Annals of Operations Research manuscript No.
(will be inserted by the editor)

Recovering All Generalized Order-Preserving Submatrices: New Exact Formulations and Algorithms

Andrew C. Trapp · **Chao Li** · **Patrick Flaherty**

Received: date / Accepted: date

Abstract Cluster analysis of gene expression data is a popular and successful way of elucidating underlying biological processes. Typically, cluster analysis methods seek to group genes that are differentially expressed across experimental conditions. However, real biological processes often involve only a subset of genes and are activated in only a subset of environmental or temporal conditions. To address this limitation, Ben-Dor et al. (2003) developed an approach to identify order-preserving submatrices (OPSMs) in which the expression levels of included genes induce the sample linear ordering of experiments. In addition to gene expression analysis, OPSMs have application to recommender systems and target marketing. While the problem of finding the largest OPSM is $\mathscr{NP}$-hard, there have been significant advances in both exact and approximate algorithms in recent years. Building upon these developments, we provide two exact mathematical programming formulations that generalize the OPSM formulation by allowing for the reverse linear ordering, known as the *generalized* OPSM pattern, or GOPSM. Our formulations incorporate a constraint that provides a margin of safety against detecting spurious GOPSMs. Finally, we provide two novel algorithms that iteratively solve mathematical programming formulations to global optimality to recover, for any given level of significance, *all* GOPSMs from a given data matrix. We demonstrate the computational performance and accuracy of our algorithms on real gene expression data sets showing the capability of our developments.

Andrew C. Trapp
Foisie School of Business, Worcester Polytechnic Institute, 100 Institute Rd., Worcester, MA 01609, USA
Tel.: +1-508-831-4935
E-mail: atrapp@wpi.edu

Chao Li
Department of Computer Science, Worcester Polytechnic Institute, 100 Institute Rd., Worcester, MA 01609, USA

Patrick Flaherty
Department of Mathematics and Statistics, University of Massachusetts, Amherst, MA 01003, USA

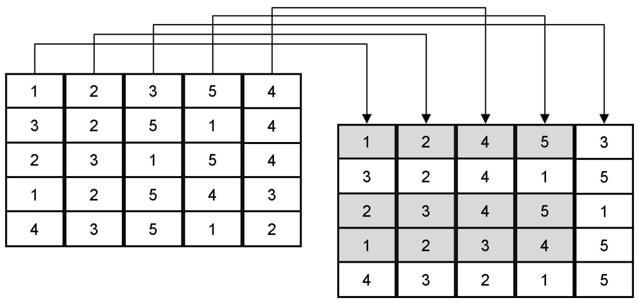
Keywords Order-Preserving Submatrix · Integer Programming · Data Mining · Biclustering

1 Introduction and Background

Given a data matrix $A = (a_{ij})_{m \times n}$, the order-preserving submatrix (OPSM) problem is to identify a progression of features (rows) across a subset of experiments (columns) represented as a “hidden” submatrix within A. In an OPSM the expression levels of all included rows induce a linear ordering across all included columns. The origins of the OPSM problem are in DNA microarray data analysis, where a coherent tendency in multiple features (here, genes) across a number of participating experiments may be indicators of the presence of disease [1]. The decision version of the OPSM problem is $\mathscr{NP}$-Complete [1].

Traditional two-dimensional clustering algorithms attempt to group in a single dimension (i.e., features *or* experiments) across the entirety of the other dimension. In contrast, the OPSM problem is a type of biclustering problem, which allows for a *strict subset* of features to be related to a *strict subset* of experiments. All features in a submatrix have the same coherent tendency (i.e., “up” and “down” pattern) across all included experiments, potentially highlighting regulatory mechanisms that appear in subsets of both features and experiments. For further information on biclustering, we refer to [15] and [2].

A simple example of a data matrix together with a corresponding embedded order-preserving submatrix is illustrated in Figure 1. On the left, the entries in rows 1, 3, 4 exhibit an {“up”, “up”, “up”, “down”} pattern across columns 1, 2, 4, and 5. Alternatively, by permuting columns 3 and 5, it can be seen on the right that an OPSM exists with three rows exhibiting progressively increasing values across the four included columns.



1	2	3	5	4
3	2	5	1	4
2	3	1	5	4
1	2	5	4	3
4	3	5	1	2

1	2	4	5	3
3	2	4	1	5
2	3	4	5	1
1	2	3	4	5
4	3	2	1	5

Fig. 1 Example of an OPSM (right) found in a data matrix A (left) by simple column permutation.

1.1 Some Variations to the OPSM Problem

The OPSM problem has been considered from a variety of perspectives. The original work of Ben-Dor et al. acknowledges that the explicit definition of the OPSM pattern can be somewhat overly restrictive due to the lack of both neatness in biological patterns as well as accuracy in DNA microarray observations [1]. This has led to generalizations that consider approximate order-preserving submatrices [6, 19] as well as OPSMs using fractional and probabilistic support [4, 7, 18].

Other works focus on finding long OPSMs (many columns) with few features. These so-called “twig clusters” may be missed by alternative methods, but have definite biological significance, as pathways/processes exist that require as little as two genes to act in concert across many conditions [8–10].

Another variation is the *GOPSM* pattern, a generalization of the original OPSM pattern, first introduced in [8]. The GOPSM pattern extends the OPSM pattern, which locates subsets of rows and columns obeying the same linear ordering, to allow for the exact reverse linear ordering pattern to be included among the order-preserving rows. The GOPSM framework enables the inclusion of both positively and negatively correlated features among selected columns, thereby generalizing the OPSM problem.

The GOPSM pattern is more applicable to biological gene expression data than the OPSM pattern because genes are both activated and deactivated in response to stimulus or in varying environmental conditions. For example, the response of S. cerevisiae (yeast) to salt stress induces the upregulation of cell stress response genes and the simultaneous downregulation of protein synthesis and RNA processing genes [3]. In cancer, the simultaneous activation of oncogenes and repression of tumor-suppressor genes can lead to more aggressive clinical phenotypes.

An exemplary GOPSM pattern can be seen in Figure 2 for the same data matrix *A* as in Figure 1. We build upon the GOPSM pattern in this work.

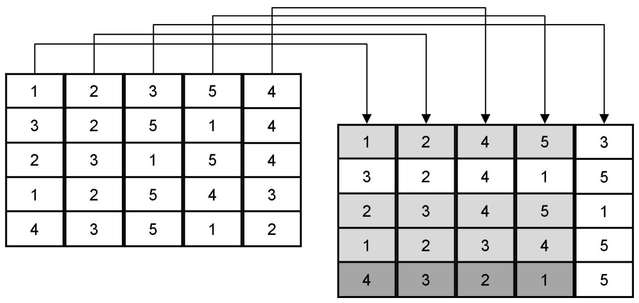


Fig. 2 Example of a GOPSM (right) found in a data matrix *A* (left) by simple column permutation.

1.2 Recent Methodological Developments to Solving the OPSM Problem

Accompanying extensions to the base OPSM problem has been significant methodological progress. Subsequent to the original method proposed by Ben-Dor et al. for finding OPSMs [1], which is essentially a greedy heuristic, various other heuristic approaches have appeared [4, 6–10, 18, 19].

Less frequently, solution approaches to the OPSM problem have been developed that provide guarantees on solution quality. These prefer the exactness of solution at, possibly, the expense of computational runtime. Approximation algorithms have been discussed for the OPSM problem [11]; the authors consider a minimization variant which attempts to remove the least number of rows and columns to ensure that the remaining submatrix satisfies the order-preserving criteria. Trapp and Prokopyev propose and implement the first exact approaches to find a globally maximal OPSM using mixed-integer programming techniques in an integrated algorithmic framework [17]; subsequently, Humrich et al. discuss a number of enhancements that substantially improve tractability [12].

