Triclustering Algorithms for Three-Dimensional Data Analysis: A Comprehensive Survey

RUI HENRIQUES, INESC-ID and Instituto Superior Técnico, Universidade de Lisboa, Portugal
SARA C. MADEIRA, LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal

Three-dimensional data are increasingly prevalent across biomedical and social domains. Notable examples are gene-sample-time, individual-feature-time, or node-node-time data, generally referred to as observation-attribute-context data. The unsupervised analysis of three-dimensional data can be pursued to discover putative biological modules, disease progression profiles, and communities of individuals with coherent behavior, among other patterns of interest. It is thus key to enhance the understanding of complex biological, individual, and societal systems. In this context, although clustering can be applied to group observations, its relevance is limited since observations in three-dimensional data domains are typically only meaningfully correlated on subspaces of the overall space. Biclustering tackles this challenge but disregards the third dimension. In this scenario, triclustering—the discovery of coherent subspaces within three-dimensional data—has been largely researched to tackle these problems. Despite the diversity of contributions in this field, there still lacks a structured view on the major requirements of triclustering, desirable forms of homogeneity (including coherency, structure, quality, locality, and orthonormality criteria), and algorithmic approaches. This work formalizes the triclustering task and its scope, introduces a taxonomy to categorize the contributions in the field, provides a comprehensive comparison of state-of-the-art triclustering algorithms according to their behavior and output, and lists relevant real-world applications. Finally, it highlights challenges and opportunities to advance the field of triclustering and its applicability to complex three-dimensional data analysis.

CCS Concepts: • **Computing methodologies** → *Machine learning*; *Cluster analysis*; *Motif discovery*;

Additional Key Words and Phrases: Triclustering, three-dimensional data analysis, subspace clustering, multivariate time series analysis, multidimensional clustering

ACM Reference format:
Rui Henriques and Sara C. Madeira. 2018. Triclustering Algorithms for Three-Dimensional Data Analysis: A Comprehensive Survey. *ACM Comput. Surv.* 51, 5, Article 95 (September 2018), 43 pages.
https://doi.org/10.1145/3195833

1 INTRODUCTION

Three-dimensional data—also referred as tridiac, three-way, tensor, or cubic data—are increasingly prevalent in several domains (Mankad and Michailidis 2014). In biomedical domains, the periodic profiling of biological entities and clinical monitoring of individuals are giving rise to

---

This work was partially supported by *Fundação para a Ciência e Tecnologia* (FCT) under the project NEUROCLINOMICS2 PTDC/EEI-SII/1937/2014, and INESC-ID (UID/CEC/50021/2013) and LASIGE (UID/CEC/ 00408/2013) pluriannual funding.

Authors' addresses: Prof. R. Henriques, DEI, IST, Av. Rovisco Pais 1, 1900-001 Lisboa, Portugal; email: rmch@tecnico.ulisboa.pt; Prof. S. C. Madeira, Room 6.3.19, FCUL, Campo Grande 016, 1749-016 Lisboa, Portugal; email: sacmadeira@ciencias.ulisboa.pt.

ACM 0360-0300/2018/09-ART95 $15.00
https://doi.org/10.1145/3195833

multivariate time series (Amar et al. 2015). Here, the analysis of *gene-sample-time* expression data, *compound-sample-time* concentration data, and *patient-record-time* data (as well as other forms of three-dimensional data, such as *residues-position-time* protein data, *entity-entity-time* interaction data, and *drug-gene-dose* expression data) is essential to understand complex biological and physiological processes underlying disease progression, development, and responses to stimuli, drugs, or therapy (Tchagang et al. 2012; Bhar et al. 2015; Heylen et al. 2016). Two-dimensional biomedical data also have been augmented into three-dimensional (3D) data by combining heterogeneous data sources (such as *specie-metabolite-condition* concentration (Hu and Bhatnagar 2010)) or by decomposing the observed values against knowledge-driven annotations (such as *gene-sample-term* or *gene-sample-regulator* expression data (Alqadah and Bhatnagar 2008; Li and Tuck 2009)). In social domains, understanding consumerism, web usage, work performance, and social activity can be reliably done using *individual-feature-time* behavioral data or *node-node-time* interaction data. Moreover, financial decisions can be supported by the analysis of *stock-ratio-time* data or *society-society-trade* data (Sim et al. 2010a; Mankad and Michailidis 2014; Guigourès et al. 2015). In this work, the aforementioned forms of 3D data are generally referred to as *observation-attribute-context* data, and the term *object* is used to indistinctly denote an observation, attribute, or context.

Due to the inherent complexity of biomedical and social 3D data analysis, a natural first step is to identify potentially relevant subspaces (subsets of objects). Clustering can be used to group objects from one dimension at a time. However, this is usually performed globally considering all the attributes describing a given object, an undesirable restriction in 3D data spaces with locally correlated values. Illustrating, groups of biological entities or individuals are often only meaningfully correlated on a specific subset of conditions/records and time points (Zhao and Zaki 2005). Subspace clustering was originally proposed to address this problem. In the context of two-dimensional data, subspace clustering is known as biclustering (Madeira and Oliveira 2004). In the context of 3D data, subspace clustering is termed triclustering (Zhao and Zaki 2005). Given a (real-valued or symbolic) 3D dataset, the triclustering task aims to discover subsets of observations, attributes, and contexts—referred to as triclusters—satisfying certain homogeneity and statistical significance criteria. Triclustering has been applied to unravel putative regulatory modules, disease patterns, communities with shared behavior, or profitable financial profiles (Amar et al. 2015; Mankad and Michailidis 2014), thus being key to study complex biological, individual, and societal systems. Furthermore, when observations are labeled, triclustering can be applied with additional discriminative criteria to differentiate classes and support real-world decisions (Li and Ngom 2010).

The aforementioned points explain the increasing attention given to triclustering (Narmadha and Rathipriya 2016), an emerging research topic with algorithmic proposals nearly duplicating every year (see Table 1) since the first algorithm proposal by Zhao and Zaki (2005). Despite its relevance, triclustering faces major challenges:

1. *Efficiency*: triclustering, similar to biclustering, is an NP-hard problem (Zhao and Zaki 2005) involving heavy combinatorial optimization to group subsets of objects against all possible subspaces.
2. *Homogeneity*: coherence criteria need to be placed in accordance with the target problem and desirable outputs (Gutiérrez-Avilés and Rubio-Escudero 2014b). In this context, agreement on what is an adequate merit function has not yet been reached (Amar et al. 2015)
3. *Avoidance of biases*: triclustering should prevent forms of dimension dominance (biases toward one or more dimensions) and guarantee coherence between objects from each pair of dimensions (Mahanta et al. 2011).

(4) *Robustness*: triclustering should be able to handle varying types (and degrees) of noise and missing values inherent to real-world 3D biomedical and social data (Jiang et al. 2006).
(5) *Flexibility*: the ability to discover a nonfixed number of triclusters with arbitrary size, shape, and positioning should be pursued to guarantee that all relevant subspaces are found. Eventually, data can be described according to the cumulative effects of overlapping triclusters (Mankad and Michailidis 2014).
(6) *Statistical significance*: by definition, triclusters should be statistically significant; i.e., their probability to occur should be unexpectedly low. In this context, optimizing or testing their homogeneity and/or size is insufficient (Sim et al. 2010a).
(7) *Temporality* (when applicable): triclustering observation-attribute-time data requires adequate homogeneity criteria to capture coherent progression patterns and the ability to accommodate meaningful time lags on observations (Xu et al. 2009).
(8) *Data specificities*: the inherent aspects of 3D data should not be disregarded. Illustrating, the analysis of *heterogeneous* 3D data (data combined from multiple sources) should consider source-specific forms of coherence and noise (Supper et al. 2007), while the analysis of *spatial* 3D data depends on meaningful contiguity and distance criteria (Guigourès et al. 2015). Complementarily, 3D data attributes might be symbolic, real-valued, or a (non-i.i.d.) combination of them.
(9) *Others*: additional challenges include the need to guarantee that triclustering algorithms are not overly dependent on parameterizations (Guigourès et al. 2015) and that the learned triclustering models are actionable (Sim et al. 2010b).

In this context, this work proposes the first comprehensive survey on triclustering, providing six major contributions:

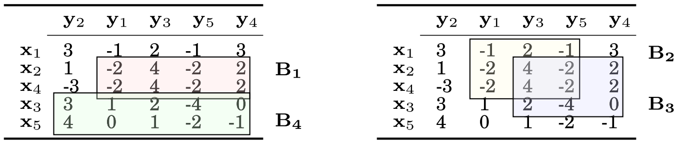
- Formalization of the triclustering task and its major properties
- Comparison of triclustering against peer tasks for multidimensional data analysis
- Taxonomy for a structured and critical assessment of existing/future contributions
- Principles to support the design and unbiased evaluation of triclustering algorithms
- Comparison of the state-of-the-art triclustering algorithms according to their behavior, output, strengths, and limitations
- Overview of real-world applications, together with insights to understand the intrinsic potential of triclustering across domains

Accordingly, this survey is organized as follows. *Section 2* provides the formal ground on triclustering and introduces peer tasks for 3D data analysis. *Section 3* proposes a taxonomy for the comprehensive understanding and guided development of triclustering algorithms. *Section 4* formalizes the properties of triclustering solutions, *Section 5* categorizes and describes the behavior of state-of-the-art triclustering algorithms, and *Section 6* establishes principles for their adequate evaluation. *Section 7* lists relevant triclustering applications. *Section 8* summarizes the contributions in the literature against the introduced taxonomy. Finally, the implications of this work are described together with key directions for future work.

2 PROBLEM FORMULATION

Definition 2.1. Given a dataset with n observations, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, the **clustering** task aims to find subsets of observations (clusters), $\{\mathbf{I}_1, \ldots, \mathbf{I}_r\}$, where $\mathbf{I}_i \subseteq \mathbf{X}$ satisfies certain intracluster and intercluster criteria of (dis)similarity over the whole space.

Despite the relevance of the clustering task, the (dis)similarity between observations becomes blurred in the presence of a high number of attributes per observation. A way to tackle this



	y_2	y_1	y_3	y_5	y_4	
x_1	3	-1	2	-1	3	
x_2	1	-2	4	-2	2	B_1
x_4	-3	-2	4	-2	2	
x_3	3	1	2	-4	0	
x_5	4	0	1	-2	-1	B_4

	y_2	y_1	y_3	y_5	y_4	
x_1	3	-1	2	-1	3	B_2
x_2	1	-2	4	-2	2	
x_4	-3	-2	4	-2	2	
x_3	3	1	2	-4	0	B_3
x_5	4	0	1	-2	-1	

Fig. 1. Biclusters with varying homogeneity criteria. $B_1 = (I_1 = \{x_2, x_4\},\ J_1 = \{y_1, y_3, y_4, y_5\})$, $B_2 = (I_2 = \{x_1, x_2, x_4\},\ J_2 = \{y_1, y_3, y_5\})$, $B_3 = (I_3 = \{x_2, x_3, x_4\},\ J_3 = \{y_3, y_4, y_5\})$, and $B_4 = (I_4 = \{x_3, x_5\},\ J_4 = \{y_1, y_2, y_3, y_4, y_5\})$ respectively satisfy a constant, multiplicative, additive, and order-preserving coherence without noise.

problem is to perform clustering in data subspaces (Madeira and Oliveira 2004) so that a group of observations needs only to be similar on a subset of attributes.

Definition 2.2. Let a two-dimensional dataset (matrix), **A**, be defined by n observations (rows) $\mathbf{X} = \{x_1, \ldots, x_1\}$, m attributes (columns) $Y = \{y_1, \ldots, y_m\}$, and $n \times m$ elements (values) a_{ij}. Given a real-valued or symbolic matrix A, a **bicluster** $\mathbf{B} = (\mathbf{I}, \mathbf{J})$ is a subspace given by a subset of rows, $\mathbf{I} \subseteq \mathbf{X}$, and a subset of columns, $\mathbf{J} \subseteq Y$. The **biclustering** task aims to find a set of biclusters $\{\mathbf{B}_1, \ldots, \mathbf{B}_q\}$, such that each bicluster $\mathbf{B}_i$ satisfies specific criteria of homogeneity (Henriques et al. 2015b) and statistical significance (Henriques and Madeira 2018).

Figure 1 provides a matrix (with reordered rows and columns) with biclusters satisfying constant, multiplicative, additive, and order-preserving homogeneities on rows for an unexpectedly large area (statistical significance). Henriques et al. (2015b) and Henriques and Madeira (2018) survey available criteria of homogeneity and statistical significance. In the presence of temporal two-dimensional data, contiguity can be assumed on attributes (time points) (Madeira et al. 2010) and lags accommodated (Gonçalves and Madeira 2014). *Definitions 2.3* and *2.4* extend this formulation to 3D data.

Definition 2.3. A **three-dimensional dataset A** (also referred as tridiac data, cube data, or 3D data in short) is defined by n observations $\mathbf{X} = \{x_1, \ldots, x_n\}$, m attributes $Y = \{y_1, \ldots, y_m\}$, and p contexts $\mathbf{Z} = \{z_1, \ldots, z_p\}$. Elements a_{ijk} relate observation x_i, attribute y_j, and context z_k. 3D data can be *real-valued* ($a_{ijk} \in \mathbb{R}$), *symbolic* ($a_{ijk} \in \Sigma$, where Σ is a set of nominal or ordinal symbols), *integer* ($a_{ijk} \in \mathbb{R}$), or *non-identically distributed* ($a_{ijk} \in \mathcal{A}_j$, where $\mathcal{A}_j$ is the domain of y_j's attribute). *Sparse 3D data* are 3D data with an arbitrary number of missing elements, $a_{ijk} \in \mathcal{A}_j \cup \emptyset$.

When contexts correspond to time points, we are in the presence of a *temporal 3D dataset* (also referred as three-way time series), where each observation is a multivariate time series with m order. A *temporal network* is a sparse temporal 3D dataset given by a sequence of adjacency matrices derived from (directed or undirected) weighted graphs.

The values of an object (whether an observation, attribute, or context) from a 3D dataset define a matrix, referred to in this survey as a slice. For instance, $\mathbf{x_i}$ defines an observation slice, also referred as an attribute-context plane (Xu et al. 2009).

Definition 2.4. Given a 3D dataset **A** with n observations **X**, m attributes Y, and p contexts **Z**, a **tricluster** $\mathbf{B} = (\mathbf{I}, \mathbf{J}, \mathbf{K})$ is a subspace of the original space, where $\mathbf{I} \subseteq \mathbf{X}$, $\mathbf{J} \subseteq Y$, and $\mathbf{K} \subseteq \mathbf{Z}$ are subsets of observations, attributes, and contexts, respectively.

Given **A**, the **triclustering** task aims to find a set of triclusters $\{\mathbf{B}_1, \ldots, \mathbf{B}_l\}$ such that each tricluster $\mathbf{B}_i$ satisfies specific criteria of *homogeneity* and statistical *significance*.

Figure 2 illustrates some of the aforementioned concepts. A tricluster is defined by subsets of objects in each of the three data dimensions. *Definitions 2.5* and *2.6* extend matured concepts on biclustering homogeneity and statistical significance (Madeira and Oliveira 2004; Henriques and Madeira 2018) toward triclustering.

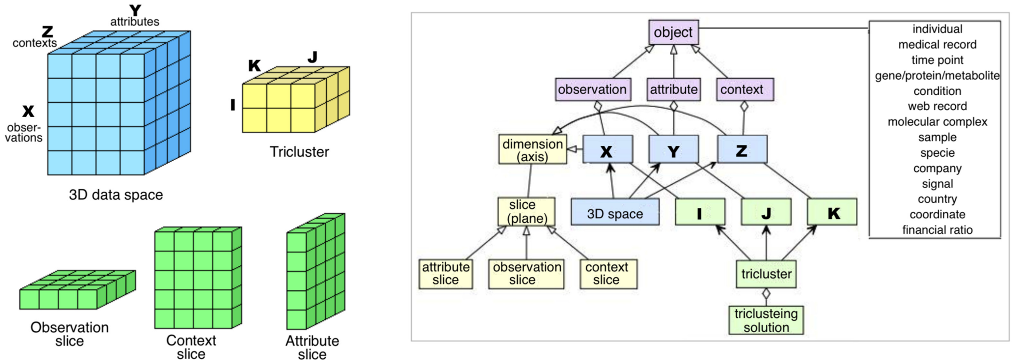


Fig. 2. 3D data analysis: basic concepts.

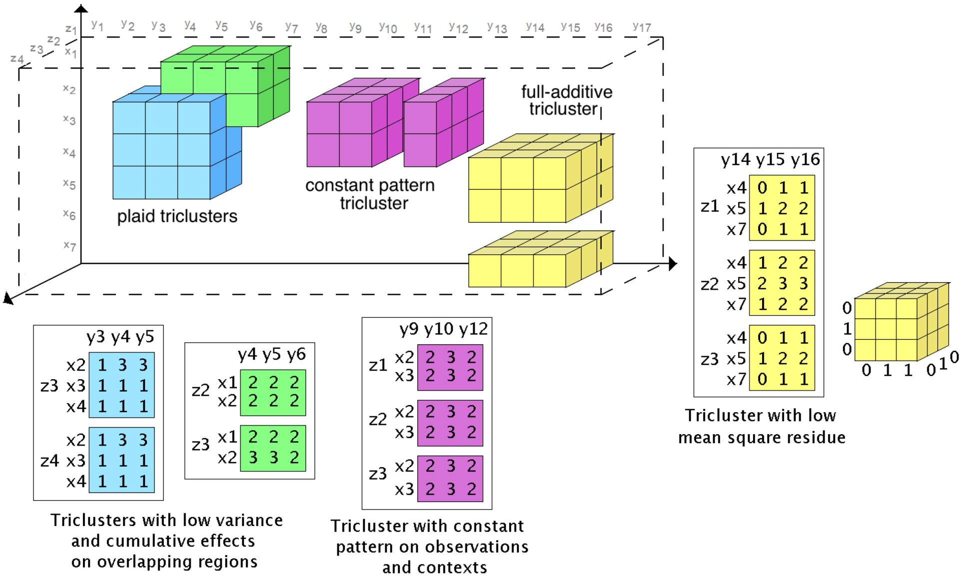


Fig. 3. Triclustering solution: flexible structure of triclusters with different coherencies.

Definition 2.5. The **homogeneity** criterion determines the structure, coherence, and quality of a triclustering solution, where

- the *structure* is described by the number, size, shape, and position of triclusters;
- the *coherence* of a tricluster is defined by the observed correlation of values (coherence assumption) and the allowed deviation from expectations (coherence strength); and
- the *quality* of a tricluster is defined by the type and amount of tolerated noise.

Flexible structures of triclusters are characterized by an arbitrary number of (possibly overlapping) triclusters. Figure 3 illustrates a flexible structure of triclusters with different coherencies and perfect quality (no noise allowed).

Definition 2.6. A tricluster is **statistically significant** if its probability to occur deviates from expectations (i.e., is unexpectedly low against a null data model).

Definition 2.7. Let $\mathcal{B}$ be the set of triclusters that satisfy a given homogeneity and statistical significance criteria; $(\mathbf{I}, \mathbf{J}, \mathbf{K}) \in \mathcal{B}$ is a **maximal tricluster** if and only if there is no other tricluster $(\mathbf{I}', \mathbf{J}', \mathbf{K}')$ such that $\mathbf{I} \subseteq \mathbf{I}' \wedge \mathbf{J} \subseteq \mathbf{J}' \wedge \mathbf{K} \subseteq \mathbf{K}'$ satisfies the given criteria.

Although an *optimal triclustering solution* is one containing all maximal triclusters satisfying predefined homogeneity and statistical significance criteria, the high number of (possibly redundant) maximal triclusters is often undesirable and thus the formulation of triclustering can be augmented to satisfy dissimilarity criteria.

Definition 2.8. Let the *priority* of a tricluster be defined by specific criteria of interest, such as size, statistical significance, and/or homogeneity. Given a 3D dataset, $\mathcal{B}$ set of triclusters, a specific priority criterion, and a subspace similarity function (e.g., the fraction of overlapping elements), **dissimilarity criteria** can be placed to guarantee that any tricluster similar to another tricluster with higher priority is removed from $\mathcal{B}$ and (possibly) used to refine similar triclusters in $\mathcal{B}$.

