Ann Math Artif Intell (2014) 70:55–79
DOI 10.1007/s10472-013-9379-1

Biclustering meets triadic concept analysis

**Mehdi Kaytoue · Sergei O. Kuznetsov ·
Juraj Macko · Amedeo Napoli**

Published online: 6 September 2013

Abstract Biclustering numerical data became a popular data-mining task at the beginning of 2000's, especially for gene expression data analysis and recommender systems. A bicluster reflects a strong association between a subset of objects and a subset of attributes in a numerical object/attribute data-table. So-called biclusters of similar values can be thought as maximal sub-tables with close values. Only few methods address a complete, correct and non-redundant enumeration of such patterns, a well-known intractable problem, while no formal framework exists. We introduce important links between biclustering and Formal Concept Analysis (FCA). Indeed, FCA is known to be, among others, a methodology for biclustering binary data. Handling numerical data is not direct, and we argue that Triadic Concept Analysis (TCA), the extension of FCA to ternary relations, provides a powerful mathematical and algorithmic framework for biclustering numerical data. We discuss hence both theoretical and computational aspects on *biclustering numerical data with triadic concept analysis*. These results also scale to *n*-dimensional numerical datasets.

M. Kaytoue (✉)
Université de Lyon, CNRS, INSA-Lyon, LIRIS, 20, Avenue Albert Einstein,
69621 Villeurbanne Cedex, France
e-mail: mehdi.kaytoue@insa-lyon.fr

S. O. Kuznetsov
Higher School of Economis (HSE), Pokrovskiy Bd. 11, 109028 Moscow, Russia
e-mail: skuznetsov@hse.ru

J. Macko
Palacky University, 17. listopadu, 77146 Olomouc, Czech Republic
e-mail: juraj.macko@upol.cz

A. Napoli
Laboratoire Lorrain de Recherche en Informatique et ses Applications (LORIA),
B.P. 239, 54500 Vandœuvre-lès-Nancy, France
e-mail: amedeo.napoli@loria.fr

Keywords Numerical biclustering · Similarity relation · Formal concept analysis · Triadic concept analysis · N-ary relations

Mathematics Subject Classifications (2010) 06 · 68

1 Introduction

Taking roots in the work of Hartigan [13] in 1972 and extended by Mirkin in 1996 [27], numerical data biclustering then strongly attracted attention from the beginning of 2000's as a first answer to new challenges raised by gene expression data analysis [11] and recommender systems design [1]. Starting from an object/attribute numerical data-table, the goal is to group together some objects with some attributes according to the values taken by these attributes for these objects. The main idea of biclustering is to overcome the limitation of standard clustering techniques producing partitions of objects where distance functions that use all the attributes may be ineffective and hard to interpret [2]. For example, in gene expression data, it is known that genes (objects) may share a common behavior for a subset of biological situations (attributes) only: one should accordingly produce local patterns to characterize biological processes, the latter should possibly overlap, since a gene may be involved in several processes. The same remark applies for recommender systems, where the taste of users for some items is realized by a so-called utility matrix (usually very sparse): one is interested in local patterns characterizing groups of users that strongly share almost the same tastes for a subset of items [1].

Accordingly, a bicluster is formally defined as a pair composed of a set of objects and a set of attributes. Such a pair can be represented as a rectangle in a numerical table, modulo rows and columns permutations. Table 1 is a numerical dataset with objects in rows and attributes in columns, while each table entry corresponds to the value taken by the attribute in column for the object in row. Table 2 illustrates bicluster $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\})$ as a grey rectangle that can be understood as a sub-table of the original one. There are several types of biclusters in the literature, depending on the relation between the values taken by their attributes for their objects (as surveyed by Madeira and Oliveira [26]). The most simple case can be understood as rectangles of equal values: a bicluster corresponds to a set of objects whose attributes take exactly the same value, e.g. $(\{g_1, g_2, g_3\}, \{m_5\})$. Constant biclusters only appear in idyllic situations. Accordingly, a straightforward generalization of such biclusters lies in so-called biclusters of similar values: they are represented by rectangles with almost identical, say similar, values (see [6, 19, 26] and to a similar extent [9]). Table 2 illustrates a bicluster of similar values $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\})$ where two values are said to be similar if their difference is no more than one. Moreover, this bicluster is maximal: neither an object nor an attribute can be

Table 1 A numerical dataset

	m_1	m_2	m_3	m_4	m_5
g_1	1	2	2	1	6
g_2	2	1	1	0	6
g_3	2	2	1	7	6
g_4	8	9	2	6	7

Table 2 A bicluster of similar values

	m_1	m_2	m_3	m_4	m_5
g_1	**1**	**2**	**2**	1	6
g_2	**2**	**1**	**1**	0	6
g_3	**2**	**2**	**1**	7	6
g_4	8	9	2	6	7

added without violating the similarity condition. The problem of biclustering that we investigate in this paper consists in extracting all pairs (A, B), such that A and B are maximal sets with respect to a similarity constraint between values.

To better understand our investigation, we recall a definition of bicluster of Prelic et al. in binary data or relation, i.e. an object has or not an attribute [32]: *inclusion-maximal biclusters* are defined as maximal sets of objects related to a maximal set of attributes. As shown in [21], this definition exactly meets the one of *formal concepts* in the Formal Concept Analysis theory (FCA, [12]). Hence, our general intuition is that FCA can be used to answer the problem of biclustering numerical data, which is not straightforward, FCA basically applying to binary data.

Formal Concept Analysis is a branch of applied lattice theory that appeared in the 1980's [12, 38] and proved to be very useful in data analysis. It aims at representing data as a formal concept hierarchy, the later being useful for many tasks of, among others, knowledge management and data-mining [4, 33, 36, 39]. Starting from a binary relation between a set of objects and their attributes, so-called formal concepts are built as maximal sets of objects in relation with a maximal set of attributes. If we represent the binary relation as a binary table (with objects as rows, attributes as columns and 0/1 as values if an object has/has not an attribute), a formal concept is represented as a maximal rectangle of 1 values (or crosses $\times$ in the following of this paper such as in Fig. 1). The ordering of concepts among a complete lattice makes overlapping of such local and maximal patterns natural. Then a complete enumeration of patterns respecting some constraints like closure and minimal frequency is possible [7, 24]. Indeed, the subsets of patterns satisfying these constraints is an order ideal of the lattice of patterns.

It is now natural to argue that FCA can be considered as a kind of biclustering method for binary data. As such, it has been applied to numerical data, and especially to gene expression data after an adequate transformation, see e.g. [7, 30–32]. The process that turns numerical data into binary data (discretization), usually called

	m_1	m_2	m_3
g_1			$\times$
g_2	$\times$	$\times$	
g_3		$\times$	$\times$
g_4		$\times$	$\times$

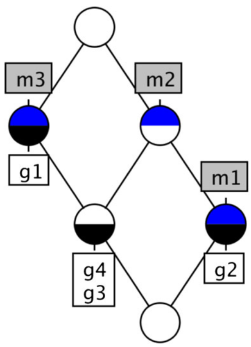


Fig. 1 A dyadic context (*with a concept in grey*) and its concept lattice

conceptual scaling in FCA, generally comes with a loss of information, and thus the obtained formal concepts are not exactly and formally related with biclusters (although they are good representatives). This being stated, biclustering binary data is still attracting a lot of attention, to cope with several issues such as the number of produced patterns and enabling a fault tolerance to leverage the strict notion of maximality of formal concepts, see e.g. [5, 10, 14, 28, 29]. Biclustering directly numerical data, without a priori binarization, has also been widely studied, and several ad hoc algorithms have been proposed to extract specific kind of biclusters with different algorithmic strategies (such as *divide-and-conquer*, *greedy iterative search*, *exhaustive enumeration* as deeply surveyed in [26]). Indeed enumerating all biclusters of a given type is an intractable problem and complete approaches generally fail. Our main contribution states that such approach is possible when considering the problem of extracting maximal bicluster of similar values in formal concept analysis settings, outperforming the other existing algorithms for this task [6, 19]. Other concerns of biclustering are to be able to consider multi-dimensional data (e.g. when the expression of a gene is monitored in several situations across time [35, 40]) and parallelization of the algorithms [8] which both are important issues we address in this paper. This leads us to our main contributions.

Problem We consider here maximal biclusters of similar values, denoted by (A, B) where A and B are respectively maximal sets of objects and attributes, such that the values taken by these attributes for these objects are pairwise similar. Given a similarity parameter θ, the similarity relation is defined as $a \simeq_\theta b \iff |b - a| \leq \theta$, for any numbers a and b. The problem is to design an approach that allows an exact, correct and complete extraction of maximal biclusters of similar values.

Contribution 1 Triadic Concept Analysis (TCA) [25] is an extension of FCA to handle ternary relations: an object has an attribute under a given condition. This leads to triadic contexts, i.e. data are represented as a "box", where so-called triadic concepts can be seen as maximal sets of objects in relation with a maximal set of attributes under a maximal set of conditions, i.e. a maximal "sub-box" of $\times$ in the context (still with rows, columns and layers permutations). We show then, that after turning the original numerical data in a triadic context without loss of information (with interordinal scaling [12]), the resulting triadic concepts are in 1-1-correspondence with the maximal biclusters of similar values for any similarity parameter θ (stating if two values are similar or not). Then, such concepts can be organized in a trilattice whose diagram gives a visualization of biclusters in the numerical dataset. Finally, we show that this result naturally holds when considering n-dimensional numerical datasets.

