserted in/removed from each bicluster at a time. Therefore the biclusters are likely to suffer an incremental growth toward both ends of the non-dominated front (low and high volume), thus tending to be equally distributed.

Two constraints can be imposed on the biclustering problem and controlled on these steps of the algorithm. The first one is associated with the mean-squared residue that can be limited to a specified value (δ), being every bicluster with a residue higher than δ considered dominated by any other feasible bicluster. And the other is the occupancy rate (for sparse matrices), that measures the proportion of non-null values in the bicluster. When an insertion action is chosen by the mutation process, the set of candidate rows/columns to be inserted is reduced to those that make the bicluster occupancy rate stay above a threshold γ.

After the cloning and mutation processes, from time to time a suppression operation is performed, so that the largest biclusters of each subpopulation are compared, based on the degree of overlapping. When a pair of biclusters has a degree of overlapping higher than a given threshold σ, the two subpopulations are combined and the non-dominance selection is performed, creating a single subpopulation. Only the largest bicluster of each population is taken for comparison, because they tend to be more representative. So, two subpopulations will only be merged when they start to express the same correlations among rows and columns of the data matrix. In other words, they will be merged when both subpopulations start to converge to the same region of the data matrix.

Finally, besides the suppression of similar subpopulations, it is also performed an insertion of new randomly generated subpopulations, in order to increase diversity and promote the exploration capability of the algorithm. This random insertion is performed in the same way as the initial subpopulations are generated, but with the difference that first it is chosen a pair of row and column that is not contained in any existing bicluster.

8 Experimental Results

In order to evaluate the quality of the biclusters generated by MOM-aiNet, when compared to those obtained by the other three algorithms, we have chosen two well-known gene expression datasets and one collaborative filtering experimental dataset, which presents a high degree of sparsity.

The first gene expression dataset is the *Saccharomyces cerevisiae* cell cycle expression data from Cho et al. [34]. This dataset contains $2,884$ genes under 17 conditions, and it was adapted in this work according to Cheng & Church [8], with the null values being replaced by random numbers in the range $[0, 600]$. This dataset is popularly known as *Yeast*.

The second dataset is the *Human B-Cell Lymphoma* expression data, first employed in [35]. This problem holds $4,026$ genes under 96 conditions, and the expression levels were treated in the same way as the Yeast dataset, with missing values being replaced by random numbers in $[-750, 650]$. Both the Yeast and Human datasets can be found at http://arep.med.harvard.edu/biclustering.

The third dataset is called *Movielens* (available at http://www.grouplens.org/node/73), and it was built with ratings in the range of $[1, 5]$ given by 943 users to different subsets of a database of $1,650$ movies, resulting in a highly sparse dataset with $80,000$ ratings (approximately 6.4% of occupancy). This dataset was first intended to be used with collaborative filtering algorithms, where the objective is to extract information from the dataset to infer a list of movie recommendations to a given user. This dataset is challeging to biclustering algorithms due to its lack of information (high sparsity) so, in this case, good quality biclusters must not only present low mean-squared residue and high volume but also a high rate of occupancy (non-null values).

For each experiment, the mean values taken in five independent runs of each algorithm were calculated for the average mean-squared residue, volume, coverage and, for the sparse dataset, occupancy of the final set of biclusters. Box plots for these criteria were also generated to allow better conclusions to be drawn from the results. Additionally, the quantitative comparisons between the algorithms were verified with the Student's t-test [36] with p-value threshold equal to 10^{-3}.

It is important to notice that the goal of the analysis that will be made here is not purely quantitative, as we will also focus on the quality of the biclusters generated by each algorithm. It is not possible to establish a single numeric criteria to determine which bicluster should be picked as the best one in the population, since multiple objectives are being considered in this approach and the quality of the biclusters should be seen as an interconnection of such different criteria, which is application-dependent.

The parameters adopted for each dataset were chosen experimentally for each algorithm, in order to lead to the best possible set of biclusters. Most of the parameters adopted were the same for all the three datasets being only a few of them problem-specific. The BIC-aiNet and MOM-aiNet algorithms were run for $2,000$ iterations, with a maximum of 6 individuals in each sub-population of the MOM-aiNet. For Mitra & Banka's algorithm, the number of iterations adopted was equal to the number of biclusters required, the offspring size was set to 50 and the probability of crossover and mutation were 75% and 3%, respectively. For CC, the only fixed parameter was α which held a value of 1.2.

For the Yeast dataset, each algorithm was set to generate 300 biclusters with δ (upper boundary of the mean-squared residue) set to 300 for MOM-aiNet, CC and Mitra & Banka, and to 100 for BIC-aiNet. As it was seen on Section 5.3, the BIC-aiNet algorithm tries to control the maximum mean-squared residue in the fitness equation (see Eq. 13), so the meaning of parameter δ is different in this algorithm. However, δ was properly set to 100 in order to lead BIC-aiNet to generate a set of biclusters that attend the same upper bound of the residue (300). Also, the α parameter of the Mitra & Banka's algorithm (which is associated with the rate of node deletion in the local search) was set to 1.8. The row (w_r) and column (w_c) degree of relevance of the BIC-aiNet algorithm were set to 2 and 3, respectively.

