Pattern-Based Biclustering with Constraints for Gene Expression Data Analysis

Rui Henriques(✉) and Sara C. Madeira

Inesc-ID, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal
{rmch,sara.madeira}@tecnico.ulisboa.pt

Abstract. Biclustering has been largely applied for gene expression data analysis. In recent years, a clearer understanding of the synergies between pattern mining and biclustering gave rise to a new class of biclustering algorithms, referred as pattern-based biclustering. These algorithms are able to discover exhaustive structures of biclusters with flexible coherency and quality. Background knowledge has also been increasingly applied for biological data analysis to guarantee relevant results. In this context, despite numerous contributions from domain-driven pattern mining, there is not yet a solid view on whether and how background knowledge can be applied to guide pattern-based biclustering tasks.

In this work, we extend pattern-based biclustering algorithms to effectively seize efficiency gains in the presence of constraints. Furthermore, we illustrate how constraints with succinct, (anti-)monotone and convertible properties can be derived from knowledge repositories and user expectations. Experimental results show the importance of incorporating background knowledge within pattern-based biclustering to foster efficiency and guarantee non-trivial yet biologically relevant solutions.

1 Introduction

Biclustering, the task of finding subsets of rows with a coherent pattern across subsets of columns in real-valued matrices, has been largely used for expression data analysis [9,11]. Biclustering algorithms based on pattern mining methods [9,11,12,18,22,25], referred in this work as pattern-based biclustering, are able to perform flexible and exhaustive searches. Initial attempts to use background knowledge for biclustering based on user expectations [5,7,15] and knowledge-based repositories [18,20,26] show its key role to guide the task and guarantee relevant solutions. In this context, two valuable synergies can be identified based on these observations. First, the optimality and flexibility of pattern-based biclustering provide an adequate basis upon which knowledge-driven constraints can be incorporated. Contrasting with pattern-based biclustering, alternative biclustering algorithms place restrictions on the structure (number, size and positioning), coherency and quality of biclusters, which may prevent the incorporation of certain constraints [11,16]. Second, the effective use of background knowledge to guide pattern mining searches has been largely researched in the context of domain-driven pattern mining [4,23].

F. Pereira et al. (Eds.) EPIA 2015, LNAI 9273, pp. 1–14, 2015.
DOI: 10.1007/978-3-319-23485-4_34

Despite these synergies, there is a lack of literature on the feasibility and impact of integrating domain-driven pattern mining and biclustering. In particular, there is a lack of research on how to map the commonly available background knowledge in the form of parameters or constraints to guide the biclustering task. Additionally, the majority of existing pattern-based biclustering algorithms rely on searches dependent on bitset vectors [18,22,25], which may turn their performance impracticable for large and dense biological datasets. Although new searches became recently available for biclustering large and dense data [13], there are not yet contributions on how these searches can be adapted to seize the benefits from the available background knowledge.

In this work, we address these problems. First, we list an extensive set of key constraints with biological relevance and show how they can be specified for pattern-based biclustering. Second, we extend F2G [13], a recent pattern-growth search that tackles the efficiency bottlenecks of peer searches, to bed able to effectively use constraints with succinct, (anti-)monotone and convertible properties.

To achieve these goals, we propose BiC2PAM (**BiC**lustering with **C**onstraints using **PA**ttern **M**ining), an algorithm that integrates recent breakthroughs on pattern-based biclustering [9,11,12] and extends them to effectively incorporate constraints. Experimental results confirm the role of BiC2PAM to foster the biological relevance of pattern-based biclustering solutions and to seize large efficiency gains by adequately pruning the search space.

The paper is structured as follows. *Section 2* provides background on pattern-based biclustering and domain-driven pattern mining. *Section 3* surveys key contributions and limitations from related work. *Section 4* lists biologically meaningful constraints and proposes BiC2PAM for their effective incorporation. *Section 5* provides initial empirical evidence of BiC2PAM's efficiency and ability to unravel non-trivial yet biologically significant biclusters from gene expression data. Finally, concluding remarks are synthesized.

2 Background

Definition 1. *Given a matrix, $A{=}(X,Y)$, with a set of rows $X{=}\{x_1,..,x_n\}$, a set of columns $Y{=}\{y_1,..,y_m\}$, and elements $a_{ij}{\in}\mathbb{R}$ relating row i and column j: the* ***biclustering task*** *aims to identify a set of biclusters $\mathcal{B}{=}\{B_1,..,B_m\}$, where each* bicluster *$B_k=(I_k,J_k)$ is a submatrix of A ($I_k \subseteq X$ and $J_k \subseteq Y$) satisfying specific criteria of* homogeneity *and* significance *[11].*

A real-valued matrix can thus be described by a (multivariate) distribution of background values and a *structure* of biclusters, where each bicluster satisfies specific criteria of *homogeneity* and *significance*. The *structure* is defined by the number, size and positioning of biclusters. Flexible structures are characterized by an arbitrary-high set of (possibly overlapping) biclusters. The *coherency* (homogeneity) of a bicluster is defined by the observed correlation of values (see Definition 2). The *quality* of a bicluster is defined by the type and amount of accommodated noise. The statistical *significance* of a bicluster determines the deviation of its probability of occurrence from expectations.