Contribution 2 Maximal biclusters of similar values for a user-defined similarity parameter have been studied with complete approaches in [6, 19]. In [6], an algorithm for extracting such biclusters is presented, while [19] shows how such biclusters can be characterized by post-processing a concept lattice built from the numerical data directly. We show that our first contribution can be easily adapted to answer this problem, with a new generic algorithm TriMax that shows better results than its competitors and can be naturally parallelized.

For summarizing, this research article is two-fold: first, theoretical new links are emphasized between biclustering and FCA in general, and TCA in particular, for a better understanding of numerical pattern mining with closure operators. Secondly, a computational aspect is investigated using these links: it allows one to bring back a problem of biclustering into well known-settings (i.e. FCA and pattern-mining) and comes with better computational properties and several perspectives of research. Note that this paper paper extends our previous work [18] by adapting the methodology to n-dimensional data and showing how the set of concepts can be represented by line diagrams.

The paper is organized as follows. Firstly, we present the preliminaries regarding FCA and TCA in Section 2. Thanks to the introduced notations, we formally state the problem in Section 3. The Sections 4, 5 respectively tackle our two main contributions. The paper ends with a conclusion suggesting further research.

2 Formal concept analysis

Formal Concept Analysis (FCA) [12] is a mathematical framework for allowing one to derive implicit relationships from a set of objects and their attributes. Starting from a relation stating which objects have which attributes, it allows one to build a so-called concept lattice. A concept is there seen as a maximal set of objects sharing a maximal set of attributes. Ordering concepts with a specialization/generalization relation gives rise to a concept lattice. This structure can be represented by a diagram where classes of objects/attributes and ordering relations between classes can be drawn, interpreted and used for data-mining, knowledge management and discovery [36, 39].

2.1 Dyadic concept analysis

We use standard definitions from [12]. Let G and M be arbitrary sets and $I \subseteq G \times M$ be an arbitrary binary relation between G and M. The triple (G, M, I) is called a formal context, or *dyadic context*. Each $g \in G$ is interpreted as an object, each $m \in M$ is interpreted as an attribute. The fact $(g, m) \in I$ is interpreted as "g has attribute m". The two following derivation operators $(\cdot)'$:

$$A' = \{m \in M \mid \forall g \in A : gIm\} \qquad for\ A \subseteq G,$$

$$B' = \{g \in G \mid \forall m \in B : gIm\} \qquad for\ B \subseteq M$$

define a Galois connection between the powersets of G and M. The derivation operators $\{(\cdot)', (\cdot)'\}$ put in relation elements of the lattices $(\wp(G), \subseteq)$ of objects and $(\wp(M), \subseteq)$ of attributes and vice-versa. A Galois connection induces closure operators $(\cdot)''$ and realizes a one-to-one correspondence between all closed sets of objects and all closed sets of attributes. For $A \subseteq G$, $B \subseteq M$, a pair (A, B) such that $A' = B$ and $B' = A$, is called a *formal concept*, or *dyadic concept*. Concepts are partially ordered by $(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2$ $(\Leftrightarrow B_2 \subseteq B_1)$. (A_1, B_1) is a sub-concept of (A_2, B_2), while the latter is a super-concept of (A_1, B_1). With respect to this partial order, the set of all formal concepts forms a complete lattice called the *concept lattice* of the formal context (G, M, I), i.e. any subset of concepts has both a supremum (join) and an infimum (meet), see Theorem 1. For a concept (A, B) the set A is called the *extent* and the set B the *intent* of the concept.

Theorem 1 (The Basic Theorem on Concept Lattices [12]) *The concept lattice of a context* (G, M, I) *is a complete lattice in which infimum and supremum are given by:*

$$\bigwedge_{t\in T}(A_t, B_t) = \left(\bigcap_{t\in T} A_t, \left(\bigcup_{t\in T} B_t\right)''\right)$$

$$\bigvee_{t\in T}(A_t, B_t) = \left(\left(\bigcup_{t\in T} A_t\right)'', \bigcap_{t\in T} B_t\right)$$

Example Figure 1 shows a dyadic context and its concept lattice. Starting from an arbitrary set of objects, say $\{g_3\}$, one obtains concept $(\{g_3\}'', \{g_3\}') = (\{g_3, g_4\}, \{m_2, m_3\})$ (in grey). The diagram shows the resulting concept lattice: each node denotes a concept while a line denotes an order relation between two concepts. The top (resp. bottom) concept is the highest (resp. lowest) concept w.r.t. $\leq$. *Reduced labeling* avoids to display the whole concept extents and intents. The extent of a concept has to be considered as composed of all object labels attached to it and its sub-concepts; the intent of a concept is composed of all attributes attached to it and its super-concepts.[1]

2.2 Triadic concept analysis

Lehmann and Wille introduced Triadic Concept Analysis (TCA [25]) to handle ternary relations between objects, attributes and conditions. Data are formalized by a triadic context in which triadic concepts are defined.

Definition 1 (Triadic context) Data are represented by a triadic context $\mathbb{K} = (G, M, B, Y)$, where G, M, and B are respectively called sets of objects, attributes and conditions, and $Y \subseteq G \times M \times B$. The fact $(g, m, b) \in Y$ is interpreted as the statement "Object g has attribute m under condition b".

Example An example of such triadic context is given in Table 3 where the very first cross (to the left) denotes the fact "Object g_2 has attribute m_1 under the condition b_1, i.e. $(g_2, m_1, b_1) \in Y$. In this tabular representation, each table corresponds to the projection of the triadic context for one condition. Another choice could have been made.

Definition 2 (Triadic concept) A triadic concept of (G, M, B, Y) is a triple (A_1, A_2, A_3) with $A_1 \subseteq G$, $A_2 \subseteq M$ and $A_3 \subseteq B$ satisfying the two following statements: (i) $A_1 \times A_2 \times A_3 \subseteq Y$ and (ii) for $X_1 \times X_2 \times X_3 \subseteq Y$, we have $A_1 \subseteq X_1$, $A_2 \subseteq X_2$ $A_3 \subseteq X_3$ implies $(A_1, A_2, A_3) = (X_1, X_2, X_3)$. If (G, M, B, Y) is represented by a three dimensional table, (i) means that a concept stands for a rectangular parallelepiped full of crosses while (ii) characterizes component-wise maximality of

[1]More details on the ConExp software: http://conexp.sourceforge.net/.

Table 3 A triadic context (G, M, B, Y) with the triadic concept $(\{g_3, g_4\}, \{m_2, m_3\}, \{b_1, b_2, b_3\})$

	b_1			b_2			b_3		
	m_1	m_2	m_3	m_1	m_2	m_3	m_1	m_2	m_3
g_1			×	×	×	×	×		×
g_2	×	×		×	×		×		
g_3		×	×	×	×	×	×	×	×
g_4		×	×		×	×		×	×

concepts. For a triadic concept (A_1, A_2, A_3), A_1 is called the extent, A_2 the intent and A_3 the modus.

Example $(\{g_3, g_4\}, \{m_2, m_3\}, \{b_1, b_2, b_3\})$ is a triadic concept in the triadic context represented by Table 3. Representing the triadic context as a box, where each condition is a layer, one can observe that this triadic concept denotes a maximal rectangular parallelepiped of crosses (modulo lines, columns and layers permutations).

Definition 3 (Outer derivation operators) To describe the derivation operators, it is convenient to represent a triadic context as (K_1, K_2, K_3, Y). Then, for $\{i, j, k\} = \{1, 2, 3\}$, $j < k$, $X \subseteq K_i$ and $Z \subseteq K_j \times K_k$, (i)-derivation operators are defined by:

$$\Phi : X \rightarrow X^{(i)} : \{(a_j, a_k) \in K_j \times K_k \mid (a_i, a_j, a_k) \in Y \text{ for all } a_i \in X\}$$
$$\Phi^{'} : Z \rightarrow Z^{(i)} : \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_j, a_k) \in Z\}$$

This definition leads to dyadic contexts

$$\mathbb{K}^{(1)} = \langle K_1, K_2 \times K_3, Y^{(1)} \rangle$$
$$\mathbb{K}^{(2)} = \langle K_2, K_1 \times K_3, Y^{(2)} \rangle$$
$$\mathbb{K}^{(3)} = \langle K_3, K_1 \times K_2, Y^{(3)} \rangle$$

where $gY^1(m, b) \iff mY^2(g, b) \iff bY^3(g, m)$.

Example Consider $i = 1$, $j = 2$ and $k = 3$, i.e. $K_1 = G$, $K_2 = M$ and $K_3 = B$. Given an arbitratry set of objects $X = \{g_4\}$, we have:

$$\Phi(X) = \{(m_2, b_1), (m_3, b_1), (m_2, b_2), (m_3, b_2), (m_2, b_3), (m_3, b_3)\}$$
$$\Phi'\Phi(X) = \{g_3, g_4\}$$

Definition 4 (Inner derivation operators) Further derivation operators are defined as follows: for $\{i, j, k\} = \{1, 2, 3\}$, $X_i \subseteq K_i$, $X_j \subseteq K_j$ and $A_k \subseteq K_k$, the (i, j, A_k)-derivation operators are defined by:

$$\Psi : X_i \rightarrow X_i^{(i,j,A_k)} : \{a_j \in K_j \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_i, a_k) \in X_i \times A_k\}$$
$$\Psi^{'} : X_j \rightarrow X_j^{(i,j,A_k)} : \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_j, a_k) \in X_j \times A_k\}$$

This definition yields the derivation operators of dyadic contexts defined by

$$\mathbb{K}^{ij}_{A_k} = \langle K_i, K_j, Y^{ij}_{A_k} \rangle$$
$$\text{where } (a_i, a_j) \in Y^{ij}_{A_k} \iff a_i, a_j, a_k \text{ are related by } Y \text{ for all } a_k \in A_k$$

Example Consider $i = 1$, $j = 2$ and $k = 3$, i.e. $K_1 = G$, $K_2 = M$ and $K_3 = B$, $A_3 = \{b_1, b_2\}$ and $X = \{g_3\}$:

$$\Psi(X) = \{m_2, m_3\} \qquad \Psi'\Psi(X) = \{g_3, g_4\}$$

Operators Φ and Φ' are called outer operators, a composition of both operators is called outer closure. Operators Ψ and Ψ' are called inner operators, a composition of them is called inner closure.