Notably absent from the aforementioned exact studies are explicit discussions to ensure that recovered OPSMs are biologically significant. Moreover, as is customary with optimization routines, these methods return one optimal solution, i.e., the single largest OPSM in A (or possibly per fixed number of columns). On the other hand, it may be of great interest to recover multiple submatrices, so long as they are distinct and meaningful.

1.3 Contributions

We now highlight our contributions. First, we implement as a basis for subsequent extension an exact minimization-based formulation from [11] that attacks the complementary problem of removing the fewest number of rows and columns. The problem is formulated as a binary integer program that, for any fixed number of columns and column ordering, minimizes the number of rows to be excluded to ensure the resultant submatrix is order-preserving. Similar to [17] and [12], the formulation can be embedded within an algorithmic framework that ensures recovering a globally optimal OPSM over all rows and columns.

Second, we expand upon existing exact formulations for solving the OPSM problem [11, 12, 17] by demonstrating how to incorporate the aforementioned *GOPSM* pattern [8]. We extend both maximization-based [12, 17] and minimization-based [11] OPSM optimization formulations to accommodate the GOPSM pattern. Moreover, these GOPSM extensions can be maintained, or omitted, without affecting the validity of the two subsequent contributions, thereby providing modeling flexibility.

Third, we explicitly guard against the possibility of finding false correlations in recovered GOPSMs [see, e.g., 1] for both the maximization- and minimization-variant formulations of the GOPSM problem. Briefly, to satisfy a specified level of significance, there exists a corresponding minimum size (expressed via the number of rows and columns) to which a GOPSM pattern must adhere. Such restrictions can be represented explicitly using constraints in the mathematical formulations, thereby ensuring, for arbitrarily strict criteria, that recovered GOPSMs are of sufficient size.

Fourth, and perhaps most important, we provide two new and complementary algorithms that repeatedly solve the maximization- and minimization-variant formulations to global optimality to recover, for any given level of significance, *all* GOPSMs from a given data matrix. We believe this to be a particularly meaningful contribution due to the potential practical implications. For example, in the context of DNA microarray data analysis, there is value in recovering all GOPSMs that satisfy minimum size criteria; each may indicate distinct sets of genes that are closely coregulated across many experiments, likely revealing unique and previously undiscovered pathways or processes. The fact that any arbitrary strictness can be used to guard against spurious correlation makes the approach especially powerful.

The remainder of this paper is organized as follows. In Section 2 we discuss maximization- and minimization-based formulations to find the largest OPSM for a fixed number of columns and column ordering, and demonstrate how to extend these models to incorporate the GOPSM pattern. In Section 3 we outline how to go beyond state-of-the-art exact methods of recovering a single optimal submatrix, introduce explicit constraints that ensure minimum meaningful size thresholds of any recovered submatrix, and provide two new algorithmic frameworks that can identify *all* corresponding GOPSMs for a given data matrix A. Section 4 covers the computational testing of our proposed methodologies on real biological data. We discuss our computational results in Section 5, and conclude by summarizing our findings in Section 6.

2 Mathematical Formulations

To motivate our discussion, we provide a formal definition of the specific biclustering task that the OPSM problem addresses.

2.1 Formal Definition

The OPSM problem is to identify p rows and ℓ columns from a data set $A = (a_{ij})_{m\times n}$ in which there exists a permutation of the selected columns such that in every supporting row the values corresponding to included columns are strictly increasing [1]. More formally, let $\mathscr{F}_0$ be a set of row indices $\{f_1, f_2, \dots, f_p\}$. Then there exists a permutation of a subset of column indices $\mathscr{S}_0 = \{s_1, s_2, \dots, s_\ell\}$ such that for all $i = 1, \dots, p$ and $j = 1, \dots, \ell - 1$ we have that

$$a_{f_i,s_j} < a_{f_i,s_{j+1}}. \tag{1}$$

The corresponding submatrix $(\mathscr{F}_0, \mathscr{S}_0) \in \mathbb{N}^{p\times\ell}$ is *order-preserving*.

2.2 Existing Exact Formulations and Algorithmic Frameworks

The $n!$ possible permutations of columns, even for small values of n, rapidly becomes prohibitive for exhaustive consideration. One of the key insights in solving the OPSM problem is that the matter of importance is really only over the m column permutations for which at least one of the rows is ordered in increasing fashion. Thus, the $n!$ permutations can be reduced considerably to no more than m orderings, namely just the specific orderings that coincide with those induced from permuting the columns so that the entries of a given row are in increasing order. The exact approach of [17] proposed a mathematical program to find the largest order-preserving submatrix in data matrix A for a fixed column ordering according to specific row h, and coupled it with an algorithmic framework to search over all necessary column orderings to recover a largest OPSM based on submatrix area.

In [12] the authors demonstrate a significant simplification in the variable scope and dimension, introducing a mathematical programming formulation that contains only n binary column variables and m continuous row variables. In addition to iterating over $O(m)$ rows, the simplification of [12] does come with the additional algorithmic expense of iterating over $O(n)$ column levels – in fact, a total of $n-2$, because OPSMs ought to include more than 2 columns to be considered as a legitimate pattern. Nevertheless, the substantial computational savings from their reduced formulation appear to largely offset this modest algorithmic expense.

As first discussed in [17] and later in [12], all of the mathematical programming formulations included in the present work propose to find OPSMs (or GOPSMs) for a *fixed* number of columns γ, where the columns have been permuted so that, for a particular row $h \in \{1,\ldots,m\}$, the entries appear in increasing order. It can be seen from (1) that at least 2 columns are required for a pattern to exist across columns. Because of the somewhat pathological case of $\gamma = 2$, where each row has an equal probability of being in an OPSM for the two included columns, we require $\gamma \geq 3$.

2.2.1 Exact Formulation for the OPSM Problem via Maximization

We next introduce and discuss a formulation that bears resemblance to that of (4) and (5) in [12]. Consider permuting the columns of data matrix A to ensure the entries of a given row h are in increasing order, thereby forming $\hat{A}^h = (\hat{a}^h_{ij})_{m\times n}$. With respect to $\hat{A}^h$, define the index set $I^h_{jk} = \left\{ i : \hat{a}^h_{i,j} > \hat{a}^h_{i,k} \right\}$, so that I^h_{jk} contains the indices of all rows which exhibit a decreasing order across each column pair $(j,k), j<k$ when the columns are permuted according to row h. Then for a fixed number of columns γ, the following formulation will recover a largest OPSM contained in $\hat{A}^h$ that has exactly γ columns:

$$\text{maximize} \quad z = \sum_{i=1}^{m} x_i \tag{2a}$$

$$\text{subject to} \quad \sum_{j=1}^{n} y_j = \gamma, \tag{2b}$$

$$\sum_{i\in I^h_{jk}} x_i + |I^h_{jk}|\, y_j + |I^h_{jk}|\, y_k \leq 2\, |I^h_{jk}|, \quad \forall\, j,k : j<k \tag{2c}$$

$$x \in [0,1]^m,\ y \in \{0,1\}^n. \tag{2d}$$

Objective (2a) maximizes the number of rows in any OPSM corresponding to $\hat{A}^h$, cardinality constraint (2b) ensures that exactly γ out of n columns are chosen, while constraint set (2c) forbids decreasing patterns across included rows and columns.

Formulation (2a)–(2d) differs from that of [12] primarily in the technical update to clear the denominators in the constraint set of [12] that corresponds to our constraint set (2c) above. While on the surface this appears to be a minor modification, its significance is twofold: it improves the numerical stability of the formulation by eliminating representations of fractional coefficients and avoiding roundoff error, and moreover, in the extreme case where $I^h_{jk} = \emptyset$, it ensures the constraints are well-formed.

The optimal objective function value in (2a) is $z^\star$, indicating the maximum number of rows included in an OPSM corresponding to $\hat{A}^h$ containing exactly γ columns; thus the overall size (area) of the recovered OPSM will be $z^\star \cdot \gamma$. For a given γ and h, we refer to formulation (2a)–(2d) as MAXOPSM^h_γ. To identify a globally maximal OPSM in A, one could solve MAXOPSM^h_γ over all values of $h = 1,\dots,m$ and $\gamma = 3,\dots,n$, retaining an OPSM with the largest value of $z^\star \cdot \gamma$.