Definition 2. *Let the elements in a bicluster $a_{ij} \in (I, J)$ have* coherency across rows *given by $a_{ij}{=}k_j{+}\gamma_i{+}\eta_{ij}$, where k_j is the expected value for column j, γ_i is the adjustment for row i, and η_{ij} is the noise factor* [16]*. For a given real-valued matrix A and coherency strength δ: $a_{ij}{=}k_j{+}\gamma_i{+}\eta_{ij}$ where $\eta_{ij}{\in}[\quad -\delta/2, \quad +\delta/2]$.*

As motivated, the discovery of exhaustive and flexible structures of biclusters satisfying certain homogeneity criteria (Definition 2) is a desirable condition to effectively incorporate knowledge-driven constraints. However, due to the complexity of such biclustering task , most of the existing algorithms are either based on greedy or stochastic approaches, producing sub-optimal solutions and placing restrictions (e.g. fixed number of biclusters, non-overlapping structures, and simplistic coherencies) that prevent the flexibility of the biclustering task [16]. Pattern-based biclustering appeared in recent years as one of various attempts to address these limitations. As follows, we provide background on this class of biclustering algorithms, as well as on constraint-based searches.

Pattern-Based Biclustering. Patterns are itemsets, rules, sequences or other structures that appear in symbolic datasets with frequency above a specified threshold. Patterns can be mapped as a bicluster with constant values across rows ($a_{ij}{=}c_j$), and specific coherency strength determined by the number of symbols in the dataset, $\delta{=}1/|\mathcal{L}|$ where $\mathcal{L}$ is the alphabet of symbols. The relevance of a pattern is primarily defined by its support (number of rows) and length (number of columns). To allow this mapping, the pattern mining task needs to output not only the patterns but also their supporting transactions (*full-patterns*). Definitions 3 and 4 illustrate the paradigmatic mapping between full-pattern mining and biclustering.

Definition 3. *Let $\mathcal{L}$ be a finite set of items, and P an itemset $P \subseteq \mathcal{L}$. A symbolic matrix D is a finite set of transactions in $\mathcal{L}$, $\{P_1, .., P_n\}$. Let the* coverage *Φ_P of an itemset P be the set of transactions in D in which P occurs, $\{P_i \in D \mid P \subseteq P_i\}$, and its* support *$sup_P$ be the coverage size, $|\Phi_P|$.*

A full-pattern *is a pair (P, Φ_P), where P is an itemset and Φ_P the set of all transactions that contain P. A* closed full-pattern *(P, Φ_P) is a full-pattern where P is not subset of another itemset with the same support, $\forall_{P' \supset P} |P'| < |P|$.*

*Given D and a minimum support threshold θ, the **full-pattern mining** task* [13] *consists of computing: $\{(P, \Phi_P) \mid P \subseteq \mathcal{L}, sup_P \geq \theta, \forall_{P' \supset P} |P'| < |P|\}$.*

Given an illustrative symbolic matrix $D{=}\{(t_1, \{a, c, e\}), (t_2, \{a, b, d\}), (t_3, \{a, c, e\})\}$, we have $\Phi_{\{a,c\}}{=}\{t_1, t_3\}$, $sup_{\{a,c\}}{=}2$. For a minimum support $\theta{=}2$, the full-pattern mining task over D returns the set of closed full-patterns, $\{(\{a\}, \{t_1, t_2, t_3\}),\ (\{a, c, e\}, \{t_1, t_3\})\}$ (note that $|\Phi_{\{a,c\}}|{\leq}|\Phi_{\{a,c,e\}}|$). Fig.1 illustrates how full-pattern mining can be used to derive constant biclusters[1].

[1] Association rule mining, sequential pattern mining and graph mining can be also used to respectively mine biclusters with noisy, order-preserving and differential coherencies [9,12].

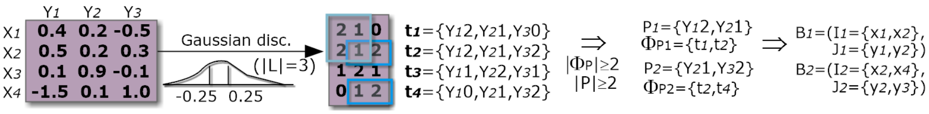

Fig. 1. Discovery of biclusters with constant coherency on rows from full-patterns.

Definition 4. *Given a symbolic matrix D in $\mathcal{L}$, let a matrix A be the concatenation of D elements with their column indexes. Let Ψ_P be the column indexes of an itemset P, and Υ_P be the original items of P in $\mathcal{L}$. The set of* ***maximal biclusters*** *$\cup_k B_k = (I_k, J_k)$ can be derived from the set of closed full-patterns $\cup_k P_k$ from A, by mapping I_k=Φ_{P_k} and J_k=Ψ_{P_k}, to compose constant biclusters with coherency across rows with pattern Υ_P* [11].

The inherent simplicity, efficiency and flexibility of pattern-based biclustering explains the increasing attention [11,12,18,22,25]. The major contributions of pattern-based approaches for biclustering include: *1)* efficient analysis of large matrices due to the monotone search principles (and the support for distributed/partitioned data settings and approximate patterns [8]). *2)* biclusters with parameterizable coherency strength (beyond differential assumption) and type (possibility to accommodate additive, multiplicative, order-preserving and plaid models) [9,11,12]; *3)* flexible structures of biclusters (arbitrary positioning of biclusters) and searches (no need to fix the number of biclusters apriori) [22,25]; and *4)* robustness to noise, missings and discretization problems [11].