For the Human dataset, the algorithms were adjusted to generate 200 biclusters with maximum mean-squared residue (δ) of $1,200$ (in BIC-aiNet, δ was set to 200), and α was set to 1.2 in Mitra & Banka's technique. As for the Yeast dataset, the row (w_r) and column (w_c) degree of relevance of the BIC-aiNet algorithm were set to 2 and 3, respectively.

Finally, for the Movielens dataset, all the algorithms were adjusted to generate 100 biclusters with a maximum residue δ of 2 (including BIC-aiNet) and α was set to 1.2 in Mitra & Banka's algorithm, and the occupancy threshold γ of the MOM-aiNet algorithm was set to 0.6. For this problem, the row (w_r) and column (w_c) degree of relevance of the BIC-aiNet algorithm were set to 3 and 2, respectively.

As the number of subpopulations and of individuals in each of these subpopulations of biclusters can vary during the execution of MOM-aiNet, which generally leads this algorithm to return a much larger number of biclusters at the end of its execution than the other three techniques, in this work only the largest bicluster of each sub-population of the MOM-aiNet algorithm was considered in the evaluation of the results that will be presented in the following sub-sections.

8.1 Yeast Microarray Dataset

In order to correctly analyze the quality of the generated biclusters, it is first necessary to establish what are the expected characteristics desired from the set of biclusters. Sometimes it is required an equilibrium among minimal mean-squared residue, maximum volume and maximum coverage. In other situations, biclusters with maximum volume and coverage may be needed, being only required that they attend a maximum mean-squared residue constraint. A third possible situation is that the generated biclusters present a minimum degree of overlap, so that they can represent the correlations in the dataset with a minimum amount of redundancy. The ideal set of requirements to be imposed on the bicluster generation procedure is highly problem-specific. For this reason, in the analysis that will be performed here, we will try to highlight the main characteristics of each algorithm according to several criteria, to allow users who intent to apply any of these techniques to a specific problem to identify which proposal presents the characteristics required by his/her situation.

From Table 1 it is possible to see that both the MOM-aiNet and Mitra & Banka's algorithm presented the highest coverage for this problem, being the coverage of the biclusters generated by MOM-aiNet a little bit higher than those returned by the Mitra & Banka's proposal.

In order to be sure that the comparisons performed in this work were statistically significant, Student's t-tests were made (with p-value threshold equal to 10^{-3}) comparing the MOM-aiNet algorithm against all the other three techniques, with respect to the mean-squared residue, volume, coverage, overlap and occupancy (just for the Movielens dataset) of the returned biclusters. It is important to highlight here that these tests do not indicate that MOM-aiNet is better or worse than the other algorithms, but only that their results are statistically different. The results of these t-tests are given in Table 2 and, from this table, it is possible to see that the comparisons made in this paper are conclusive regarding all the criteria, except the overlap among biclusters in the Movielens dataset.

It can also be seen from Table 1 that the average volume of the biclusters generated by Mitra & Banka's algorithm is much higher than those obtained by MOM-aiNet, which is due to a much higher overlap presented by the biclusters of Mitra & Banka. This same situation can be observed when the results obtained by MOM-aiNet are compared to those of the BIC-aiNet algorithm. The MOM-aiNet presents mechanisms to maintain a low overlap between biclusters, as it stimulates both the diversity among different subpopulations and the diversity within the individuals in the same subpopulation (see Section 7).

Table 1 Mean and standard deviation values of the mean-squared residue, mean volume, coverage and overlap of the biclusters obtained by CC, BIC-aiNet, Mitra & Banka's algorithm and MOM-aiNet, for the Yeast dataset

	CC	BIC-aiNet	Mitra & Banka	MOM-aiNet
Residue	144.96 ± 2.9	232.04 ± 25.30	288.62 ± 3.35	294.56 ± 1.53
Volume	126.10 ± 0.2	$3,659.00 \pm 1,214.80$	$4,800.40 \pm 76.40$	$1,834.70 \pm 30.70$
Coverage (%)	76.75 ± 0.12	68.20 ± 11.28	83.26 ± 0.83	85.91 ± 1.06
Overlap (%)	10.67 ± 0.01	20.67 ± 7.58	20.76 ± 0.72	7.27 ± 0.26