Definition 5 (Triadic concept formation) A concept having X_1 in its extent can be constructed as follows.

$$\left(X_1^{(1,2,A_3)(1,2,A_3)}, X_1^{(1,2,A_3)}, (X_1^{(1,2,A_3)(1,2,A_3)} \times X_1^{(1,2,A_3)})^{(3)}\right)$$

Example In the previous example, we have $(\{g_3, g_4\}, \{m_2, m_3\}, \{b_1, b_2, b_3\})$.

From a computational point of view, [15] developed the algorithm Trias for extracting frequent triadic concepts, i.e. whose extent, intent and modus cardinalities are higher than user-defined thresholds (see also [16]). Cerf et al. presented a more efficient algorithm called Data-peeler able to handle n-ary relations [10], the formal definitions being given in terms of Polyadic Concept Analysis [37].

3 Problem settings

A numerical dataset is formalized by a many-valued context [12] and we define accordingly (maximal) biclusters of similar values.

Definition 6 (Many-valued context) (G, M, W, I) is called many-valued context, or simply numerical dataset in this paper, with G being a set of objects, M a set of attributes, W the set of attribute values and I a ternary relation defined on $G \times M \times W$. The fact $(g, m, w) \in I$, also written $m(g) = w$, means that "Attribute m takes the value w for the object g".

Example 1 Table 1 is a numerical dataset, or many-valued context, with objects $G = \{g_1, g_2, g_3, g_4\}$, attributes $M = \{m_1, m_2, m_3, m_4, m_5\}$, attribute values $W = \{0, 1, 2, 6, 7, 8, 9\}$ and for example $m_5(g_2) = 6$.

Definition 7 (Bicluster) In a numerical dataset (G, M, W, I), a bicluster is a tuple (A, B) with $A \subseteq G$ and $B \subseteq M$.

Definition 8 (Similarity relation and bicluster of similar values) Let $w_1, w_2 \in W$ be two attribute values and $\theta \in \mathbb{R}$ be a user-defined parameter, called *similarity parameter* or *threshold*. w_1 and w_2 are said to be similar iff $|w_1 - w_2| \leq \theta$, which we denote by $w_1 \simeq_\theta w_2$. (A, B) is bicluster of similar values if $m(g) \simeq_\theta n(h)$ for all $g, h \in A$ and for all $m, n \in B$.

Definition 9 (Maximal bicluster of similar values) A bicluster of similar values (A, B) is maximal if adding either an object in A or an attribute in B does not result in a bicluster of similar values.

Example 2 (From Table 1) $(\{g_1, g_4\}, \{m_2, m_4\})$ is a bicluster. $(\{g_1, g_2\}, \{m_2\})$ is a bicluster of similar values with $\theta \geq 1$. However, it is not maximal. With $1 \leq \theta < 5$, $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\})$ is maximal. Finally, with $\theta = 7$ the bicluster $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3, m_4, m_5\})$ is maximal. Note that a constant (maximal) bicluster is a (maximal) bicluster of similar values with $\theta = 0$.

Thus the problem that we address in this article is the extraction of all maximal biclusters of similar values from a numerical dataset. We desire the extraction to be complete, correct and non-redundant compared to most of existing methods of the literature based on heuristics [26]. We will show that FCA is a good candidate as a formal framework for such a task.

4 Biclusters of similar values in triadic concept analysis

This first contribution considers the problem of generating maximal biclusters for any θ with TCA after a scaling procedure. We then show how to represent the resulting set of concepts with line diagrams, and extend the methodology to n-dimensional numerical datasets.

4.1 Scaling numerical data into a triadic context

Starting from a numerical dataset (G, M, W, I), the basic idea lies in building a triadic context (G, M, T, Y) where the two first dimensions remain formal objects and formal attributes, while W is scaled into a third dimension denoted by T. This new dimension T is called the *scale dimension*: intuitively, it gives different "spaces of values" that each object-attribute pair $(g, m) \in G \times M$ can take. Once the scale is given, a triadic context is derived and it gives rise to triadic concepts.

We use the *interordinal scaling* [12] to build the scale dimension. It allows one to encode in 2^T all possible intervals of values in W. This scale allows one to derive a triadic context from which any bicluster of similar values can be characterized as a triadic concept. We make these statements more precise and illustrate the whole procedure with examples.

Definition 10 (Interordinal Scaling) A scale is a binary relation $J \subseteq W \times T$ associating original elements from the set of values W to their derived elements in T. In the case of interordinal scaling, $T = \{[min(W), w], \forall w \in W\} \cup \{[w, max(W)], \forall w \in W\}$. Then $(w, t) \in J$ iff $w \in t$.

Example 3 Table 4 gives the tabular representation of the interordinal scale for Table 1. Each row describes a single value, while dyadic concepts represent all possible intervals over W. An example of dyadic concept in this table is given

Table 4 Interordinal scale of the set of attribute values W

J	$t_1 = [0, 0]$	$t_2 = [0, 1]$	$t_3 = [0, 2]$	$t_4 = [0, 6]$	$t_5 = [0, 7]$	$t_6 = [0, 8]$	$t_7 = [0, 9]$	$t_8 = [1, 9]$	$t_9 = [2, 9]$	$t_{10} = [6, 9]$	$t_{11} = [7, 9]$	$t_{12} = [8, 9]$	$t_{13} = [9, 9]$
0	×	×	×	×	×	×	×						
1		×	×	×	×	×	×	×					
2			×	×	×	×	×	×	×				
6				×	×	×	×	×	×	×			
7					×	×	×	×	×	×	×		
8						×	×	×	×	×	×	×	
9							×	×	×	×	×	×	×

by $(\{6, 7, 8\}, \{t_6, t_7, t_8, t_9, t_{10}\})$, rewritten as $(\{6, 7, 8\}, \{[6, 8]\})$ since $\{t_6, t_7, t_8, t_9, t_{10}\}$ represents the interval $[0, 8] \cap [0, 9] \cap [1, 9] \cap [2, 9] \cap [6, 9] = [6, 8]$.

Definition 11 (Triadic scaled context) Let Y be a ternary relation $Y \subseteq G \times M \times T$. Then $(g, m, t) \in Y$ iff $(m(g), t) \in J$, or simply $m(g) \in t$. We call the tuple (G, M, T, Y) the triadic scaled context of the numerical dataset (G, M, W, I).

Example 4 The object-attribute pair (g_1, m_1) taking value $m_1(g_1) = 1$ is scaled into triples $(g_1, m_1, t) \in Y$, where t takes any interval in $\{[0, 1], [0, 2], [0, 6], [0, 7], [0, 8], [0, 9], [1, 9]\}$. The intersection of intervals in this set is the original value itself, i.e. $m_1(g_1) = 1$, a basic property of interordinal scaling. As a result, Table 5 illustrates the whole scaled triadic context derived from the numerical dataset given in Table 1 using interordinal scaling. The very first cross ($\times$) in this table (upper left) represents the tuple (g_2, m_4, t_1), meaning that $m_4(g_2) \in [0, 0]$.

We present now our first main result: there is a one-to-one correspondence between (i) the set of maximal biclusters of similar values in a given numerical dataset for any similarity parameter θ and (ii) the set of all triadic concepts in the triadic context derived with interordinal scaling. Consider first the following definition and notations.

Definition 12 (Standard order of interordinal scale attributes) The values of the interordinal scale are intervals. Define the *standard order* on $2k - 1$ attributes of the interordinal scale based on k first natural numbers as follows: $[1, 1], [1, 2], \ldots, [1, k], [2, k], \ldots, [k, k]$. Having the standard order on the attributes of the interordinal scale one can think of attributes having numbers from 1 to $2k + 1$. Note the obvious *main property of the standard order on attributes of the interordinal scale*: if an object has two scale attributes with numbers r and s, $r < s$, then it has all scale attributes with numbers in $[r, s]$.

For a many-valued context (G, M, W, I), let the set W ($|W| = q$) be the set of numerical values enumerated in the ascending order from 1 to q, and let $g(m)$ be a map taking attribute m to its value $w \in W$ for object g. Let the numerical values from W be interordinally scaled with the standard order on the scale attributes, so we can denote the scale attributes by $m_1, \ldots, m_q, \ldots, m_{2q-1}$. Let $B = \{m_1, \ldots, m_q, \ldots m_{2q-1}\}$ and (G, M, B, Y) be the triadic context such that $(g, m, b) \in Y$ iff $g(m)$ lies in the interval given by the interordinal attribute b.