Constraint-Based Pattern Mining. A *constraint* is a predicate on the powerset of items $C : 2^{\mathcal{L}} \rightarrow \{\mathit{true,false}\}$. A full-pattern (P, Φ_P) satisfies C if $C(P)$ is true. Minimum support is the default constraint in full-pattern mining, $C_{freq}(P)$ $=|\Phi_P| \geq \theta$. Typical constraints with interesting properties include: regular expressions on the items in the pattern, and inequalities based on aggregate functions, such as length, maximum, minimum, range, sum, average and variance [24].

Definition 5. *Let each item have a correspondence with a real value, $\mathcal{L} \rightarrow \mathbb{R}$, when numeric operators are considered. C is* ***monotone*** *if for any P satisfying C, P supersets satisfy C (e.g. $range(P) \geq v\}$). C is* ***anti-monotone*** *if for any P not satisfying C, P supersets do not satisfy C (e.g. $max(P) \leq v$). Let P_1 satisfy C, C is* ***succint*** *if for any P_2 satisfying C, $P_1 \subseteq P_2$ (e.g. $min(P_2) \leq v$). C is* ***convertible*** *w.r.t. an ordering of items R_Σ if for any P satisfying C, P suffixes satisfy C or/and itemsets with P as suffix satisfy C (e.g. $avg(P) \geq v$).*

To illustrate these constraints, consider $\{(t_1, \{a, b, c\}), (t_2, \{a, b, c, d\}), (t_3, \{a, d\})\}$, θ=1 and {a:0,b:1,c:2,d:3} value correspondence. The set of closed full-patterns under the monotone $range(P) \geq 2$ is $\{(\{a, b, c\}, \{t_1, t_2\}), (\{a, d\}, \{t_1, t_3\})\}$; the anti-monotone $sum(P) \leq 1$ is $\{(\{a, b\}, \{t_1, t_2\})\}$; the succint $P \supseteq \{c, d\}$ is $\{(\{a, b, c, d\}, \{t_2\})\}$; and the convertible $avg(P) \geq 2$ is $\{(\{b, c, d\}, \{t_2\})\}$.

3 Related Work

Knowledge-Driven Biclustering. The use of background knowledge to guide biclustering has been increasingly motivated since solutions with good homogeneity and statistical significance may not necessarily be biologically relevant. However, only few biclustering algorithms are able to incorporate background knowledge. AI-ISA [26], GenMiner [18] and scatter biclustering [20] are able to annotate data with functional terms retrieved from repositories with ontologies, and use these annotations to guide the search. COBIC [19] is able to adjust its behavior (maximum-flow/minimum-cut parameters) in the presence of background knowledge. Similarly, the priors and architectures of generative biclustering algorithms can also incorporate background knowledge [10]. However, COBIC and generative peers are not able to deliver flexible biclustering solutions and only consider simplistic constraints. Fang et al. [5] propose a constraint-based algorithm that turns possible the discovery of dense biclusters associated with high-order combinations of single-nucleotide polymorphisms (SNPs). Data-Peeler [7], as well as algorithms from formal concept analysis [15] and bi-sets mining [1], are able to efficiently discover dense biclusters in binary matrices in the presence of (anti-)monotone constraints. However, these last sets of algorithms impose a very restrictive form of homogeneity in the delivered biclusters.

Full-Pattern Mining for Biclustering. The majority of existing full-pattern miners rely on frequent itemset mining with implementations based on bitset vectors to represent transaction-sets. There are two major classes of searches with this behavior. First, Apriori-based searches [8], generally suffering from costs of candidate generation for low support thresholds (commonly required for biological tasks [22]). Efficient implementations include LCM and CLOSE, used respectively by BiModule [22] and GenMiner [18] biclustering algorithms. Second, vertical-based searches, such as Eclat and Carpenter [8]. These searches rely on intersection operations over transaction-sets to generate candidates, requiring structures such as bitset vectors or diffsets. However, for datasets with a high number of transactions the bitset cardinality becomes large, these structures consume a significant amount of memory and operations become costly. MAFIA is an implementation used by DeBi [25]. Only in recent years, a third class of searches without the bottlenecks associated with bitset vectors were made available by extending pattern-growth searches for the discovery of full-patterns using frequent-pattern trees (FP-Trees) annotated with transactions. F2G [13] used by default in BicPAM [11] implements this third type of searches.

Constraint-Based Pattern Mining. A large number of studies explore how constraints can be used with pattern mining. Two major paradigms are available: constraint-programming (CP) and dedicated searches. First, CP allows the pattern mining task to be declaratively defined according to sets of constraints [4,14]. These declarative models are expressive as they can allow mathematical expressions over itemsets and transaction-sets. Nevertheless, due to the poor scalability of CP methods, they have been only used in highly constrained settings, small-to-medium data, or to mine approximative patterns [4,14].