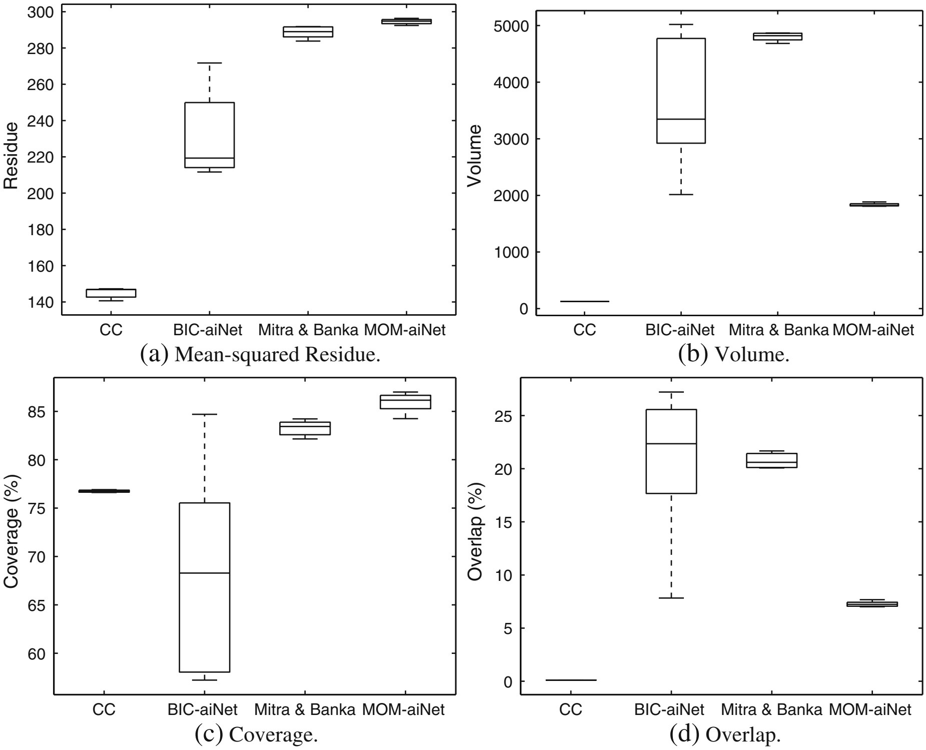
Table 2 Hypothesis Student's t-test for the comparisons of MOM-aiNet with the three other proposals, for each dataset studied in this paper using a p-value threshold of 10^{-3}

Dataset	Criterion	CC		BIC-aiNet		Mitra & Banka	
		H	p-value	H	p-value	H	p-value
Yeast	Mean-squared residue	True	0.0000	True	0.0001	True	0.0000
	Volume	True	0.0000	True	0.0001	True	0.0000
	Coverage	True	0.0000	True	0.0001	True	0.0000
	Overlap	True	0.0000	True	0.0001	True	0.0000
Human	Mean-squared residue	True	0.0000	True	0.0000	True	0.0000
	Volume	True	0.0000	True	0.0000	True	0.0000
	Coverage	True	0.0000	True	0.0000	True	0.0000
	Overlap	True	0.0000	True	0.0000	True	0.0000
Movielens	Mean-squared residue	True	0.0000	True	0.0000	True	0.0001
	Volume	True	0.0000	True	0.0000	True	0.0001
	Coverage	True	0.0000	True	0.0000	True	0.0001
	Overlap	False	0.1121	True	0.0000	False	0.8043
	Occupancy	True	0.0000	True	0.0000	True	0.0000

Considering the mean-squared residue, an interesting characteristic of the MOM-aiNet and Mitra & Banka's algorithm can be noticed. These algorithms allow the increase of the mean-squared residue of the biclusters up to values close to the upper boundary δ, so that their volume can also be increased. Thinking in the Pareto front in the "mean-squared residue × volume" objective space, it could be observed in this experiment that these two multi-objective algorithms tend to generate more biclusters of high volumes and high mean-squared residues than individuals with low mean-squared residues and also low volumes. This behavior is directly reflected in their respective average volume and average mean-squared residue obtained in the five independent runs.

In Fig. 6a, b, c and d it can be seen that, apart from BIC-aiNet, all algorithms present low variance in their results, which means that they are consistent and do not tend to present much variability from one run to another. It is important to highlight here that, although the variance for the volume and mean-squared residue shown in the box plots are small, this does not mean that the algorithms are not capable of finding one (or more) non-dominated fronts. Such variance is small because, at each run, we calculate the average volume and mean-square residue of the final set of biclusters obtained. Therefore, a small variance of these two criteria may indicate that the non-dominated front (or fronts) found by each algorithm in each run are within the same range.

The results presented in Table 1 and Fig. 6 illustrate well the different characteristics of each biclustering algorithm. For this problem MOM-aiNet presented the highest coverage of the data matrix, highlighting more correlations among the data than any other algorithm, while BIC-aiNet and Mitra & Banka's proposals presented biclusters with much higher volumes, but at the cost of a much higher overlap between such biclusters. All these three algorithms seem to have favored the volume and coverage characteristics in detriment of the mean-squared residue, differently from the CC algorithm, which led to a set of much smaller biclusters but also with a much smaller average mean-square residues (see Fig. 7). Therefore, it is not possible to assert that one of these algorithms is better than the other without



(a) Mean-squared Residue.
(b) Volume.
(c) Coverage.
(d) Overlap.

Fig. 6 Box plots of the results for the Yeast dataset (**a**–**d**)

considering the characteristics of the practical problem, as all of them present distinct advantages and disadvantages, when distinct criteria are considered.