Proposition 1 *(A, D) is a maximal bicluster of similar values $(A \subseteq G,\ D \subseteq M)$ with the values lying in the interval $[t, t + \theta]$ for $t \in \mathbb{N}, \theta \geq 0$ iff (A, D, U) is a triadic concept of the context (G, M, B, Y), where $U = \{t + \theta, \ldots q, \ldots, q + t - 1\}$. Moreover, every triadic concept of the interordinally scaled triadic context (G, M, B, Y) is of the form (A, D, U), where $A \subseteq G$, $D \subseteq M$, and $U = \{t + \theta, \ldots q, \ldots, q + t - 1\}$ for some $t \in \mathbb{N}$ and $\theta \geq 0$.*

Proof Let (A, D) be a maximal bicluster of similar values $(A \subseteq G,\ D \subseteq M)$, then the values of attributes of the bicluster are lying in the interval $[t, t + \theta]$ for some $t \in \mathbb{N}, \theta \geq 0$, i.e. $g(m) \in [t, t + \theta]$ for every $g \in A$, $m \in D$. Due to the standard order on interordinal attributes, this implies that in the triadic context (G, M, B, Y)

Table 5 Triadic scaled context from Table 1 with interordinal scaling

	$t_1 = [0, 0]$					$t_2 = [0, 1]$					$t_3 = [0, 2]$					$t_4 = [0, 6]$					$t_5 = [0, 7]$				
	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5
g_1						×			×		×	×	×	×		×	×	×	×	×	×	×	×	×	×
g_2				×			×	×	×		×	×	×	×		×	×	×	×	×	×	×	×	×	×
g_3								×			×	×	×			×	×	×		×	×	×	×	×	×
g_4													×					×	×				×	×	×

	$t_6 = [0, 8]$					$t_7 = [0, 9]$					$t_8 = [1, 9]$					$t_9 = [2, 9]$					$t_{10} = [6, 9]$				
	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5
g_1	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×		×	×		×					×
g_2	×	×	×	×	×	×	×	×	×	×	×	×	×		×	×				×					×
g_3	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×				×	×
g_4	×		×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×	×		×	×

	$t_{11} = [7, 9]$					$t_{12} = [8, 9]$					$t_{13} = [9, 9]$				
	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5	m_1	m_2	m_3	m_4	m_5
g_1															
g_2															
g_3				×											
g_4	×	×			×	×	×					×			

In gray lies the triadic concept $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\}, \{t_3, t_4, t_5, t_6, t_7, t_8\})$ corresponding to bicluster $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\})$ with values in $[1, 2]$ and thus maximal for $\theta = 2 - 1 = 1$

one has $(g, m, b) \in Y$ for all $g \in A, m \in D$ and $b \in \{t+\theta, \ldots q, \ldots, q+t-1\}$ and there is a rectangular parallelepiped $(A, D, \{t+\theta, \ldots q, \ldots, q+t-1\})$ filled with crosses in the triadic cross-table of Y, i.e. $(A, D, \{t+\theta, \ldots q, \ldots, q+t-1\} \subseteq Y$. This parallelepiped is inclusion-maximal, since otherwise this would mean that one can add either another object, or another attribute, or another scale value to its respective component. The possibility of adding another object or attribute would contradict the fact that (A, D) is a maximal bicluster, the possibility of adding another scale value would contradict the fact that the attribute values of the bicluster lie strictly in the interval $[t, t+\theta]$. Thus, $(A, D, \{t+\theta, \ldots q, \ldots, q+t-1\})$ is a triadic concept.

In the opposite direction, consider a triadic concept (A, D, V) in the interordinally scaled three-dimensional context, the attributes of V being ordered in the standard way. By the main property of the standard order on attributes of the interordinal scale, this would mean that for any two values r and s of V, the set V also contains all values in the interval $[r, s]$. Hence there are some t and q such that the values of V lie in the interval $[t, t+\theta]$ for all object-attribute pairs from $A \times D$. This means that (A, D) is a bicluster of similar values, which is maximal, since otherwise (A, D, V) would not have been a triadic concept. □

Example 4 $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\}, \{t_3, t_4, t_5, t_6, t_7, t_8\})$ is a triadic concept corresponding to the maximal bicluster $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\})$ with $\theta = 1$ since $\{t_3, t_4, t_5, t_6, t_7, t_8\}$ is a modus characterizing interval $[1, 2]$ of length 1.

4.2 Trilattice diagram

In their seminal paper on TCA, Lehman and Wille proposed a way to visualize the ordered structure of triadic concepts [25]. This visualization possibility has not attracted a lot of attention since, hence we propose to illustrate it with derived triadic contexts from numerical data. Let us firstly recall notations of TCA: a triadic context is denoted by $\mathbb{K} = (K_1, K_2, K_3, Y)$, the set of all its triadic concepts by $\mathcal{I}(\mathbb{K})$ and its corresponding triadic diagram by $\underline{\mathcal{I}}(\mathbb{K})$.

Definition 13 (Quasi-order $\lesssim_i$ and equivalence relation $\sim_i$ on $\mathcal{I}(\mathbb{K})$) Given two triadic concepts (A_1, A_2, A_3) and (B_1, B_2, B_3), three quasi-order and three equivalence are defined as follows, for $i = 1, 2, 3$

$$(A_1, A_2, A_3) \lesssim_i (B_1, B_2, B_3) \iff A_i \subseteq B_i, \tag{1}$$

$$(A_1, A_2, A_3) \sim_i (B_1, B_2, B_3) \iff A_i = B_i. \tag{2}$$

Definition 14 (Anti-ordinal dependencies) A triadic concept is uniquely determined by two of its components since the three quasi-orders satisfy the anti-ordinal dependencies: For $\{i, j, k\} = \{1, 2, 3\}$, $(A_1, A_2, A_3) \lesssim_i (B_1, B_2, B_3)$ and $(A_1, A_2, A_3) \lesssim_j (B_1, B_2, B_3)$ imply $(A_1, A_2, A_3) \gtrsim_k (B_1, B_2, B_3)$ for any two concepts (A_1, A_2, A_3) and (B_1, B_2, B_3).

Definition 15 (Equivalence and factor sets) For $i = 1, 2, 3$, the equivalence class of the relation $\sim_i$ which contains the concept (A_1, A_2, A_3) is denoted by $[(A_1, A_2, A_3)]_i$. $\lesssim_i$ induces an order $\leq_i$ on the factor set $\mathcal{I}(\mathbb{K})/\sim_i$:

$$[(A_1, A_2, A_3)]_i \leq_i [(B_1, B_2, B_3)]_i \iff A_i \subseteq B_i.$$

Accordingly, $(\mathcal{I}(\mathbb{K})/\sim_i, \leq_i)$ is the ordered set of all extents ($i = 1$), or intents ($i = 2$) and modi ($i = 3$) of $\mathbb{K}$.

Definition 16 (Triadic diagram) This relational structure $\underline{\mathcal{I}}(\mathbb{K})$ can be understood as two types of structures:

- The geometric structure: $(\mathcal{I}(\mathbb{K}), \sim_1, \sim_2, \sim_3)$: It is represented as a partial 3-net, i.e. a triangular pattern. The three equivalence relations are here represented by 3 systems of parallel lines. For example, consider the equivalence relation on concepts with $i = 1$: concepts of an equivalence class have same extent and are depicted on the same line. As such, the classes of equivalence meet at most in one element for a given concept.
- The ordered structures: $(\mathcal{I}(\mathbb{K})/\sim_i, \leq_i)$: Each of them is represented by a Hasse diagram.

Figure 2 presents the trilattice obtained from our running example (i.e. Table 5). For sake of readability, we highlight there only the biclusters that are maximal for $\theta = 1$. Taking the concept $(\{g_4\}, \{m_1, m_5\}, [6, 11])$ from the Table 6, the three (pairwise non parallel) lines, corresponding respectively to the equivalence class of the extent $\{g_4\}$, the intent $\{m_1, m_5\}$ and the modus $[6, 11]$, only meet in one point of

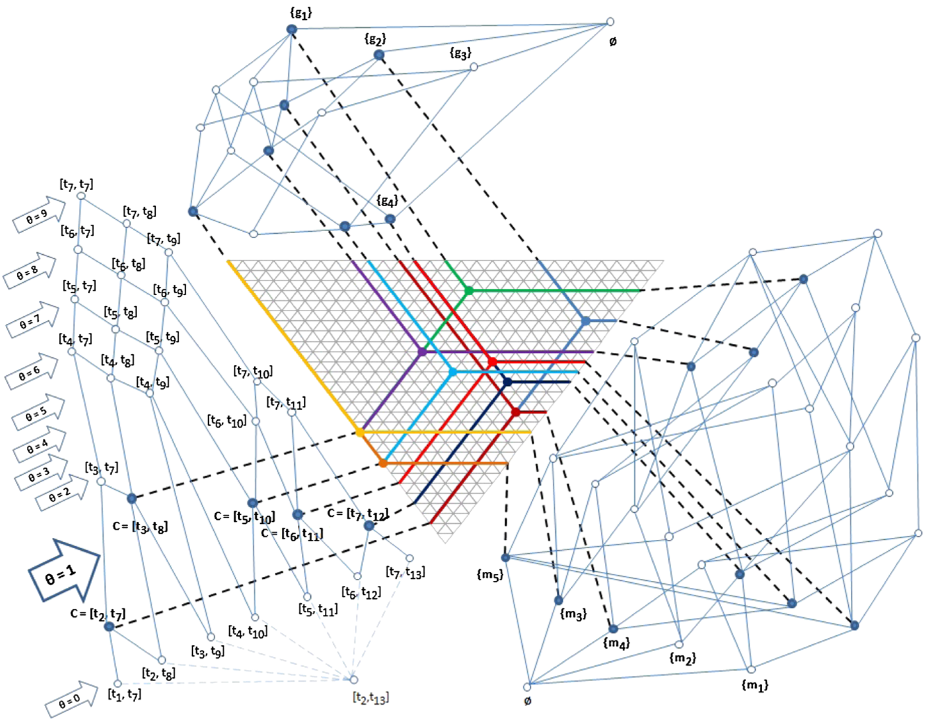


Fig. 2 Trilattice from multi-valued context (Table 1) interordinally scaled to Table 5. Note, that only biclusters maximal for $\theta = 1$ are depicted. Consider the purple point that is the meet of the three purple lines: it represents the concept $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\}, \{[t_3, t_8]\})$, i.e. with values in $[1, 2]$