Second, pattern mining methods have been adapted to optimally seize efficiency gains from different types of constraints. Such efforts replace naïve solutions: post-filtering patterns that satisfy constraints. Instead, the constraints are pushed as deeply as possible within the mining step for an optimal pruning of the search space. The nice properties exhibited by constraints, such as anti-monotone and succinct properties, have been initially seized by Apriori methods [21] to affect the generation of candidates. Convertible constraints, can hardly be pushed in Apriori but can be handled by FP-Growth approaches [23]. FICA, FICM, and more recently MCFPTree, are FP-Growth extensions to seize the properties of anti-monotone, succinct and convertible constraints [23]. The inclusion of monotone constraints is more complex. Filtering methods, such as ExAnte, are able to combine anti-monotone and monotone pruning based on reduction procedures [2]. Reductions are optimally handled in FP-Trees [3].

4 Pattern-Based Biclustering with Constraints

BicPAM [11], BicSPAM [12] and BiP [9] are the state-of-the-art algorithms for pattern-based biclustering. They integrate the dispersed contributions of previous pattern-based algorithms and extend them to discover non-constant coherencies and to guarantee their robustness to discretization (by assigning multi-items to a single element [11]), noise and missings. In this section, we propose BiC2PAM (**BiC**lustering with **C**onstraints using **PA**ttern **M**ining) to integrate their contributions and adapt them to effectively incorporate constraints. BiC2PAM is a composition of three major steps: 1) *preprocessing* to itemize real-valued data; 2) *mining* step, corresponding to the application of full-pattern miners; and 3) *postprocessing* to merge, reduce, extend and filter similar biclusters. As follows, *Section 4.1* lists native constraints supported by parameterizations along these steps. *Section 4.2* lists biologically meaningful constraints with properties of interest. Finally, we extend a pattern-growth search to seize efficiency gains from succinct, (anti-)monotone and convertible constraints (*Section 4.3*).

4.1 Native Constraints

Below we list a set of structural constraints that can be incorporated by adapting the parameters that control the behavior of pattern-based biclustering algorithms along their three major steps.

Relevant constraints provided in the pre-processing step:

– combined inclusion of annotations (such as functional terms) with succinct constraints. A functional term is associated with an interrelated group of genes, and thus it can be appended as a new dedicated symbol to the respective transactions/genes, possibly leading to a set of transactions with varying length. Illustrating, consider T_1 and T_2 terms to be respectively associated with genes $\{g_1, g_3, g_4\}$ and $\{g_3, g_5\}$, an illustrative dataset for this scenario would be $\{(g_1, \{a_{11}, .., a_{1m}, T_1\}), (g_2, \{a_{21}, .., a_{2m}\}), (g_3\{a_{31}, .., a_{3m}, T_1, T_2\}), ...\}$. Pattern mining can then be applied on top of these annotated transactions with succinct constraints to guarantee the inclusion of certain terms (such as $P \cap \{T_1, T_2\} \neq 0$). This is useful to discover, for instance, biclusters with genes participating in specific functions of interest.

- ranges of values (or symbols) to ignore from the input matrix, $remove(S)$ where $S \subseteq \mathbb{R}^+$ (or $S \subseteq \mathcal{L}$). In gene expression, elements with default/non-differential expression are generally less relevant and thus can be removed. This is achieved by removing these elements from the transactions. Despite the simplicity of this constraint, this option is not easily supported by peer biclustering algorithms [16].
- minimum coherency strength (or number of symbols) of the target biclusters: $\delta = 1/|\mathcal{L}|$. Decreasing the coherency strength (increasing the number of symbols) reduces the noise-tolerance of the resulting set of bilusters and it is often associated with solutions composed by a larger number of biclusters with smaller areas.
- level of relaxation to handle noise by increasing the η_{ij} noise range (Definition 2). This constraint is used to adjust the behavior of BiC2PAM in the presence of noise or discretization problems (values near a boundary of discretization). By default, one symbol is associated with an element. Yet, this constraint gives the possibility to assign an additional symbol to an element when its value is near a boundary of discretization, or even a parameterizable number of symbols per element for a high tolerance to noise (proof in [11]).

Relevant constraints provided in the mining step:

- minimum pattern length (minimum number of columns in the bicluster).
- stopping criteria: either the anti-monotone minimum support length (minimum number of rows in the bicluster), or iteratively decreasing support until minimum number of biclusters is discovered or minimum area of the input matrix is coverage by the discovered biclusters.
- type of coherency and orientation. Currently, BiC2PAM supports the selection of constant, additive, multiplicative, symmetric, order-preserving and plaid models with coherency on rows or columns (according to [9,11]).
- pattern representation: simple (all coherent biclusters), closed (all maximal biclusters), and maximal (solutions with a compact number of biclusters with a preference towards a high number of columns).

Understandably, constraints addressed at the postprocessing stage are not desirable since they are not able to seize major efficiency gains. Nevertheless, BiC2PAM supports two key types of constraints that could imply additional computational costs, but are addressed with heightened efficiency: *1)* maximum percentage of noisy and missing elements per bicluster (based on merging procedures [11]), and *2)* minimum homogeneity of the target biclusters (using extension and reduction procedures with a parameterizable merit function [11]).