Triclustering involves non-trivial combinatorial optimization of higher complexity than the NP-hard biclustering task (Zhao and Zaki 2005). As such, triclustering is similarly NP-hard, a property that can be proven by mapping the triclustering task into the problem of finding maximal hypercliques (constant triclusters) in graphs with multiwise interactions (Lin et al. 2009). The complexity increases when considering non-trivial forms of coherence, flexible structures, and tolerance to noise. As a result, most triclustering algorithms are based on greedy or stochastic searches (producing suboptimal solutions) (Mahanta et al. 2011) and place restrictions on the allowed structure, coherence, and quality of triclusters (Sim et al. 2010a).

2.1 Merit Functions

As aforementioned, the triclustering task is primarily guided by the placed homogeneity and statistical significance criteria (*Definitions 2.5 and 2.6*), and possibly combined with dissimilarity criteria (*Definition 2.8*). These criteria are essentially defined by the selected merit functions and how they are applied. A merit function is a function that evaluates how good a tricluster is based on the values of its elements. The variance of values in a tricluster is an illustrative merit function, which when minimized leads to the discovery of subspaces with approximately constant values. Merit functions vary according to

- *the way they are applied*: to guide greedy iterative searches (Bhar et al. 2012), optimize multiple objectives (Liu et al. 2008), or learn parametric models describing the target solution (Amar et al. 2015)), for example;
- *their scope*: whether they are used to assess a single tricluster or the overall triclustering solution (Mankad and Michailidis 2014); and
- *the correlation extent*: whether they (1) jointly assess the three dimensions (Sim et al. 2010a), (2) concern two dimensions at a time followed by consensus (Greco et al. 2010), or (3) target maximal triclusters with (symbolic) patterns (Liu et al. 2010).

Definition 2.9. The homogeneity criteria of a triclustering task is determined by the placed merit functions: **3D merit functions** correlate all values or slices of a tricluster within a single function; **2D merit functions** correlate values within a slice of a tricluster and are often followed by the search for consensus among slices; **pattern-based merit functions** assess the maximality of triclusters with well-defined patterns.

To facilitate the understanding of existing merit functions, some relevant notation follows. Given a real-valued 3D dataset $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ and a tricluster $(\mathbf{I} \subseteq \mathbf{X}, \mathbf{J} \subseteq \mathbf{Y}, \mathbf{K} \subseteq \mathbf{Z})$, $a_{iJK} = \frac{1}{|\mathbf{J}| \times |\mathbf{K}|} \sum_{j \in \mathbf{J}}$

$\sum_{k\in\mathbf{K}} a_{ijk}$ is the mean of values for the $\mathbf{x}_i$ observation. Similarly, a_{IJk} is the mean of values for the $\mathbf{z}_k$ context. Following this notation, a_{ijK} is the mean of values associated with the $\mathbf{x}_i$ observation and $\mathbf{y}_j$ attribute, while $a_{IJK}=\frac{1}{|\mathbf{I}||\mathbf{J}||\mathbf{K}|}\sum_{i\in\mathbf{I}}\sum_{j\in\mathbf{J}}\sum_{k\in\mathbf{K}} a_{ijk}$ is the mean of all values of a tricluster.

Given a symbolic 3D dataset, prior notation is also applicable by replacing the mean calculus with either the mode (symbolic data) or median (integer data) operators.

3D Merit Functions. Examples of 3D merit functions include algebraic distances, correlation coefficients, and residue-based functions, among others (Gutiérrez-Avilés and Rubio-Escudero 2014b; Sim et al. 2010a).

A *perfect constant* tricluster is a subspace $(\mathbf{I},\mathbf{J},\mathbf{K})$ with $a_{ijk} = c$, where c is a symbol or expected numeric value. Although constant triclusters are common in symbolic 3D data, they are often masked by noise in real-valued 3D data.

Given a real-valued 3D dataset, a *constant* tricluster $(\mathbf{I},\mathbf{J},\mathbf{K})$ has $a_{ijk} = c + \eta_{ijk}$, where $c \in \mathbb{R}$ is the expected value and $\eta_{ijk} \in \mathbb{R}$ is the noise factor.

An illustrative merit function to find constant triclusters in real-valued data is the variance of their values,

$$\sigma^2_{(\mathbf{I},\mathbf{J},\mathbf{K})} = \frac{1}{|\mathbf{I}||\mathbf{J}||\mathbf{K}|}\sum_{i\in\mathbf{I}}\sum_{j\in\mathbf{J}}\sum_{k\in\mathbf{K}}(a_{ijk} - a_{IJK})^2. \quad (1)$$

The allowed level of variance can be calibrated to find larger triclusters, $\sigma^2_{(\mathbf{I},\mathbf{J},\mathbf{K})} < \delta$, where δ is the coherence strength (controlling deviations from expected values).

Residue-based functions can be used to guarantee more flexible forms of homogeneity, including coherence assumptions that can accommodate shifts and scales on one, two, or three dimensions. In this context, the mean squared residue (MSR), originally proposed for the biclustering task (Cheng and Church 2000), was extended to guide triclustering algorithms (Bhar et al. 2012; Dede and Oğul 2013). Given a real-valued 3D dataset, the elements of a tricluster can be described by

$$a_{ijk} = c + \alpha_i + \beta_j + \gamma_k + \eta_{ijk}, \quad (2)$$

where $c \in \mathbb{R}$ is the background value of the tricluster; α_i, β_j, and γ_k are contributions from the $\mathbf{x}_i$ observation, $\mathbf{y}_j$ attribute, and $\mathbf{z}_k$ context, respectively; and η_{ijk} is the noise factor (residue).

The residue is an indicator of the correlation of the values in the tricluster. Hence, the difference between expectations, $a_{ijk} = c + \alpha_i + \beta_j + \gamma_k + \eta_{ijk} = a_{IJK} + (a_{iJK} - a_{IJK}) + (a_{IjK} - a_{IJK}) + (a_{IJk} - a_{IJK})$, leads to a residue given by $\eta_{ijk} = a_{ijk} - (a_{iJK} + a_{IjK} + a_{IJk} - 2a_{IJK})$. The 3D MSR of a subspace is the average of the squared residues,

$$MSR_{(\mathbf{I},\mathbf{J},\mathbf{K})} = \frac{1}{|\mathbf{I}||\mathbf{J}||\mathbf{K}|}\sum_{i\in\mathbf{I}}\sum_{j\in\mathbf{J}}\sum_{k\in\mathbf{K}}\eta_{ijk}, \quad \eta_{ijk} = a_{ijk} - (a_{iJK} + a_{IjK} + a_{IJk} - 2a_{IJK}). \quad (3)$$

A tricluster is a subspace with MSR below a given coherence strength δ (Gutiérrez-Avilés and Rubio-Escudero 2014b). A perfect tricluster exhibits no bias (MSR = 0), with each a_{ijk} value defined by dimension-wise means ($\alpha_i = a_{iJK}, \beta_j = a_{IjK}, \gamma_k = a_{IJk}$) and overall mean ($c = a_{IJK}$).

The previous merit functions do not take into consideration the relationships between triclusters. However, the values a_{ijk} can be seen as a sum of the contributions of the different triclusters to which they belong. Mankad and Michailidis (2014) extend the plaid model proposed by Lazzeroni and Owen (2002), originally prepared to consider cumulative contributions in real-valued matrices, toward 3D data. In this context, real-valued 3D data can be described as a sum of q layers (triclusters),

$$a_{ijk} = \mu_0 + \sum_{t=0}^{q} \theta_{ijkt}\rho_{it}\kappa_{jt}\tau_{kt} \ + \eta_{ijk} \quad | \quad \theta_{ijkt} = \mu_t + \alpha_{it} + \beta_{jt} + \gamma_{kt} + \eta_{ijkt}, \tag{4}$$

where θ_{ijkt} specifies the contribution of each tricluster. The Boolean ρ_{ik}, κ_{jk}, and τ_{jk} variables define the membership of observation $\mathbf{x}_i$, attribute $\mathbf{y}_j$, and context $\mathbf{z}_k$ in tricluster $\mathbf{B}_t$. When assuming η_{ijk} is approximately Gaussian, the triclusters can be obtained by minimizing a merit function based on the quadratic error,

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{p}\left(\hat{a}_{ijk} - \theta_{ijk0} - \sum_{t=1}^{T}\theta_{ijkt}\rho_{it}\kappa_{jt}\tau_{kt}\right)^2. \tag{5}$$

Some works consider the possibility to specify multiple (possibly conflicting) merit functions (Bhar et al. 2015; Gutiérrez-Avilés et al. 2014). Liu et al. (2008) define triclustering as a multiobjective optimization problem with three merit functions: triclusters with 3D MSR below δ, high volume, and high variance.

Complementary works combine merit functions with additional criteria to handle further challenges, such as high imbalance on the number of objects per dimension (e.g., thousands of genes for dozens of samples and time points in gene-sample-time data) (Bhar et al. 2012) or arbitrarily high overlapping areas (Xu et al. 2009). Gutiérrez-Avilés and Rubio-Escudero (2014b) extended the 3D MSR to incorporate two new terms controlling both the imbalance on the size of each dimension and the overlapping degree between triclusters.

In the context of symbolic 3D data analysis, most 3D merit functions were proposed for binary 3D data (Ignatov et al. 2015). An illustrative 3D function is the product of the fraction and number of ones in the tricluster to maximize both its density and size.

In the context of temporal 3D data analysis, a tricluster may show temporal lags on observations. A wide multiplicity of similarity metrics to compare time series have been proposed (Ding et al. 2008; Gonçalves and Madeira 2014). However, only a few are prepared to compare multivariate time series. Planar mean residue similarity (PMRS) was proposed in Ahmed et al. (2011) to compare two observations (each given by a multivariate time series). A tricluster in this context is a subspace with pairs of observations, $\mathbf{x}_{i_1}$ and $\mathbf{x}_{i_2}$, strongly correlated on a subset of attributes, $\mathbf{J}$, and time points, $\mathbf{K}$:

$$\underset{\mathbf{x}_{i_1}\in\mathbf{I},\mathbf{x}_{i_2}\in\mathbf{I}}{\forall}\left(\frac{\sum_{j\in\mathbf{J}}\sum_{k=1}^{p}|(a_{i_1jk} - a_{i_1\mathbf{JK}}) - (a_{i_2jk} - a_{i_2\mathbf{JK}})|}{2\times max(\sum_{j=1}^{m}\sum_{k=1}^{p}|a_{i_1jk} - a_{i_1\mathbf{JK}}|, \sum_{j=1}^{m}\sum_{k=1}^{p}|a_{i_2jk} - a_{i_2\mathbf{JK}}|)} > \delta\right). \tag{6}$$

Sim et al. (2010a) proposed a mutual information score to correlate real-valued matrices that are able to accommodate time lags. Unlike PRMS, this score correlates time points (contexts) instead of observations. In Sim et al. (2010b), the similarity between two observations is not directly computed on their values but rather on the set of k-nearest neighbors from each observation. An observation is seen as a set of Gaussian distributions (one per attribute $\mathbf{y}_j \in \mathbf{J}$) inferred from the k-nearest observations. The correlation between two observations is then measured using their covariances. The proposed merit function further measures distances between observations and the centroid of triclusters, $(a_{iJK}, a_{IjK}, a_{IJk})$, promoting dissimilarity between triclusters.

Alternative merit functions were proposed to focus on the coherent variation of values within a tricluster (independently of their amplitude) (Gutiérrez-Avilés and Rubio-Escudero 2014a, 2015), being also adequate to deal with temporal misalignments. Least Square Lines (LSL) and Multislope Measure (MSL) compute the similarity by averaging differences on the angles of the plane slopes for all pairs of observations, attributes, and contexts in a given subspace. In LSL, this similarity is inferred from a least square calculus. MSL further considers the influence of neighboring objects.

2D Merit Functions with Consensus. An alternative is to specify a homogeneity criterion for two-dimensional slices of a 3D dataset in order to find subspaces on each slice, and then search for consensus (also referred as agreements) between the identified subspaces across the third dimension (Tchagang et al. 2012). The extensive research on biclustering merit functions (Madeira and Oliveira 2004; Henriques et al. 2015b) can be considered for this aim, including Euclidean distances, pattern support, correlation coefficients, permutation consistency, and residue-based functions.

Zhao and Zaki (2005) proposed the use of the Pearson correlation coefficient (PCC) to identify subspaces per context slice (observation-attribute plane) together with a consensus based on the extent of the overlapping objects to infer the final triclusters. Although PCC is flexible enough to accommodate shifts and scales, it disregards alternative (possibly relevant) forms of homogeneity. To address this challenge, Jiang et al. (2006) proposed Spearman rank correlation (SRC) as the target 2D merit function.

When a slice is given by a multivariate time series, 2D merit functions can enforce contiguity of time points and accommodate temporal misalignments. Xu et al. (2009) placed an order-preserving assumption on attribute slices (observation-time planes) to guarantee that observations follow similar variations along time (independently of their amplitude or change ratio), thus elegantly handling time-related misalignments.

Complementarily, a dedicated portion of literature is focused on consensus measures to infer triclusters from (two-dimensional) subspaces (Greco et al. 2010), ranging from simple Jaccard-based indexes to more complex forms of correlations (able to guarantee that coherence is also preserved between slices) (Xu et al. 2009). The inferred triclusters might be further assessed to verify whether they preserve the desirable homogeneity and statistical significance criteria (Supper et al. 2007).

Pattern-based merit functions. Pattern-based merit functions are typically applied on symbolic or integer 3D data in order to find triclusters with well-defined patterns (Liu et al. 2010). Ji et al. (2006) introduced the concept of frequent closed cube, where a pattern is a set of symbols on a subset of objects from one dimension and repeated across the subsets of objects from the remaining dimensions. In this context, merit functions generally aim to maximize the volume of a tricluster while still respecting a given pattern. Noisy elements can be further accommodated (Ignatov et al. 2015). Pattern-based functions can also rely on the widely researched principles from pattern-based biclustering (Henriques et al. 2015b; Henriques and Madeira 2014; Madeira and Oliveira 2009; Gonçalves and Madeira 2014) in order to handle non-constant patterns, lags in temporal data, sparse data, and the drawbacks of discretization procedures.

2.2 Related Tasks

Variants of the triclustering task can be found in the literature, including (1) revised formulations prepared to handle deviant observations, specify alternative criteria of interest, or retrieve triclusters with non-fixed objects on some dimensions; (2) peer tasks, such as coclustering or motif discovery, or alternative tasks to discover subspaces within different data structures; and (3) generalized formulations to identify subspaces within N-dimensional data where $N \in \mathbb{N}^+$.

Hu and Bhatnagar (2010) generalized the triclustering task for the analysis of heterogeneous 3D data with attributes shared by all observations as well as attributes specific to a single observation. Observations correspond to data sources, with some sources having unique attributes (e.g., genes pertaining to a specific species).

The work of Sim et al. (2010b) extended the definition of a tricluster—referred as an actionable subspace—to further guarantee that it respects prespecified utility criteria.

Amar et al. (2015) further extended the tricluster definition $(\mathbf{I}, \mathbf{J}, \mathbf{K})$ to accommodate observation-specific augmentations on attributes $\{\mathbf{J}\}_i|_{\mathbf{x}_i \in \mathbf{I}}$, contexts $\{\mathbf{K}\}_i|_{\mathbf{x}_i \in \mathbf{I}}$, or both.

In addition, triclustering should not be confused with the peer clustering tasks for the unsupervised analysis of 3D data, such as coclustering or motif discovery.

Definition 2.10. The task of **coclustering** 3D data aims to partition each dimension, ($\mathcal{I} = \{\mathbf{I}_1, \ldots, \mathbf{I}_r\}$, $\mathcal{J} = \{\mathbf{J}_1, \ldots, \mathbf{J}_s\}$, $\mathcal{K} = \{\mathbf{K}_1, \ldots, \mathbf{K}_t\}$), so that either (1) the partitions on each dimension optimize some criteria of (dis)similarity or (2) the $\mathcal{I} \times \mathcal{J} \times \mathcal{K}$ subspaces from the intersecting partitions optimize some homogeneity criteria.

3D coclustering requires all data elements to belong to a subspace (exhaustive condition) and to a single subspace only (exclusive condition), thus limiting the inherent flexibility of the triclustering task (Schepers et al. 2006; Sim et al. 2010b).

Definition 2.11. Given temporal 3D data, **motif discovery** aims to find subspaces (where $|\mathbf{J}| \approx |\mathbf{Y}|$) that are frequent either within an observation or across observations.

Motif discovery can be seen as a specialization of the triclustering task since motifs generally span all the attributes and most contributions are only able to handle multivariate time series with low m order (Vahdatpour et al. 2009; Minnen et al. 2007).

Subspace clustering also can be applied on alternative data structures: (1) *relational data* (Long et al. 2006; Banerjee et al. 2007; Bekkerman et al. 2005; Gao et al. 2006; Chiaravalloti et al. 2006; Balasubramanian et al. 2016), (2) *vertical partitioned data* (Alqadah and Bhatnagar 2008), and (3) *time-annotated graphs* (Guigourès et al. 2015).

Subspace clustering can be generalized for spaces with arbitrary N dimensionality order, often referred as N-way clustering or simply N-clustering (*Definition 2.12*). *Appendix A* extends the proposed triclustering taxonomy for data contexts where $N > 3$.

Definition 2.12. Given an N-dimensional dataset, the ***N*-way clustering task** aims to find subspaces—subsets of objects on each N dimensions—satisfying certain input criteria of interest.

3 TRICLUSTERING TAXONOMY

Figure 4 specifies the proposed taxonomy of concepts to offer a structured understanding of the diverse aspects of triclustering problems, as well as to guide the development or adaptation of triclustering algorithms. The gathered concepts—expanded throughout *Sections 4, 5*, and *6*—offer principles to:

(1) characterize triclustering solutions in accordance with their locality, homogeneity, statistical significance, and data-specific aspects (*Section 4*);
(2) design algorithms according to state-of-the-art triclustering approaches (*Section 5*); and
(3) assess the quality and relevance of the triclustering outputs (*Section 6*).

4 PROPERTIES OF TRICLUSTERING SOLUTIONS

A primary step when tackling a triclustering problem is to determine the desired locality degree (*Section 4.1*), homogeneity criteria (*Section 4.2*), and statistical significance guarantees (*Section 4.3*) of triclustering solutions, as well as to identify the data-specific challenges to address (*Section 4.4*). These topics are expanded below, thus completing the formal ground on triclustering started in *Section 2*.

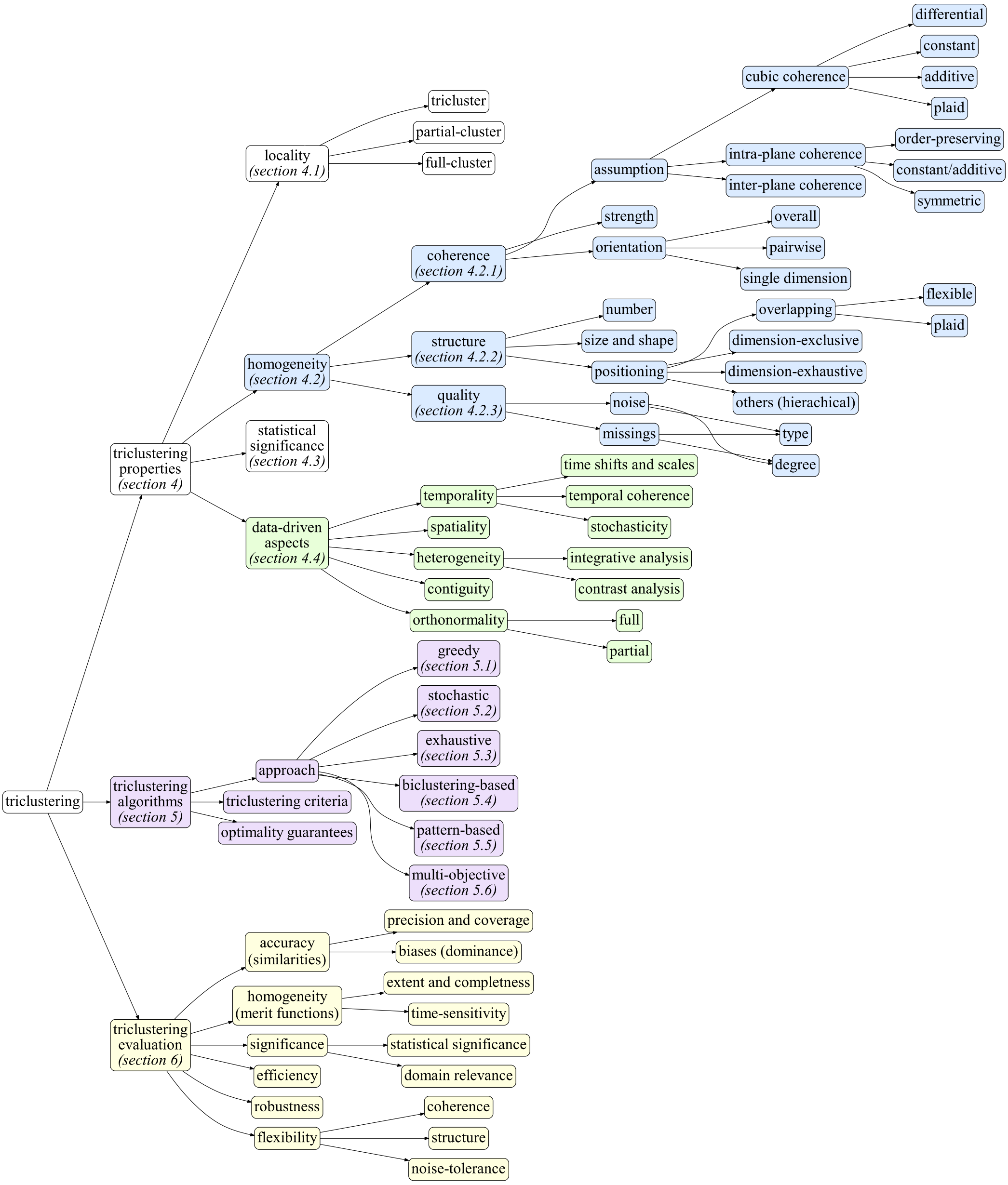


Fig. 4. Taxonomy of triclustering concepts: structured view on the aspects affecting the behavior of triclustering algorithms.