Table 6 Triadic concepts with $\theta = 1$

$G \supseteq A$ – extent	$M \supseteq B$ – intent	$T \supseteq C$ – modus	Interval over W
$A = \{g_1\}$	$B = \{m_1, m_2, m_3, m_4\}$	$C = [t_3, t_8]$	$[1, 2]$
$A = \{g_1, g_2\}$	$B = \{m_4\}$	$C = [t_2, t_7]$	$[0, 1]$
$A = \{g_1, g_2, g_3\}$	$B = \{m_1, m_2, m_3\}$	$C = [t_3, t_8]$	$[1, 2]$
$A = \{g_1, g_2, g_3, g_4\}$	$B = \{m_3\}$	$C = [t_3, t_8]$	$[1, 2]$
$A = \{g_1, g_2, g_3, g_4\}$	$B = \{m_5\}$	$C = [t_5, t_{10}]$	$[6, 7]$
$A = \{g_2\}$	$B = \{m_2, m_3, m_4\}$	$C = [t_2, t_7]$	$[0, 1]$
$A = \{g_3, g_4\}$	$B = \{m_4, m_5\}$	$C = [t_5, t_{10}]$	$[6, 7]$
$A = \{g_4\}$	$B = \{m_1, m_2\}$	$C = [t_7, t_{12}]$	$[8, 9]$
$A = \{g_4\}$	$B = \{m_1, m_5\}$	$C = [t_6, t_{11}]$	$[7, 8]$

the triangular pattern which represents this concept. The three quasi-order structures of extents, intents and modi, i.e. Hasse diagrams of all $(\mathcal{I}(\mathbb{K})/\sim_i, \leq_i)$, lie around the trilattice.

4.3 Handling n-ary numerical dataset

A straightforward generalization of the presented approach lies in its potential extension to n-ary numerical datasets. The basic idea is as follows. Consider a numerical dataset with n dimensions, e.g. *genes* $\times$ *biological situations* $\times$ *timestamps* when $n = 3$. Then, one can extract n-clusters of similar values by scaling the numerical data into a $n+1$-dimensional binary dataset. So-called polyadic concepts [37] in the binary dataset are here again in 1-to-1-correspondence with maximal n-clusters of similar values of the numerical dataset. We present here theoretical aspects while computing aspects can be regarded with the existing algorithms Data-Peeler [10].

Recall that the *standard order* on $2k-1$ attributes of the interordinal scale is as follows: $[v_1, v_1], [v_1, v_2], \ldots, [v_1, v_k], [v_2, v_k] \ldots, [v_k, v_k]$. Having the standard order on the attributes of the interordinal scale one can enumerate them from 1 to $2k+1$. Let $(G_1, \ldots, G_n, W, I)$ be an n-dimensional many-valued context, i.e., an $n+1$-dimensional relation $I \subseteq G_1 \times \ldots \times G_n \times W$ and W $(|W| = q)$ be the set of numerical values enumerated in the ascending order from 1 to q, and let $v(g_1, \ldots, g_n)$ be a map taking the tuple $g_1, \ldots, g_n$ to the value $w \in W$. Let the numerical values from W be interordinally scaled with the standard order on the scale attributes, so we can denote the scale attributes by $m_1, \ldots, m_q, \ldots, m_{2q-1}$. Let $B = \{m_1, \ldots, m_q, \ldots m_{2q-1}\}$ and $Y \subseteq G_1 \times \ldots \times G_n \times B$ be an $n+1$-ary relation such that $(g_1, \ldots, g_n, m) \in Y$ iff the value w of the n-tuple $g_1, \ldots, g_n$ lies in the interval given by the interordinal attribute m.

Proposition 2 *$(A_1, \ldots, A_n)$ is a maximal n-way cluster of similar values $(A_i \subseteq G_i)$ with the values lying in the interval $[t, t+\theta]$ for $t \in \mathbb{N}, \theta \geq 0$ iff $(A_1, \ldots, A_n, U)$ is an $n+1$-adic concept of the $n+1$-dimensional context $(G_1, \ldots, G_n, U, Y)$, where $U = \{t+\theta, \ldots q, \ldots, q+t-1\}$. Moreover, every $n+1$-dimensional concept of the interordinally scaled $n+1$-dimensional context $(G_1, \ldots, G_n, W, Y)$ is of the form $(A_1, \ldots, A_n, U)$, where $A_i \subseteq G_i$, and $U = \{t+\theta, \ldots q, \ldots, q+t-1\}$ for some $t \in \mathbb{N}$ and $\theta \geq 0$.*

The proof is similar as in the triadic case and hence is omitted.

4.4 Remarks

We showed that extracting biclusters of similar values for any θ in a numerical dataset can be achieved by (i) scaling the attribute value dimension and (ii) extracting the triadic concepts in the resulting derived triadic context. The same applies when considering n-ary numerical datasets.

On the one hand, triadic concepts (A, B, U) with the largest sets A, B or C represent large biclusters of similar values. Indeed, the larger $|A|$ and $|B|$ the larger the data covering of the corresponding bicluster. Furthermore, the larger $|U|$, the more similar values for bicluster (A, B). Indeed, by the properties of interordinal scaling, the more intervals in U, the smaller their interval intersection. Mining so-called top-k frequent triadic concepts can accordingly be achieved with the existing algorithm Data-Peeler [10].

On the other hand, extracting maximal biclusters for all θ may be neither efficient nor effective with large numerical data: their number tends to be very large and not all biclusters are relevant for a given analysis. Furthermore, both size and density of contexts derived with interordinal scaling are known to be problematic w.r.t algorithmic scalability, see e.g. [20]. In existing methods of the literature, θ is set *a priori*. We show now how to handle this case with slight modifications, this is our second main result.

5 Extracting biclusters of similar values for a given θ

In this section, we present our second contribution. We consider the problem of extracting maximal biclusters of similar values in TCA for a given θ only. It comes with slight modifications of the methodology presented in the previous section, but requires more algorithmic considerations: although all triadic concepts correspond to biclusters of similar values with a new transformation procedure, it is not sure that such concepts correspond to maximal biclusters. In this way, it is not possible to use concepts extraction algorithms directly (or it would require post-processing which is always a solution to avoid). Accordingly, a modified scaling procedure will lead us to the design of the algorithm TriMax for a complete and correct extraction of maximal biclusters for a given θ. Finally, we experiment with this new algorithm.

5.1 Scaling numerical data in a triadic context

Consider the previous scaling applied to a numerical dataset (G, M, W, I). It scales W into a dimension T and all subsets of T characterize all intervals of values over W. To get maximal biclusters for a given θ only, we should not consider all possible intervals in W, but rather all intervals (i) having a range size that is less or equal than θ to avoid biclusters with non similar values, and (ii) having a range size the closest as possible to θ to avoid non-maximal biclusters. For example, if we set $\theta = 2$, it is probably not interesting to consider interval $[0, 8]$ in the scale dimension since $8 - 0 > \theta$. Similarly, considering the interval $[6, 6]$ may not be interesting as well, since a bicluster with all its values equal to 6 may not be maximal. As introduced in [17], the maximal intervals of similar values used for the scale are called *blocks* *of tolerance* over the set of numbers W with respect to the tolerance relation $\simeq_\theta$. We now recall basics on tolerance relations over a set of numbers. This allows us to define a simpler scaling procedure. The resulting triadic context is then mined with a new TCA algorithm called TriMax to extract maximal biclusters of similar values for a given θ.

Blocks of tolerance over W are as maximal sets of pairwise similar values:

Definition 17 (Tolerance relation and blocks) A binary relation $\simeq$ is a tolerance if it is reflexive, symmetric but not necessarily transitive. Given a set W, a subset $V \subseteq W$, and a tolerance relation $\simeq$ over W, V is a *block of tolerance* if:

(i) $\forall w_1, w_2 \in V,\ w_1 \simeq w_2$ (pairwise similarity)

(ii) $\forall w_1 \notin V, \exists w_2 \in V,\ w_1 \not\simeq w_2$ (maximality).