4.2 Biologically Meaningful Constraints

Different types of constraints were introduced in Definition 5. In order to illustrate how such constraints can be specified and instantiated, a symbolic gene expression matrix (and associated "price table") is provided in Fig.2, where the rows correspond to different genes and the values correspond to observed levels of expressions for a specific condition (column). The {-3,-2}, {-1,0,1} and {2,3} sets

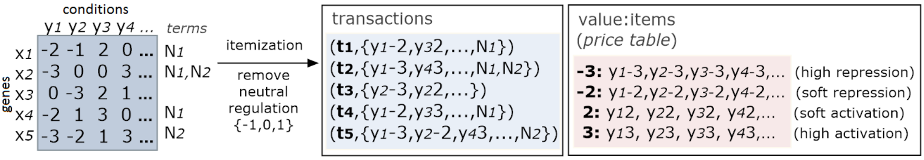
conditions
y1 y2 y3 y4 ...
terms
genes
x1 -2 -1 2 0 ... N1
x2 -3 0 0 3 ... N1,N2
x3 0 -3 2 1 ...
x4 -2 1 3 0 ... N1
x5 -3 -2 1 3 ... N2
itemization
remove neutral regulation {-1,0,1}
transactions
(t1,{y1-2,y32,...,N1})
(t2,{y1-3,y43,...,N1,N2})
(t3,{y2-3,y22,...})
(t4,{y1-2,y33,...,N1})
(t5,{y1-3,y2-2,y43,...,N2})
value:items
(price table)
-3: y1-3,y2-3,y3-3,y4-3,... (high repression)
-2: y1-2,y2-2,y3-2,y4-2,... (soft repression)
2: y12, y22, y32, y42,... (soft activation)
3: y13, y23, y33, y43,... (high activation)

Fig. 2. Illustrative symbolic dataset and "price table" for expression data analysis.

of symbols are respectively associated with repressed (down-regulated), default (preserved) and activated (up-regulated) levels of expression.

First, **succinct** constraints in gene expression analysis allow the discovery of genes with specific constrained levels of expression across a subset of conditions. Illustrating, $min(P)$=-3 implies an interest in biclusters (biological processes) where genes are at least highly repressed in one condition. Alternatively, succinct constraints can be used to discover non-trivial biclusters by focusing on non-highly differential expression (e.g. patterns with symbols {-2,2}). Such option contrasts with the large focus on dense biclusters [16]. Finally, succinct constraints can also be used to guarantee that a specific condition of interest appears in the resulting set (e.g. $P \cap \{y_2\text{-}3, y_2\text{-}2, y_2 2, y_2 3\} \neq \emptyset$ to include y_2), or a specific annotation ($P \cap \{N_1, N_2\} \neq \emptyset$).

Second, **(anti-)monotone** constraints are key to capture background knowledge and guide biclustering. Illustrating, the non-succinct monotonic constraint $countVal(P) \geq 2$ implies that at least two different levels of expression must be present within a bicluster (biological process). In gene expression analysis, biclusters should be able to accommodate genes with different degrees of up-regulation and/or down-regulation. Yet, the majority of existing biclustering approaches are only able to model constant values across conditions [11,16]. When constraints, such as the value-counting inequality, are available, the pruning of the search space allows an efficient handling of very low support thresholds for these non-trivial biclusters to be discovered.

Finally, **convertible** constraints also play an important role in biological settings to guarantee, for instance, that the observed patterns have an average of values within a specific range. Illustrating, the anti-monotonic convertible constraint $avg(P) \leq 0$ indicates a preference for patterns with repression mechanisms without a strict exclusion of activation mechanisms. These constraints are useful to focus the discovery on specific expression levels, while still allowing for noise deviations. Understandably, they are a robust alternative to the use of strict bounds from succinct constraints with maximum-minimum inequalities.

4.3 Effective Use of Constraints in Pattern-based Biclustering

Although native constraints are supported through adequate parameterizations of pattern-based biclustering algorithms, the previous (non-native) constraints are not directly supported. Nevertheless, as surveyed, pattern mining searches have been extended to seize efficiency gains when succinct, (anti-)monotone or convertible constraints are considered. Although there is large consensus that

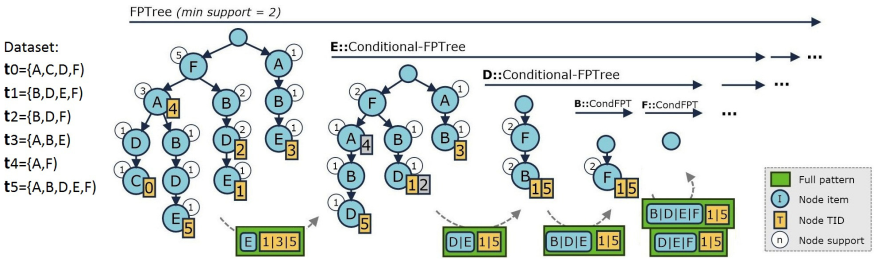

Fig. 3. Illustrative behavior of F2G [13].

pattern-growth searches are better positioned to seize efficiency gains from constraints than peer methods based on bitset vectors, there is not yet proof whether this observation remains valid in the context of full-pattern mining. As such, we extend the recently proposed F2G algorithm to guarantee an optimal pruning of the search space in the presence of constraints and integrate F2G in BiC2PAM. F2G implements a pattern-growth search that does not suffer from efficiency bottlenecks since it relies on tree structures where transaction-IDs are stored without duplicates[2]. F2G behavior is illustrated in Fig.3. In this section, we first show the compliance of F2G with principles to handle succinct and convertible constraints [23]. Second, we show compliance of F2G with principles to handle difficult combinations of monotone and anti-monotone constraints [3].