4.1 Locality

According to *Definition 2.4*, triclusters are subspaces defined by subsets of objects on all dimensions of a 3D dataset. Restrictions on the locality of subspaces can nevertheless be placed.

Definition 4.1. A **full cluster** is a subspace defined by a subset of objects from one dimension that (implicitly) contains all the objects from the remaining dimensions. A **partial cluster** is a

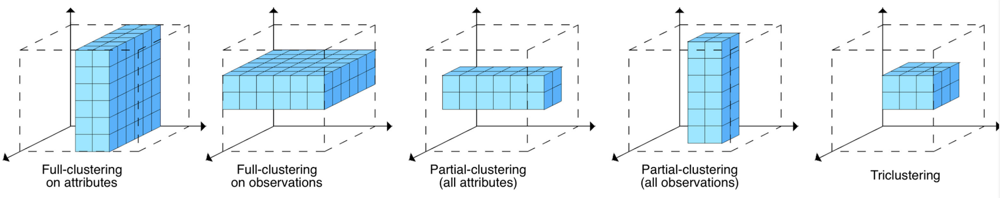


Fig. 5. Subspace clustering of 3D data with varying locality criteria: full, partial, and triclustering.

subspace defined by subsets of objects from two dimensions that (implicitly) contain all the objects from the remaining dimension.

Given a 3D dataset, the discovery of full clusters and partial clusters is here thus termed *full clustering* and *partial clustering*, respectively.

Given $\mathbf{I} \subseteq \mathbf{X}, \mathbf{J} \subseteq \mathbf{Y}$, and $\mathbf{K} \subseteq \mathbf{Z}$, then $(\mathbf{I}, \mathbf{Y}, \mathbf{Z})$, $(\mathbf{X}, \mathbf{J}, \mathbf{Z})$, and $(\mathbf{X}, \mathbf{Y}, \mathbf{K})$ are illustrative full clusters, typically referred to as clusters, while $(\mathbf{I}, \mathbf{J}, \mathbf{Z})$, $(\mathbf{I}, \mathbf{Y}, \mathbf{K})$, and $(\mathbf{X}, \mathbf{J}, \mathbf{K})$ are illustrative partial clusters.

Contrasting to full clustering, the traditional task of clustering 3D data imposes all objects of the dimension under partitioning to belong to one and only one cluster.

While some partial-clustering algorithms consider the overall values along the context dimension (Jiang et al. 2004; Supper et al. 2007), others collapse this dimension into a compact set of metrics (Sim et al. 2013). Partial clustering is useful in the following cases: (1) one dimension has a significantly low number of objects (e.g., gene-sample-time with few time points), and (2) one dimension can be meaningfully collapsed.

Still, the assumption made by full and partial clustering that subspaces should contain all objects from at least one dimension is restrictive for real-world scenarios, where meaningful correlations can only be observed for subsets of objects on all dimensions. In this context, and according to *Definition 2.4*, a *tricluster* is a subspace with locality on all dimensions, $(\mathbf{I}, \mathbf{J}, \mathbf{K})$. Figure 5 shows subspaces with varying locality criteria.

4.2 Homogeneity

The placed homogeneity defines the coherence, structure, and quality of triclusters.

4.2.1 Coherence. The coherence criterion of a tricluster is defined by the observed correlation of its values, being essentially determined by the underlying merit function. The coherence can be divided according to *assumption*, *strength*, and *orientation*.

Coherence Assumption. Coherence can be observed either among all the values of a tricluster or within and between the slices of a tricluster.

Definition 4.2.1. The **coherence assumption** determines the type of correlation between the values of a tricluster: (1) *cubic* when established among all values in a tricluster, $f(\mathbf{I}, \mathbf{J}, \mathbf{K})$; (2) *intraplane* if established for each slice of a tricluster, e.g., $\forall_{\mathbf{x}_i \in \mathbf{I}} f(\mathbf{J}, \mathbf{K}|\mathbf{x}_i)$; and (3) *interplane* when established between the slices of a tricluster.

1. Cubic Coherence. In accordance with the 3D merit functions surveyed in *Section 2.1*, different forms of cubic coherence can be considered.

Definition 4.2.2. Let $\mathbf{B} = (\mathbf{I}, \mathbf{J}, \mathbf{K})$ be a tricluster with a_{ijk} values. If $\mathbf{B}$ has categoric values, $a_{ijk} \in \Sigma$, then $a_{ijk} = c$ and $a_{ijk} = c_j$ (or c_i or c_k) correspond respectively to triclusters with a **constant symbol** and a **constant pattern**. If $\mathbf{B}$ has real values, $a_{ijk} \in \mathbb{R}$, then unexpected a_{ijk} values and $a_{ijk} = c + \eta_{ijk}$ correspond to triclusters with **differential** and **low-variance** (constant) assumptions. A real-valued tricluster $\mathbf{B}$ with values respecting $a_{ijk} = c + \alpha_i + \beta_j + \gamma_k + \eta_{ijk}$ (where

$c \in \mathbb{R}$ and α_i, β_j, and γ_k in $\mathbb{R}$ are contributions from $\mathbf{x}_i$ observation, $\mathbf{y}_j$ attribute, and $\mathbf{z}_k$ context) is said to follow a **fully additive** assumption when $\alpha_i \neq 0, \beta_j \neq 0$, and $\gamma_k \neq 0$, and **partially additive** otherwise. When the values of a tricluster are better described by $a_{ijk} = c \times \alpha_i \times \beta_j \times \gamma_k + \eta_{ijk}$, then the tricluster is said to follow a (either fully or partial) **multiplicative** assumption. A tricluster considering the cumulative effects from other triclusters follows a **plaid** assumption: $a_{ijk} = \mu_0 + \sum_{t=0}^{q} \theta_{ijkt}\rho_{it}\kappa_{jt}\tau_{kt}$, where θ_{ijkt} defines the contribution from tricluster $\mathbf{B}_t = (\mathbf{I}_t, \mathbf{J}_t, \mathbf{K}_t)$ to a_{ijk} when ρ_{it}, κ_{jt}, and τ_{kt} are true, i.e., $\mathbf{x}_i \in \mathbf{I}_t$, $\mathbf{y}_j \in \mathbf{J}_t$ and $\mathbf{z}_k \in \mathbf{K}_t$.

A real-valued tricluster with a *differential* assumption has unexpectedly high or low values. Sim et al. (2010a) replaced the differential condition by random *shaking noise profiles*, where the values in a tricluster follow an unexpected distribution (e.g., Gaussian with unexpected mean). In the context of binary 3D data, this assumption is often termed *dense* assumption (tricluster with high degree of ones) (Ignatov et al. 2015).

A generalization of these assumptions is the *constant overall* assumption, where most elements in the tricluster either have the same symbol (discrete 3D data) or are approximately constant (real-valued data), also referred to as *low -variance* triclusters (Hu and Bhatnagar 2010). The generalized *constant* assumption considers different values along the objects of a given dimension, typically the attribute dimension ($a_{ijk} \approx c_j$).

Given a real-valued or integer 3D dataset, the elements of a *fully additive* (multiplicative) tricluster with a constant assumption can accommodate shifting (scaling) factors per observation, attribute, or context in accordance with *Definition 4.2.2*. Partially additive (multiplicative) triclusters have shifting (scaling) factors on two dimensions. Alternative formulations combining shifting and scaling factors on different dimensions can be envisioned.

A *plaid* assumption can be alternatively placed according to *Definition 4.2.2* to account for meaningful overlapping effects between triclusters.

Figure 3 illustrates triclusters with constant, plaid, and additive cubic coherencies.

2. Intraplane Coherence. In accordance with the surveyed 2D merit functions, alternative forms of coherence can be formulated using the largely researched contributions in biclustering (Madeira and Oliveira 2004; Henriques and Madeira 2016). Similarly to cubic forms of coherence, differential, constant, additive, and plaid coherencies can be observed within the slices of a tricluster. In addition, the order-preserving assumption can be placed to guarantee a coherent variation of values, tolerant to misalignments. Figure 1 provides illustrative examples of these intraplane coherence assumptions.

Definition 4.2.3. Consider a slice (matrix) of a real-valued or integer 3D data space defined by the $(\mathbf{X}, \mathbf{Y})$ dimensions and let the elements in a bicluster $a_{ij} \in (\mathbf{I}, \mathbf{J})$ have coherence across rows $a_{ij} = c_j + \gamma_i + \eta_{ij}$, where c_j is the expected value on column $\mathbf{y}_j$, γ_i is the adjustment on row $\mathbf{x}_i$, and η_{ij} is the noise factor of a_{ij}. The γ factors define the intraplane coherence assumption: **constant** when $\gamma = 0$; **multiplicative** if a_{ij} is better described by $c_i\gamma_j + \eta_{ij}$ (or $c_j\gamma_i + \eta_{ij}$); and **additive** otherwise. **Symmetries** can be further allowed on rows, $a_{ij} \times s_i$, where $s_i \in \{1, -1\}$. A **plaid** assumption is sensitive to the cumulative effect of contributions from multiple biclusters on areas where their columns and rows overlap. The **order-preserving** assumption is verified when the values for each row have the same relative order on the $\mathbf{J}$ columns.

Intraplane coherence can be enforced for the slices along a single dimension, pair of dimensions, or all dimensions. For instance, LagMiner (Xu et al. 2009) imposes that the triclusters found in gene-sample-time data have intraplane coherence on each gene slice, sample slice (given by an order-preserving coherence), and time slice.

3. Interplane Coherence. This type of coherence guarantees that the different slices that compose a tricluster are correlated. By solely placing an intraplane form of coherence, dissimilar slices can be included as part of a tricluster.

Definition 4.2.4. Given a tricluster **B** described by a set of slices with intraplane coherence on one (or more) dimensions, **B** has **interplane coherence** if its slices satisfy some similarity criteria.

Ahmed et al. (2011) enforces interplane coherence based on the distance between each slice of a candidate tricluster and its planar mean representative.

Coherence Strength. Coherence strength is the criterion that guarantees whether the correlation of values within a tricluster is sufficiently strong ($|\eta_{ijk}| < \delta$ in accordance with *Definition 4.2.5*). Coherence strength can only be placed in convex domains and is thus not applicable to symbolic and integer 3D data analysis.

Definition 4.2.5. Given a real-valued 3D dataset, **A**, let $\bar{A}$ be the amplitude of a finite range of domain values. Given triclusters with values a_{ijk} described by a form of coherence plus an uncertainty/noise factor η_{ijk}, the **coherence strength** of a tricluster in **A** is a real value $\delta \in [0, \bar{A}]$, such that $\eta_{ijk} \in [-\delta/2, \delta/2]$.

The coherence strength is applicable to both (1) greedy searches (addition of objects to a subspace if δ-inequality is preserved or removal of objects until δ-inequality becomes satisfied) and (2) stochastic searches (as a stopping criterion for convergence).

The possibility to parameterize the coherence strength can be useful for some triclustering problems (Mahanta et al. 2011). For instance, when considering gene-sample-time expression data $a_{ijk} \in [-1, 1]$ (where $\bar{A} = 2$), $\delta = 2/3 = 0.(6)$ is able to distinguish three expression levels (preserved, activated, and repressed), while $\delta = 2/5 = 0.4$ is able to recognize five expression levels (further separating strong from soft regulation).

Coherence Orientation. Most cubic coherence assumptions are applied without dimension-specific considerations. An illustrative exception is the constant coherence, where the orientation is determined by the dimension with constant values (e.g., orientation on attributes when $a_{ijk} \approx c_j$). Contrasting, all forms of intra- and interplane coherence have a well-defined orientation. When considering observation slices, an intraplane coherence can be applied with orientation on attributes, contexts, or both dimensions. An interplane coherence between context slices is oriented along the context dimension.

4.2.2 Structure. In accordance with the proposed taxonomy, the structure of a triclustering solution is defined by the number, size, shape, and positioning of triclusters.

Number, size, and shape. The triclustering task can be applied either (1) with a fixed number of triclusters that maximize homogeneity, significance, and dissimilarity criteria (Gutiérrez-Avilés and Rubio-Escudero 2014b) or (2) without restrictions on the number of triclusters in an attempt to discover all triclusters satisfying the input criteria (Jiang et al. 2006).

The triclustering task may require the found triclusters to cover a minimum τ portion of data elements, $|\cup_{\mathbf{B}_i \in \mathcal{B}} \mathbf{B}_i|/(nmp) > \tau$, and can be guided by minimum size expectations, such as min$|\mathbf{I}|$, min$|\mathbf{J}|$, min$|\mathbf{K}|$, min$|\mathbf{I}|\times|\mathbf{J}|$, min$|\mathbf{I}|\times|\mathbf{K}|$, min$|\mathbf{J}|\times|\mathbf{K}|$, or min$|\mathbf{I}|\times|\mathbf{J}|\times|\mathbf{K}|$.

The *size* and *shape* of triclusters are highly dependent on the given (1) coherence assumption, strength, and orientation; (2) tolerance to noise; and (3) dimension-specific biases of the searches. Larger triclusters are associated with flexible coherence assumptions, loose coherence strength, and/or high noise tolerance.

Positioning (overlapping properties). Triclustering algorithms generally place constraints on the allowed positioning of triclusters in order to reduce the search space. These constraints determine whether the resulting structure is flexible, plaid, dimension-exclusive or exhaustive, hierarchical, or non-overlapping, among other possibilities.

Definition 4.2.6. A triclustering structure is determined by the constraints placed on the **positioning** of triclusters:

- *No constraints* (flexible structure): arbitrarily positioned triclusters
- *Plaid constraint*: triclusters with well-defined cumulative effects on overlapping areas (Mankad and Michailidis 2014)
- Dimension-specific *exclusivity constraint*: each object from a given dimension is assigned to at most one tricluster (e.g., $\forall_{\mathbf{B}_1 \in \mathcal{B} \wedge \mathbf{B}_2 \in \mathcal{B} \wedge \mathbf{B}_1 \neq \mathbf{B}_2} \mathbf{I}_1 \cap \mathbf{I}_2 = \emptyset$)
- Dimension-specific *exhaustive* constraint: each object from a given dimension belongs to at least one tricluster (e.g., $(\bigcup_{k=1}^{|\mathcal{B}|} \mathbf{I}_k) = \mathbf{X}$)
- *Hierarchical constraint*: triclusters totally contained in a larger tricluster/3D space
- *Nonoverlapping constraint* on all dimensions ($\forall_{\mathbf{B}_1 \in \mathcal{B} \wedge \mathbf{B}_2 \in \mathcal{B} \wedge \mathbf{B}_1 \neq \mathbf{B}_2} \mathbf{B}_1 \cap \mathbf{B}_2 = \emptyset$) or pair of dimensions (e.g., $\forall_{\mathbf{B}_1 \in \mathcal{B} \wedge \mathbf{B}_2 \in \mathcal{B} \wedge \mathbf{B}_1 \neq \mathbf{B}_2} (\mathbf{I}_1, \mathbf{J}_1) \cap (\mathbf{I}_2, \mathbf{J}_2) = \emptyset$)

These constraints are useful to categorize 3D subspace clustering algorithms. Illustrating, the structures targeted by the coclustering task are both exclusive and exhaustive in all three dimensions (also referred to as a chessboard structure) (Sim et al. 2010a). The hierarchical constraint is generally placed by hierarchical subspace clustering algorithms (Achtert et al. 2006).

The pursuit of an exhaustive structure should not be confused with the possible exhaustive nature of a triclustering search. In addition, the non-overlapping constraint differs from dimension exclusivity since a triclustering solution might be composed of nonoverlapping triclusters and break the exclusivity constraint on all dimensions.

Although non-overlapping and exclusivity constraints are restrictive, they can be found among different triclustering approaches, including (1) algorithms that find one tricluster at a time and remove or mask its area with random noise for subsequent discoveries (Waltman et al. 2010); (2) most algorithms relying on reoderings of objects or covariance matrices (Schepers et al. 2006); and (3) stochastic algorithms that parametrically model triclusters disregarding their interdependencies (Gerber et al. 2007).

Flexible structures, such as those produced by algorithms with pattern-based merit functions (Ji et al. 2006), are non-exhaustive, are non-exclusive, and allow for overlaps.

Plaid constraints define the overlapping properties of a triclustering solution. These constraints are often placed to model biological or social entities that simultaneously participate in multiple modules or communities (triclusters). A triclustering structure with plaid constraints is characterized by the:

- overlapping degree per dimension (degree of overlapping objects), slice (degree of overlapping areas), or all dimensions (degree of overlapping volume);
- composition functions that explain the cumulative contributions, such as the additive plaid formulation (Equation (4)) or alternative formulations sensitive to nonlinear effects (Henriques and Madeira 2015); and
- distribution of overlaps (whether explained by pairwise or multiwise contributions).

4.2.3 Quality. In symbolic and integer 3D data, a tricluster might have symbols that differ from the expected pattern. In real-valued 3D data, although the natural oscillation of values within a tricluster is bounded by the placed coherence strength ($|\eta_{ijk}| \leq \delta$), some values in a tricluster might deviate from expectations ($|\eta ijk| > \delta$). Understandably, the presence of a few noisy elements can lead to the undesirable exclusion of relevant observations, attributes, and contexts from a tricluster. In this context, the triclustering task can further place robustness considerations.

Definition 4.2.7. The **quality** of a tricluster is defined by the tolerated type and amount of noisy and missing elements. Given a noisy real-valued 3D dataset, (1) deviations on η_{ijk} can be bounded, $|\eta_{ijk}| < \epsilon - \frac{\delta}{2}$; (2) the average error of a single tricluster can be bounded, $(\frac{1}{|\mathbf{I}||\mathbf{J}||\mathbf{K}|} \sum_{\mathbf{x}_i \in \mathbf{I}} \sum_{\mathbf{y}_j \in \mathbf{J}} \sum_{\mathbf{z}_k \in \mathbf{K}} |\eta_{ijk}|) < \epsilon - \frac{\delta}{2}$; and/or (3) the average error of a set of triclusters can be bounded to allow deviations while still establishing a delineate desirable level of coherence strength.

Robustness to noise. Given a real-valued or integer tricluster, two major forms of robustness to noise can be considered: (1) sensitivity to values in a tricluster showing slight deviations from the expected coherence (Sim et al. 2010a) and (2) ability to accommodate a certain distribution of noisy elements independently of their noise factor (Ignatov et al. 2015). In the context of symbolic 3D data, the first criterion is not applicable. The quality of a tricluster is determined not only by the criteria placed on the search but also by preprocessing and postprocessing stages. Normalization and discretization are common preprocessing steps impacting quality (Supper et al. 2007; Tchagang et al. 2012; Sim et al. 2013). The possibility to postprocess triclusters by merging, filtering, extending, and/or reducing triclusters is often considered to calibrate their quality (Zhao and Zaki 2005; Xu et al. 2009).