It follows that $\simeq_\theta$ is a tolerance relation. From Table 1 we have $W = \{0, 1, 2, 6, 7, 8, 9\}$. With $\theta = 2$, one has $0 \simeq_2 2$ but $2 \not\simeq_2 6$. Accordingly, one obtains 3 blocks of tolerance, namely the sets $\{0, 1, 2\}$, $\{6, 7, 8\}$ and $\{7, 8, 9\}$. These three sets can be renamed as the convex hull of their elements on $\mathbb{N}$: respectively, $[0, 2]$, $[6, 8]$ and $[7, 9]$: any number lying between the minimal and the maximal elements (w.r.t. natural number ordering) of a block of tolerance is naturally similar to any other element of the block. Then, to derive a triadic context from a numerical dataset, we simply use tolerance blocks over W to define the scale dimension.

Definition 18 (TriMax scale relation) The scale relation is a binary relation $J \subseteq W \times C$, where C is the set of blocks of tolerance over W renamed as their convex hulls. Then, $(w, c) \in J$ iff $w \in c$.

Example 6 From Table 1 we have: $C = \{[0, 1], [1, 2], [6, 7], [7, 8], [8, 9]\}$ with $\theta = 1$, and $C = \{[0, 2], [6, 8], [7, 9]\}$ with $\theta = 2$.

In this way, we can apply the same context derivation as in the previous section: scaling is still based on intervals, but this time it uses tolerance blocks.

Definition 19 (TriMax triadic scaled context) Let $Y \subseteq G \times M \times C$ be a ternary relation. Then $(g, m, c) \in Y$ iff $(m(g), c) \in J$, or simply $m(g) \in c$, where J is the scale relation. (G, M, C, Y) is called the TriMax triadic scaled context.

Example 7 Table 7 is the TriMax triadic scaled context derived from the numerical dataset lying in Table 1 with $\theta = 1$.

Definition 20 (Dyadic context associated with a block of tolerance) Consider a block of tolerance $c \in C$. The dyadic context associated with this block is given by (G, M, Z) where Z denotes the set of all $(g, m) \in G \times M$ such that $m(g) \in c$.

Table 7 Triadic scaled context using tolerance blocks over W and $\theta = 1$ (empty columns are not displayed)

	[0, 1]				[1, 2]				[6, 7]		[7, 8]			[8, 9]	
	m_1	m_2	m_3	m_4	m_1	m_2	m_3	m_4	m_4	m_5	m_1	m_4	m_5	m_1	m_2
g_1	×			×	×	×	×	×		×					
g_2		×	×	×	×	×	×			×					
g_3			×		×	×	×		×	×		×			
g_4							×		×	×	×		×	×	×

In gray, the bicluster $(\{g_1, g_2, g_3\}, \{m_1, m_2, m_3\})$ with values in [1, 2] and maximal for $\theta = 1$ corresponds to a dyadic concept in the dyadic context labeled [1, 2]

Example 8 In Table 7, each dyadic context is labeled by its corresponding block of tolerance for $\theta = 1$.

Now, note that blocks of tolerance over W are totally ordered: let $[v_1, v_2]$ and $[w_1, w_2]$ be two blocks of tolerance, one has $[v_1, v_2] < [w_1, w_2]$ iff $v_1 < w_1$. Hence, associated dyadic contexts are also totally ordered and we can use an indexing set to label them (as done in the algorithm pseudo-code later).

We now present our next results: the scaled triadic context supports the extraction of maximal biclusters of similar values for a given θ. In this case however, existing algorithms of TCA cannot be applied directly. For example, in Table 7, the triadic concept $(\{g_3\}, \{m_4\}, \{[6, 7], [7, 8]\})$ corresponds to a bicluster of similar values which is not maximal. Hence we present hereafter a new TCA algorithm for this task, called TriMax.

The basic idea of TriMax relies on the following facts. Firstly, since each dyadic context corresponds to a block of tolerance, we do not need to compute intersections of contexts, such as classically done in TCA. Hence each dyadic context is processed separately. Secondly, a dyadic concept of a dyadic context necessarily represents a bicluster of similar values, but we cannot be sure it is maximal (see previous example). Hence, we need to check if a concept is still a concept in other dyadic contexts, corresponding to other classes of tolerance. This is made precise with the following proposition.

Proposition 3 *Let (A, B, U) be a triadic concept from* TriMax *triadic scaled context (G, M, C, Y), such that U is the outer closure of a singleton $\{c\} \subseteq C$. If $|U| = 1$, (A, B) is a maximal bicluster of similar values. Otherwise, (A, B) is a maximal bicluster of similar values iff there is no $\in [min(U); max(U)]$, $y < c$ such that $(A, B) \neq \Psi'_y(\Psi_y((A, B)))$, where $\Psi'_y(\cdot)$ and $\Psi_y(\cdot)$ correspond to inner derivation operators associated with y^{th} dyadic context.*

Proof When $|U| = 1$, (A, B) is a dyadic concept only in one dyadic context corresponding to a block of tolerance. By properties of tolerance blocks, (A, B) is a maximal bicluster. If $|U| \neq 1$, (A, B) is a dyadic concept in $|U|$ dyadic contexts. Since the tolerance block set is totally ordered, it directly implies that modus U is the interval $[min(U); max(U)]$. Hence, if there is $y \in [min(U); max(U)]$ such that $(A, B) = \Psi'_y(\Psi_y((A, B)))$, then (A, B) is not a maximal bicluster of similar values. □

5.2 The TriMax algorithm

TriMax starts with scaling initial numerical data into several dyadic contexts, each one standing for a block of tolerance over W with given θ. The set of all dyadic contexts forms accordingly a triadic context. Then, each dyadic context is mined with any FCA algorithm (or closed itemset mining algorithm), and all formal concepts are extracted. For a given concept (A, B), we compute outer derivation $\Phi^{'}((A, B))$, i.e. to obtain the set of dyadic contexts labels in which the current dyadic concept holds. If this set is a singleton, this means that (A, B) is a concept for the current block of tolerance only, i.e. it is a maximal bicluster of similar values, and it has been, or will never be, generated twice. Otherwise, (A, B) is a concept in other contexts, and can be generated accordingly several times (as much as the number of contexts in which it holds). Then, we only consider (A, B) if we are sure it is the last time it is computed. Finally, we need to check if current concept represents a maximal bicluster, i.e. there should not exist a context labeled by an element of the modus where (A, B) is not a dyadic concept.

Algorithm 1 TriMax

```
input  : Numerical dataset (G, M, W, I), tolerance parameter θ
output: Maximal biclusters of similar values
Let C = {[a_i, b_i]} be the totally ordered set of all blocks over W for given θ. Indices
i form an indexing set.
forall the [a_i, b_i] ∈ C do
    Build context (G, M, Z_i) such that (g, m) ∈ Z_i ⇔ m(g) ∈ [a_i, b_i]
forall the (G, M, Z_i) do
    Use any FCA algorithm to extract all its concepts (A, B)
    forall the dyadic concepts (A, B) in the current context (G, M, Z_i) do
        if |Φ'((A, B))| = 1 then
            print (A, B)
        else if max(Φ'((A, B)) = i then
            x ← min(Φ'((A, B))
            if ∄y ∈ [x, i[ s.t. (A, B) ≠ Ψ'_y(Ψ_y((A, B))) then
                print (A, B)
```

Proposition 4 TriMax *outputs a* (*i*) *complete,* (*ii*) *correct and* (*iii*) *non redundant collection of all maximal biclusters of similar values for a given numerical dataset and similarity parameter* θ.

Proof (i) and (ii) follow directly from Proposition 3. Statement (iii) is ensured by the second *if* condition of the algorithm: a dyadic concept (or equivalently bicluster) is considered iff it has been extracted in the last dyadic context in which it holds. □

5.3 Experimenting with TriMax

In this section, we present experiments carried out with the algorithm TriMax and highlight various aspects of its practical complexity.

Data We explore a gene expression dataset of the species *Laccaria bicolor* available at NCBI.[2] More details on this dataset can be found in [20]. This gene expression dataset monitors the behaviour of 11,930 genes in 12 biological situations, reflecting various stages of *Laccaria bicolor* biological cycle. Attribute values in W vary between 0 and 60,000.

TRIMAX *implementation* TRIMAX is written in C++. It uses the BOOST library 1.42 for data structures and INCLOSE,[3] an implementation of the algorithm CLOSEBYONE [23] for dyadic concept extraction. At each iteration of the main loop, i.e. each tolerance block, the current scaled dyadic context is produced: We do not generate the whole triadic context which cannot fit into memory for large databases. It turns out that the modus computation for a given dyadic concept requires to compute scaling "on the fly", i.e. when computing the set of dyadic contexts in which a current concept holds. The experiments were carried out on an Intel CPU 2.54 Ghz machine with 8 GB RAM running under Ubuntu 11.04.

Experiment settings The goal of the present experiments is not to give a qualitative evaluation of the present approach (say biological interpretation), but rather a quantitative evaluation in terms of computational efficiency. Indeed, the present work aims at showing how an existing type of biclusters can be mined with Triadic Concept Analysis. For a qualitative evaluation, the reader may refer e.g. to [6, 20].