Compliance with Different Types of Constraints. Unlike candidate generation methods, pattern growth methods (such as FP-Growth) provide further pruning opportunities. Pruning principles can be standardly applied on both the original database (full FP-Tree) and on each projected database (conditional FP-Tree). CFG extensions to FP-Growth [23] seize the properties of such constraints under three simple principles. First, supersets of itemsets violating anti-monotone constraints are removed for each (conditional) FP-Tree (e.g. for $y_1 2$ conditional database, remove conflicting items $\cup_{i=1}^{m}\{y_i 2, y_i 3\}$ as their sum violates $sum(P) \leq 3$). For an effective pruning, it is recommended to order the symbols in the header table according to their value and support [23,24]. F2G is compliant with these removals, since it allows the rising of transaction-IDs in the FP-Tree according to the order of candidate items for removal in the header table (property explained in [13]).

For the particular case of an anti-monotone convertible constraint, itemsets that satisfy the constraint are efficiently generated under a pattern-growth search [24] (e.g. $\{y_1\text{-}3, y_2 2, y_4 2\}$ itemset is not included in the generated pattern set respecting $avg(P) \leq 0$), and provide a simple criterion to either stop FP-tree projections or prune items in a (conditional) FP-Tree.

[2] The FP-tree is recursively mined to enumerate all full-patterns. Unlike peer pattern-growth searches, transaction-IDs are not lost at the first scan. Full-patterns are generated by concatenating the pattern suffixes with the full-patterns discovered from conditional FP-trees where suffixes are removed. F2G is applicable on top of FP-Close trees to mine closed full-patterns [13].

Finally, the removal of conflicting transactions (e.g. t_1 and t_4 does not satisfy the illustrated succinct constraint) and of individual items (e.g. $\cup_{i=1}^{m}\{y_i\text{-}1, y_i0, y_i1\}$) do not cause changes in the FP-Tree construction methods. Additionally, constraint checks can be avoided for subsets of itemsets satisfying a monotone constraint (e.g. no further checks of $countVal(P) \geq 2$ constraint when the range of values in the suffix is $\geq$2 under the $\{y_10, y_11\}$-conditional FP-Tree).

Combination of Constraints. The previous extensions of pattern-growth searches are not able to effectively comply with monotone constraints when anti-monotone constraints (such as minimum support) are also considered. In FP-Bonsai [3], principles to further explore the monotone properties for pruning the search space are considered without reducing anti-monotone pruning opportunities. This method is based on the ExAnte synergy of two data-reduction operations that seize the properties of monotone constraints: μ-reduction, which deletes transactions not satisfying C; and α-reduction, which deletes from transactions single items not satisfying C. Thanks to the recursive projecting approach of FP-growth, the ExAnte data-reduction methods can be applied on each conditional FP-tree to obtain a compact number of smaller FP-Trees (FP-Bonsais). The FP-Bonsai method can be combined with the previously introduced principles, which are particularly prone to handle succinct and convertible anti-monotone constraints. Since F2G can be extended to support the pruning of FP-Trees, it complies with the FP-Bonsai extension.

5 Results and Discussion

In this section, we assess the performance of BiC2PAM on synthetic and real datasets with different types of constraints and three distinct full-pattern miners: AprioriTID[3], Eclat[3] and F2G. BiC2PAM is implemented in Java (JVM v1.6.0-24). The experiments were computed using an IC i5 2.30GHz with 6GB of RAM.

Results on Synthetic Data. The generated data settings are described in Table 1. Biclusters with different shapes and coherency strength ($|\mathcal{L}|\in\{4,7,10\}$) were planted by varying the number of rows and columns using Uniform distributions with ranges in Table 1. For each setting we instantiated 20 matrices with background values generated with Uniform and Gaussian distributions.

Table 1. Properties of the generated dataset settings.

Matrix size ($\sharp$rows × $\sharp$columns)	500×50	1000×100	2000×200	4000×400
Nr. of hidden patterns	5	10	15	25
Nr. transactions for the hidden patterns	[10,14]	[14,30]	[30,50]	[50,100]
Nr. items for the hidden patterns	[5,7]	[6,8]	[7,9]	[8,10]

BiC2PAM was applied with a default merging option (70% of overlapping) and a decreasing support until a minimum number of 50 (maximal) biclusters

[3] http://www.philippe-fournier-viger.com/spmf/

was found. Fig.4 provides the results of parameterizing BiC2PAM with different pattern miners and two simple constraints defining the target coherency strength and symbols to remove. We observe that the proposed F2G miner is the most efficient option for denser data settings (looser coherency). Also, in contrast with existing biclustering algorithms, BiC2PAM seizes large efficiency gains from neglecting specific ranges of values (symbols) from the input matrix.

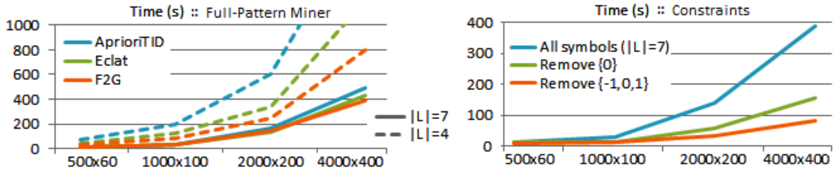

Fig. 4. BiC2PAM performance in the presence of simplistic native constraints.