Regarding the computational costs, all the algorithms presented equivalent times for this problem, except CC: MOM-aiNet, BIC-aiNet and Mitra & Banka's proposal

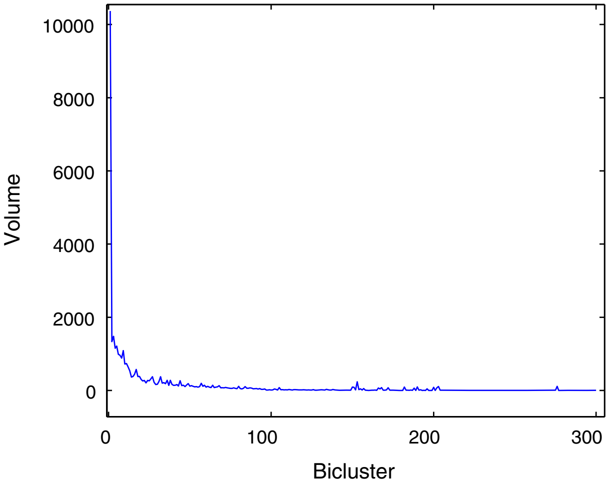


Fig. 7 Volume of each bicluster returned in a single execution of the CC algorithm for the Yeast problem, in descending order. Note that only a few biclusters with high volume are generated, being the majority of the proposals biclusters with low volume and also low mean-squared residue

spent about 1 hour on an AMD Turion 64 2GHz with 1GB RAM and CC about 25 minutes on the same machine.

8.2 Human Microarray Dataset

Concerning the Human microarray dataset results, given in Table 3, it is possible to see that the CC algorithm led to a set of biclusters that present the lowest average mean-squared residue, together with a good coverage of the dataset (the second highest rate) and no overlapping among the biclusters. The next best average mean-squared residue on this experiment was obtained with Mitra & Banka's algorithm, which has also presented biclusters with the second highest average volume, being more than twelve times larger than those obtained by CC in average. However, the algorithm of Mitra & Banka has also presented the smallest coverage of the dataset, and the largest amount of overlapping among the biclusters, which indicates that most of the biclusters generated remained concentrated in a small portion of the data matrix.

The MOM-aiNet algorithm, on the other hand, did not present the best results concerning the residue and the volume of the biclusters for this problem, but the mean-squared residue constraint was respected and the deficiency on the volume of the biclusters was compensated with the largest coverage of the dataset, with minimal overlapping (2.87%). This large coverage of the dataset increases the efficiency of the low-volume set of biclusters, as the goal of generating high-volume biclusters is akin to the one of providing a good coverage of the dataset: highlight the highest amount of correlations present in the data.

With a mean-squared residue just slightly higher than the one presented by Mitra & Banka's proposal, BIC-aiNet was able to find a set of biclusters with much higher average volume. Also, BIC-aiNet had a good average coverage compared to the other algorithms, although corresponding to half the coverage of the MOM-aiNet. However, the overlapping of the biclusters generated by BIC-aiNet was almost as high as the one of Mitra & Banka.

Then again we have different characteristics on each set of biclusters generated by each algorithm. If the application requires a high coverage of the dataset with little or no overlapping and the maintenance of average to high volume biclusters, the MOM-aiNet is the most indicated algorithm. If you look for a set of biclusters with the highest volume, that just respects the residue threshold and presents a standard coverage with overlaps, the BIC-aiNet is the best choice. And also, if you want to cover the dataset with biclusters with the smallest possible mean-squared residue that do not overlap, then CC may be the best option, although experiments have

Table 3 Mean and standard deviation values of the mean-squared residue, mean volume, coverage and overlap obtained by CC, BIC-aiNet, Mitra & Banka's proposal and MOM-aiNet, for the Human dataset

	CC	BIC-aiNet	Mitra & Banka	MOM-aiNet
Residue	915.50 ± 6.27	1,156.70 ± 2.62	1,052.90 ± 2.33	1,097.50 ± 18.24
Volume	610.00 ± 3.61	11,606.00 ± 38.39	7,559.00 ± 83.41	2,953.00 ± 126.62
Coverage (%)	31.57 ± 0.19	23.08 ± 0.11	17.08 ± 0.20	41.65 ± 0.76
Overlap (%)	0.00 ± 0.00	32.00 ± 0.21	34.57 ± 0.27	2.87 ± 0.21

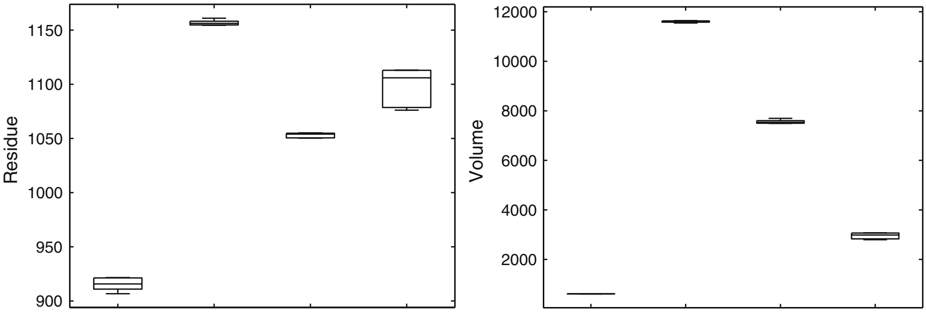
pointed out that this algorithm cannot go much further regarding coverage, even if the number of generated biclusters is raised.