2.2.2 Exact Formulation for the OPSM Problem via Minimization

The work of [11] introduces a complementary viewpoint of the OPSM problem, that is, that of *excluding the fewest* number of rows and columns to obtain an order-preserving submatrix. They introduce a minimization-based mathematical programming formulation that resembles a set covering problem. The authors do not directly implement the optimization model, but rather design and analyze approximation algorithms to find approximate solutions to the OPSM problem.

We next introduce, and subsequently build upon, the original formulation presented in [11]. The formulation finds a largest OPSM in $\hat{A}^h$, indicated by binary variables r and c that represent whether a particular row or column is *excluded*, taking a value of 1 if so, and 0 otherwise.

$$\begin{aligned} \text{minimize} \quad & n\sum_{i=1}^{m} r_i + m\sum_{j=1}^{n} c_j - \sum_{i=1}^{m}\sum_{j=1}^{n} r_i c_j && (3)\\ \text{subject to} \quad & r_i + c_j + c_k \geq 1, \ \ \forall\, i,\ \forall\, j < k \text{ and } a_{ij} \geq a_{ik}, && (4)\\ & r \in \{0,1\}^m,\ c \in \{0,1\}^n. && (5) \end{aligned}$$

Objective function (3) has nonlinearities due to mn bilinear terms $r_i c_j$. In a manner similar to MAXOPSM^h_γ we can introduce linearity into the objective by fixing the number of columns to γ (here, *to be excluded*); this requires an additional cardinality constraint and considerations over all (practical) fixed levels of γ:

$$\text{minimize} \quad n\sum_{i=1}^{m} r_i - \gamma\sum_{i=1}^{m} r_i + m\gamma. \qquad (6)$$

Objective (6) can be further simplified by eliminating constants and combining (and dropping) coefficients, as shown below in (7a). We can also use the index set I^h_{jk} in the same manner as MAXOPSM^h_γ to represent an equivalent set of constraints to (4) that enforces the OPSM pattern restriction. Specifically, (7c) forbids increasing patterns across included rows and columns. Moreover, we will show in Proposition 1 that the domain of the row variables r can be relaxed to continuous, i.e., $r \in [0,1]^m$. This gives our final minimization-based formulation for finding an OPSM in $\hat{A}^h$:

$$\text{minimize} \quad \zeta = \sum_{i=1}^{m} r_i \tag{7a}$$

$$\text{subject to} \quad \sum_{j=1}^{n} c_j = \gamma, \tag{7b}$$

$$\sum_{i \in I_{jk}^h} r_i + |I_{jk}^h| c_j + |I_{jk}^h| c_k \geq |I_{jk}^h|, \quad \forall\, j,k : j < k, \tag{7c}$$

$$r \in [0,1]^m, \ c \in \{0,1\}^n. \tag{7d}$$

Proposition 1 *Any optimal solution* $(r^\star, c^\star)$ *to formulation* (7a)–(7d) *has* $r^\star \in \{0,1\}^m$.

Proof. Objective (7a) drives the values of the r_i variables toward their lower bound of 0; only constraint set (7c) potentially impedes this. Consider an optimal solution $(r^\star, c^\star)$ to formulation (7a)–(7d). For any column pair $(j,k), j < k$ for which $c_j^\star = 1$ or $c_k^\star = 1$, constraint set (7c) is trivially satisfied, and so does not restrict values of $r_i : i \in I_{jk}^h$. Suppose column pair $(j,k), j < k$ has $c_j^\star = c_k^\star = 0$. This implies that $\sum_{i \in I_{jk}^h} r_i^\star \geq |I_{jk}^h|$, which can only occur precisely when $r_i^\star = 1 \ \forall\, i \in I_{jk}^h$. This shows that $r^\star \in \{0,1\}^m$. ■

Formulation (7a)–(7d) features a linear objective, but algorithmically must now be solved over all m rows as well as $n-2$ columns. The optimal objective function value is $\zeta^\star$, indicating the minimum number of rows excluded from $\hat{A}^h$ that also excludes exactly γ columns; thus the overall size (area) of a recovered maximum OPSM will be $(m - \zeta^\star) \cdot (n - \gamma)$. In the ensuing discussion, for a specific γ and h we refer to (7a)–(7d) as MINOPSM$_\gamma^h$.

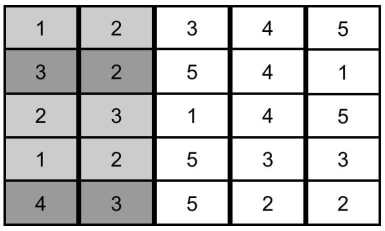

1	2	3	4	5
3	2	5	4	1
2	3	1	4	5
1	2	5	3	3
4	3	5	2	2

Fig. 3 $I_{12}^1 = \{2,5\}$ in darker shade, versus $H_{12}^1 = \{1,3,4\}$ in lighter shade.

2.3 Incorporating the Generalized OPSM Pattern (GOPSM)

We now extend the formal definition of the OPSM pattern from Section 2.1 to incorporate the GOPSM pattern [8]. We introduce $\mathscr{G}_0$ as a (possibly empty) set of q additional, distinct row indices. Let $\mathscr{G}_0 = \{g_1, g_2, \ldots, g_q\}$ if $\mathscr{G}_0 \neq \emptyset$. Then there exists a permutation of a subset of column indices $\mathscr{S}_0 = \{s_1, s_2, \ldots, s_\ell\}$ such that for all $i = 1, \ldots, p$, $h = 1, \ldots, q$, and $j = 1, \ldots, \ell - 1$

$$a_{f_i, s_j} < a_{f_i, s_{j+1}}, \text{ and} \tag{8}$$

$$a_{g_h, s_j} > a_{g_h, s_{j+1}}. \tag{9}$$

First, note that $\mathscr{F}_0 \cap \mathscr{G}_0 = \emptyset$. We term the corresponding submatrix $(\mathscr{F}_0 \cup \mathscr{G}_0, \mathscr{S}_0) \in \mathbb{N}^{(p+q)\times \ell}$ as *generalized order-preserving* (GOPSM). Complementary to I^h_{jk}, let $H^h_{jk} = \left\{ i : \hat{a}^h_{i,j} < \hat{a}^h_{i,k} \right\}$, so that H^h_{jk} contains the indices of all rows exhibiting an increasing order across each column pair $(j,k), j < k$. Thus for all column pairs $(j,k), j < k$, $I^h_{jk} \cap H^h_{jk} = \emptyset$. We next extend MAXOPSM$^h_\gamma$ and MINOPSM$^h_\gamma$ to identify maximum-sized GOPSMs in an arbitrary data matrix A. Figure 3 depicts I^1_{jk} and H^1_{jk} for $j = 1, k = 2$.