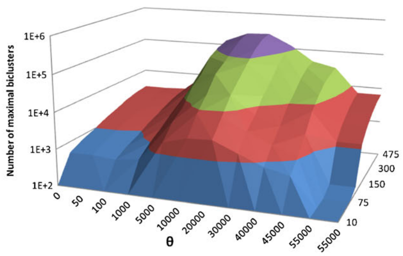
Accordingly, we designed the following experiments to monitor various aspects of the TRIMAX algorithm. For most of the experiments, the dataset used is composed of an increasing number of objects and all attributes. The objects are chosen randomly once and for all so that the different experiment results can be compared. We also vary the parameter θ in the same way across all experiments. Then, we monitor the following aspects, as presented in Fig. 3:

- i Number of maximal biclusters of similar values
- ii Execution time (in seconds)
- iii Number of tolerance blocks
- iv Density of the triadic context, where density is defined as $d(G, M, C, Y) = |Y|/(|G| \times |M| \times |C|)$. This information is important, since contexts with high density are known to be hard to process with FCA algorithms [24].
- v Comparison between the number of non-maximal biclusters produced by TRIMAX (i.e. dyadic concepts that do not correspond to maximal biclusters) with the number of maximal biclusters.
- vi Execution time profiling of the main procedures of TRIMAX. This is achieved with the tool GNU GPROF and gives us which parts of the algorithm are the most time consuming.

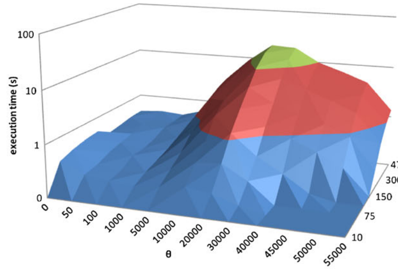
Experiment results Figure 3 presents the results of our experiments with different settings. In these settings, we vary the number of objects $|G|$ and the parameter θ. A first observation arises from graph (i): the number of biclusters is the highest when $\theta \simeq 30{,}000$. A first explanation is that 30,000 is the half of the maximal value of W

[2] http://www.ncbi.nlm.nih.gov/geo/ as series GSE9784.

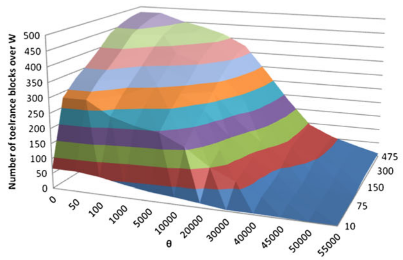
[3] http://sourceforge.net/projects/inclose/



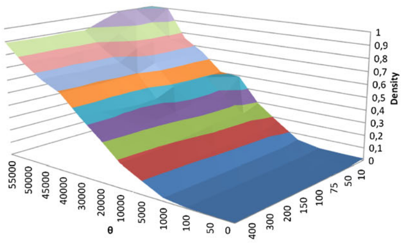
(i) Numbers of patterns (Y-axis) w.r.t. θ (X-axis) and $|G|$ (Z-axis)



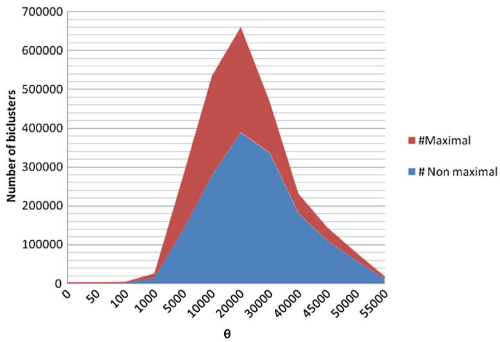
(ii) Execution times in seconds (Y-axis) w.r.t. θ (X-axis) and $|G|$ (Z-axis)



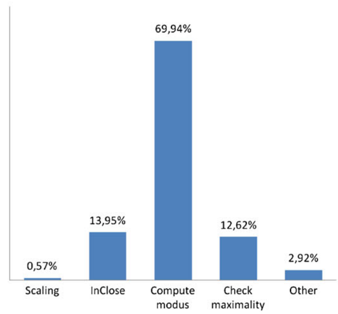
(iii) Numbers of blocks of tolerance (Y-axis) w.r.t. θ (X-axis) and $|G|$ (Z-axis)



(iv) Density of triadic contexts (Y-axis) w.r.t. θ (X-axis) and $|G|$ (Z-axis)



(v) Comparing the number of generated dyadic concepts w.r.t. the actual number of maximal biclusters varying θ with $|G| = 500$



(vi) Repartition of execution time w.r.t main steps of TriMax with $\theta = 33,000$ and $|G| = 500$

Fig. 3 Monitoring with different settings (**i**) the number of maximal biclusters, (**ii**) the execution times of TriMax, (**iii**) the number of tolerance blocks, (**iv**) the derived triadic context density, (**v**) the number of non-maximal biclusters generated as dyadic-concepts w.r.t. the number of maximal biclusters, and (**vi**) repartition of execution time in the TriMax algorithm

and almost all multiples of 100 in [0; 60,000] belong to W. In the figure (ii), execution time has the same behavior as in the figure (i). This fact can be understood by paying attention to the next figures (iii) and (iv). In (iii) the number of tolerance blocks is monitored. The maximal number is reached when $\theta = 0$, i.e. $|C| = |W|$. When $\theta = max(W)$, we have $|C| = 1$. Now we observe in (iv) that the density follows a reverse behavior: When $\theta = 0$, the density tends towards 0 %; when $\theta = max(W)$, then density equals exactly 1 %. Combining both graph (iii) and (iv), the worst cases happen when both density and tolerance block count are high.

Another observation, which explains also the execution times, arises from graph (v). Here the number of maximal biclusters and the number of non-maximal biclusters generated as dyadic concepts are compared. Here again, the worst case is reached when $\theta \simeq 30{,}000$. Looking at figure (vi), we learn that this is however not the major problem. The mostly consuming procedure of TriMax is the computation of the modus of a dyadic concept. The explanation is that we compute modus with “on the fly scaling”.

Therefore, the bottleneck of our algorithm appears to be the modus computation. In practical applications however, the analyst is not interested in all biclusters of similar values. Some constraints are generally defined, such as a minimal (resp. maximal) number of objects (resp. attributes) in a bicluster (A, B), or a minimal area $|A| \times |B|$, etc. Interestingly, most of those constraints can be evaluated on a generated dyadic concept. Therefore, before computing the modus of such concept, we can check such properties and discard the concept if it does not respect the constraints. Although not reflected in this paper, we tested how adding minimal (resp. maximal) size constraints on a bicluster affects both the number of biclusters and the execution times. The results are very interesting: for example with $\theta = 33{,}000$, $|G| = 500$, and minimal (resp. maximal) size for $|A|$ set to 10 (resp. 40), TriMax produces only 5,332 maximal biclusters in 2.1 s compared to 104,226 maximal biclusters extracted in 16.130 s without any constraint.

Finally, the most interesting aspect of TriMax is the possibility of its distributed execution. Indeed, each iteration, i.e. for each block of tolerance, can be achieved independently from the others. Furthermore, the core of TriMax consisting in extracting dyadic contexts can also be distributed, see e.g. [22]. A deeper investigation remains to be done in this case. Note that although the method description involves W as a set of natural numbers, TriMax can directly handle numerical data with real (floating point) numbers (since W is a finite set).

Comparison with existing methods Two methods in the literature also consider the problem of extracting all maximal biclusters of similar values from a numerical dataset. The first method is called *Numerical Biset Miner* (NBS-Miner [6]). The second method is based on *interval pattern structures* (IPS [19]). We compared the execution times of NBS-Miner, IPS and TriMax. Algorithms have been implemented in C++. Figure 4 display three experiments showing that NBS-Miner is not

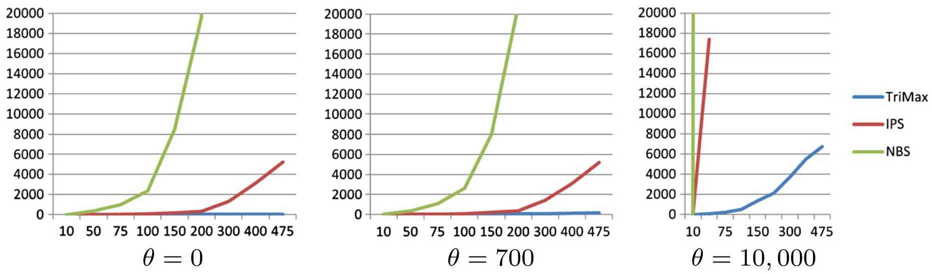


Fig. 4 Comparing performance of TriMax, NBS and the IPS approach (ms)

scalable compared to IPS and TriMax. Although TriMax outperforms IPS, a deeper investigation is required: the main problem with IPS is to find an efficient algorithm able to compute tolerance blocks over a set of intervals. Other experiments show the same behavior.

6 Conclusion

We addressed the problem of biclustering numerical data with Formal Concept Analysis. So-called (maximal) biclusters of similar values can be characterized and extracted with Triadic Concept Analysis, which turns out to be a novel mathematical framework for this task. We have defined a scaling procedure turning original numerical data into triadic contexts from which biclusters can be extracted as triadic concepts with existing algorithms. This approach allows a correct, complete and non-redundant extraction of all maximal biclusters, for any similarity parameter θ and can be extended to n-ary numerical datasets while their computation can be directly distributed. The interpretation of triadic concepts is powerful: both extent and intent allow one to characterize a bicluster (i.e. the rectangle), while the modus gives the range of values of the biclusters, and for which θ is the bicluster maximal. Moreover, the larger the modus, the more similar the values within a current bicluster. This fact gives a particular semantics to the notion of *support* as defined in itemset-mining [3]. We also adapted the TCA machinery with algorithm TriMax to extract maximal biclusters for a user-defined threshold θ. It appears that TriMax is a fully customizable algorithm: any concept extraction algorithm can be used as a core module (along with several constraints on produced dyadic concepts), while its distributed computation is direct.