The desirable quality might not be enforced per tricluster but assessed for a group of triclusters, thus allowing some triclusters to deviate from quality expectations.

Robustness to missing elements. Although most state-of-the-art triclustering algorithms are not able to learn from sparse 3D data (Zhou et al. 2014; Henriques 2016), different strategies can be established based on the locality and amount of missings:

1. Imputation procedures for 3D data (Donders et al. 2006) (considering the values of the closest observations, attributes, and contexts from the missing element)
2. Accommodation of missing elements within triclusters (with bounded tolerance) using a dedicated interpretation, similarly to mature biclustering algorithms prepared for sparse data analysis (Henriques and Madeira 2016)
3. Within pattern mining-based approaches by replacing a missing value by a range of values or multiple symbols, in accordance with principles in Henriques and Madeira (2014).

4.3 Statistical Significance

The triclustering problem is further challenged by the need to guarantee that the retrieved triclusters deviate from expectations. In other words, ensure that the probability of a tricluster to occur against a null data model is unexpectedly low.

In this context, statistical assessments are necessary to measure and minimize the risk of *false-positive triclusters* (retrieving triclusters that appear by chance on the sample data) without increasing the risk of *false-negative triclusters* (excluding relevant triclusters).

Despite the relevance of this criterion, there is not yet an agreed ground truth on how to verify and promote the statistical significance of triclustering solutions. In fact, most triclustering algorithms are guided by merit functions that explore the homogeneity of triclusters but do not subject them to a sound statistical assessment (Moise and Sander 2008a; Henriques 2016) (see Table 2). Understandably, optimizing homogeneity is of limited use since good homogeneity levels can appear by chance in the observed data (common for small triclusters).

In what follows, we survey some of the few contributions toward this goal.

Moise and Sander (2008a) defined a tricluster as statistically significant when the distribution of its values a_{ijk} diverges from the Gaussian distribution approximated for the overall 3D data space. This statistical test is only able to assess specific types of triclusters and is only suited for 3D data well approximated by (a possibly multivariate) Gaussian distribution.

Sim et al. (2010a) proposed a correlation information metric to find triclusters with unexpected values yet independent of the data distribution. To this aim, a probability density function (pdf) for the 3D space is approximated using kernel density estimation (shown to be able to converge to the true pdf). Integral calculus is then used to test the probability of the values in a tricluster to deviate from the expected pdf.

Inspired by the statistical principles proposed by Moise and Sander (2008b), significance thresholds are dynamically adjusted in Sim et al. (2010b). Similarly, Li and Tuck (2009) relied on automatic boundary searches to detect the boundary between non-significant and significant triclusters by iteratively correcting thresholds.

Tchagang et al. (2012) extended the statistical tests originally proposed by Ben-Dor et al. (2003) to assess order-preserving biclusters. Assuming 3D data to have i.i.d. $\mathbf{Z}$ contexts (time points) according to a uniform distribution, the probability that a tricluster contains an order-preserving slice with $n \geq |\mathbf{I}|$ observations is given by $P(x \geq p) < \alpha/n!$, where $x \sim \text{Bin}(1/n!, |\mathbf{Z}|)$ and α is the significance threshold. In this context, a tricluster is statistically significant if its slices are statistically significant.

Finally, Mankad and Michailidis (2014) proposed a resampling procedure on the input 3D data to obtain a null data model to statistically test triclusters. The size of the candidate tricluster is compared against the size of triclusters found in randomized data, being considered statistically significant if its size deviates from expectations.

As the aforementioned statistical tests are deemed for triclusters with very specific forms of coherence, additional statistical views are expected in upcoming research.

4.4 Data-Driven Aspects

Real-world 3D data is abundant and diverse. We now pinpoint data-related specificities with impact on the triclustering task: size, regularities, temporality, spatiality, heterogeneity, contiguity, and orthonormality.

4.4.1 Size and Regularities. In terms of *size*, 3D data can vary with regard to *volume* (determined by the number of observations, attributes, and contexts) as well as by the *ratio* between each pair of dimensions ($|\mathbf{X}|/|\mathbf{Y}|$, $|\mathbf{X}|/|\mathbf{Z}|$, and $|\mathbf{Y}|/|\mathbf{Z}|$). In this context, triclustering algorithms should pursue scalability principles to mine very large subspaces while preventing biases toward specific dimensions when 3D data have imbalances on the number of objects per dimension (Gutiérrez-Avilés and Rubio-Escudero 2014b). In the presence of temporal 3D data, Tchagang et al. (2012) further categorize triclustering algorithms based on whether they are deemed for short or long multivariate time series according to the number of time points (contexts).

The data *regularities* are defined by the domain of attributes (whether real-valued, symbolic, or nonidentically distributed) and by the distribution of values. 3D data values may be well described by a single univariate distribution or object-conditional distributions. Understandably, the desirable homogeneity and statistical significance criteria should depend on the regularities of the input data.

4.4.2 Temporality. A large portion of currently available 3D data results from the periodic monitoring of biological, individual, and societal systems. The intrinsic temporal nature of such 3D data presents unique challenges, including the need to (1) place adequate homogeneity criteria to capture meaningful forms of temporal progression, (2) handle arbitrarily high temporal lags on observations, (3) place proper contiguity criteria, and (4) deal with the complex stochasticity inherent to temporal 3D data.

First, triclusters are only properly captured when coherence is both verified within and between the time slices. Although algorithms based on 3D merit functions satisfy this requirement, only a few of the remaining algorithms address this issue (Xu et al. 2009). Illustrating, the first triclustering algorithm disregards coherence between time slices (Zhao and Zaki 2005), offering no guarantees of meaningful temporal progression.

Second, shifts and scales can be observed on observations. For instance, when analyzing gene-sample-time data, the responsiveness of a set of coregulated genes may differ (both in starting time and duration). Observations in the context of patient-record-time data may show arbitrarily high time shifts in accordance with the health profile of the individual. Individual-individual-time data often capture arbitrary delayed behavioral responses to a social event. In this context, triclustering searches might break the orthonormality constraints.

Third, contiguity can be optionally pursued on the time dimension given the high probability of homogeneous subspaces to be verified along time intervals. Nevertheless, gap-based relaxations or the removal of contiguity constraints can be considered for a more flexible exploration of noise-susceptible data contexts.

Finally, the inherent stochasticity of complex evolving systems are associated with misalignments that should be properly handled. In this context, triclustering tasks should place flexible coherence assumptions (Ahmed et al. 2011), such as order preserving, and adequate merit functions based on largely surveyed similarity functions between multivariate time series, in particular similarity criteria sensitive to frequency, temporal, or geometric misalignments (Ding et al. 2008).

4.4.3 Spatiality. 3D data can be further categorized according to whether they are spatial or not. Examples of spatial 3D data candidates for triclustering include video data (coordinate-coordinate-time), structural molecular data (residues-position-time), imaging data (observations-space-space), geophysical data, or location data (observations-contiguous locations-time) (Guigourès et al. 2015; Achtert et al. 2006). Triclustering-related tasks have been abundantly researched in the context of computer vision, primarily propelled by image and video segmentation (Fu and Mui 1981; McInerney and Terzopoulos 1996; Tremeau and Borel 1997; Zhu et al. 2016). In this context, triclustering is applied with the aim of finding homogeneous regions in the 3D space. As such, it often imposes the discovery of contiguous objects on the spatial dimensions. In the presence of a time dimension, shifts might be additionally observed with the purpose of detecting moving or adaptive regions.

4.4.4 Heterogeneity. Heterogeneous 3D data are 3D data with observations extracted from different sources or collected under different experimental settings. Species-gene-sample expression data, patient-record-time data for patients monitored in different hospitals, or biological networks from different knowledge bases are notable cases. In this context, two major tasks can be envisioned: integrative and contrast analysis.

Definition 4.4.5. Given a heterogeneous 3D dataset, **integrative triclustering** aims to find triclusters with observations $\mathbf{x}_i$ satisfying specific $(\mathbf{x}_i, \mathbf{J}, \mathbf{K})$-plane coherence and similarity criteria, $\forall_{\mathbf{x}_{i_1}, \mathbf{x}_{i_2} \in \mathbf{I}} (\mathbf{x}_{i_1}, \mathbf{J}, \mathbf{K}) \approx (\mathbf{x}_{i_2}, \mathbf{J}, \mathbf{K})$, while **contrast triclustering** aims to find triclusters with coherent observations satisfying specific deviating/dissimilarity criteria, $\forall_{\mathbf{x}_i \in \mathbf{I}} \forall_{\mathbf{x}_l \in \mathbf{X} \wedge l \neq i} (\mathbf{x}_i, \mathbf{J}, \mathbf{K}) \not\approx (\mathbf{x}_l, \mathbf{J}, \mathbf{K})$.

Integrative analysis can be applied using traditional triclustering principles, with particular attention on the placed homogeneity to tolerate differences explained by biases pertaining to the data source or collection procedure. Preprocessing options can be further considered to minimize source-specific biases. In the context of *contrast analysis*, the coherence of each observation from a given tricluster necessarily differs from remaining observations. In previous work (Hu and

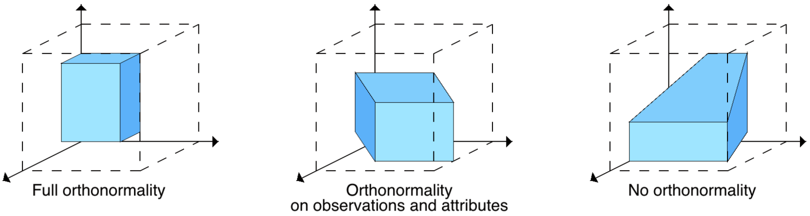


Fig. 6. 3D subspaces with orthonormality constraints.

Bhatnagar 2010), a tricluster is seen as a composition of slices differing with regard to their average value.

Additional variants for triclustering heterogeneous data can be found in the literature. Supper et al. (2007) define three types of triclusters: (1) general responses where similarity between observations is enforced, (2) independent responses where observations are not required to be similar or dissimilar as long as the intracoherence of each observation is guaranteed, and (3) specific responses where dissimilarity between observations is enforced. These options offer a compromise between integrative analysis and contrast analysis of heterogeneous 3D data.

4.4.5 Contiguity. Contiguity constraints are often applied on time and spatial dimensions, meaning that the selected subset of time points or coordinates per tricluster forms a convex set (Jiang et al. 2006). Contiguity is used to enhance the consistency and interpretability of the results and reduce the complexity of the triclustering task. The contiguity condition can be relaxed or even disregarded since triclusters can be also meaningfully described by non-contiguous objects as long as the underlying homogeneity captures some form of temporal progression or spatial meaning (Bhar et al. 2012; Amar et al. 2015). The allowance of gaps between contiguous objects (Ji et al. 2007) is a common relaxation to contiguity constraints that can be placed to tolerate local inconsistencies while still preserving desirable homogeneity criteria.

4.4.6 Orthonormality. Orthonormality is by default considered within the triclustering task to guarantee that a tricluster is well defined by a triplet $(\mathbf{I}, \mathbf{J}, \mathbf{K})$. A tricluster is thus fully orthonormal by default. However, full orthonormality may be undesirable for specific data contexts. For instance, temporal 3D data characterized by arbitrarily high temporal lags is better described by the pair $(\mathbf{I}, \mathbf{J})$ since the time points where coherence is observed vary for each observation.

Definition 4.4.8. A tricluster is **fully orthonormal** (also referred as axis parallel) when the subset of objects for a given dimension (either $\mathbf{I}$, $\mathbf{J}$, or $\mathbf{K}$) is preserved across the remaining dimensions, and it is **partially orthonormal** when the previous condition is not preserved in one dimension (e.g., $\mathbf{B} = \cup_{\mathbf{x}_i \in \mathbf{I}}(\mathbf{x}_i, \mathbf{J}, \mathbf{K}_i)$).

Partially orthonormal triclusters are proposed in Xu et al. (2009) for gene-sample-time expression data analysis, where a tricluster is described by coherent time-sample curves orthonormal to the gene axis, coherent gene-time curves orthonormal to the sample axis, and coherent gene-sample curves nonorthonormal to the time axis due to temporal lags. Partial orthonormality also can be observed for movie data (position-position-time) analysis to capture a sliding object (e.g., aircraft) in a subset of contiguous frames. Figure 6 illustrates triclusters with varying degrees of orthonormality.

5 TRICLUSTERING ALGORITHMS

Given the numerous aspects of the triclustering task (diversity of data inputs, behavioral options, desirable homogeneity criteria), a high number of triclustering algorithms have been proposed in the last decade. These algorithmic efforts can be divided according to (1) whether their behavior is based on iterative searches (*greedy; Section 5.1*) or on distribution parameter identification (*stochastic; Section 5.2*) and (2) whether they are able to offer guarantees of optimality (*exhaustive; Section 5.3*) or not. Each approach can be further categorized according to whether the behavior relies on biclustering algorithms (*Section 5.4*), pattern mining procedures (*Section 5.5*), or evolutionary multiobjective optimization (*Section 5.6*), among other possibilities.

Triclustering algorithms can also be classified according to whether they aim to identify one or multiple triclusters, and in this latter case whether they identify a tricluster at a time (Ahmed et al. 2011; Waltman et al. 2010), a subset of all triclusters at a time (Amar et al. 2015), or all triclusters at a time (common case). Finding one or a subset of triclusters at a time implies that the algorithm is able to either (1) shift the focus into new subspaces or (2) mask previous discoveries.

The way homogeneity criteria are pursued further determines the susceptibility of triclustering searches to different biases, explaining why the found triclusters might differ from the desirable set of statistically significant and dissimilar triclusters. Often, these biases lead to imbalance in the number of selected objects per dimension, creating forms of *dimension dominance*. For instance, the first triclustering algorithm (Zhao and Zaki 2005) finds large 2D subspaces per context slice without considering their impact on candidate triclusters, thus being prone to context dominance. It can thus miss smaller 2D subspaces of observations and attributes that are part of triclusters with a number of contexts above average.

5.1 Greedy Approaches

Cubic forms of coherence can be used within iterative schemes where objects are incrementally added and removed from candidate subspaces in order to (locally) maximize/satisfy a predefined 3D merit function (*Section 2.1*). We discuss two major classes of these greedy approaches: divide-and-conquer and seed growth.

5.1.1 Divide and Conquer. Bhar et al. (2012) proposed the use of the 3D mean-squared residue (MSR) within a greedy approach, termed *δ-TRIMAX*, to find triclusters with low MSR (according to (3)) in temporal 3D data. This is performed in two steps: (1) starting from the whole 3D space, objects (observations, attributes, and contexts) are iteratively deleted from the tricluster until its MSR becomes lower than a prespecified threshold δ, and (2) new objects are attemptively added to the tricluster if they are able to preserve MSR $< \delta$. Following similar principles, the *TriWClustering* (Three-Way Clustering) algorithm (Dede and Oğul 2013) and the subsequent TriClust tool (Dede and Oğul 2014) were proposed to find triclusters with low 3D MSR.

Li and Tuck (2009) proposed *TRI-Clustering* (3D Regulated expression Iterative Clustering) to analyze gene-sample-regulator data. TRI-Clustering uses a divide-and-conquer procedure with an Automatic Boundary Search (ABS) to discover statistically significant triclusters with a 2D differential coherence on all object slices of a candidate subspace. To this end, it starts with a fixed number of randomly generated seeds and iteratively trims triclusters (according to their merit) until convergence. Contrasting with δ-dependent approaches, ABS is able to preserve flexibility (by iteratively adjusting the percentage of candidate differential elements until no changes are observed between two iterations). Despite its merits, the differential coherence is restrictive for most triclustering applications and the number of triclusters needs to be prespecified.

5.1.2 Seed Growth. An alternative schema is to find relevant seeds (building blocks) within 3D data and greedily merge and extend them to build candidate triclusters. Sim et al. (2010a) proposed MIC (MIning 3D subspace Clusters) to find statistically significant triclusters in temporal 3D data with an approximate constant coherence. These triclusters, referred to as Correlated 3D Subspace Clusters (CSCs), show high correlation within each context slice and between each pair of contiguous slices. To this end, MIC first generates seeds (pairs of correlated objects) with residual p-values. Then it grows CSCs from these seeds by greedily extending and merging seeds that maximize the proposed correlation scores. Unlike most triclustering algorithms, MIC is parameter-free and enforces statistical significance criteria (*Section 4.3*).

Xu et al. (2009) proposed *LagMiner* to find S^2D^3-Clusters (shift-and-scale correlated triclusters) in gene-sample-time expression data. Given a coherence strength δ, LagMiner aims to find triclusters that satisfy (1) a shifting-and-scaling intraplane coherence for each time point (gene-sample plane) and observation (time-gene plane) using the S^2Score in Equation (7) – $\forall_{z_k \in K} S^2Score(I, J, z_k) \leq \delta$ and $\forall_{x_i \in I} S^2Score(K, J, x_i) \leq \delta$ –, and (2) an order-preserving intraplane coherence within each sample (gene-time plane). Given an ordered triplet of attributes $\rho(y_j) = y_{j1} \prec y_{j2} \prec y_{j3}$ for a slice:

$$S^2Score(\mathbf{I}, \mathbf{J}, \mathbf{z}_k) = max_{\mathbf{x}_i \in \mathbf{I} \rho(y_j) \subseteq \mathbf{J}} \frac{a_{ij_2k} - a_{ij_1k}}{a_{ij_3k} - a_{ij_1k}} - min_{\mathbf{x}_i \in \mathbf{I}, \rho(y_j) \subseteq \mathbf{J}} \frac{a_{ij_2k} - a_{ij_1k}}{a_{ij_3k} - a_{ij_1k}}. \tag{7}$$

LagMiner starts by discovering subspaces described by a triplet of samples with as many genes as possible. Then, it proceeds by including more samples and time points that satisfy the introduced forms of coherence. Triclusters are thus enforced to have coherent time-sample objects per gene slice (orthonormal to the gene axis), coherent gene-time objects per sample slice (orthonormal to the sample axis), and coherent gene-sample objects per time slice (not necessarily orthonormal to the time axis due to the allowed time lags). Nevertheless, by enforcing three forms of intraplane coherence, S^2D^3-Clusters are often small and are susceptible to the risk of being described by simplistic forms of correlation such as low variance.

5.2 Stochastic Approaches

Amar et al. (2015) proposed *TWIGS* (Three-Way module Inference via Gibbs Sampling) to find large triclusters in temporal 3D data, where observations may be temporally misaligned. Due to the presence of arbitrarily high time lags, a tricluster (referred to as a core module) is primarily defined by subsets of observations and attributes, $\forall_{\mathbf{x}_i \in \mathbf{I}} (\mathbf{x}_i, \mathbf{J}, \mathbf{K}_i)$. TWIGS uses a hierarchical Bayesian model with a Bernoulli-β assumption for binary 3D data and a Normal-γ assumption for real-valued 3D data. A biclustering solution is found per observation slice as a starting point and then iterative improvements are performed in accordance with the Bayesian model. Gibbs sampling is then applied to infer the candidate triclusters from the learned membership vectors. TWIGS is further able to detect observation-specific augmentations of a tricluster to capture additional attributes for some observations as long as coherence is preserved. Despite being a pioneer in its probabilistic formulation, TWIGS imposes the homogeneity of triclusters to be characterized by two Gaussian distributions and can thus miss large overlapping subspaces within the 3D data space.

Gerber et al. (2007) proposed a different stochastic approach, *GeneProgram*, consisting of three major steps: (1) discretization of real-valued 3D data, (2) Markov Chain Monte Carlo sampling to estimate the model posterior probability distribution of subspaces (accomplished with a three-level hierarchy of Dirichlet processes), and (3) summarization of the learned posterior distributions to infer the final triclusters.

Mankad and Michailidis (2014) proposed the triclustering plaid model introduced in *Section 2.1*, here termed *3D-Plaid*. The goal is to model the entire data structure as a set of overlapping triclusters (according to Equation (4)). The algorithm detects subspaces exhibiting strong deviations and estimates their dependence with an iterative procedure aiming to minimize the sum of squares of the 3D data values. To this end, a background layer is first fitted and triclusters are then added one at a time until no more statistically significant triclusters can be found under a given permutation test.