2.3.1 Exact Formulations for the GOPSM Problem via Maximization

To accommodate the GOPSM pattern, we introduce a new binary variable vector $v \in \{0,1\}^m$, where $v_i = 1$ indicates that row i is chosen to be in *decreasing* order. The following formulation will find the largest number of rows in a GOPSM according to the permuted data matrix $\hat{A}^h$:

$$\text{maximize} \quad z_G = \sum_{i=1}^{m} (x_i + v_i) \tag{10a}$$

$$\text{subject to} \quad \sum_{j=1}^{n} y_j = \gamma, \tag{10b}$$

$$\sum_{i \in I^h_{jk}} x_i + |I^h_{jk}|\, y_j + |I^h_{jk}|\, y_k \le 2\, |I^h_{jk}|, \;\; \forall\, j,k : j < k, \tag{10c}$$

$$\sum_{i \in H^h_{jk}} v_i + |H^h_{jk}|\, y_j + |H^h_{jk}|\, y_k \le 2\, |H^h_{jk}|, \;\; \forall\, j,k : j < k, \tag{10d}$$

$$x \in [0,1]^m, \; v \in [0,1]^m, \; y \in \{0,1\}^n. \tag{10e}$$

Objective (10a) can attain a value of at most m if all rows are included as either increasing ($x_i = 1$) or decreasing ($v_i = 1$) and there are exactly γ columns in the recovered GOPSM. Because $I^h_{jk} \cap H^h_{jk} = \emptyset$ for all column pairs $(j,k), j < k$, the form of constraint sets (10c) and (10d) naturally ensure that, for all i, at most one of x_i or v_i can be set to 1 (i.e., taken together, it is impossible for $x_i = v_i = 1$). Moreover, in a manner analogous to [12], we can relax the domain of the v variables to continuous without changing the optimal solution, i.e. $v \in [0,1]^m$. A recovered GOPSM will have area of $z^\star_G \cdot \gamma$. For a given γ and h, we refer to formulation (10a)–(10e) as MAXGOPSM$^h_\gamma$.

2.3.2 Exact Formulations for the GOPSM Problem via Minimization

The minimization-based formulation (7a)–(7d) can also be extended to handle the GOPSM pattern. Introduce new binary variable vector $s \in \{0,1\}^m$, where $s_i = 1$ indicates that row i is excluded from being in *decreasing* order. It is not difficult to see that the domain of the s variables can also be relaxed to $s \in [0,1]^m$ without affecting the binary nature of the optimal solution. The following formulation will exclude the fewest rows in a GOPSM according to the permuted data matrix $\hat{A}^h$:

$$\begin{aligned}
\text{minimize} \quad & \zeta_G = \sum_{i=1}^{m} (r_i + s_i) && \text{(11a)}\\
\text{subject to} \quad & \sum_{j=1}^{n} c_j = \gamma, && \text{(11b)}\\
& \sum_{i \in I^h_{jk}} r_i + |I^h_{jk}|\, c_j + |I^h_{jk}|\, c_k \geq |I^h_{jk}|, \;\; \forall\, j,k : j < k, && \text{(11c)}\\
& \sum_{i \in H^h_{jk}} s_i + |H^h_{jk}|\, c_j + |H^h_{jk}|\, c_k \geq |H^h_{jk}|, \;\; \forall\, j,k : j < k, && \text{(11d)}\\
& r \in [0,1]^m, \; s \in [0,1]^m, \; c \in \{0,1\}^n. && \text{(11e)}
\end{aligned}$$

We next show that, for any i, at most one of r_i or s_i can take the value of 0. Consequently, formulation (11a)–(11e), by construction, naturally avoids the prohibitive result of retaining (i.e., not excluding) both increasing and decreasing orders for row i.

Proposition 2 *For any GOPSM corresponding to optimal solution $(r^\star, s^\star, c^\star)$ to formulation* (11a)–(11e) *and all rows $i \in \{1,\dots,m\}$, at most one linear ordering can be chosen; that is, no more than one of $r^\star_i$ or $s^\star_i$ can be* 0.

Proof. Objective function (11a) attempts to set to zero as many r_i and s_i variables as possible. Constraint (11b) ensures any optimal solution $(r^\star, s^\star, c^\star)$ has exactly γ columns excluded, thus $n-\gamma$ columns are not excluded in an optimal GOPSM. Consider two of these non-excluded columns, e.g. j and k, so that $c^\star_j = c^\star_k = 0$, and consider any row $i \in \{1,\dots,m\}$. Suppose $\hat{a}^h_{i,j} > \hat{a}^h_{i,k}$, so that $i \in I^h_{jk}$. The values $c^\star_j = c^\star_k = 0$ in (11c) imply $r_i = 1$. Similarly, if $i \in H^h_{jk}$, then $s_i = 1$ by (11d), so that in either case, no more than one of $r^\star_i$ or $s^\star_i$ is 0. ■

Similar to (7a)–(7d), formulation (11a)–(11e) must be algorithmically solved over all m rows and $n-2$ columns. The recovered GOPSM will have a maximized area of $(m-\zeta^\star_G)\cdot(n-\gamma)$. Hereafter, for a given γ and h we refer to (11a)–(11e) as MINGOPSM$^h_\gamma$.

3 Towards Extracting *All* Meaningful GOPSM Patterns

Our algorithmic procedures are able to identify a *single* largest GOPSM: solve MAXGOPSM$^h_\gamma$ or MINGOPSM$^h_\gamma$ according to each nontrivial fixed column level γ and row $h \in \{1,\dots,m\}$. Moreover, for the goal of finding a GOPSM of globally maximum size, a variety of algorithmic improvements exist to enhance such an implementation, e.g. in [17], where the authors extend this idea to recover a single largest OPSM pattern, *one that corresponds to various levels of γ*. This idea is further expanded upon in [12], where they provide an algorithm to recover a single largest OPSM for every nontrivial level of γ.

So, whether with respect to a fixed column level, or over all column levels, the aforementioned methods return a single optimal solution (if one exists). We expand on these ideas as follows. In the context of DNA microarray data analysis, consider the case of multiple (e.g., two) optimal solutions for MAXGOPSM$_\gamma^h$, a common occurrence in combinatorial optimization problems. Although both patterns may have biological significance, the selection of a "single" optimal solution is left completely to the jurisdiction of the solver, and only one is reported. This realistic setting highlights the importance of identifying multiple GOPSMs, as long as certain size thresholds are met, and the recovered GOPSMs are not submatrices of other recovered GOPSMs (i.e., they should be maximal in the row and column dimensions).

We next demonstrate how to explicitly integrate a size threshold into the MAXGOPSM$_\gamma^h$ and MINGOPSM$_\gamma^h$ formulations.

3.1 Guarding Against Spurious Correlation in Recovered GOPSMs

A general challenge in data mining is not being fooled by randomness, that is, revealed patterns should have a negligible probability of appearing in random data. [1] propose the following method to serve as a proxy for assessing the statistical significance of any obtained order-preserving submatrix. They introduce an upper bound on the probability of having found, at random, an increasing OPSM pattern with γ columns and at least ρ rows as:

$$U(\gamma,\rho) = n\cdots(n-\gamma+1)\sum_{i=\rho}^{m}\binom{m}{i}\left(\frac{1}{\gamma!}\right)^{i}\left(1-\frac{1}{\gamma!}\right)^{(m-i)}. \tag{12}$$

By adapting (12) to accommodate the GOPSM pattern, for which we need to account for precisely two linear orderings (one increasing, and one decreasing), we obtain:

$$U_G(\gamma,\rho) = n\cdots(n-\gamma+1)\sum_{i=\rho}^{m}\binom{m}{i}\left(\frac{2}{\gamma!}\right)^{i}\left(1-\frac{2}{\gamma!}\right)^{(m-i)}. \tag{13}$$

While we recognize that the approach of [1] implies the testing of combinatorially many hypotheses, the upper bounds in (12) and (13) are still useful to guard against spurious correlation. For a fixed number of columns γ and arbitrary significance level α, let ρ_γ^α be the smallest integer number of rows for which $U_G(\gamma,\rho_\gamma^\alpha) \le \alpha$. Then for MAXGOPSM$_\gamma^h$ we can introduce a new constraint that requires a minimum necessary number of rows ρ_γ^α to satisfy a size threshold:

$$\sum_{i=1}^{m}(x_i+v_i) \ge \rho_\gamma^\alpha. \tag{14}$$

We can use (14) to serve as a margin of safety against being fooled by randomness. Note that, for constant α, ρ_γ^α is nonincreasing as γ increases. To accommodate such a large number of hypotheses, in our computational experiments we require very stringent significance levels of α to be observed.