From Fig. 8a, b, c and d we can see again that the results were stable enough (presented low variance) to discard any doubts about the behavior of each studied algorithm. It is also possible to notice that, for this problem, the variance of MOM-aiNet results was a little higher than the variance of the other proposals, but this is still not enough to raise uncertainties about the given results, which were corroborated by the Student's t-test given in Table 2.

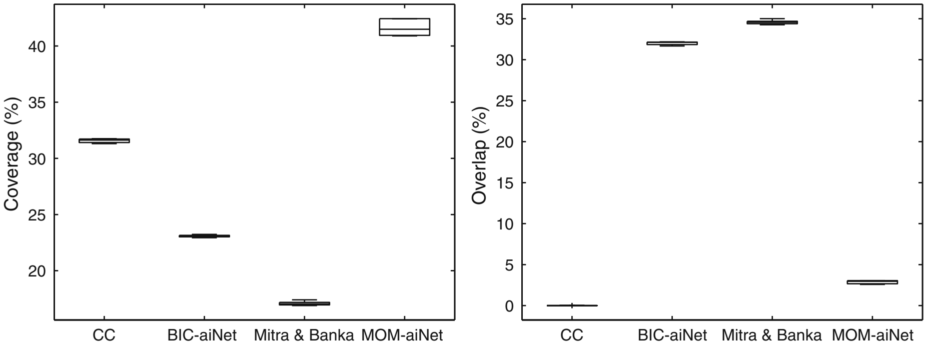
Concerning the computational costs associated with each algorithm for this problem, MOM-aiNet, BIC-aiNet and Mitra & Banka's proposal spent about 5 hours on an AMD Turion 64 2 GHz with 1 GB RAM, while CC spent about 80 minutes to generate the final set of biclusters.

8.3 Movielens Dataset

In the Movielens dataset, which is highly sparse, another characteristic of the biclusters must be controlled by the algorithms: their occupancy rate. Since we expect to have enough information to create inferences or draw conclusions from the correlations represented in the biclusters, the amount of non-null values in each



(a) Mean-squared Residue. (b) Volume.



(c) Coverage. (d) Overlap.

Fig. 8 Box plots of the results for the Human dataset (**a**–**d**) bicluster is very important as, according to the application, biclusters with high or very low amounts of null values may not be desired.

The results obtained by each algorithm for the Movielens problem are given in Table 4. Analyzing the CC performance, it is possible to see that, although this algorithm provides 100% of occupancy and the best mean-squared residue, it also generates biclusters with very low volume and coverage, which indicates that CC is only capable of generating small biclusters positioned in a small region of the dataset.

Regarding the other three algorithms, it is possible to see that they generate sets of biclusters with very similar mean-squared residues, but with significantly different characteristics, as they differ in volume, coverage, overlap and occupancy. In this problem the MOM-aiNet algorithm has led to biclusters with the highest volume and the highest coverage of the dataset while maintaining a fairly low overlap rate (although it was almost twice the overlap presented by BIC-aiNet) and a good occupancy, given the high volume of the biclusters. As Movielens is a sparse dataset, the higher the volume of the bicluster, the higher the probability that this bicluster will have a null value, which reduces its occupancy rate. Nevertheless, the MOM-aiNet algorithm still respected the imposed occupancy threshold established ($\gamma = 0.6$).

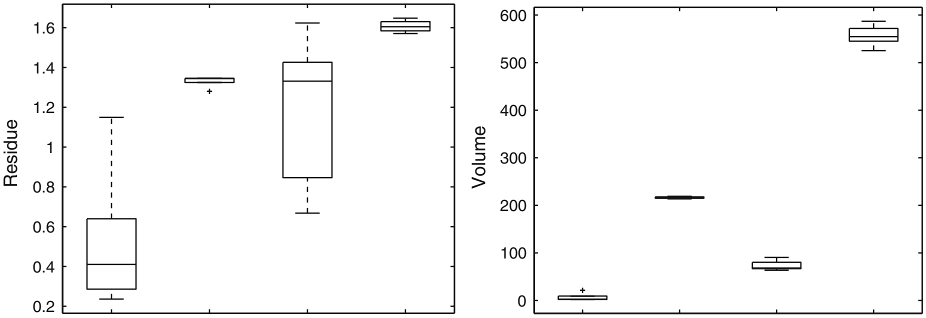
Therefore, for problems that present characteristics similar to Movielens, if biclusters with high occupancy rate are required, the BIC-aiNet algorithm seems to perform better (even though it has not led to the highest occupancy rate in Movielens, it presents a good trade-off between volume and residue), but if such applications do not require high occupancy rates but they demand high-volume biclusters, the MOM-aiNet should be the best choice.

Although the results obtained by the MOM-aiNet algorithm for this dataset may indicate that this algorithm is not suitable for some applications, as the occupancy rate obtained by this algorithm is smaller than those obtained by the other three techniques, this is the best scenario for collaborative filtering problems, as one of the goals of such problems is to extract the highest amount of information from the dataset in order to predict the missing values. For this reason, it is required that the biclusters obtained present a high occupancy rate so that the missing information can be inferred. However, if a bicluster presents 100% of occupancy rate, it will be meaningless to this kind of application, as it does not contribute to the prediction of any missing value.