Perspectives of further research are numerous. Firstly, a deeper algorithmic study has to be carried out: Could we avoid discretization and apply TCA directly on the numerical data as it is done with interval pattern structures [20]? Is it more efficient? Secondly, consider constraint-based itemset-mining (e.g. [34]). The goal is to extract only patterns that respect a given predicate, e.g. cardinality of the extent should be greater than a given minimal support. Concerning triadic concepts (and even polyadic concepts), several constraints can be handled with the algorithm Data-Peeler [10]. An interesting investigation is to list all additional constraints that could be handled easily in our framework. Thanks to the genericity, i.e. using FCA and existing algorithms, many of existing constraints can be handled directly: for example Data-Peeler can be used as a core module of TriMax. Finally, one should remark that we focused our study on a particular type of biclusters. Accordingly, can we handle other types of maximal biclusters with TCA? If so, what would be the corresponding scaling? Can we characterize properties that a bicluster definition should follow so that TCA can be applied?

Acknowledgements Authors would like to thank Dmitry A. Morozov for implementing the algorithms NBS-Miner and IPS. The first author was partially supported by CNPq, Fapemig and the Brazilian National Institute for Science and Technology for the Web (InWeb). The second author was supported by the Basic Research Program of the National Research University Higher School of Economics, project "Mathematical models, algorithms, and software for knowledge discovery in structured and text data". The third author acknowledges support by Grant No. P103/10/1056 of the Czech Science Foundation.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. **17**(6), 734–749 (2005)
2. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data. Data Min. Knowl. Discov. **11**(1), 5–33 (2005)
3. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S. (eds.) Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216. ACM Press (1993)
4. Alqadah, F., Bhatnagar, R.: Similarity measures in formal concept analysis. Ann. Math. Artif. Intell. **61**(3), 245–256 (2011)
5. Besson, J., Robardet, C., Boulicaut, J.-F.: Mining a new fault-tolerant pattern type as an alternative to formal concept discovery. In: Schärfe, H., Hitzler, P., hrstrøm, P. (eds.) Conceptual Structures: Inspiration and Application, 14th International Conference on Conceptual Structures (ICCS). Lecture Notes in Computer Science, vol. 4068, pp. 144–157. Springer (2006)
6. Besson, J., Robardet, C., Raedt, L.D., Boulicaut, J.-F.: Mining bi-sets in numerical data. In: Dzeroski, S., Struyf, J. (eds.) KDID. Lecture Notes in Computer Science, vol. 4747, pp. 11–23. Springer (2007)
7. Blachon, S., Pensa, R., Besson, J., Robardet, C., Boulicaut, J.-F., Gandrillon, O.: Clustering formal concepts to discover biologically relevant knowledge from gene expression data. Silico Biology **7**(4–5), 467–483 (2007)
8. Braga Araújo, R., Trielli Ferreira, G., Orair, G., Meira, J., Wagner, R., Ferreira, C., Olavo Guedes Neto, D., Zaki, M.: The partricluster algorithm for gene expression analysis. Int. J. Parallel Prog. **36**, 226–249 (2008)
9. Califano, A., Stolovitzky, G., Tu, Y.: Analysis of gene expression microarrays for phenotype classification. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB), pp. 75–85. AAAI (2000)
10. Cerf, L., Besson, J., Robardet, C., Boulicaut, J.-F.: Closed patterns meet *n*-ary relations. TKDD **3**(1), 3:1–3:36 (2009)
11. Cheng, Y., Church, G.: Biclustering of expression data. In: Proc. 8th International Conference on Intelligent Systems for Molecular Biology (ISBM), pp. 93–103 (2000)
12. Ganter, B., Wille, R.: Formal Concept Analysis. Springer (1999)
13. Hartigan, J.A.: Direct clustering of a data matrix. J. Am. Stat. Assoc. **67**(337), 123–129 (1972)
14. Ignatov, D.I., Kuznetsov, S.O., Poelmans, J.: Concept-based biclustering for internet advertisement. In: Vreeken, J., Ling, C., Zaki, M.J., Siebes, A., Yu, J.X., Goethals, B., Webb, G.I., Wu, X. (eds.) ICDM Workshops, pp. 123–130. IEEE Computer Society (2012)
15. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Trias—an algorithm for mining iceberg tri-lattices. In: ICDM, pp. 907–911 (2006)
16. Ji, L., Tan, K.-L., Tung, A.K.H.: Mining frequent closed cubes in 3d datasets. In: Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB), pp. 811–822. ACM (2006)
17. Kaytoue, M., Assaghir, Z., Napoli, A., Kuznetsov, S.O.: Embedding tolerance relations in formal concept analysis: an application in information fusion. In: CIKM, pp. 1689–1692. ACM (2010)
18. Kaytoue, M., Kuznetsov, S.O., Macko, J., Meira, W., Napoli, A.: Mining biclusters of similar values with Triadic Concept Analysis. In: Napoli, A., Vychodil, V. (eds.) The Eighth International Conference on Concept Lattices and their Applications—CLA 2011. INRIA Nancy Grand Est - LORIA, Nancy, France (2011)
19. Kaytoue, M., Kuznetsov, S.O., Napoli, A.: Biclustering numerical data in formal concept analysis. In: Valtchev, P., Jäschke, R. (eds.) ICFCA. LNCS, vol. 6628, pp. 135–150. Springer (2011)
20. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. Inf. Sci. **181**(10), 1989–2001 (2011)
21. Kaytoue-Uberall, M., Duplessis, S., Kuznetsov, S.O., Napoli, A.: Two fca-based methods for mining gene expression data. In: Ferré, S., Rudolph, S. (eds.) Proceedings of the 7th International Conference on Formal Concept Analysis (ICFCA). Lecture Notes in Computer Science, vol. 5548, pp. 251–266. Springer (2009)
22. Krajca, P., Vychodil, V.: Distributed algorithm for computing formal concepts using map-reduce framework. In: IDA, pp. 333–344. Springer (2009)
23. Kuznetsov, S.O.: A fast algorithm for computing all intersections of objects in a finite semi-lattice. Autom. Doc. Math. Linguist. **27**(5), 11–21 (1993)

24. Kuznetsov, S.O., Obiedkov, S.A.: Comparing performance of algorithms for generating concept lattices. J. Exp. Theor. Artif. Intell. **14**(2–3), 189–216 (2002)
25. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. In: ICCS. LNCS, vol. 954, pp. 32–43. Springer (1995)
26. Madeira, S., Oliveira, A.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans. Comput. Biol. Bioinforma. **1**(1), 24–45 (2004)
27. Mirkin, B.: Mathematical Classification and Clustering. Kluwer Academic Publisher, Boston (1996)
28. Mirkin, B.: Clustering for Data Mining: A Data Recovery Approach. Chapman & Hall/Crc Computer Science (2005)
29. Mirkin, B., Kramarenko, A.V.: Approximate bicluster and tricluster boxes in the analysis of binary data. In: Kuznetsov, S.O., Slezak, D., Hepting, D.H., Mirkin, B. (eds.) Proceedings of the 13th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC 2011). Lecture Notes in Computer Science, vol. 6743, pp. 248–256. Springer (2011)
30. Motameny, S., Versmold, B., Schmutzler, R.: Formal concept analysis for the identification of combinatorial biomarkers in breast cancer. In: Medina, R., Obiedkov, S.A. (eds.) Formal Concept Analysis, 6th International Conference (ICFCA). Lecture Notes in Computer Science, vol. 4933, pp. 229–240. Springer (2008)
31. Pensa, R.G., Leschi, C., Besson, J., Boulicaut, J.-F.: Assessment of discretization techniques for relevant pattern discovery from gene expression data. In: Zaki, M.J., Morishita, S., Rigoutsos, I. (eds.) Proceedings of the 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD 2004), pp. 24–30 (2004)
32. Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Buhlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics **22**(9), 1122–1129 (2006)
33. Raïssi, C., Pei, J., Kister, T.: Computing closed skycubes. PVLDB **3**(1), 838–847 (2010)
34. Soulet, A., Raïssi, C., Plantevit, M., Crémilleux, B.: Mining dominant patterns in the sky. In: Cook, D.J., Pei, J., Wang, W., Zaïane, O.R., Wu, X. (eds.) In: 11th IEEE International Conference on Data Mining (ICDM), pp. 655–664. IEEE (2011)
35. Tchagang, A.B., Phan, S., Famili, F., Shearer, H., Fobert, P.R., Huang, Y., Zou, J., Huang, D., Cutler, A., Liu, Z., Pan, Y.: Mining biological information from 3d short time-series gene expression data: the optricluster algorithm. BMC Bioinforma. **13**, 54 (2012)
36. Valtchev, P., Missaoui, R., Godin, R.: Formal concept analysis for knowledge discovery and data mining: the new challenges. In: Eklund, P.W. (ed.) ICFCA. LNCS, vol. 2961, pp. 352–371. Springer (2004)
37. Voutsadakis, G.: Polyadic concept analysis. Order **19**(3), 295–304 (2002)
38. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (eds.) Ordered Sets, pp. 445–470. Reidel (1982)
39. Wille, R.: Why can concept lattices support knowledge discovery in databases? J. Exp. Theor. Artif. Intell. **14**(2–3), 81–92 (2002)
40. Zhao, L., Zaki, M.J.: Tricluster: an effective algorithm for mining coherent clusters in 3d microarray data. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, SIGMOD '05, pp. 694–705. ACM, New York, USA (2005)