Similarly, for MINGOPSM$^h_\gamma$ an upper bound on the number of rows excluded to ensure statistical significance of a resulting GOPSM can be expressed as:

$$\sum_{i=1}^{m}(r_i + s_i) \leq m + (m - \rho^\alpha_\gamma) = 2m - \rho^\alpha_\gamma, \tag{15}$$

where the right-hand side is derived from the fact that at least one of $r^\star_i = 1$ or $s^\star_i = 1$ for every row $i = 1,\dots,m$, as can be seen from Proposition 2. A constraint in the form of (14) and (15) exists for any level α and every level of γ considered, and each ensure that every recovered GOPSM pattern is of sufficient size.

3.2 Ensuring Maximality of Recovered GOPSMs

For any fixed column level γ and level α, there may be many distinct GOPSMs that satisfy constraints (14) and (15). We now propose a method to discover all such GOPSMs, so long as they are maximal – that is, for the given rows and columns that constitute such a GOPSM, it is not possible to expand in either dimension. This will ensure that any recovered GOPSM is not a proper subset of another. Figure 4 highlights the two dimensions that a submatrix could be non-maximal – in the rows, and in the columns.

Without loss of generality, we assume the perspective of MAXGOPSM$^h_\gamma$ for ease of exposition (a parallel argument exists for MINGOPSM$^h_\gamma$). Consider an algorithmic procedure that iterates over all values of $\gamma \in \{3,\dots,n\}$ and $h \in \{1,\dots,m\}$. For any GOPSM optimal for MAXGOPSM$^h_\gamma$ and (14), say $\Gamma^\star = (x^\star, v^\star, y^\star)$, objective function (10a) already ensures we are maximal with respect to the number of rows, so it is not possible for the row dimension to be suboptimal. To ensure that we are maximal in the column dimension, we can prioritize recovering the largest column-wise GOPSMs first in the algorithmic procedure, by stepping the value of γ from the largest value to the smallest, i.e., $\gamma = n,\dots,3$. Suppose, in a process of iterating γ from n to 3 and $h = 1$ to m, we solve MAXGOPSM$^h_\gamma$ with (14) and recover $\Gamma^\star = (x^\star, v^\star, y^\star)$. We refer to the particular number of columns as $\gamma^\star$. By (10b) $\Gamma^\star$ has exactly $\gamma^\star$ columns and as per (10a) it maximizes the number of included rows z_G. Define $\mathscr{X}^-_{\Gamma^\star} = \{i : x^\star_i = 0\}$, $\mathscr{X}^+_{\Gamma^\star} = \{i : x^\star_i = 1\}$, $\mathscr{V}^-_{\Gamma^\star} = \{i : v^\star_i = 0\}$, $\mathscr{V}^+_{\Gamma^\star} = \{i : v^\star_i = 1\}$, $\mathscr{Y}^-_{\Gamma^\star} = \left\{j : y^\star_j = 0\right\}$, and $\mathscr{Y}^+_{\Gamma^\star} = \left\{j : y^\star_j = 1\right\}$. For this level of $\gamma^\star$, there is no subset of columns for which a greater number of rows exists. Still, for this level of h there may be other GOPSMs that satisfy (14), and we would like to avoid recovering the same $\Gamma^\star$ for this level of h and $\gamma^\star$.

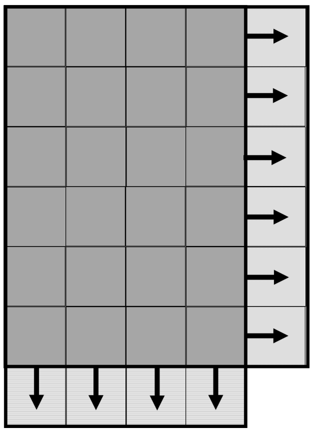


Fig. 4 Two non-maximal possibilities of a GOPSM.

Moreover, at lower levels of $\bar{\gamma} < \gamma^\star$, we would also like to avoid finding a new GOPSM consisting of a strict subset of the columns in $\mathscr{Y}^{+}_{\Gamma^\star}$, if there is no accompanying change in newly included rows (i.e., if no new increasing or decreasing rows are added beyond those appearing in $\Gamma^\star$). However, this concern is irrelevant if at $\bar{\gamma}$ the corresponding level of $\rho^{\alpha}_{\bar{\gamma}}$ exceeds z_G for $\Gamma^\star$– because from (14) there will necessarily be additional rows included in any optimal GOPSM to maintain feasibility.

Hence we can forbid the recovery of $\Gamma^\star$, as well as any GOPSM formed from a strictly smaller subset of its column set $\mathscr{Y}^{+}_{\Gamma^\star}$ together with no accompanying change in new rows, by adding the following family of inequalities, one for each unique subset of column indices:

$$\sum_{i \in \mathscr{X}^{-}_{\Gamma^\star}} x_i + \sum_{i \in \mathscr{V}^{-}_{\Gamma^\star}} v_i + \sum_{j \in \bar{\mathscr{Y}}^{+}_{\Gamma^\star}} (1 - y_j) \geq 1, \ \forall \ \bar{\mathscr{Y}}^{+}_{\Gamma^\star} \subseteq \mathscr{Y}^{+}_{\Gamma^\star} : |\bar{\mathscr{Y}}^{+}_{\Gamma^\star}| \geq 3. \tag{16}$$

Constraints of form (16) can be readily understood through the use of a small example. Suppose we have a data matrix with $m = 10$ rows, $n = 6$ columns, and we are presently considering exactly $\gamma = 4$ columns. For the first row $h = 1$, the MIP according to the sort order $\hat{A}^1$ (constructed using MAXGOPSM$^h_\gamma$ with (14)) is generated and solved. Suppose the recovered GOPSM $\Gamma^\star$ has five included rows, three that are increasing, and two that are decreasing. Suppose that the column indices of this GOPSM are 2, 3, 5, and 6; that rows 1, 7, and 10 are increasing; and that rows 5 and 8 are decreasing. Then we have $\mathscr{X}^{+}_{\Gamma^\star} = \{1,7,10\}$, $\mathscr{X}^{-}_{\Gamma^\star} = \{2,3,4,5,6,8,9\}$, $\mathscr{V}^{+}_{\Gamma^\star} = \{5,8\}$, $\mathscr{V}^{-}_{\Gamma^\star} = \{1,2,3,4,6,7,9,10\}$, $\mathscr{Y}^{+}_{\Gamma^\star} = \{2,3,5,6\}$, and $\mathscr{Y}^{-}_{\Gamma^\star} = \{1,4\}$.

Now for all $\bar{\mathscr{Y}}^{+}_{\Gamma^\star} \subseteq \mathscr{Y}^{+}_{\Gamma^\star} : |\bar{\mathscr{Y}}^{+}_{\Gamma^\star}| \geq 3$, we have a constraint of the form (16). There are five such subsets: $\{2,3,5,6\}$, $\{2,3,5\}$, $\{2,3,6\}$, $\{2,5,6\}$, and $\{3,5,6\}$. This constraint family serves a dual purpose. Consider the first subset, with $\bar{\mathscr{Y}}^{+}_{\Gamma^\star} = \{2,3,5,6\}$. It yields the following constraint: $(x_2 + x_3 + x_4 + x_5 + x_6 + x_8 + x_9) + (v_1 + v_2 + v_3 + v_4 + v_6 + v_7 + v_9 + v_{10}) + (1 - y_2) + (1 - y_3) + (1 - y_5) + (1 - y_6) \geq 1$. Once added, it ensures that the same GOPSM $\Gamma^\star$ cannot be recovered again; further, it forbids *only* this $\Gamma^\star$: it is the only GOPSM for which the left-hand side equals zero.