In order to test the ability of BiC2PAM to seize further efficiency gains in the presence of non-trivial constraints, we fixed the 2000×200 setting with 6 symbols/values {-3,-2,-1,1,2,3}. In the baseline performance, constraints were satisfied using post-filtering procedures. Fig.5 illustrates this analysis. As observed, the use of constraints can significantly reduce the search complexity when they are properly incorporated within the full-pattern mining method. In particular, CFG principles [23] are used to seize efficiency gains from convertible constraints and FP-Bonsai [3] to seize efficiency gains from monotonic constraints.

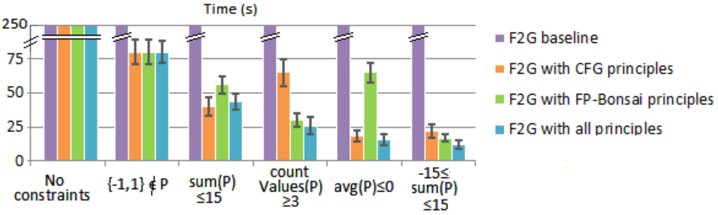

Fig. 5. Efficiency gains of considering constraints in F2G using different principles.

Results on Real Data. Fig.6 shows the (time and memory) efficiency of applying BiC2PAM in the yeast[4] expression dataset with different pattern miners and varying support thresholds for a desirable coherency strength of 10% ($|\mathcal{L}|$=10). The proposed F2G is the most efficient option in terms of time and, along with Apriori, a competitive choice for efficient memory usage.

Finally, Figs.7 and 8 show the impact of biologically meaningful constraints in the efficiency and effectiveness of BiC2PAM. For this purpose, we used the complete gasch dataset (6152×176) [6] with six levels of expression ($|\mathcal{L}|$=6). The effect of constraints in the efficiency is shown in Fig.7. This analysis supports their key role of providing opportunities to solve hard biomedical tasks.

[4] http://www.upo.es/eps/bigs/datasets.html

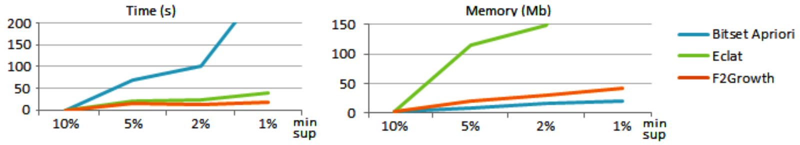

Fig. 6. Computational time and memory of full-pattern miners for yeast (2884×17).

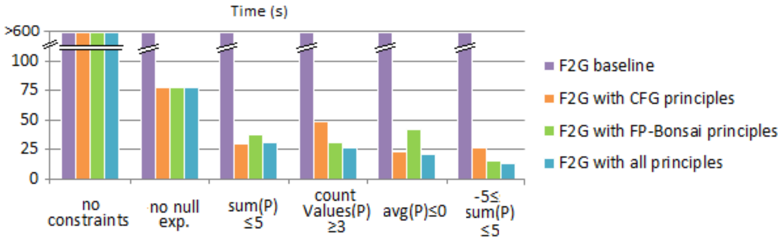

Fig. 7. Efficiency gains from using biological constraints for gasch (6152×176).

The impact of these constraints in the relevance of pattern-based biclustering solutions is illustrated in Fig.8. The biological relevance of each bicluster was derived from the functionally enriched terms using an hypergeometric test of Gene Ontology (GO) annotations [17]. As a measure of significance, we counted the number of terms with Bonferroni corrected p-values below 0.01 [17]. Two major observations can be retrieved. First, when focusing on properties of interest (e.g. differential expression), the average significance of biclusters increases as their genes have higher propensity to be functionally co-regulated. This trend is observed despite the smaller size of the constrained biclusters. Second, when focusing on rare expression profiles (≥3 distinct levels of expression), the average relevance of biclusters slightly decreases as their co-regulation is less obvious. Yet, such non-trivial biclusters hold unique properties with potential interest.

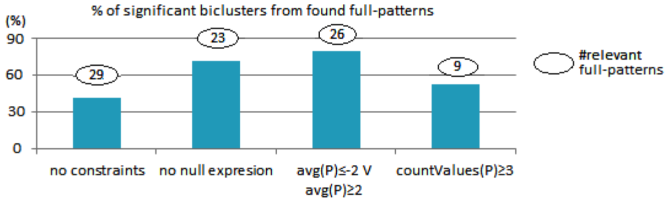

Fig. 8. Biological relevance of F2G for multiple constraint-based profiles of expression.

6 Conclusions

This work motivates the task of biclustering biological data in the presence of constraints. To answer this task, we explore the synergies between pattern-based biclustering and domain-driven pattern mining. As a result, BiC2PAM algorithm is proposed to effectively incorporate constraints derived from user expectations and available background knowledge.

Two major sets of constraints were proposed for the discovery of biclusters with specific interestingness criteria. First, native constraints to guarantee the discovery of biclusters with parameterizable coherency, noise-tolerance and shape, and to consider annotations from knowledge-based repositories. Second, constraints with succinct, monotone, anti-monotone and convertible properties to focus the search space on non-trivial yet biologically meaningful patterns.

In this context, we extended a recent pattern-growth search to optimally explore efficiency gains in the presence of different types of constraints.