MultiFacTV (Li et al. 2013) is an alternative stochastic triclustering algorithm for temporal 3D data under a plaid assumption. MultiFacTV uses a tensor factorization objective function (holding an approximate constant coherence assumption with plaid effects), together with a time-axis regularization (enforcing subspaces to have objects as contiguous as possible in the time dimension). Triclusters are then derived from the factorized tensors. Although pioneer on bridging tensor factorization and triclustering views, MultiFacTV requires the specification of the target number of triclusters and thresholding parameters per dimension. *Pacifier* (PAtient reCord densIFIER) (Zhou et al. 2014) combines low-rank factorization with block coordinate descent optimization to explore the latent structure of sparse 3D data. An alternative tensor factorization method for triclustering image-tag-user data analysis, named TTC (triclustered tensor completion), was more recently proposed (Tang et al. 2017). TTC relies on three novel principles to (1) minimize the computational challenges of large-scale tensor factorization, (2) allow assumptions on the independence among subtensors, and (3) solve the tensor completion problem by selecting and postrefining the discovered subspaces.

Recently, Wu et al. (2018) developed the Bregman cuboid average triclustering algorithm with I-divergence (*BCAT_I*) for the analysis of 3D georeferenced time series. BCAT_I partitions the 3D space in a set of coclusters by optimizing the loss of mutual information using I-divergence and subsequently refines the gathered subspaces using k-means to capture triclusters unraveling spatiotemporal patterns of interest.

In the context of coclustering tasks, Heylen et al. (2016) proposed *2M-KSC* (two-mode K-Spectral Centroid) for patient-symptom-time data. 2M-KSC clusters individuals $\mathcal{I} = \{\mathbf{I}_1, \ldots, \mathbf{I}_n\}$ and symptoms $\mathcal{J} = \{\mathbf{J}_1, \ldots, \mathbf{J}_m\}$ for all time points $\mathbf{K} = \mathbf{Z}$, such that the resulting subspaces show coherent time profiles (same pattern but eventually varying in amplitude). The 3D data values are described by $\sum_{c_1 \in \mathcal{I}} \sum_{c_2 \in \mathcal{J}} p_{ic_1c_2} p_{jc_1c_2} f(\{a_{ij1}, \ldots, a_{ijp}\}) + \eta_{ijK}$, where $f(\{a_{ij1}, \ldots, a_{ijp}\})$ defines the time profile, and $p_{ic_1c_2}$ and $p_{jc_1c_2}$ are respectively the probability of patient $\mathbf{x}_i$ and symptom $\mathbf{y}_j$ to belong to the subspace given by partitions c_1 and c_2. The final subspaces are inferred from the membership vectors obtained by minimizing the error η_{ijK} using a least squares loss function. The *Three-Mode Partitioning* algorithm (Schepers et al. 2006) also minimizes a least squares loss function between 3D data values and a parametric coclustering model. Contrasting with 2M-KSC, it is able to simultaneously partition the three dimensions.

An alternative class of stochastic algorithms maps triclustering algorithms into the task of clustering tripartite graphs. Guigourès et al. (2015) proposed a maximum a posteriori algorithm for coclustering temporal 3D data, referred to as *MAP*, where the distributions of values on the edges of the graph show similar connectivity patterns. The learned coclusters can be postprocessed to attemptively answer triclustering tasks.

Lin et al. (2009) proposed *MetaFac* (Meta-graph Factorization) to extract communities (partial clusters) from hypergraphs able to capture multiwise interactions between sets of objects along time using non-negative multitensor factorization.

5.3 Exhaustive Approaches

Contrasting with previously surveyed approaches, some triclustering algorithms rely on the exhaustive exploration of the 3D search space. Restrictive homogeneity criteria are commonly placed to guarantee a tractable computational complexity. As such, these algorithms could be better referred as quasi-exhaustive. Jiang et al. (2004) proposed an algorithm, here referred to as *GST-Miner*, to find subspaces from temporal 3D data spanning all time points. Pearson correlation coefficient is used in GST-Miner to discover either the maximal coherent sample subset per gene or maximal coherent gene subset per sample. This coherence is applied within a recursive depth-first search that relies on an enumeration tree of maximal coherent subsets of samples and gene slices to infer the candidate triclusters.

The discovery of triclusters in binary 3D data, termed triadic formal concepts (closed ternary sets), was first proposed by Krolak-Schwerdt et al. (1994) and Lehmann and Wille (1995), a decade before the first wave of triclustering algorithms for real-valued 3D data analysis. The discovery of triclusters is here guided by quality scores based on the density, cardinality, coverage, and diversity of the subspaces. The work by Ignatov et al. (2015) formalizes and experimentally compares five triclustering algorithms for binary 3D data using principles from formal concept analysis. All these algorithms have polynomial computational time on data volume. The *Trias* algorithm was initially proposed to search for dense triclusters without noise (Jäschke et al. 2006), and *OAC-triclustering* (based on box operators) as a noise-tolerant extension (Ignatov and Kuznetsov 2009). More recently, *Tribox*, *SpecTric*, *OAC-triclustering* (based on primes of pairs), and *Krimp-triclustering* were proposed to overcome some of the drawbacks of previous algorithms, such as pattern explosion and quality (Ignatov et al. 2015; Mirkin and Kramarenko 2011; Yurov and Ignatov 2017). As prime-based operators satisfy single-pass and linearity search conditions, OAC-triclustering was recently extended within a new algorithm, here referred to as *Online OAC*, that further explores efficiency gains and can be parallelized (Gnatyshak 2015). A generalization of this task for the discovery of *n*-ary formal concepts from *n*-dimensional data was proposed by Cerf et al. (2009) with the *Data-Peeler* algorithm and later extended to tolerate noise (Cerf et al. 2013).

Graph-based approaches have been alternatively proposed for the exhaustive discovery of subspaces. Palla et al. (2007) proposed a clique percolation search for 3D data given by temporal networks by exploiting the overlap of nodes in contiguous snapshots to study their evolution. Sun et al. (2007) proposed an information-theoretic approach, termed Graphscope, to segment snapshots using the minimum description length framework within an agglomerative process. Hopcroft et al. (2004) proposed a two-stage triclustering algorithm, originally deemed for the analysis of evolving nodes within a temporal network (sequence of graphs), where similarity is given by the cosine of vectors. Patterns of node progression across snapshots are investigated by comparing similar subspaces in different snapshots.

5.4 Biclustering-Based Approaches

2D merit functions can be applied to discover a set of biclusters for each slice of a 3D data space, from which a set of triclusters (satisfying some form of intraplane coherence) can be inferred. These approaches, here referred to as biclustering-based approaches, can either be greedy, stochastic, or quasi-exhaustive depending on the underlying algorithms used for the discovery of biclusters and consensus. Intraplane coherence can be placed for more than one dimension and optionally complemented with a form of interplane coherence to further promote the correlation between the slices of a subspace.

5.4.1 Quasi-Exhaustive. Zhao and Zaki (2005) proposed the first triclustering algorithm, termed *TriCluster*, for gene-sample-time data analysis. TriCluster uses Pearson correlation to place intraplane coherence on context slices and guarantee tolerance to value shifts. To this end, TriCluster (1) slices the 3D space along the time dimension and for each time point constructs a multigraph to store similar value ranges between all pairs of samples, (2) exhaustively searches for maximal cliques in these multigraphs to find a set of biclusters per time point (boosted by a symmetry constraint), and (3) composes triclusters by merging similar biclusters from different time points by finding maximal cliques in a new graph (where nodes correspond to biclusters). Despite being pioneer, it disregards intertemporal coherence (genes or samples may not coherently vary across time yet appear in sequent slices of a tricluster) and is sensitive to parameterizations.

Shortly after, Jiang et al. (2006) proposed *gTricluster* to guarantee intertemporal coherence and parameter insensitivity. gTricluster slices the 3D space into observation/gene slices and uses the Spearman correlation coefficient across time points to capture more flexible forms of intraplane coherence than Pearson correlation. Similarly to TriCluster, gTricluster searches for maximal cliques using dedicated data structures (set enumeration trees) to identify candidate biclusters, from which triclusters are exhaustively inferred. Despite its strengths, gTricluter neglects intergene coherence and triclusters can be biased due to sample and gene size dominance.

Hu and Bhatnagar (2010) proposed *low-variance 3clustering* to find all maximal 2D subspaces with variance below an upper bound. These 2D subspaces are maintained within a prefix-tree structure to detect maximum overlapping areas for the subsequent inference of triclusters. Data subspaces with significant value differences across observations are removed to minimize the computational complexity of the search.

Wang et al. (2010) proposed the *TD-Clustering* algorithm to find time-delayed scaling, shifting or inverting triclusters in temporal 3D data. The algorithm is a depth-first algorithm (classified as quasi-exhaustive due to the presence of pruning strategies) that constructs and traverses a tree structure to mine all maximal biclusters (referred to as bi-td-clusters) for each attribute slice that satisfies all the given thresholds: minimum number of observations and time points and coherence strength δ. Merging procedures are used with sliding and jumping windows to accommodate time delays. A tree structure is then used to infer all maximal tri-td-clusters above a prespecified number of attributes with delineated interplane coherence.

5.4.2 Approximate. Ahmed et al. (2011) proposed *SubCubeMiner* to discover triclusters by performing order-preserving biclustering per context slice (approximate search) and extending these biclusters along the time dimension using the planar mean residue similarity (Equation (6)). A connectivity-based approach with dynamic ranges is then applied to infer the final triclusters.

Contrasting, *OPTricluster* (Tchagang et al. 2012) aims to find triclusters where the order-preserving coherence is verified on attribute slices. OPTricluster consists of five main steps: (1) data quantization, (2) ranking of gene expression along time, (3) order-preserving biclustering, (4) graph-based inference of triclusters (using TriCluster's procedures), and (5) statistical significance assessment (using the order-preserving statistical tests in *Section 4.3*). OPTricluster further removes the contiguity constraint on the time dimension and the placed coherence guarantees less susceptibility to variations caused by measurement procedures and gene responsiveness. Depending on the goal, OPTricluster can be applied with swapped dimensions. Illustrating, by swapping the time and sample dimensions, OPTricluster is able to capture groups of genes with expression coherently varying on a subset of samples for specific time points.

Multi-Species C-Monkey (*MSCM*) (Waltman et al. 2010) was proposed for triclustering species-gene-time data. Each tricluster is discovered at a time, where a tricluster is a composition of species-specific biclusters greedily grown from (semi)random seeds (given by orthologous genes) using an optimization algorithm. Changes to the tricluster are applied by iteratively adding or removing objects according to a conservation score until convergence (no improvements can be applied). When a tricluster converges, new triclusters are randomly seeded and built until no triclusters are discoverable or a prespecified maximum number of triclusters is found. In this context, a tricluster is termed perfectly conserved if all its orthologous genes are likely to be coherently found in all the related species.

Supper et al. (2007) proposed EDISA (Extended Dimension Iterative Signature Algorithm), a greedy algorithm to find partial clusters ($\mathbf{I}, \mathbf{J}, \mathbf{K} = \mathbf{Z}$) in gene-sample-time data. Three criteria of homogeneity are considered: (1) triclusters with independent response profiles (genes coregulated under a set of conditions yet with possibly different response patterns under each condition),

(2) triclusters with similar responses under a set of conditions (a subset of previous ones), and (3) condition-specific response (a more rare subset of the first set of triclusters). These types of triclusters are respectively termed response-independent, coherent, and response-specific modules. Pearson correlation is considered to quantify the similarity between the profiles of two genes. The EDISA algorithm is inspired in the ISA biclustering algorithm (Ihmels et al. 2004): initial modules of samples and genes are greedily formed according to their overall time responses and then refined by removing genes and samples until the subspaces respect a criterion of homogeneity. An extension step is then applied to ensure gene and sample maximality.

Similarly, Meng et al. (2009) proposed a new triclustering algorithm based on the ISA biclustering algorithm—*ECTDISA* (enrichment-constrained time-dependent iterative signature algorithm)—to identify temporal transcription modules in gene-sample-time data. Contrasting with EDISA, ECTDISA applies a sliding time window to capture temporal dependencies and imposes an enrichment constraint to guarantee the biological significance of the discovered triclusters (TTMs).

More recently, Kakati et al. (2016) proposed a distributed triclustering algorithm, here termed *SS-SimTri* (Shifting-and-Scaling Similarity Triclustering), to identify coexpressed patterns with shifts and/or scales in gene-sample-time expression data. A seed-growth algorithm is proposed to extract biclusters from each time slice in parallel. A "shared nothing" client-server architecture is then proposed to guarantee a cost-effective distribution of the computational load, opening new considerations on the development of scalable triclustering algorithms.

5.5 Pattern-Based Approaches

Pattern-based triclustering approaches have been alternatively proposed to discover subspaces with well-defined patterns on a subset of objects. Ji et al. (2006) introduced the notion of Frequent Closed Cube (FCC), generalizing the concept of 2D frequent closed patterns. An FCC is a maximal tricluster in a symbolic (or discretized) 3D dataset. RSM and CubeMiner algorithms (Ji et al. 2006) were proposed to find FCCs in 3D data. *RSM* (Representative Slice Mining) mines each slice using frequent closed pattern mining, infers frequent cubes (similarly to biclustering-based approaches), and prunes cubes that are not closed. Contrasting, *CubeMiner* operates on the 3D space directly by exhaustively and recursively splitting the space according to well-defined monotonic support constraints.

Liu et al. (2010) proposed the *3D-TDAR-Mine* algorithm to identify triclusters given by Temporal Dependency Association Rules in gene-sample-time data, revealing regulatory relations between genes. Contrasting with peer pattern-based algorithms, 3D-TDAR-Mine does not require discretization and the placed coherence is able to accommodate shifts, scales, and trends. To this end, a novel merit function was proposed by ranking differences over min-max normalized values. First, patterns are computed per gene, where a pattern is given by a continuous time segment coherent for at least two samples. Sliding windows are used to accommodate time lags up to a prespecified length. Then, frequent pattern mining is applied to exhaustively find patterns supported by multiple genes. Finally, association rule mining is applied to discover dependencies between patterns (using contributions from Tatavarty et al. (2007)). Merging procedures are considered for largely overlapping TDARs.

TimesVector (Jung et al. 2017) has been recently proposed to find similarly and differentially expressed patterns in gene-sample-time data. To this end, the sample and time dimensions are first concatenated into a single dimension and spherical k-means applied to measure the similarity between observations under a silhouette score. Then, knowledge regarding the expression of genes on different samples is reintroduced to recognize delineated patterns of interest.

Hu and Bhatnagar (2011) proposed *Versatile Temporal Subspace* discovery for temporal 3D data analysis. A tricluster is described by a clear temporal pattern given by the longest possible smooth sections of nonoverlapping waveforms using the LOESS residue (computed from the lowest-order polynomial fitting the observed values). An exhaustive prefix-based search algorithm is then proposed with Pearson correlation (parameterized with the LOESS residue) to discover all maximal triclusters from closed temporal patterns. Due to the possible high number of triclusters, the top k-triclusters can be alternatively outputted.

In the context of coclustering tasks, Sim et al. (2013) proposed *CATSeeker*, combining background knowledge (preferred objects/centroids) with numerical optimization (score objects in subspaces with regard to each centroid) and 3D frequent itemset mining to find coclusters satisfying homogeneity, significance, and actionability criteria. The peer *MASC* (Mining Actionable Subspace Clusters) (Sim et al. 2010b) algorithm bypasses the need to prespecify the (generally unknown) preferred objects.

5.6 Multiobjective Optimization Approaches

Multiple objectives conflicting with each other, such as the volume and homogeneity of the triclusters, can be jointly optimized. In this context, multiobjective optimization is suitable to answer triclustering tasks.

Gutiérrez-Avilés et al. (2014) proposed *TriGen* (Triclustering Genetic-based algorithm), a multiobjective algorithm that maximizes volume and minimizes 3D mean square residue (fitness criteria according to Equation (3)). TriGen identifies a set of triclusters at a time, where each candidate tricluster corresponds to an individual from a population subjected to well-defined operators (exchanges, mutations, and selections of objects).

TriGen was extended in more recent works with two alternative merit functions: *MSL* (Gutiérrez-Avilés and Rubio-Escudero 2015) and LSL (Gutiérrez-Avilés and Rubio-Escudero 2014a) (discussed in *Section 2.1*). As these functions measure angle differences of the slopes between observation slices, they provide a sufficiently flexible form of coherence to accommodate time lags on observations.

Bhar et al. (2015) developed an evolutionary multiobjective optimization search for δ-TriMax, termed EMOA-δ-TriMax, sensitive to overlapping subspaces that prevents local optima by maximizing volume while minimizing 3D MSR using a non-dominated sorting genetic algorithm (NSGA-II (Deb et al. 2002)).

Despite these recent contributions, the initial multiobjective evolutionary algorithm, *MOGA3C* (Multi-Objective Genetic Algorithm for 3D Clustering) (Liu et al. 2008), is still considered state of the art. Three objectives—triclusters' volume, homogeneity, and gene-dimension variance—are simultaneously satisfied/optimized. In particular, MOGA3C focuses on finding triclusters with maximum volume, bounded 3D MSR, and high gene-dimension variance. These criteria guide selection, crossover, and mutation until stabilization or a prespecified number of generations is reached.

6 TRICLUSTERING EVALUATION

Section 2.1 introduced diverse merit functions to guide the discovery of triclusters. The assessment of the found triclusters should rely on independent metrics to produce unbiased performance views. However, evaluating triclustering algorithms is challenged by two issues. First, although synthetic data with planted triclusters can be generated, there are no consensual similarity metrics, and generation procedures are typically biased toward specific forms of homogeneity. Second, there is no ground truth to describe triclusters present in real 3D data. Accordingly, triclustering solutions can be assessed in the presence of hidden triclusters (accuracy-based views) and in their absence (homogeneity and significance views).

Accuracy-based views. In synthetic data, a set of triclusters $\mathcal{H} = \{\mathbf{H}_1, \ldots \mathbf{H}_r\}$ (referred to as true or hidden triclusters) can be planted. Objective metrics can be formulated since the true solution is known. These include similarity metrics to assess one dimension at a time, two dimensions at a time, or the overall dimensions. Clustering similarity metrics can then be used to evaluate one dimension at a time, including measures of intracluster similarity, entropy, or the F-measure (and its precision and recall components) (Assent et al. 2008). The principle is that triclusters should cover most objects of a specific hidden tricluster but few objects from remaining hidden triclusters.

Similarity metrics from biclustering can be applied to evaluate two dimensions at a time (Padilha and Campello 2017), including relative nonintersecting area (RNAI) (Bozdağ et al. 2010), subspace clustering error (Patrikainen and Meila 2006), and Jaccard-based scores (MS) (Prelić et al. 2006). Jaccard-based scores typically assess the similarity between the learned and planted triclusters ($\mathcal{B}$ and $\mathcal{H}$) on observation and attribute dimensions: $MS(\mathcal{B}, \mathcal{H})$ defines the extent to which the found triclusters cover the hidden triclusters (*completeness*), while $MS(\mathcal{H}, \mathcal{B})$ defines how well the hidden triclusters are recovered (*precision*).

Jaccard-based scores can also be extended to assess all dimensions at a time. Ignatov et al. (2015) define coverage as a fraction of the elements of the found tricluster included in at least one of the hidden triclusters. However, this measure cannot distinguish if a found tricluster is covered by one or multiple hidden triclusters. Alternative similarity metrics have been proposed (Gonçalves 2013), including an extension of the clustering F-measure (referred to as the E4SC metric) and their recall-precision components. Below, we propose a new score, the 3D revised match score (RMS3), to offer an adequate penalization of non-matched volume between triclusters:

$$\mathbf{RMS3}(\mathcal{B}, \mathcal{H}) = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{B}_1 \in \mathcal{B},\, maxarg\{Jac(\mathbf{B}_1, \mathbf{B}_2) \,|\, \mathbf{B}_2 \in \mathcal{H}\}} \sqrt[3]{\frac{|\mathbf{I}_1 \cap \mathbf{I}_2|}{|\mathbf{I}_1 \cup \mathbf{I}_2|} \frac{|\mathbf{J}_1 \cap \mathbf{J}_2|}{|\mathbf{J}_1 \cup \mathbf{J}_2|} \frac{|\mathbf{K}_1 \cap \mathbf{K}_2|}{|\mathbf{K}_1 \cup \mathbf{K}_2|}}. \tag{8}$$

Sim et al. (2013) proposed three measures: recoverability (a measure of completeness), spuriousness (a measure of precision), and a significance indicator that trades off recoverability and spuriousness. Bhar et al. (2012) estimated similarity using the product of a similarity score per dimension: $MS_{\mathbf{I}}(\mathbf{B}_1, \mathbf{B}_2) \times MS_{\mathbf{J}}(\mathbf{B}_1, \mathbf{B}_2) \times MS_{\mathbf{K}}(\mathbf{B}_1, \mathbf{B}_2)$.