The second purpose of (16) is now discussed. Constraints (16) for the second, third, fourth, and fifth subsets are very similar in form; we detail only the second, with $\bar{\mathscr{Y}}^{+}_{\Gamma^\star} = \{2,3,5\}$. It yields the following constraint: $(x_2 + x_3 + x_4 + x_5 + x_6 + x_8 + x_9) + (v_1 + v_2 + v_3 + v_4 + v_6 + v_7 + v_9 + v_{10}) + (1 - y_2) + (1 - y_3) + (1 - y_5) \geq 1$. When these four constraints are taken together, they forbid those GOPSMs having the same increasing and decreasing rows, and only a subset of three of the four columns of $\mathscr{Y}^{+}_{\Gamma^\star}$. That is, they forbid the four GOPSMs that are strict subsets with respect to columns. However, they are designed to allow a GOPSM with any subset of three of the columns of $\mathscr{Y}^{+}_{\Gamma^\star}$, as long as a previously inactive row becomes active. This would correspond to a GOPSM involving additional rows, which is distinct from $\Gamma^\star$.

While on the one hand there are combinatorially many constraints of the form (16), mitigating this growth is the fact that they are only relevant when, for some $\bar{\gamma} < \gamma^\star$, the corresponding level of $\rho^{\alpha}_{\bar{\gamma}}$ does not exceed the value of $z^\star_G$ for $\Gamma^\star$. In light of the rate of growth of $\rho^{\alpha}_{\bar{\gamma}}$ for stringent α as $\bar{\gamma}$ decreases from $\gamma^\star$ (see, e.g., Tables 4 and 5), this appears to be rather manageable, as we observe later in our computational experiments.

Theorem 1 *For fixed γ and h, let $\Gamma^\star = (x^\star, v^\star, y^\star)$ represent an optimal solution to* MAXGOPSM$^h_\gamma$ *with* (14)*. Adding inequalities* (16) *to subsequent* MAXGOPSM$^h_\gamma$ *formulations renders infeasible precisely (i) the GOPSM patterns specified by $\Gamma^\star$ as well* *as (ii) those formed from a strictly smaller subset of its column set $\mathscr{Y}^+_{\Gamma^\star}$ together with no accompanying change in new rows. Further, it does not affect feasibility of any other GOPSM patterns.*

Proof. The left-hand side of (16) evaluates to zero for $\Gamma^\star$, so clearly it is forbidden. Suppose there exists at this level of γ another GOPSM pattern $\tilde{\Gamma}$ where $\tilde{\Gamma} \neq \Gamma^\star$, for which the left-hand side of (16) also evaluates to zero. Further, suppose that $\tilde{\Gamma}$ has the identical column set $\mathscr{Y}^+_{\Gamma^\star}$ in common with $\Gamma^\star$, so that $\tilde{\Gamma}$ must differ from $\Gamma^\star$ in $\mathscr{X}^-_{\Gamma^\star}$ or $\mathscr{V}^-_{\Gamma^\star}$. Yet $\Gamma^\star$ already represents the largest GOPSM over this particular column set $\mathscr{Y}^+_{\Gamma^\star}$; it is impossible to increase $\sum_{i\in\mathscr{X}^-_{\Gamma^\star}} x_i + \sum_{i\in\mathscr{V}^-_{\Gamma^\star}} v_i$ over the same column set $\mathscr{Y}^+_{\Gamma^\star}$, for this would imply that $\Gamma^\star$ is suboptimal. So it must be that $\tilde{\Gamma}$ differs from $\Gamma^\star$ in the column set $\mathscr{Y}^+_{\Gamma^\star}$, immediately implying its feasibility in inequality (16).

Now consider a lower level $\bar{\gamma} < \gamma$ for which $\rho^\alpha_{\bar{\gamma}} \leq z^\star_G$. Suppose there exists a GOPSM $\hat{\Gamma} = (\hat{x}, \hat{v}, \hat{y})$ with $\mathscr{X}^-_{\hat{\Gamma}}$, $\mathscr{X}^+_{\hat{\Gamma}}$, $\mathscr{V}^-_{\hat{\Gamma}}$, $\mathscr{V}^+_{\hat{\Gamma}}$, $\mathscr{Y}^-_{\hat{\Gamma}}$, and $\mathscr{Y}^+_{\hat{\Gamma}}$ defined analogously, and with $\mathscr{Y}^+_{\hat{\Gamma}} \subset \mathscr{Y}^+_{\Gamma^\star}$. We want to show that $\hat{\Gamma}$ cannot be a strict submatrix of $\Gamma^\star$. Consider the particular constraint of the form (16) that corresponds to the column subset $\mathscr{Y}^+_{\hat{\Gamma}}$; here, $\sum_{j\in\mathscr{Y}^+_{\hat{\Gamma}}}(1-y_j)$ also evaluates to zero. Thus, the constraint implies that there must be a change in $\mathscr{X}^-_{\hat{\Gamma}}$ or $\mathscr{V}^-_{\hat{\Gamma}}$, thereby preventing $\hat{\Gamma}$ from being a strict submatrix of $\Gamma^\star$. ■

For MINGOPSM$^h_\gamma$, let an optimal $\Gamma^\star$ be denoted by $(r^\star, s^\star, c^\star)$, and similarly define $\mathscr{R}^-_{\Gamma^\star} = \{i : r^\star_i = 0\}$, $\mathscr{R}^+_{\Gamma^\star} = \{i : r^\star_i = 1\}$, $\mathscr{S}^-_{\Gamma^\star} = \{i : s^\star_i = 0\}$, $\mathscr{S}^+_{\Gamma^\star} = \{i : s^\star_i = 1\}$, $\mathscr{C}^-_{\Gamma^\star} = \left\{j : c^\star_j = 0\right\}$, and $\mathscr{C}^+_{\Gamma^\star} = \left\{j : c^\star_j = 1\right\}$. The following inequality forbids $\Gamma^\star$, and can be used to forbid any GOPSMs formed by strict subsets of the indices of $\mathscr{C}^-_{\Gamma^\star}$ if there are no corresponding changes in rows (again, supposing size thresholds specified in (15) remain satisfied for the previous $\Gamma^\star$):

$$\sum_{i\in\mathscr{R}^+_{\Gamma^\star}\cap\mathscr{S}^+_{\Gamma^\star}} \{(1-r_i)+(1-s_i)\} + \sum_{j\in\bar{\mathscr{C}}^-_{\Gamma^\star}} c_j \geq 1,\ \forall\, \bar{\mathscr{C}}^-_{\Gamma^\star} \subseteq \mathscr{C}^-_{\Gamma^\star} : |\bar{\mathscr{C}}^-_{\Gamma^\star}| \geq 3. \qquad (17)$$

We omit the associated proof for MINGOPSM$^h_\gamma$ because of its similarity to Theorem 1.

3.3 Algorithms to Find All GOPSMs of Sufficient Size

We now present two algorithms to find all GOPSMs in a given data matrix A with respect to a prespecified significance α. They are complementary to one another, and are based on the idea of iterating (and so fixing) all nontrivial values of γ columns to include, iterating over rows $h = 1, \ldots m$, and for each level of γ and h, solving either MAXGOPSM$^h_\gamma$ using constraint (14) or MINGOPSM$^h_\gamma$ with constraint (15). For each level of γ and h, each algorithm continues to recover all associated GOPSMs of sufficient size, so long as they are unique and not contained in larger GOPSMs (i.e., they must be maximal). The termination condition is reached when all nontrivial levels of columns and rows have been considered, and the list $\mathscr{L}$ of recovered GOPSMs is returned.