Results from synthetic and real data show that biclustering benefits from large efficiency gains in the presence of constraints derived from background knowledge. We further provide evidence of the relevance of the supported types of constraints to discover non-trivial yet meaningful biclusters in expression data.

Acknowledgments. This work was supported by *FCT* under the project UID/CEC/50021/2013 and the PhD grant SFRH/BD/75924/2011 to RH.

References

1. Besson, J., Robardet, C., De Raedt, L., Boulicaut, J.-F.: Mining Bi-sets in numerical data. In: Džeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 11–23. Springer, Heidelberg (2007)
2. Bonchi, F., Giannotti, F., Mazzanti, A., Pedreschi, D.: Exante: a preprocessing method for frequent-pattern mining. IEEE Intel. Systems **20**(3), 25–31 (2005)
3. Bonchi, F., Goethals, B.: FP-Bonsai: the art of growing and pruning small FP-trees. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 155–160. Springer, Heidelberg (2004)
4. Bonchi, F., Lucchese, C.: Extending the state-of-the-art of constraint-based pattern discovery. Data Knowl. Eng. **60**(2), 377–399 (2007)
5. Fang, G., Haznadar, M., Wang, W., Yu, H., Steinbach, M., Church, T.R., Oetting, W.S., Van Ness, B., Kumar, V.: High-Order SNP Combinations Associated with Complex Diseases: Efficient Discovery, Statistical Power and Functional Interactions. Plos One 7 (2012)
6. Gasch, A.P., Werner-Washburne, M.: The genomics of yeast responses to environmental stress and starvation. Functional & integrative genomics **2**(4–5), 181–192 (2002)
7. Guerra, I., Cerf, L., Foscarini, J., Boaventura, M., Meira, W.: Constraint-based search of straddling biclusters and discriminative patterns. JIDM **4**(2), 114–123 (2013)
8. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Min. Knowl. Discov. **15**(1), 55–86 (2007)
9. Henriques, R., Madeira, S.: Biclustering with flexible plaid models to unravel interactions between biological processes. IEEE/ACM Trans, Computational Biology and Bioinfo (2015)
10. Henriques, R., Antunes, C., Madeira, S.C.: Generative modeling of repositories of health records for predictive tasks. Data Mining and Knowledge Discovery, pp. 1–34 (2014)

11. Henriques, R., Madeira, S.: Bicpam: Pattern-based biclustering for biomedical data analysis. Algorithms for Molecular Biology **9**(1), 27 (2014)
12. Henriques, R., Madeira, S.: Bicspam: Flexible biclustering using sequential patterns. BMC Bioinformatics **15**, 130 (2014)
13. Henriques, R., Madeira, S.C., Antunes, C.: F2g: Efficient discovery of full-patterns. In: ECML /PKDD IW on New Frontiers to Mine Complex Patterns. Springer-Verlag, Prague, CR (2013)
14. Khiari, M., Boizumault, P., Crémilleux, B.: Constraint programming for mining n-ary patterns. In: Cohen, D. (ed.) CP 2010. LNCS, vol. 6308, pp. 552–567. Springer, Heidelberg (2010)
15. Kuznetsov, S.O., Poelmans, J.: Knowledge representation and processing with formal concept analysis. Wiley Interdisc. Reviews: Data Mining and Knowledge Discovery **3**(3), 200–215 (2013)
16. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Trans. Comput. Biol. Bioinformatics **1**(1), 24–45 (2004)
17. Martin, D., Brun, C., Remy, E., Mouren, P., Thieffry, D., Jacq, B.: Gotoolbox: functional analysis of gene datasets based on gene ontology. Genome Biology (12), 101 (2004)
18. Martinez, R., Pasquier, C., Pasquier, N.: Genminer: Mining informative association rules from genomic data. In: BIBM, pp. 15–22. IEEE CS (2007)
19. Mouhoubi, K., Létocart, L., Rouveirol, C.: A knowledge-driven bi-clustering method for mining noisy datasets. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012, Part III. LNCS, vol. 7665, pp. 585–593. Springer, Heidelberg (2012)
20. Nepomuceno, J.A., Troncoso, A., Nepomuceno-Chamorro, I.A., Aguilar-Ruiz, J.S.: Integrating biological knowledge based on functional annotations for biclustering of gene expression data. Computer Methods and Programs in Biomedicine (2015)
21. Ng, R.T., Lakshmanan, L.V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. SIGMOD R. **27**(2), 13–24 (1998)
22. Okada, Y., Fujibuchi, W., Horton, P.: A biclustering method for gene expression module discovery using closed itemset enumeration algorithm. IPSJ T. on Bioinfo. **48**(SIG5), 39–48 (2007)
23. Pei, J., Han, J.: Can we push more constraints into frequent pattern mining? In: KDD. pp. 350–354. ACM, New York (2000)
24. Pei, J., Han, J.: Constrained frequent pattern mining: a pattern-growth view. SIGKDD Explor. Newsl. **4**(1), 31–39 (2002)
25. Serin, A., Vingron, M.: Debi: Discovering differentially expressed biclusters using a frequent itemset approach. Algorithms for Molecular Biology **6**, 1–12 (2011)
26. Visconti, A., Cordero, F., Pensa, R.G.: Leveraging additional knowledge to support coherent bicluster discovery in gene expression data. Intell. Data Anal. **18**(5), 837–855 (2014)