Amar et al. (2015) proposed a weighted maximum of all pairwise Jaccard-based scores:

$$\frac{1}{|\mathcal{B}| + |\mathcal{H}|} \left(\sum_{\mathbf{B}_1 \in \mathcal{B}} max_{\mathbf{B}_2 \in \mathcal{H}} MS(\mathbf{B}_1, \mathbf{B}_2) + \sum_{\mathbf{B}_2 \in \mathcal{H}} max_{\mathbf{B}_1 \in \mathcal{B}} MS(\mathbf{B}_2, \mathbf{B}_1) \right). \tag{9}$$

Homogeneity views. Merit functions can be used not only to guide the triclustering task but also to evaluate the quality of triclustering solutions. Diverse merit functions were listed in *Section 2.1* to be used in the mining process. Despite their relevance, the application of these functions to evaluate triclustering solutions is biased toward specific homogeneity criteria. For this reason, and since different merit functions can provide radically distinct views, multiple functions can be combined for fair assessments. Ahmed et al. (2011) proposed the joint analysis of two merit functions, respectively prepared to assess intra- and interplane coherence. Merit functions may also benefit small triclusters due to their greater chance of having high homogeneity. Liu et al. (2008) and Gutiérrez-Avilés and Rubio-Escudero (2014b) proposed functions that minimize this effect by weighting merits according to the size of triclusters.

Statistical significance views. Statistical tests should alternatively be considered to assess the statistical significance of triclusters and bound the propensity of triclustering algorithms toward false-positive and false-negative risks. *Section 4.3* surveyed statistical views based on deviations between the observed and (either expected or unknown) underlying data distributions (Moise and Sander 2008a; Sim et al. 2010a), thresholding methods (Li and Tuck 2009), exact calculations (Tchagang et al. 2012), and size expectations collected from randomized data (Mankad and Michailidis 2014). However, the applicability of previous methods is limited by the placed assumptions either on the 3D data distributions or the homogeneity of triclusters, hence the open need for flexible and robust statistical tests, possibly extending the statistical views recently proposed in the context of biclustering tasks (Henriques and Madeira 2018).

Domain significance views. Despite the relevance of using real data to assess the performance of triclustering algorithms, there is no ground truth that describes the true triclusters in real datasets. In this context, it is essential to complement previous performance views with indicators gathered from assessing triclustering solutions against background knowledge. For this purpose, annotations L associated with groups of objects can be extracted from knowledge bases, semantic sources, or bibliographic databases. Illustrating, in biological domains, gene ontology terms, transcription factors, molecular associations, or metabolic pathways provide well-established annotations. Enrichment measures can then be computed by, for instance, testing hypergeometric hypotheses for the subsets of objects per tricluster against these annotations.

Complementary performance views. The aforementioned performance views primarily assess the effectiveness of triclustering algorithms. Given the diversity of available metrics, it is important to find consensual views. Recent work on consensus have been conducted by Gutiérrez-Avilés and Rubio-Escudero (2016). In their work, a new measure combining correlation scores, graphic validation, and enrichment tests was proposed.

In addition to effectiveness views, six alternative performance views are suggested for an in-depth assessment of triclustering algorithms:

- Scalability (Kakati et al. 2016) together with measures of optimality guarantees
- Dissimilarity between triclusters and their distribution in the 3D space (Ignatov et al. 2015)
- Robustness to different forms and degrees of noise and missings on generated data
- Flexibility: ability to customize coherence, structure, and quality of triclusters
- Actionability: ability to find triclusters suggesting concrete actions/profitable decisions
- Sensitivity to parameters and numerosity of parameters

7 TRICLUSTERING APPLICATIONS

This section lists prominent real-world problems across biomedical and social domains where triclustering has been shown to be an essential tool.

7.1 Biological Data Analysis

The following biological applications are found in the literature: analysis of 3D omic, augmented, network, multispecious, chemical, drug, and sample-by-sample data.

First, triclustering has been largely applied over omic data, including gene expression time series (*gene-sample-time* data) or the concentration of molecular compounds (such as proteins, protein complexes, and metabolites) along time (Gutiérrez-Avilés and Rubio-Escudero 2014b; Bhar et al. 2015). Samples are taken either from the same tissue under varying conditions (e.g. yeast responses to abiotic stress (Supper et al. 2007) or human responses to septic shock (Parnell et al. 2013)) or from multiple individuals (e.g., individuals after drug intake (Li et al. 2013) or extended wakefulness (Möller-Levet et al. 2013)). Triclustering offers the opportunity to enhance the understanding of functional genomics (coregulated entities on specific processes and pathways), improve diagnoses, and support the creation of new drug targets (Tchagang et al. 2012). To this end, triclustering has been applied with additive, order-preserving, and plaid assumptions (Hu and Bhatnagar 2011; Ahmed et al. 2011; Mankad and Michailidis 2014) to detect putative modules with flexible yet meaningful coherence.

Second, 2D biological data have been augmented into 3D data by distributing the observed values along a third dimension given by annotations from knowledge bases. Alqadah and Bhatnagar (2008) applied triclustering over *disease-gene-GO term* expression data to unravel associations between diseases and the regulated biological processes, cellular components, and molecular functions. Li and Tuck (2009) analyzed *gene-sample-regulator* expression data, where binding information was incorporated to create the third dimension. The understanding of which transcription factors coherently regulate groups of genes is unraveled in the found triclusters.

Third, triclustering has been applied over *dynamic biological networks* (*node-node-time* data) (Li et al. 2013) to discover coherent modules of biological entities along time. Triclusters can capture relevant cellular signaling events, thus holding the potential to unravel the organization and dynamics of cell functions. Triclustering has been alternatively considered for the integrative analysis of biological networks from heterogeneous data sources (*source-node-node* data) (Li et al. 2011).

Fourth, triclustering also has been applied over *heterogeneous 3D data* from different sources or distinct species. Hu and Bhatnagar (2010) placed the focus on coherent expression modules within each species followed by cross-species consensus, while Dede and Oğul (2014, 2013) analyzed all gene-sample-organism dimensions simultaneously. Waltman et al. (2010) analyzed expression data for all species closely related to Firmicutes, addressing the fact that orthology is not in many cases a one-to-one mapping, as well as the fact that modules within a tricluster may differ regarding their regulation or function and suffer from species-specific nuances that mask their discovery.

Fifth, triclustering also holds intrinsic opportunities for the analysis of *chemical 3D data*, such as protein structure in the form of *residues-position-time* (Sim et al. 2013).

Triclustering has been additionally suggested (yet not comprehensively applied) for the analysis of drug, genome-wide, and sample-by-sample 3D data. Triclustering drug 3D data, where the third dimension indexes dose levels, can be used to support pharmacological decisions (understand the impact of drug compounds and concentrations on biological systems) and personalized oncology (defined as "getting the right treatment to the right patient at the right dose and time" (Hayes 2013)). Triclusters from genome-wide and translational 3D data can disclose groups of mutations, conserved functional subsequences (alignments), or transcription factor binding sites correlated for a group of individuals on a subset of all chromosomes (Wang et al. 2007; Fang et al. 2012). Sample-by-sample 3D data records similarities between samples (first and second dimension) with a third dimension given by biological entities, similarity criteria, or time. Triclustering sample-by-sample 3D data was suggested to find coherent biological indicators from individuals with identical pathologies (Fan et al. 2010).

7.2 3D Medical Data Analysis

In medical domains, we highlight three major triclustering applications: (1) multivariate physiological signal (*individual-feature-time signal* data) analysis, where triclusters can capture coherent physiological responses for a group of individuals; (2) neuroimaging data analysis, where triclusters can capture hemodynamic response functions and connectivity between brain regions; and

(3) clinical records analysis, where triclusters correspond to groups of patients with correlated clinical features along time.

Balasubramanian et al. (2016) applied a variant of the triclustering task over e-mobile physiological data periodically monitored through multiple body sensors. Triclusters correspond to statistically significant deviations from healthy indicators for subsets of patients sharing a specific risk profile. Fan et al. (2010) analyzed *patient-electrode-time* EEG data to guide the design of electrical stimulation protocols (by parameterizing the location, intensity, duration, pulse width, and frequency of stimulation based on locally correlated brain activity). Amar et al. (2015) and Vaisvaser et al. (2013) triclustered three-dimensional fMRI data (*patient-brain parcel-time* blood flow intensity data) to localize regions with properties of interest and characterize functional connectivity. Triclustering also has been suggested for the analysis of local field potentials (LFPs) and spiking activity (Pourahmadi and Noorbaloochi 2016).

Contrasting to signal and spatial 3D data analysis, only few triclustering algorithms are prepared for the analysis of clinical record data since clinical attributes are typically nonidentically distributed. In Zhou et al. (2014), the goal was to infer phenotypic patterns (given by triclusters) from noisy and sparse 3D data (given by electronic medical records for a monitored population of individuals) to support the prediction of heart and renal failure. Henriques et al. (2015a) focused on subspaces from integrative healthcare databases (sparse patient-record-time data given by a sequence of events), where a subspace is mapped from arrangements of events learned using generative and deterministic views. In Heylen et al. (2016), 3D clinical data is given by periodically measuring the severity of a set of symptoms for a population of patients with the aim of understanding the effects of an intervention or prescription. In this work, triclusters are given by subsets of patients and symptoms with coherent time profiles that allow both amplitude and temporal misalignments.

7.3 3D Social Data Analysis

Triclustering has been applied over social networks (*individual-individual-time* data) (Mankad and Michailidis 2014; Gnatyshak et al. 2012) to either discover communities of individuals with correlated activity and interaction (often referred to as coherently evolving communities) or group contents according to the accessors' profile (Banerjee et al. 2007; Mankad and Michailidis 2014). Temporal network data analysis is challenging since changes along time can profoundly affect the network topology. Lin et al. (2009) tackled the problem of discovering community structure over rich media social networks (*individual-media object-time* data) through the analysis of time-varying user actions and their social context. The community structure in these networks captures the social scope of user actions, thus having relevant applications in multiple tasks such as description and recommendation. Triclustering is also relevant for the analysis of alternative types of networks, such as transportation or communication networks (Guigourès et al. 2015). For instance, in a subway network (where interactions correspond to journeys from an origin to a destination station for a specific time window), triclustering can reveal local patterns that explain the evolving distribution of journeys along time to support planning decisions (Guigourès et al. 2015).

In the context of web usage, triclustering has also been applied to group users with coherent patterns of navigation (Zhou et al. 2009). Zudin et al. (2015) triclustered *user-tag-bookmark* data to find groups of users coherently assigning similar tags on shared bookmarks. Alqadah and Bhatnagar (2008) applied triclustering over sparse *keyword-ad campaign-site* data. The same authors also introduced the problem of finding static communities from context-conditional social networks (*individual-individual-issue* data) to gain insight on consensual or disjoint views within a community on certain issues (such as Republican-Democrat views on political issues).

Social image tag refinement is another application (Tang et al. 2017) where triclustering patterns from *image-user attributes-tag* data are used to complete missing tags and rectify noise-corrupted ones, an essential component for image search.

Triclustering has also been applied for the analysis of (1) e-commerce data to find hidden browsing patterns from correlated sets of (web) users, visited (web) pages, and operations along time (Cerf et al. 2008); (2) marketing research data to study the perceived utility of diverse products for different ends as judged by different age groups (Schepers et al. 2006); and (3) collaborative filtering data (*user-item/action-time/region* data) to discover actionable correlations for recommendation systems (Bobadilla et al. 2013) or group users with similar preferences (or similar behavior) on a subset of items (or available actions) (Banerjee et al. 2007; Zudin et al. 2015).

7.4 Other Applications

Triclustering has also been applied over financial and trading data (Sim et al. 2010a) to identify indicators that produce consistent profitability profiles for specific trading decisions (buy, hold, and sell) in the stock market, as well as to understand deviant economic patterns. In this context, *stock-ratio-time* data has been considered to discover triclusters, groups of stocks with similar indicators across time, for investment decisions (Sim et al. 2010b). Financial ratios (and performance indicators) are seen as indicative of future price movements (and company behavior).

In econometrics, resource allocation or policy decisions can be supported by studying responses to the ebbs and flows of the economy. Mankad and Michailidis (2014) applied triclustering to *indicator-covariate-time* data. In addition, the same authors analyzed *country-country-time* world trade data given by the annual total bilateral trade flows between countries. Here, triclustering provided an alternative to traditional growth curves of global trade trends over time, promoting the understanding of economic strategies in the context of specific groups of countries.

Triclustering has been recently applied for the analysis of folksonomies, text, or unstructured (web) data (Gnatyshak 2015; Ignatov et al. 2015) to identify user-sensitive consensus in *source-user-tag* data and correlated content cubes measuring the frequency of categories/words across text segments.

Finally, triclustering has been additionally applied for geophysical data analysis, including (1) seismic data (such as *latitude-longitude-attribute* data) to capture geophysical patterns from cubes capturing the number of occurring earthquakes at different magnitudes (Martínez-Álvarez et al. 2015) and (2) atmospheric time-series data (*location-attribute-time* data) to identify spatiotemporal patterns, such as the annual regional variability of meteorological conditions (Wu et al. 2018, 2017).

8 SUMMARY

Tables 1 and 2 synthesize the properties of the surveyed triclustering algorithms in accordance with the proposed taxonomy (Figure 4). The triclustering algorithms are chronologically ordered and characterized according to their (1) locality (whether local or partial, and when local whether aimed at triclustering or coclustering), (2) homogeneity criteria, (3) algorithmic approach, (4) guarantees of statistical significance, (5) susceptibility to forms of dimension dominance, (6) ability to tackle time-related challenges, and (7) targeted type of 3D data.

As can be observed, state-of-the-art algorithms fail to satisfy relevant requirements: either homogeneity is restrictive, optimality guarantees are not pursued, statistical significance is disregarded, triclustering structures are constrained, efficiency rapidly degrades with data size, applicability is constrained to binary data, or unintended biases toward specific dimensions are observed.

In this context, numerous synergies remain to be explored. As such, and given the contradictory goals of triclustering tasks, the promises of upcoming contributions might instead reside in

Table 1. State-of-the-Art Triclustering Algorithms: Locality, Placed Homogeneity Criteria, and Algorithmic Approach

Name	Year	Locality	Homogeneity	Algorithmic Approach
GST-Miner	2004	partial	Pearson on **X**-**Y** plane with all time points	Biclustering based (quasi-exhaustive)
TriCluster	2005	tricluster	Pearson on **Z** slices (no interplane coherence)	Biclustering based (quasi-exhaustive) with graph consensus
gTriCluster	2006	tricluster	Spearman on **X** slices (no interplane coherence)	Biclustering based (quasi-exhaustive) with graph consensus
CubeMiner	2006	tricluster	Constant pattern	Pattern based (exhaustive)
RSM	2006	tricluster	Constant (multiwise intraplane coherence)	Pattern based (quasi-exhaustive) with closed-form pruning
TRIAS	2006	tricluster	Dense (binary and no noise)	Exhaustive (formal concept analysis)
Three-mode partitioning	2006	cocluster	Approximately constant	Stochastic (minimization of least squares)
EDISA	2007	partial	Pearson on **X**-**Y** plane with all time points	Greedy (seed growth and reduction)
Gene Program	2007	tricluster	Approximately constant patterns on **X**	Stochastic (hierarchical Bayesian model)
MOGA3C	2008	tricluster	Fully additive (3D MSR)	Multiobjective optimization (quasi-exhaustive)
TRI-Clustering	2009	tricluster	Differential values	Greedy (divide-and-conquer)
ECTDISA	2009	tricluster	Pearson on **Z** slices along time windows	Greedy (seed growth and removal within sliding windows)
LagMiner	2009	tricluster	Shifts/scales on **X** and **Z** slices; order-preserving on **Y**	Greedy (seed growth with multiwise intraplane coherence)
Box OAC	2009	tricluster	Dense (binary)	Exhaustive (formal concept analysis)
Data-Peeler	2009	tricluster	Dense (binary)	Exhaustive (formal concept analysis)
MSCM	2010	tricluster	Conservation and elaboration on **X** slices	Biclustering based (greedy)
Low-variance 3clustering	2010	tricluster	Approx. constant (low variance)	Biclustering based (quasi-exhaustive)
MASC	2010	cocluster	Constant patterns with shifts	Pattern based (quasi-exhaustive)
TD-Clustering	2010	tricluster	Shifts, scales, or symmetries on **Y** slices	Biclustering based (quasi-exhaustive) with sliding windows
MIC	2010	tricluster	Approximately constant (3D correlation information)	Greedy (seed growth)
3D-TDAR-Mine	2010	tricluster	Associative subspaces with shifts, scales, and trends on **X**	Pattern based (quasi-exhaustive)
SubCubeMiner	2011	tricluster	Order-preserving on **Z** slices and planar-based coherence	Biclustering based (greedy)
Versatile T. Subspace	2011	tricluster	LOESS-based Pearson on **X** slices	Pattern based (quasi-exhaustive)
Tribox	2011	tricluster	Dense (binary)	Exhaustive (formal concept analysis)
OPTricluster	2012	tricluster	Order-preserving on **X** slices and constant interplane	Biclustering-based (greedy)
δ-TRIMAX	2012	tricluster	Fully additive (3D MSR)	Greedy (divide-and-conquer)
TriWClustering	2013	tricluster	Fully additive (3D MSR)	Greedy (divide-and-conquer)
SpecTric	2013	tricluster	Dense (binary)	Exhaustive (formal concept analysis)
MultiFacTV	2013	tricluster	Plaid assumption	Stochastic (tensor factorization objective)
CATSeeker	2013	cocluster	Constant patterns with shifts	Pattern based (quasi-exhaustive)
TriGen	2014	tricluster	Fully additive (3D MSR)	Multiobjective optimization (quasi-exhaustive)
LSL-TriGen	2014	tricluster	Slope-based variation on **Z** slices	Multiobjective optimization (quasi-exhaustive)
3D-Plaid	2014	tricluster	Plaid assumption	Stochastic (kernel-based fitting)

(Continued)

Table 1. Continued

Name	Year	Locality	Homogeneity	Algorithmic Approach
MSL-TriGen	2015	tricluster	Slope-based variation on **Z** slices	Multiobjective optimization (quasi-exhaustive)
EMOA-δ-TRIMAX	2015	tricluster	Fully additive (3D MSR)	Multiobjective optimization (quasi-exhaustive)
TWIGS	2015	tricluster	**X** slices explained by 2 Gaussians	Stochastic (hierarchical Bayesian model)
MAP	2015	cocluster	Temporal patterns on a denormalized **X**×**Y** plane	Stochastic (graph-based maximum a posteriori estimation)
Prime OAC	2015	tricluster	Dense (binary)	Exhaustive (formal concept analysis)
Online OAC	2015	tricluster	Dense (binary)	Exhaustive (formal concept analysis)
2M-KSC	2016	cocluster	Additive on **Z** slices	Stochastic (minimization of least squares)
SSSimTri	2016	tricluster	Shifts and/or scales on **Z** slices	Parallelized biclustering based (greedy)
Krimp-triclustering	2017	tricluster	Dense (binary)	Exhaustive (frequent itemset mining)
TimesVector	2017	tricluster	Similar and differential patterns (cosine distance)	Pattern based (approximate searches on clustered data)
TTC	2017	tricluster	Low-rank/variance patterns	Stochastic (tensor factorization) and postcompletion
BCAT_I	2018	tricluster	I-divergence patterns	Stochastic (tensor factorization) followed by k-means

exploring the right balance between three major requirements: (1) *quality* (ability to offer guarantees of optimality and ensure noise robustness), (2) *scalability* (ability to handle the increasingly large, complete, and precise 3D datasets), and (3) *flexibility* (ability to find unconstrained structures and customize the desirable homogeneity and dissimilarity criteria in accordance with the targeted application).

9 CONCLUSIONS AND DIRECTIONS FOR FUTURE WORK

This work provides a structured view on the problem of triclustering three-dimensional data and surveys the existing contributions in the field. We first introduced a formal ground to understand triclustering and contrast it against alternative tasks. We further proposed a taxonomy to categorize the contributions in the field, assess the pros and cons of existing algorithms, and facilitate the design and adaptation of triclustering algorithms. The proposed taxonomy covered relevant requirements pertaining to the desirable forms of locality, homogeneity (including the need to place adequate coherence assumptions, guarantee robustness to noise, and allow for flexible structures of triclusters), statistical significance, and data-specific aspects. State-of-the-art triclustering algorithms were then comprehensively compared against this taxonomy. Finally, we listed relevant applications, together with insights to understand the intrinsic potential of the triclustering task to tackle real-world problems.