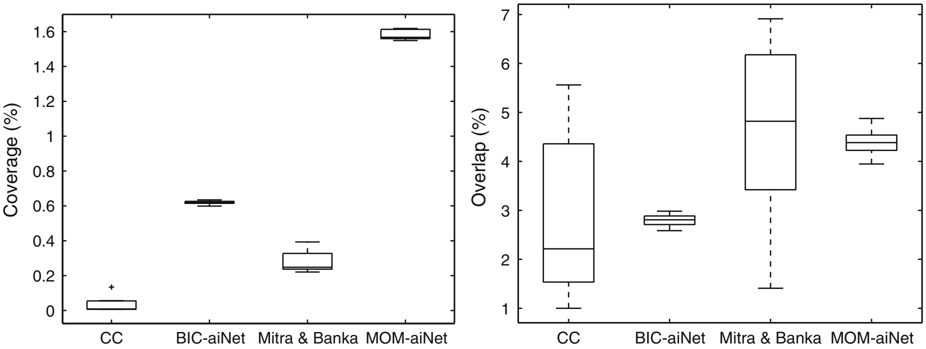
In Fig. 9a, b, c and d it is possible to see that, differently from the Yeast and Human problems, the algorithm of Mitra & Banka and CC presented a much higher variance in the mean-squared residue and overlap. This can be associated with the small coverage presented by these two algorithms, which leads them to locate different sets of biclusters in each run, extracting different correlations at each time

Table 4 Mean and standard deviation values of the mean-squared residue, average volume, coverage, overlap and occupancy of the biclusters obtained by CC, BIC-aiNet, Mitra & Banka's proposal and MOM-aiNet, for the Movielens dataset

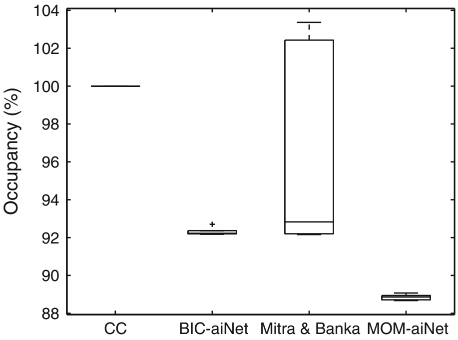
	CC	BIC-aiNet	Mitra & Banka	MOM-aiNet
Residue	0.51 ± 0.37	1.33 ± 0.03	1.18 ± 0.38	1.61 ± 0.03
Volume	6.72 ± 8.29	216.43 ± 2.05	73.42 ± 10.73	557.14 ± 22.55
Coverage (%)	0.04 ± 0.06	0.62 ± 0.01	0.28 ± 0.07	1.58 ± 0.03
Overlap (%)	2.97 ± 1.03	2.80 ± 0.14	4.63 ± 2.10	4.39 ± 0.33
Occupancy (%)	100.00 ± 0.00	92.31 ± 0.22	96.54 ± 5.69	88.84 ± 0.16



(a) Mean-squared Residue. (b)Volume.



(c) Coverage. (d) Overlap.



(e) Occupancy.

Fig. 9 Box plots of the results for the Movielens dataset (**a**–**e**)

and, consequently, presenting different residue and overlapping profiles. Also, this high variance in the overlap criterion explains why the t-test indicated that there are no statistical differences among the overlap results of CC and Mitra & Banka's proposals and the one obtained by MOM-aiNet (see Table 2).

Concerning the computational time spent by each algorithm for Movielens, it was verified that MOM-aiNet was about 6 times slower than BIC-aiNet and Mitra & Banka's technique, due to the constraint on occupancy included in MOM-aiNet. For this problem, MOM-aiNet spent about 3 hours on an AMD Turion 64 2 GHz with 1 GB RAM, while BIC-aiNet and Mitra & Banka's proposal spent about

30 minutes. For this problem, CC was again the fastest approach, obtaining its final set of biclusters in less than 10 min on the same machine.

8.4 Number of Fronts Obtained by MOM-aiNet and Mitra & Banka's Algorithm

In this work, it was also evaluated the final number of fronts in the objective space found by MOM-aiNet and Mitra & Banka's algorithm. This value is calculated by counting the total number of ranks generated by the final population of biclusters (see *non-dominated ranking* in Section 5.2), and is given in Table 5 (together with the number of biclusters in the final population). It may seem, at first sight, that the MOM-aiNet could find a better set of frontiers because of the elevated number of biclusters generated, but it is important to notice that each sub-population generally belongs entirely to a single front, which means that the algorithm is capable of maintaining and evolving several fronts in parallel. This capability of working with a large number of fronts is directly reflected in the percentage of coverage of the search space accomplished by this algorithm, as MOM-aiNet is able to explore a given region of the dataset without the risk of discarding it, during its execution, for belonging to a lower rank (and thus being dominated by solutions in different subpopulations).