Both Algorithms 1 and 2 solve a sequence of mixed-integer programs to find all GOPSMs that adhere to the minimum size thresholds of (14). Notwithstanding that

Algorithm 1 Find All GOPSMs of Sufficient Size via MAXGOPSM$^h_\gamma$

```
Input: Data matrix $A = (a_{ij})_{m\times n}$, significance level $\alpha$, and precomputed set of $\rho^\alpha_\gamma$ values for all relevant
    values of $\gamma$
 1: Set $\mathscr{L} \leftarrow \emptyset$. {List of all recovered GOPSMs.}
 2: for $\gamma = n,\dots,3$ do {$\gamma$ is (fixed) number of columns to include.}
 3:   for $h = 1,\dots,m$ do
 4:     Set CONTINUE $\leftarrow$ TRUE.
 5:     while CONTINUE do
 6:       Formulate MAXGOPSM$^h_\gamma$ with (14).
 7:       for all $\ell \in \mathscr{L}$ do
 8:         if $(|\mathscr{X}^+_{\Gamma^\star_\ell}| + |\mathscr{V}^+_{\Gamma^\star_\ell}|) \geq \rho^\alpha_\gamma$ then
 9:           Add all $\binom{|\mathscr{Y}^+_{\Gamma^\star_\ell}|}{\gamma}$ inequalities of the form (16).
10:       Solve resulting MIP to global optimality.
11:       if MIP is infeasible then
12:         CONTINUE $\leftarrow$ FALSE.
13:       else
14:         Add new solution $\Gamma^\star$ to $\mathscr{L}$, i.e., $\mathscr{L} \leftarrow \mathscr{L} \cup \Gamma^\star$.
15: return $\mathscr{L}$.
```

they are integer optimization problems, each instance encountered in Step 10 solves relatively quickly, typically in under a minute for the real data sets we later discuss.

Algorithm 2 Find All GOPSMs of Sufficient Size via MINGOPSM$^h_\gamma$

```
Input: Data matrix $A = (a_{ij})_{m\times n}$, significance level $\alpha$, and precomputed set of $\rho^\alpha_\gamma$ values for all relevant
    values of $\gamma$
 1: Set $\mathscr{L} \leftarrow \emptyset$. {List of all recovered GOPSMs.}
 2: for $\gamma = 0,\dots,n-3$ do {$\gamma$ is (fixed) number of columns to delete.}
 3:   for $h = 1,\dots,m$ do
 4:     Set CONTINUE $\leftarrow$ TRUE.
 5:     while CONTINUE do
 6:       Formulate MINGOPSM$^h_\gamma$ with (15).
 7:       for all $\ell \in \mathscr{L}$ do
 8:         if $(|\mathscr{R}^+_{\Gamma^\star_\ell}| + |\mathscr{S}^+_{\Gamma^\star_\ell}|) \leq (2m - \rho^\alpha_\gamma)$ then
 9:           Add all $\binom{|\mathscr{C}^-_{\Gamma^\star_\ell}|}{\gamma}$ inequalities of the form (17).
10:       Solve resulting MIP to global optimality.
11:       if MIP is infeasible then
12:         CONTINUE $\leftarrow$ FALSE.
13:       else
14:         Add new solution $\Gamma^\star$ to $\mathscr{L}$, i.e., $\mathscr{L} \leftarrow \mathscr{L} \cup \Gamma^\star$.
15: return $\mathscr{L}$.
```

4 Computational Experiments

Throughout our computational experiments, we considered the relative ordering of the expression levels for each gene, that is, the ranks, rather than the absolute (exact) measurements. This is consistent with [1] and [17], and alleviates any potential data-scaling issues. We also make explicit that we adhere to the strict monotonically increasing (decreasing) definition for the OPSM (GOPSM) problem of [1]. That is, in the event that $a_{ij} = a_{ik}$ in row i for two columns j and k, *no more than one entry can appear* in any OPSM (GOPSM) pattern.

4.1 Test Sets

We tested our approach on two real biological data sets from the literature. The first is the Cooper promoter data set ($m = 730 \times n = 16$), which contains gene expression levels across 16 cell lines for a variety of promoter sequences, and was introduced in [5]. This data set was previously used for testing of the KiWi algorithm to find OPSMs (subspace clusters) in [10]. The second data set is yeast cell cycle data ($m = 612 \times n = 18$) from [16]. Spellman et al. identified 799 genes that are cell-cycle regulated over 18 points in time. We further reduced the feature space by removing genes for which there was incomplete information, leaving 612 genes under the 18 time points.

4.2 Computational Strategy

We ran Algorithms 1 and 2 for varying stringent α levels to identify GOPSMs. In particular, we allow $\alpha \in \left\{10^{-25}, 10^{-50}, \ldots, 10^{-150}\right\}$. These levels, while conservative, are consistent with those prevalent the literature, and provide a buffer of safety against being fooled by randomness. By allowing α to vary over the proposed range, we can observe the behavior of the algorithm through iterative tightening of this size threshold on recovered GOPSMs.

4.3 Computational Setup

CPLEX 12.5.1 was used to conduct the optimization [13] for the mixed-integer programs (Step 10 in Algorithm 1, and Step 10 in Algorithm 2). We ran the algorithm on code on IBM x3650 server with 2 Intel Xeon E5-2690 CPUs each with 6 cores running at 2.90 GHZ and 128GB of RAM. Each individual optimization problem solved in seconds.

5 Results and Discussion

We now present the computational results of Algorithms 1 and 2 on two biological data sets from the literature.

5.1 Algorithmic Performance

The results on the 730×16 Cooper promoter data set are presented in Table 1, and the results on the 612×18 Spellman yeast cell cycle data set are presented in Table 2. In each table, we report the α level in the first column, the count of recovered

GOPSMs at this level for each fixed number of columns γ, and the computational runtimes, in seconds of CPU time, for both Algorithm 1 (penultimate column, solving the MAXGOPSM$^h_\gamma$ formulation), and Algorithm 2 (final column, solving the MINGOPSM$^h_\gamma$ formulation). We note that the column heading "GOPSMs with γ Columns" uses the convention of γ in the MAXGOPSM$^h_\gamma$ definition, i.e., these are the number of *retained* columns in the recovered GOPSM (the corresponding MINGOPSM$^h_\gamma$ convention for γ can be obtained by subtracting the column heading from n).

Significance	Number of GOPSMs with γ Columns								Runtime (sec.)	
α	3	4	5	6	7	8	9 … 16	Total	MAXGOPSM	MINGOPSM
10^{-25}	77	458	1,320	985	278	26	0	3,144	5,706,000	907,258
10^{-50}	16	153	210	57	0	0	0	436	329,320	190,654
10^{-75}	2	38	42	1	0	0	0	83	111,310	142,841
10^{-100}	0	4	6	0	0	0	0	10	84,504	127,784
10^{-125}	0	1	1	0	0	0	0	2	72,660	119,368
10^{-150}	0	0	0	0	0	0	0	0	67,464	113,110

Table 1 Performance of Algorithms 1 and 2 on (730 × 16) Cooper Promoter data set [5].

Significance	Number of GOPSMs with γ Columns								Runtime (sec.)	
α	3	4	5	6	7	8	9 … 18	Total	MAXGOPSM	MINGOPSM
10^{-25}	20	237	505	300	31	1	0	1,094	1,549,720	1,070,333
10^{-50}	1	15	14	3	0	0	0	33	162,768	190,892
10^{-75}	0	0	0	0	0	0	0	0	133,756	166,762
10^{-100}	0	0	0	0	0	0	0	0	116,391	153,245
10^{-125}	0	0	0	0	0	0	0	0	101,741	142,590
10^{-150}	0	0	0	0	0	0	0	0	92,683	134,351

Table 2 Performance of Algorithms 1 and 2 on (612 × 18) Spellman Yeast data set [16].

γ	3	4	5	6	7	8
Cooper	496	304	165	70	33	16
Spellman	395	213	104	53	27	14

Table 3 Maximum number of rows (increasing + decreasing) in recovered GOPSMs with γ columns.

Table 3 shows the largest GOPSMs recovered in the Cooper and Spellman data sets per level of γ. Figure 5 uses heatmaps [14] to depict two exemplary GOPSMs recovered in each of the Cooper (left) and Spellman (right) data sets, with $\alpha = 10^{-25}$. Each GOPSM has the increasing rows sorted to the top, immediately followed by the decreasing rows. Each pair of depicted GOPSMs was chosen so that the patterns are clearly visible by ensuring their column and row sets are disjoint, though larger GOPSMs than these were recovered (as can be seen in Table 3). The top Cooper GOPSM has 6 columns and 46 rows, while the bottom has 6 columns and 60 rows. The top Spellman GOPSM has 5 columns and 65 rows, while the bottom has 5 columns and 77 rows. Note that all GOPSMs meet or exceed the minimum threshold requirement of 36 and 65 rows for the Cooper and Spellman data sets, respectively.