This survey stresses the need for upcoming contributions on different areas, including (1) the quantitative comparison of the state-of-the-art triclustering algorithms, (2) the development of scalable triclustering searches, (3) the incorporation of statistical tests to effectively guarantee the statistical significance of triclusters, (4) the development of integrative approaches able to combine the dispersed potentialities of the surveyed algorithms, (5) superior ways to handle temporal misalignments, (6) triclustering searches for sparse 3D data, (7) the discovery of triclustering solutions where triclusters may show varying homogeneity criteria, (8) the possibility to specify the degree of tolerance to different types of noise, (9) the incorporation of background knowledge into the triclustering task to guide the search according to domain knowledge and user expectations, and (10) new (associative) classifiers based on discriminative triclusters to label 3D data.

Table 2. Overall Comparison of Triclustering Algorithms According to the Placed Homogeneity Criteria, Guarantees of Statistical Significance, Optimality, Dimension-Specific Forms of Dominance, Time Aspects, and Targeted 3D Data

	homogeneity		guarantees of **statistical significance**	**optimality guarantees**	**dominance** (dimension-specific biases)	**time challenges**		**3D data**	
	shifts or scales	order preserving				time lags	absence of contiguity	symbolic	real valued
GST-Miner	✓			(2)	**Z** and **X**				✓
TriCluster	✓			(2)	**Z** and **X**		✓		✓
gTriCluster	✓			(2)	**X** and **Y**				✓
CubeMiner				✓				✓	
RSM				(2)				✓	
TRIAS				✓				(3)	
Three-mode partitioning									✓
EDISA	✓				**Z** and **X**				✓
GeneProgram					**X**				✓
MOGA3C	✓			(2)			✓		✓
TRI-Clustering			✓						✓
ECTDISA	✓				**Z**				✓
LagMiner	✓	(1)				✓			✓
Box OAC				✓				(3)	
Data-Peeler				✓				(3)	
MSCM			✓		**X** and **Y**				✓
Low-variance 3clustering				(2)	**X**				✓
TD-Clustering	✓			(2)	**Y** and **X**	✓			✓
MIC			✓		**Z**				✓
3D-TDAR-Mine	✓			(2)	**X** and **Z**	✓			✓
MASC	(1)		✓	(2)			✓		✓
SubCubeMiner		✓	✓		**Z** and **X**	✓			✓
Versatile T. Subspace	✓			(2)	**X** and **Z**				✓

(Continued)

Table 2. Continued

	homogeneity		guarantees of **statistical significance**	**optimality guarantees**	**dominance** (dimension-specific biases)	**time challenges**		**3D data**	
	shifts or scales	order preserving				time lags	absence of contiguity	symbolic	real valued
Tribox				✓				(3)	
OPTricluster		✓	✓		**Y** and **X**	✓			✓
δ-TRIMAX	✓						✓		✓
TriWClustering	✓								✓
SpecTric				✓				(3)	
MultiFacTV	✓						✓		✓
CATSeeker	✓		✓	(2)					✓
TriGen	✓			(2)	**Z**		✓		✓
LSL-TriGen		✓		(2)	**X** and **Z**	✓	✓		✓
3D-Plaid	✓		✓		**Z**				✓
MSL-TriGen		✓		(2)		✓	✓		✓
EMOA-δ-TRIMAX	✓			(2)	**X** and **Y**		✓		✓
TWIGS	(1)					✓			✓
MAP					**Z**				✓
Prime OAC				✓				(3)	
OnlineOAC				✓				(3)	
2M-KSC	(1)				**Z**				✓
SSSimTri	✓				**Y** and **Z**				✓
Krimp-triclustering				✓				(3)	
TimesVector	(1)	(1)			**Y** and **Z**				✓
TTC									✓
BCAT_I	✓				**Z**		✓		✓

(1) Sometimes observed on the found solutions, **(2)** quasi-exhaustive algorithms, **(3)** binary data only.

APPENDIX

A STRUCTURED VIEW ON N-WAY SUBSPACE CLUSTERING

An *N-dimensional dataset* is defined by N sets of objects or dimensions, $\{\mathbf{O}_i \mid i = 1..N\}$, with real-valued or categorical elements $a_{i_1,\dots,i_N}$ associated with one object per dimension: $(\mathbf{o}_{i_1},\dots,\mathbf{o}_{i_N})$, where $N > 1$. Paradigmatic examples include *individual-gene-sample-time* expression data (where the sample dimension is given by replicas or samples extracted from different tissues or under different conditions; genes can be further replaced by any other molecular compound), *user-user-attribute-time* interaction data, or *location-depth-attribute-time* geophysical data.

Given an N-dimensional dataset, the N-way subspace clustering task, or simply the *N-clustering* task, aims to find subspaces satisfying certain input criteria. In this context, an *N-way cluster* is given by a subset of objects on each dimension, $\{\mathrm{S}_i \subseteq \mathbf{O}_i \mid i = 1..N\}$.

Similarly to biclustering ($N = 2$) and triclustering ($N = 3$) tasks, the input criteria for N-clustering tasks with $N > 3$ can be a composition of homogeneity, statistical significance, and dissimilarity criteria.

Regarding *homogeneity* criteria, the cubic merit functions surveyed in *Section 2.1* can be generalized for an arbitrary number of dimensions. Illustrating, constant, fully additive, and plaid N-clusters can be respectively given by Equations (10), (11), and (12). In addition, 2D and 3D functions can be applied on permutations of the N-dimensional space to assess the correlation of a subspace with regard to a subset of overall dimensions at a given time. Inspired by the contributions by Ji et al. (2006), frequent closed hypercubes can be specified for the application of pattern-based merit functions.

$$\sigma^2_{\{\mathrm{S}_i\}_{i=1..N}} = \frac{1}{\prod_{i=1}^{N}|\mathrm{S}_i|} \times \sum_{\substack{i_1\,=\,1..|\mathrm{S}_1|\\ \dots\\ i_N\,=\,1..|\mathrm{S}_N|}} (a_{i_1..i_N} - \mu_\mathrm{S})^2,\ \ \mathit{with}\ \ \mu_\mathrm{S} = \frac{\sum_{i_1=1..|\mathrm{S}_j|,\dots,i_N=1..|\mathrm{S}_N|}(a_{i_1..i_N})}{\prod_{i=1}^{N}|\mathrm{S}_i|} \tag{10}$$

$$\mathit{MSR}_{\{\mathrm{S}_i\}_{i=1..N}} = \frac{1}{\prod_{i=1}^{N}|\mathrm{S}_i|} \sum_{\substack{i_1\,=\,1..|\mathrm{S}_1|\\ \dots\\ i_N\,=\,1..|\mathrm{S}_N|}} \eta_{i_1..i_N},\ \ \mathit{with}\ a_{i_1..i_N} = c + \alpha^{(1)}_{i_1} + \cdots + \alpha^{(N)}_{i_N} + \eta_{i_1..i_N} \tag{11}$$

$$\mathit{plaid}_{\{\mathrm{S}_i\}_{i=1..N}} = \sum_{i_1=1..|\mathrm{S}_1|,\dots,i_N=1..|\mathrm{S}_N|} \left(\hat{a}_{i_1..i_N} - \mu_0 - \sum_{t=1}^{T} \theta_{ti_1..i_N}\rho^{(1)}_{ti_1}..\rho^{(N)}_{ti_N}\right)^2,\ \ \mathit{with}$$

$$a_{i_1..i_N} = \mu_0 + \sum_{t=0}^{T} \theta_{ti_1..i_N}\rho^{(1)}_{ti_1}..\rho^{(N)}_{ti_N} + \eta_{ti_1..i_N}\ \ \mathit{and}\ \ \theta_{ti_1..i_N} = \mu_t + \alpha^{(1)}_{ti_1} + \cdots + \alpha^{(N)}_{ti_N} + \eta_{ti_1..i_N} \tag{12}$$

Similarly to $N = 3$, a statistically significant N-way cluster is a subspace with unexpectedly low probability to be observed. Here, the *statistical significance* criteria essentially depend on (1) the null data model (whether given by randomization procedures or by the observed multivariate regularities) and (2) the applied statistical tests (whether testing size against subspaces found in randomized data or occurrence probability against the approximated data distributions).

An N-way cluster $\{\mathrm{S}_i\}_{i=1..N}$ is maximal if and only if there is no other subspace $\{\mathrm{S}'_i\}_{i=1..N}$ satisfying the input criteria and $\{\mathrm{S}_i \subseteq \mathrm{S}'_i\}_{i=1..N}$. In this context, *dissimilarity* criteria can be placed with regard to maximality or be more properly defined in terms of subspace similarity (such as fraction of shared elements) and priority criteria (such as hypervolume, statistical significance, and/or homogeneity) to select removal candidates.

The proposed triclustering taxonomic view throughout *Sections 3 to 5* can also be generalized for $N \neq 3$. Regarding *locality*, a partial cluster is a subspace defined by subsets of objects from $N-1$ or fewer dimensions and all objects from the remaining dimensions. Contrasting with N-way clustering, the subspaces found by N-way full and partial clustering tasks contain all objects from at least one dimension.

Regarding *coherence*, different coherence assumptions (with parameterizable coherence strength) can be pursued by correlating all the values within an N-way cluster at a time or by guaranteeing the intra- and intercorrelation of the values from a subset of dimensions of the original N-dimensional subspace.

Regarding the *structure* of N-way clustering solutions, constraints can be placed to affect the number, size, shape, and position of N-way clusters. For instance, N-way clustering tasks might pursue N-way clusters with preferred shapes, such as flattened subspaces with intentional biases toward the most relevant dimensions. In the context of the allowed positioning, plaid, nonoverlapping, dimension-exclusive, dimension-exhaustive, and hierarchical constraints also might be incorporated.

Regarding *quality*, N-way clustering similarly needs to be robust to varying types and amounts of noise and missings, possibly bounding $\eta_{i_1,\ldots,i_N}$ deviations or the number of noisy and missing elements, $(\prod_{i=1}^{N} \frac{1}{|\mathrm{S}_i|} \sum_{i_j=1..|\mathrm{S}_j|, j=1..N} |\eta_{i_1..i_N}|) < \epsilon - \frac{\delta}{2}$, per subspace.

The inherent aspects of the input N-dimensional dataset, such as temporality, spatiality, or heterogeneity, can be further considered to place decisions on the pursued homogeneity, as well as on orthonormality and contiguity constraints per dimension.

The assessment of N-way clustering solutions can follow a similar evaluation methodology. In the presence of synthetic N-dimensional data, accuracy-based views given by similarity metrics between the found and true N-way clusters (targeting either all dimensions or a subset of dimensions) offer objective quality criteria. Illustrating, by parameterizing Equation (9) with hypervolume match scores, this accuracy view becomes able to assess the recall and precision of an N-way clustering solution over a prespecified set of dimensions. In the presence of real data, homogeneity-based views given by merit functions (such as Equations (10), (11), and (12)) and domain significance views given by statistical tests against domain knowledge offer subjective yet important quality criteria to measure the homogeneity and relevance of N-way clustering solutions. Statistical significance views can be further formulated to measure the false-positive and false-negative risks of triclustering algorithms. In addition, efficiency, robustness, dissimilarity, parameter sensitivity, and actionability views offer complementary performance criteria. Qualitative assessments can be further applied to measure the time and space complexity of the algorithms, the adequacy of a given homogeneity criterion in the context of a specific application domain, the biases observed toward specific dimensions, and the flexibility of the targeted N-way clustering structures.

Finally, following the diversity of algorithmic approaches observed for biclustering and triclustering tasks, similar approaches for N-way clustering tasks with $N > 3$ can be envisioned: (1) greedy algorithms based on insertion and removal of objects from different dimensions, (2) stochastic approaches from multiway distributions, (3) (quasi-)exhaustive algorithms for the discovery of dense subspaces, (4) algorithms based on the application of subspace clustering algorithms with lower dimensionality followed by consensus, (5) pattern-mining algorithms applied with denormalization procedures, and (6) evolutionary multiobjective optimization, among other possibilities.

Despite the numerous possibilities, the state-of-the-art research on the unsupervised analysis of N-dimensional data when $N > 3$ is scarce and mostly driven by N-way full-clustering algorithms based on tensor factorization (Shashua et al. 2006), algebraic theory (He et al. 2010), and top-down decomposition and bottom-up composition of variable relations (Bekkerman et al. 2005). Papalexakis et al. (2013) proposed a new class of N-way coclustering algorithms using multilinear decomposition with sparse latent factors to exhaustively partition the objects on each dimension. Contrasting with N-dimensional data analysis ($N > 3$), there is abundant research on the subspace clustering of alternative multidimensional data structures. In the context of star-structured relational data, a dataset is defined by multiple entries, where each entry associates a subset of objects from each one of the N dimensions. In this context, subspace clustering relational data is often mapped into the task of discovering hypercliques (subspaces) in an N-partite graph (where each entry defines multiple interactions between a set of objects) (Long et al. 2006; Banerjee et al. 2007). N-partite graphs also have been referred as N-dimensional networks due to the possible presence of multiway interactions (Lin et al. 2009). In a concluding note, contributions from the fields of triclustering and N-way full-clustering—as well as from relational learning (to study probabilistic forms of dependency among objects), heterogeneous graph mining, and tensor factorization (Long et al. 2006, 2007)—provide key principles to develop the long-awaited generation of N-way clustering algorithms with $N > 3$.

COMPETING INTERESTS

The authors declare that they have no competing interests.

REFERENCES

Elke Achtert, Christian Böhm, Hans-Peter Kriegel, Peer Kröger, Ina Müller-Gorman, and Arthur Zimek. 2006. Finding hierarchies of subspace clusters. In *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 446–453.

H. A. Ahmed, P. Mahanta, D. K. Bhattacharyya, J. K. Kalita, and A. Ghosh. 2011. Intersected coexpressed subcube miner: An effective triclustering algorithm. In *2011 World Congress on Information and Communication Technologies (WICT'11)*. IEEE, 846–851.

Faris Alqadah and Raj Bhatnagar. 2008. An effective algorithm for mining 3-clusters in vertically partitioned data. In *IC on Information and Knowledge Management*. ACM, 1103–1112.

David Amar, Daniel Yekutieli, Adi Maron-Katz, Talma Hendler, and Ron Shamir. 2015. A hierarchical Bayesian model for flexible module discovery in three-way time-series data. *Bioinformatics* 31, 12 (2015), i17–i26.

Ira Assent, Emmanuel Müller, Ralph Krieger, Timm Jansen, and Thomas Seidl. 2008. Pleiades: Subspace clustering and evaluation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 666–671.

Arvind Balasubramanian, Jun Wang, and Balakrishnan Prabhakaran. 2016. Discovering multidimensional motifs in physiological signals for personalized healthcare. *IEEE Journal of Selected Topics in Signal Processing* 10, 5 (2016), 832–841.

Arindam Banerjee, Sugato Basu, and Srujana Merugu. 2007. Multi-way clustering on relation graphs. In *SDM*, Vol. 7. SIAM, 225–334.

Ron Bekkerman, Ran El-Yaniv, and Andrew McCallum. 2005. Multi-way distributional clustering via pairwise interactions. In *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 41–48.

Amir Ben-Dor, Benny Chor, Richard Karp, and Zohar Yakhini. 2003. Discovering local structure in gene expression data: The order-preserving submatrix problem. *Journal of Computational Biology* 10, 3–4 (2003), 373–384.

Anirban Bhar, Martin Haubrock, Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay, and Edgar Wingender. 2012. δ-TRIMAX: Extracting triclusters and analysing coregulation in time series gene expression data. In *International Workshop on Algorithms in Bioinformatics*. Springer, 165–177.

Anirban Bhar, Martin Haubrock, Anirban Mukhopadhyay, and Edgar Wingender. 2015. Multiobjective triclustering of time-series transcriptome data reveals key genes of biological processes. *BMC Bioinformatics* 16, 1 (2015), 1.

Jesús Bobadilla, Fernando Ortega, Antonio Hernando, and Abraham Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based systems* 46 (2013), 109–132.

Doruk Bozdağ, Ashwin S. Kumar, and Umit V. Catalyurek. 2010. Comparative analysis of biclustering algorithms. In *BCB*. ACM, New York, 265–274. DOI: http://dx.doi.org/10.1145/1854776.1854814

Loïc Cerf, Jérémy Besson, Kim-Ngan T. Nguyen, and Jean-François Boulicaut. 2013. Closed and noise-tolerant patterns in n-ary relations. *Data Mining and Knowledge Discovery* 26, 3 (2013), 574–619.

Loïc Cerf, Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. 2008. Data peeler: Contraint-based closed pattern mining in n-ary relations.. In *SDM*, Vol. 8. SIAM, 37–48.

Loïc Cerf, Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. 2009. Closed patterns meet n-ary relations. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 3, 1 (2009), 3.

Yizong Cheng and George M. Church. 2000. Biclustering of expression data. In *Intelligent Systems for Molecular Biology*. AAAI Press, 93–103.

Antonio D. Chiaravalloti, Gianluigi Greco, Antonella Guzzo, and Luigi Pontieri. 2006. An information-theoretic framework for high-order co-clustering of heterogeneous objects. In *European Conference on Machine Learning*. Springer, 598–605.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

Duygu Dede and Hasan Oğul. 2013. A three-way clustering approach to cross-species gene regulation analysis. In *Symposium on Innovations in Intelligent Systems and Applications (INISTA'13)*. IEEE, 1–5.

Duygu Dede and Hasan Oğul. 2014. TriClust: A tool for cross-species analysis of gene regulation. *Molecular Informatics* 33, 5 (2014), 382–387.

Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn J. Keogh. 2008. Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment* 1, 2 (2008), 1542–1552.

A. Donders, G. van der Heijden, T. Stijnen, and K. Moons. 2006. Review: A gentle introduction to imputation of missing values.*Clinical Epidemiology* 59, 10 (2006), 1087–1091.

Neng Fan, Nikita Boyko, and Panos M. Pardalos. 2010. Recent advances of data biclustering with application in computational neuroscience. In *Computational Neuroscience*. Springer, 85–112.

Gang Fang, Majda Haznadar, Wen Wang, Haoyu Yu, Michael Steinbach, Timothy R. Church, William S. Oetting, Brian Van Ness, and Vipin Kumar. 2012. High-order SNP combinations associated with complex diseases: Efficient discovery, statistical power and functional interactions. *Plos One* 7, 4 (2012), 1–15. Retrieved from http://dx.doi.org/10.1371/journal.pone.0033531.

King-Sun Fu and J. K. Mui. 1981. A survey on image segmentation. *Pattern Recognition* 13, 1 (1981), 3–16.

Bin Gao, Tie-Yan Liu, and Wei-Ying Ma. 2006. Star-structured high-order heterogeneous data co-clustering based on consistent information theory. In *6th International Conference on Data Mining (ICDM'06)*. IEEE, 880–884.

Georg K. Gerber, Robin D. Dowell, Tommi S. Jaakkola, and David K. Gifford. 2007. Automated discovery of functional generality of human gene expression programs. *PLoS Computational Biology* 3, 8 (2007), e148.

D. V. Gnatyshak. 2015. A single-pass triclustering algorithm. *Automatic Documentation and Mathematical Linguistics* 49, 1 (2015), 27–41.

Dmitry Gnatyshak, Dmitry I. Ignatov, Alexander Semenov, and Jonas Poelmans. 2012. Gaining insight in social networks with biclustering and triclustering. In *International Conference on Business Informatics Research*. Springer, 162–171.

Joana Gonçalves. 2013. *Integrative Mining of Gene Regulation and Its Perturbations*. Ph.D. Dissertation. Instituto Superior Técnico, Universidade de Lisboa, Lisboa.

Joana P. Gonçalves and Sara C. Madeira. 2014. LateBiclustering: Efficient heuristic algorithm for time-lagged bicluster identification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 11, 5 (2014), 801–813. DOI: http://dx.doi.org/10.1109/TCBB.2014.2312007

Gianluigi Greco, Antonella Guzzo, and Luigi Pontieri. 2010. Coclustering multiple heterogeneous domains: Linear combinations and agreements. *IEEE Transactions on Knowledge and Data Engineering* 22, 12 (2010), 1649–1663.

Romain Guigourès, Marc Boullé, and Fabrice Rossi. 2015. Discovering patterns in time-varying graphs: A triclustering approach. *Advances in Data Analysis and Classification* 1 (2015), 1–28.