9 Final Remarks

In this paper, a multi-objective multipopulation artificial immune network for biclustering, named MOM-aiNet, was explained in further details and had its performance evaluated and compared with three other algorithms from the literature, being one of them also developed based on multi-objective optimization concepts. The main characteristics of MOM-aiNet are the generation of more than a single non-dominated front of solutions in parallel, representing different regions of the original data, and the dynamic variation of its population size, which is automatically adjusted to cover the highest amount of the data space.

This algorithm was applied to three well known datasets from the literature, being two of them gene expression datasets (popularly known as Yeast and Human B-Cell Lymphoma), and the other one a dataset from the collaborative filtering literature (Movielens), which presents as its main feature a very high degree of sparseness. The results were analyzed with the verification of the balance among the mean-squared residue, volume, coverage, overlap and, when suited, occupancy rate of the biclusters. The scope of the analysis made here was to show that each algorithm could

Table 5 Final number of biclusters and fronts obtained by MOM-aiNet and Mitra & Banka's algorithm

Dataset	Algorithm	Final number of Biclusters	Final number of Fronts
Yeast	MOM-aiNet	1,800	155
	Mitra & Banka	300	28
Human	MOM-aiNet	1,200	192
	Mitra & Banka	200	23
Movielens	MOM-aiNet	600	122
	Mitra & Banka	100	28

fit different real world application situations but also that, in general, MOM-aiNet could generate a really good material for post analysis in practically every situation.

From the comparisons made, it was possible to conclude that MOM-aiNet, even though explicitly optimizing just two objectives (volume and residue), aimed at building a set of solutions that maximize the coverage of the data, mainly because of its multipopulation aspect and its suppression mechanism that inhibits two populations from exploring similar areas of the search space. Also because of this suppression mechanism, the overlap among biclusters was kept under control and implicitly minimized, although this can always be relaxed with a proper adjustment of the suppression threshold. Finally, regarding the volume-residue trade-off, MOM-aiNet could not always find the maximum volume on dense datasets, which is again due to the effect of the suppression mechanism inhibiting two populations from growing indefinitely and intersecting with each other. The most noticeable advantage of the proposed algorithm was on the sparse dataset where MOM-aiNet was able to find the best trade-off among the objectives (except, maybe, for the occupancy, but given the higher volume, the occupancy rate should decay inevitably).

It was also shown in this work that MOM-aiNet is capable of finding a larger number of distinct non-dominated fronts when compared to Mitra & Banka's algorithm, which is directly reflected in the coverage capability of the algorithm, as each region of the dataset presents different properties that might lead to biclusters dominated by solutions in distinct regions. Thus, even though the biclustering search procedure demands the optimization of conflicting objectives, the focus just on the generation of non-dominated solutions does not seem to be enough to obtain a set of biclusters as meaningful as possible to a given application.

As future works, we intend to explore the use of more biclusters on these datasets until a coverage close to 100% is achieved, include a local search step in order to improve each subpopulation without changing much of its structure and, finally, include more objectives in the optimization process, such as the row variance of the biclusters, to verify whether even more useful biclusters can be generated.

Acknowledgements This research was sponsored by UOL (www.uol.com.br), through its UOL Bolsa Pesquisa program, process number 20080129215000. The authors would also like to thank CNPq and CAPES for their additional financial support.

References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann, San Francisco (2006)
2. de França, F.O., Bezerra, G., Von Zuben, F.J.: New perspectives for the biclustering problem. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pp. 753–760. IEEE, Vancouver (2006)
3. Parmigiani, G., Garett, E.S., Irizarry, R.A., Zeger, S.L. (eds.): The Analysis of Gene Expression Data. Springer, New York (2003)
4. Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: Proceedings of the 1999 Conference on Research and Development in Information Retrieval, pp. 230–237, Berkeley, 15–19 August 1999
5. Feldman, R., Sanger, J.: The Text Mining Handbook. Cambridge University Press, Cambridge (2006)
6. Hartigan, J.A.: Direct clustering of a data matrix. J. Am. Stat. Assoc. (JASA) **67**(337), 123–129 (1972)