David Gutiérrez-Avilés and Cristina Rubio-Escudero. 2014a. LSL: A new measure to evaluate triclusters. In *2014 IEEE IC on Bioinformatics and Biomedicine (BIBM'14)*. IEEE, 30–37.

David Gutiérrez-Avilés and Cristina Rubio-Escudero. 2014b. Mining 3D patterns from gene expression temporal data: A new tricluster evaluation measure. *Scientific World Journal* 2014 (2014), 1–16.

David Gutiérrez-Avilés and Cristina Rubio-Escudero. 2015. MSL: A measure to evaluate three-dimensional patterns in gene expression data. *Evolutionary Bioinformation Online* 11 (2015), 121.

David Gutiérrez-Avilés and Cristina Rubio-Escudero. 2016. *TRIQ: A Comprehensive Evaluation Measure for Triclustering Algorithms*. Springer International Publishing, Cham, 673–684. DOI: http://dx.doi.org/10.1007/978-3-319-32034-2_56

David Gutiérrez-Avilés, Cristina Rubio-Escudero, Francisco Martínez-Álvarez, and José C. Riquelme. 2014. TriGen: A genetic algorithm to mine triclusters in temporal gene expression data. *Neurocomputing* 132 (2014), 42–53.

Daniel F. Hayes. 2013. OMICS-based personalized oncology: If it is worth doing, it is worth doing well! *BMC Medicine* 11, 1 (2013), 221.

Zhaoshui He, Andrzej Cichocki, Shengli Xie, and Kyuwan Choi. 2010. Detecting the number of clusters in n-way probabilistic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 11 (2010), 2006–2021.

Rui Henriques. 2016. *Learning from High-Dimensional Data Using Local Descriptive Models*. Ph.D. Dissertation. Instituto Superior Tecnico, Universidade de Lisboa, Lisboa.

Rui Henriques, Cláudia Antunes, and Sara C. Madeira. 2015a. Generative modeling of repositories of health records for predictive tasks. *Data Mining and Knowledge Discovery* 29, 4 (2015), 999–1032. DOI: http://dx.doi.org/10.1007/s10618-014-0385-7

Rui Henriques, Claudia Antunes, and Sara C. Madeira. 2015b. A structured view on pattern mining-based biclustering. *Pattern Recognition* 48, 12 (2015), 3941–3958. DOI: https://doi.org/10.1016/j.patcog.2015.06.018

Rui Henriques and Sara C. Madeira. 2014. BicPAM: Pattern-based biclustering for biomedical data analysis. *Algorithms for Molecular Biology* 9, 1 (2014), 27. DOI: http://dx.doi.org/10.1186/s13015-014-0027-z

Rui Henriques and Sara C. Madeira. 2015. Biclustering with flexible plaid models to unravel interactions between biological processes. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 12, 4 (2015), 738–752. DOI: http://dx.doi.org/10.1109/TCBB.2014.2388206

Rui Henriques and Sara C. Madeira. 2016. BicNET: Flexible module discovery in large-scale biological networks using biclustering. *Algorithms for Molecular Biology* 11, 1 (2016), 1–30. DOI: http://dx.doi.org/10.1186/s13015-016-0074-8

Rui Henriques and Sara C. Madeira. 2018. BSig: Evaluating the statistical significance of biclustering solutions. *Data Mining and Knowledge Discovery* 32, 1 (2018), 124–161. DOI: http://dx.doi.org/10.1007/s10618-017-0521-2

Joke Heylen, Iven Van Mechelen, Eiko I. Fried, and Eva Ceulemans. 2016. Two-mode K-spectral centroid analysis for studying multivariate longitudinal profiles. *Chemometrics and Intelligent Laboratory Systems* 154 (2016), 194–206.

John Hopcroft, Omar Khan, Brian Kulis, and Bart Selman. 2004. Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences* 101, Suppl 1 (2004), 5249–5253.

Zhen Hu and Raj Bhatnagar. 2010. Algorithm for discovering low-variance 3-clusters from real-valued datasets. In *2010 IEEE International Conference on Data Mining*. IEEE, 236–245.

Zhen Hu and Raj Bhatnagar. 2011. Discovery of versatile temporal subspace patterns in 3-D datasets. In *2011 IEEE 11th International Conference on Data Mining*. IEEE, 1092–1097.

Dmitry I. Ignatov, D. V. Gnatyshak, S. O. Kuznetsov, and Boris G. Mirkin. 2015. Triadic formal concept analysis and triclustering: Searching for optimal patterns. *Machine Learning* 101, 1–3 (2015), 271–302.

Dmitry I. Ignatov and Sergei O. Kuznetsov. 2009. Frequent itemset mining for clustering near duplicate web documents. In *IC on Conceptual Structures*. Springer, 185–200.

J. Ihmels, S. Bergmann, and N. Barkai. 2004. Defining transcription modules using large-scale gene expression data. *Bioinformatics* 20, 13 (2004), 1993–2003.

Robert Jäschke, Andreas Hotho, Christoph Schmitz, Bernhard Ganter, and Gerd Stumme. 2006. TRIAS-An algorithm for mining iceberg tri-lattices. In *ICDM*, Vol. 6. 907–911.

Liping Ji, Kian-Lee Tan, and Anthony K. H. Tung. 2006. Mining frequent closed cubes in 3D datasets. In *Proceedings of the 32nd International Conference on Very Large Data Bases*. VLDB Endowment, 811–822.

Xiaonan Ji, James Bailey, and Guozhu Dong. 2007. Mining minimal distinguishing subsequence patterns with gap constraints. *Knowledge and Information Systems* 11, 3 (2007), 259–286.

Daxin Jiang, Jian Pei, Murali Ramanathan, Chun Tang, and Aidong Zhang. 2004. Mining coherent gene clusters from gene-sample-time microarray data. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*. ACM, Seattle, WA, USA, 430–439.

Haoliang Jiang, Shuigeng Zhou, Jihong Guan, and Ying Zheng. 2006. gTRICLUSTER: A more general and effective 3d clustering algorithm for gene-sample-time microarray data. In *International Workshop on Data Mining for Biomedical Applications*. Springer, 48–59.

Inuk Jung, Kyuri Jo, Hyejin Kang, Hongryul Ahn, Youngjae Yu, and Sun Kim. 2017. TimesVector: A vectorized clustering approach to the analysis of time series transcriptome data from multiple phenotypes. *Bioinformatics* 33, 23 (2017), 3827–3835.

Tulika Kakati, Hasin A. Ahmed, Dhruba K. Bhattacharyya, and Jugal K. Kalita. 2016. A fast gene expression analysis using parallel biclustering and distributed triclustering approach. In *ICTCS*. ACM, New York, Article 122, 6 pages. Retrieved from http://doi.acm.org/10.1145/2905055.2905182.

Sabine Krolak-Schwerdt, Peter Orlik, and Bernhard Ganter. 1994. TRIPAT: A model for analyzing three-mode binary data. In *Information Systems and Data Analysis*. Springer, 298–307.

L. Lazzeroni and A. Owen. 2002. Plaid models for gene expression data. *Statistica Sinica* 12 (2002), 61–86.

Fritz Lehmann and Rudolf Wille. 1995. A triadic approach to formal concept analysis. In *International Conference on Conceptual Structures*. Springer, 32–43.

Ao Li and David Tuck. 2009. An effective tri-clustering algorithm combining expression data with gene regulation information. *Gene Regulation and Systems Biology* 3 (2009), 49.

Wenyuan Li, Chun-Chi Liu, Tong Zhang, Haifeng Li, Michael S. Waterman, and Xianghong Jasmine Zhou. 2011. Integrative analysis of many weighted co-expression networks using tensor computation. *PLoS Computational Biology* 7, 6 (2011), e1001106.

Xutao Li, Yunming Ye, Michael Ng, and Qingyao Wu. 2013. MultiFacTV: Module detection from higher-order time series biological data. *BMC Genomics* 14, Suppl 4 (2013), S2.

Yifeng Li and Alioune Ngom. 2010. Classification of clinical gene-sample-time microarray expression data via tensor decomposition methods. In *CIBB*. Springer, 275–286.

Yu-Ru Lin, Jimeng Sun, Paul Castro, Ravi Konuru, Hari Sundaram, and Aisling Kelliher. 2009. Metafac: Community discovery via relational hypergraph factorization. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 527–536.

Junwan Liu, Zhoujun Li, Xiaohua Hu, and Yiming Chen. 2008. Multi-objective evolutionary algorithm for mining 3D clusters in gene-sample-time microarray data. In *IEEE International Conference on Granular Computing (GrC'08)*. IEEE, 442–447.

Yu-Cheng Liu, Chao-Hui Lee, Wei-Chung Chen, J. W. Shin, Hui-Huang Hsu, and Vincent S. Tseng. 2010. A novel method for mining temporally dependent association rules in three-dimensional microarray datasets. In *2010 International Computer Symposium (ICS'10)*. IEEE, 759–764.

Bo Long, Xiaoyun Wu, Zhongfei Mark Zhang, and Philip S. Yu. 2006. Unsupervised learning on k-partite graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 317–326.

Bo Long, Zhongfei Mark Zhang, Xiaoyun Wu, and Philip S. Yu. 2006. Spectral clustering for multi-type relational data. In *IC Conference on Machine Learning*. ACM, 585–592.

Bo Long, Zhongfei Mark Zhang, and Philip S. Yu. 2007. A probabilistic framework for relational clustering. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 470–479.

Sara C. Madeira and Arlindo L. Oliveira. 2004. Biclustering algorithms for biological data analysis: A survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 1, 1 (2004), 24–45.

Sara C. Madeira and Arlindo L. Oliveira. 2009. A polynomial time biclustering algorithm for finding approximate expression patterns in gene expression time series. *Algorithms for Molecular Biology* 4, 1 (2009), 8. DOI: http://dx.doi.org/10.1186/1748-7188-4-8

Sara C. Madeira, Miguel Nobre Parreira Cacho Teixeira, Isabel Sá-Correia, and Arlindo Oliveira. 2010. Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 1 (2010), 153–165.

P. Mahanta, H. A. Ahmed, D. K. Bhattacharyya, and Jugal K. Kalita. 2011. Triclustering in gene expression data analysis: A selected survey. In *2011 2nd National Conference on Emerging Trends and Applications in Computer Science (NCETACS'11)*. IEEE, 1–6.

Shawn Mankad and George Michailidis. 2014. Biclustering three-dimensional data arrays with plaid models. *Journal of Computational and Graphical Statistics* 23, 4 (2014), 943–965.

Francisco Martínez-Álvarez, David Gutiérrez-Avilés, Antonio Morales-Esteban, Jorge Reyes, José L. Amaro-Mellado, and Cristina Rubio-Escudero. 2015. A novel method for seismogenic zoning based on triclustering: Application to the iberian peninsula. *Entropy* 17, 7 (2015), 5000–5021.

Tim McInerney and Demetri Terzopoulos. 1996. Deformable models in medical image analysis: A survey. *Medical Image Analysis* 1, 2 (1996), 91–108. DOI: http://dx.doi.org/https://doi.org/10.1016/S1361-8415(96)80007-7

Jia Meng, Shou-Jiang Gao, and Yufei Huang. 2009. Enrichment constrained time-dependent clustering analysis for finding meaningful temporal transcription modules. *Bioinformatics* 25, 12 (2009), 1521–1527.

David Minnen, Charles Isbell, Irfan Essa, and Thad Starner. 2007. Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery. In *7th IEEE International Conference on Data Mining (ICDM'07)*. IEEE, 601–606.

Boris G. Mirkin and Andrey V. Kramarenko. 2011. Approximate bicluster and tricluster boxes in the analysis of binary data. In *International Workshop on Rough Sets, Fuzzy Sets, Data Mining, and Granular-Soft Computing*. Springer, 248–256.

Gabriela Moise and Jörg Sander. 2008a. Finding non-redundant, statistically significant regions in high dimensional data: A novel approach to projected and subspace clustering. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 533–541.

Gabriela Moise and Jörg Sander. 2008b. Finding non-redundant, statistically significant regions in high dimensional data: A novel approach to projected and subspace clustering. In *Proceedings of the 14th ACM SIGKDD IC on Knowledge Discovery and Data Mining*. ACM, New York, 533–541. Retrieved from http://doi.acm.org/10.1145/1401890.1401956.

Carla S. Möller-Levet, Simon N. Archer, Giselda Bucca, Emma E. Laing, Ana Slak, Renata Kabiljo, June C. Y. Lo, Nayantara Santhi, Malcolm von Schantz, Colin P. Smith, and Derk-Jan Dijk. 2013. Effects of insufficient sleep on circadian rhythmicity and expression amplitude of the human blood transcriptome. *National Academy of Sciences* 110, 12 (2013), E1132–E1141.

N. Narmadha and R. Rathipriya. 2016. Triclustering: An evolution of clustering. In *2016 Online IC on Green Engineering and Technologies (IC-GET'16)*. 1–4.

Victor A. Padilha and Ricardo J. G. B. Campello. 2017. A systematic comparative evaluation of biclustering techniques. *BMC Bioinformatics* 18, 1 (2017), 55.

Gergely Palla, Albert-László Barabási, and Tamás Vicsek. 2007. Quantifying social group evolution. *Nature* 446, 7136 (2007), 664–667.

Evangelos E. Papalexakis, Nicholas D. Sidiropoulos, and Rasmus Bro. 2013. From k-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors. *IEEE Transactions on Signal Processing* 61, 2 (2013), 493–506.

Grant P. Parnell, Benjamin M. Tang, Marek Nalos, Nicola J. Armstrong, Stephen J. Huang, David R. Booth, and Anthony S. McLean. 2013. Identifying key regulatory genes in the whole blood of septic patients to monitor underlying immune dysfunctions. *Shock* 40, 3 (2013), 166–174.

Anne Patrikainen and Marina Meila. 2006. Comparing subspace clusterings. *IEEE TKDE* 18, 7 (2006), 902–916. DOI: http://dx.doi.org/10.1109/TKDE.2006.106

Mohsen Pourahmadi and Siamak Noorbaloochi. 2016. Multivariate time series analysis of neuroscience data: Some challenges and opportunities. *Current Opinion in Neurobiology* 37 (2016), 12–15.

Amela Prelić, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Bühlmann, Wilhelm Gruissem, Lars Hennig, Lothar Thiele, and Eckart Zitzler. 2006. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* 22, 9 (2006), 1122–1129.

Jan Schepers, Iven Van Mechelen, and Eva Ceulemans. 2006. Three-mode partitioning. *Computational Statistics & Data Analysis* 51, 3 (2006), 1623–1642.

Amnon Shashua, Ron Zass, and Tamir Hazan. 2006. *Multi-way Clustering Using Super-Symmetric Non-negative Tensor Factorization.* Springer, Berlin, 595–608. DOI: http://dx.doi.org/10.1007/11744085_46

Kelvin Sim, Zeyar Aung, and Vivekanand Gopalkrishnan. 2010a. Discovering correlated subspace clusters in 3D continuous-valued data. In *2010 IEEE International Conference on Data Mining*. IEEE, 471–480.

Kelvin Sim, Ardian Kristanto Poernomo, and Vivekanand Gopalkrishnan. 2010b. MininG actionable subspace clusters in sequential data. In *SDM*. SIAM, 442–453.

Kelvin Sim, Ghim-Eng Yap, David R. Hardoon, Vivekanand Gopalkrishnan, Gao Cong, and Suryani Lukman. 2013. Centroid-based actionable 3D subspace clustering. *IEEE Transactions on Knowledge and Data Engineering* 25, 6 (2013), 1213–1226.

Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S. Yu. 2007. Graphscope: Parameter-free mining of large time-evolving graphs. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 687–696.

Jochen Supper, Martin Strauch, Dierk Wanke, Klaus Harter, and Andreas Zell. 2007. EDISA: Extracting biclusters from multiple time-series of gene expression profiles. *BMC Bioinformatics* 8, 1 (2007), 334.

Jinhui Tang, Xiangbo Shu, Guo-Jun Qi, Zechao Li, Meng Wang, Shuicheng Yan, and Ramesh Jain. 2017. Tri-clustered tensor completion for social-aware image tag refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 8 (2017), 1662–1674.

Giridhar Tatavarty, Raj Bhatnagar, and Barrington Young. 2007. Discovery of temporal dependencies between frequent patterns in multivariate time series. In *IEEE Symposium on Computational Intelligence and Data Mining, 2007 (CIDM'07).* IEEE, 688–696.

Alain B. Tchagang, Sieu Phan, Fazel Famili, Heather Shearer, Pierre Fobert, Yi Huang, Jitao Zou, Daiqing Huang, Adrian Cutler, Ziying Liu, and Youlian Pan. 2012. Mining biological information from 3D short time-series gene expression data: The OPTricluster algorithm. *BMC Bioinformatics* 13, 1 (2012), 1.

Alain Tremeau and Nathalie Borel. 1997. A region growing and merging algorithm to color segmentation. *Pattern Recognition* 30, 7 (1997), 1191–1203.

Alireza Vahdatpour, Navid Amini, and Majid Sarrafzadeh. 2009. Toward unsupervised activity discovery using multi-dimensional motif detection in time series. In *IJCAI*, Vol. 9. 1261–1266.

Sharon Vaisvaser, Tamar Lin, Roee Admon, Ilana Podlipsky, Yona Greenman, Naftali Stern, Eyal Fruchter, Ilan Wald, Daniel Pine, Ricardo Tarrasch, Yair Bar-Haim, and Talma Hendler. 2013. Neural traces of stress: Cortisol related sustained enhancement of amygdala-hippocampal functional connectivity.

Peter Waltman, Thadeous Kacmarczyk, A. Bate, D. Kearns, D. Reiss, P. Eichenberger, and Richard Bonneau. 2010. Multi-species integrative biclustering. *Genome Biology* 11, 9 (2010), 1.

Guoren Wang, Linjun Yin, Yuhai Zhao, and Keming Mao. 2010. Efficiently mining time-delayed gene expression patterns. *IEEE Transactions on Systems, Man, and Cybernetics, Part B* 40, 2 (2010), 400–411.

Shu Wang, Robin R. Gutell, and Daniel P. Miranker. 2007. Biclustering as a method for RNA local multiple sequence alignment. *Bioinformatics* 23, 24 (2007), 3289–3296.

Xiaojing Wu, Raul Zurita-Milla, Menno-Jan Kraak, and Emma Izquierdo-Verdiguier. 2017. Clustering-based approaches to the exploration of spatio-temporal data. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences (ISPRS'17)*, 1387–1391.

Xiaojing Wu, Raul Zurita-Milla, Emma Izquierdo Verdiguier, and Menno-Jan Kraak. 2018. Triclustering georeferenced time series for analyzing patterns of intra-annual variability in temperature. *Annals of the American Association of Geographers* 108, 1 (2018), 71–87.

Xin Xu, Ying Lu, Kian-Lee Tan, and Anthony K. H. Tung. 2009. Finding time-lagged 3D clusters. In *2009 IEEE 25th International Conference on Data Engineering*. IEEE, 445–456.

Maxim Yurov and Dmitry I. Ignatov. 2017. Turning krimp into a triclustering technique on sets of attribute-condition pairs that compress. In *International Joint Conference on Rough Sets*. Springer, 558–569.

Lizhuang Zhao and Mohammed J. Zaki. 2005. Tricluster: An effective algorithm for mining coherent clusters in 3d microarray data. In *ACM SIGMOD International Conf. on Management of Data*. ACM, 694–705.

Jiayu Zhou, Fei Wang, Jianying Hu, and Jieping Ye. 2014. From micro to macro: Data driven phenotyping by densification of longitudinal electronic medical records. In *Proceedings of the 20th ACM SIGKDD IC on Knowledge Discovery and Data Mining*. ACM, 135–144.

Qingbiao Zhou, Guangdong Xu, and Yu Zong. 2009. Web co-clustering of usage network using tensor decomposition. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies (WI-IAT'09)*, Vol. 3. IEEE, 311–314.

Hongyuan Zhu, Fanman Meng, Jianfei Cai, and Shijian Lu. 2016. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation* 34, Suppl. C (2016), 12–27.

Sergey Zudin, Dmitry V. Gnatyshak, and Dmitry I. Ignatov. 2015. Putting OAC-triclustering on MapReduce. In *Proceedings of the 12th IC on Concept Lattices and Their Applications*. CEUR Workshop Proceedings, 47–58.

Received September 2017; revised January 2018; accepted March 2018