7. Mirkin, B.: Mathematical Classification and Clustering. Nonconvex Optimization and Its Applications. Springer, New York (1996)
8. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proc. of the 8th Int. Conf. on Intelligent Systems for Molecular Biology, pp. 93–103, La Jolla, 19–23 August 2000
9. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. Wiley, Chichester (2001)
10. Mitra, S., Banka, H.: Multi-objective evolutionary biclustering of gene expression data. Pattern Recogn. **39**, 2464–2477 (2006)
11. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
12. Mitra, S., Banka, H., Pal, S.K.: A MOE framework for biclustering of microarray data. In: Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), pp. 1154 – 1157, Hong Kong, 20–24 August 2006
13. Divina, F., Aguilar–Ruiz, J.S.: A multi-objective approach to discover biclusters in microarray data. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'07), pp. 385–392, London, 7–11 July 2007
14. Giráldez, R., Divina, F., Pontes, B., Aguilar–Ruiz, J.S.: Evolutionary search of biclusters by minimal intrafluctuation. In: Proceedings of the IEEE International Fuzzy Systems Conference (FUZZ–IEEE 2007), pp. 1–6. IEEE, London (2007)
15. Maulik, U., Mukhopadhyay, A., Bandyopadhyay, S., Zhang, M.Q., Zhang, X.: Multiobjective fuzzy biclustering in microarray data: method and a new performance measure. In: Proceedings of the 2008 IEEE Congress on Evolutionary Computation (CEC 2008), pp. 1536–1543. IEEE, Hong Kong (2008)
16. Krishnapuram, R., Joshi, A., Yi, L.: A fuzzy relative of the k–medoids algorithm with application to document and snippet clustering. In: Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'99), pp. 1281–1286. IEEE, Seoul (1999)
17. Coelho, G.P., de França, F.O., Von Zuben, F.J.: A multi-objective multipopulation approach for biclustering. In: Bentley, P.J., Lee, D., Jung, S. (eds.) Artificial Immune Systems, Proc. of the 7th International Conference on Artificial Immune Systems (ICARIS), Lecture Notes in Computer Science, vol. 5132, pp. 71–82, Phuket, 10–13 August 2008
18. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, New York (2002)
19. Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J.: Applying biclustering to text mining: an immune-inspired approach. In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) Artificial Immune Systems, Proc. of the 6th International Conference on Artificial Immune Systems (ICARIS), Lecture Notes in Computer Science, vol. 4628, pp. 83–94, Santos, 26–29 August 2007
20. Agrawal, R., Gehrke, J., Gunopulus, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proc. of the ACM/SIGMOD Int. Conference on Management of Data, pp. 94–105, Seattle, 2–4 June 1998
21. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proc. of the 7th Int. Con. on Knowledge Discovery and Data Mining, pp. 269–274, San Francisco, 26–29 August 2001
22. Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J.: Applying biclustering to perform collaborative filtering. In: Proc. of the 7th International Conference on Intelligent Systems Design and Applications, pp. 421–426, Rio de Janeiro, 22–24 October 2007
23. Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J.: Evaluating the performance of a biclustering algorithm applied to collaborative filtering: a comparative analysis. In: Proc. of the 7th International Conference on Hybrid Intelligent Systems, pp. 65–70, Kaiserslautern, 17–19 September 2007
24. Symeonidis, P., Nanopoulos, A., Papadopoulos, A., Manolopoulos, Y.: Nearest-biclusters collaborative filtering with constant values. In: Advances in Web Mining and Web Usage Analysis, Lecture Notes in Computer Science, vol. 4811, pp. 36–55. Springer, Philadelphia (2007)
25. Tang, C., Zhang, L., Zhang, I., Ramanathan, M.: Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In: Proc. of the 2nd IEEE Int. Symposium on Bioinformatics and Bioengineering, pp. 41–48. IEEE, Piscataway (2001)
26. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE Trans. Comput. Biol. Bioinformatics **1**(1), 24–45 (2004)
27. Edgeworth, F.Y.: Mathematical Physics. P. Keagan, London (1881)
28. Pareto, V.: Cours D'Economie Politique. F. Rouge, Lausanne (1896)

29. Burnet, F.M.: Clonal selection and after. In: Bell, G.I., Perelson, A.S., Pimgley, G.H. Jr (eds.) Theoretical Immunology, pp. 63–85. Marcel Dekker, New York (1978)
30. Jerne, N.K.: Towards a network theory of the immune system. Ann. Immunol. Inst. Pasteur **125C**, 373–389 (1974)
31. de Castro, L.N., Von Zuben, F.J.: Learning and optimization using the Clonal selection principle. IEEE Trans. Evol. Comput. **6**(3), 239–251 (2002)
32. de Castro, L.N., Von Zuben, F.J.: aiNet: an artificial immune network for data analysis. In: Abbass, H.A., Sarker, R.A., Newton, C.S. (eds.) Data mining: a heuristic approach, pp. 231–259. Idea Group, Harrisburg (2001)
33. Coelho, G.P., Von Zuben, F.J.: omni-aiNet: an immune-inspired approach for omni optimization. In: Bersini, H., Carneiro, J. (eds.) Artificial Immune Systems, Proc. of the 5th International Conference on Artificial Immune Systems (ICARIS), Lecture Notes in Computer Science, vol. 4163, pp. 294–308. Oeiras, Portugal (2006)
34. Cho, R., Campbell, M., Winzeler, E., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T., Gabrielian, A., Landsman, D., Lockhart, D., Davis, R.: A genome-wide transcriptional analysis of the mitotic cell cycle. Mol. Cell **2**, 65–73 (1998)
35. Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., Rosenwald, A., Boldrick, J.C., Sabet, H., Tran, T., Yu, X., Powell, J.I., Yang, L., Marti, G.E., Moore, T., Hudson J., Jr., Lu, L., Lewis, D.B., Tibshirani, R., Sherlock, G., Chan, W.C., Greiner, T.C., Weisenburger, D.D., Armitage, J.O., Warnke, R., Levy, R., Wilson, W., Grever, M.R., Byrd, J.C., Botstein, D., Brown, P.O., Staudt, L.M.: Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature **403**, 503–510 (2000)
36. Snedecor, G.S., Cochran, W.G.: Statistical Methods. Iowa University Press, Iowa (1989)