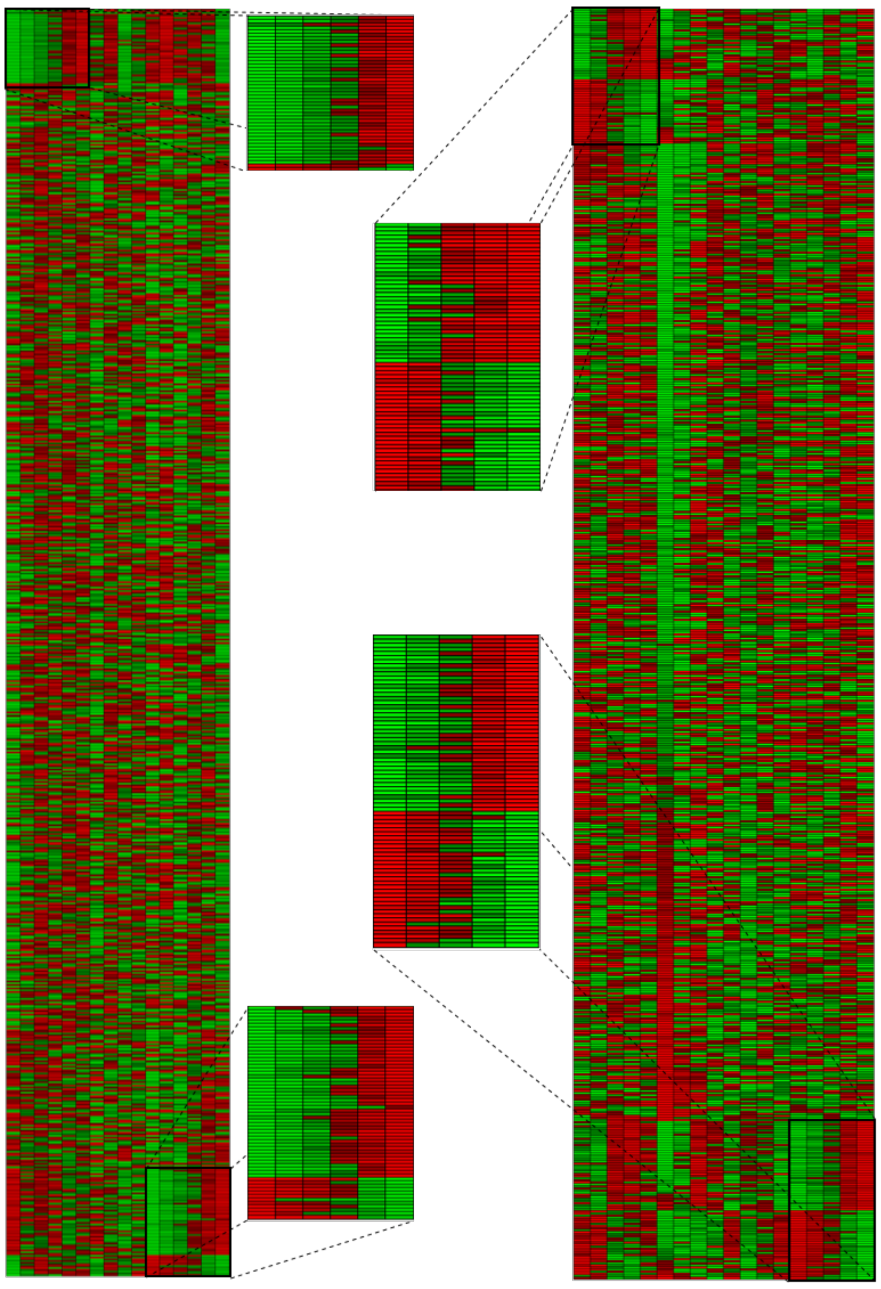


Fig. 5 Heatmaps of two GOPSMs recovered in each of the Cooper (left) and Spellman (right) data sets; $\alpha = 10^{-25}$. Rows and columns are sorted to illustrate the increasing (and decreasing) nature of the patterns.

5.2 Algorithmic Discussion

The results displayed in Tables 1 and 2 reveal several trends. First, both Algorithms 1 and 2 recovered exactly the same number of GOPSMs for both data sets, and across all levels of α. Further inspection of the two sets of GOPSMs confirmed they were identical. Second, it becomes increasingly difficult to recover GOPSMs with more stringent levels of α. This is an intuitive observation, and in fact, for the Spellman data, it can be seen in Table 2 that there were no GOPSMs at the $\alpha = 10^{-75}$ level or beyond. For the Cooper Promoter data, there were exactly two GOPSMs in the range of $10^{-125} \leq \alpha \leq 10^{-150}$, and none beyond.

Also of note is the general trend of GOPSM counts to rise for low levels of γ, peak around $\gamma = 5$, and then decrease as γ increases toward n. The reasons for the initial increase are twofold. First, the ρ_γ^α thresholds for lower γ values are higher than for larger γ values (see Tables 4 and 5). Second, constraints (16) and (17) ensure GOPSMs at lower levels of γ are not subsets of GOPSMs at larger values of γ, hence there is no double-counting. For both data sets, no GOPSMs were recovered with $\gamma \geq 9$; perhaps if the α threshold was lowered below $\alpha = 10^{-25}$, such GOPSMs would be recovered.

Concerning algorithmic performance, it appears Algorithm 2, which solves MIN-GOPSM$_\gamma^h$, excels when there are relatively many GOPSMs to solve, as can be seen for the lower levels of α. Alternatively, Algorithm 1 excels at the more stringent levels of α, exhibiting shorter overall running times to essentially prove infeasibility on the resulting mixed-integer programs found in Step 10 of Algorithm 1 and Step 10 of Algorithm 2, respectively.

6 Conclusions

We explore extensions that generalize the OPSM problem originally proposed by [1], and discuss two *exact* solution approaches to solve these generalizations. We demonstrate how to handle the generalized OPSM (GOPSM) pattern [8] in a mathematical programming context, extending both maximization- [12, 17] and minimization-based [11] OPSM optimization formulations to accommodate the GOPSM pattern. We explicitly integrate the notion of statistical significance [see, e.g., 1] to require that recovered OPSMs meet size thresholds for both the maximization- and minimization-variant formulations of the GOPSM problem. To meet a specified significance level α, there exists a corresponding minimum size (expressed via the number of rows and columns) to which a GOPSM pattern must adhere. This provides for a margin of safety against being fooled by randomness. Such restrictions are explicitly represented using constraints in our mathematical formulations, thereby ensuring, for arbitrary significance level, that recovered GOPSMs are of sufficient size.

Our most important contribution is two new and complementary algorithms that repeatedly solve the maximization- and minimization-variant formulations to global optimality to recover, for any given significance level α, *all* GOPSMs that are of sufficient size. In so doing, our algorithms exploit the properties of optimization to ensure that all GOPSMs are recovered via an iterative process, forbidding recovered GOPSMs as well as related ones that are strict subsets prior to resolving, until the threshold for significance is violated.

We believe this contribution has important practical implications. In the context of DNA microarray data analysis, there is value in recovering all such meaningful

GOPSMs – each may indicate distinct sets of genes that are closely coregulated across many experiments, likely revealing unique and previously undiscovered pathways or processes. The ability to apriori choose a desired level of strictness makes this approach especially powerful.

The findings of our study are somewhat limited by the computational complexity in the column dimension. That is, as the number of columns increases, each associated mixed-integer program in Step 10 of Algorithm 1 and Step 10 of Algorithm 2 become increasingly prohibitive to computationally solve to global optimality; a contributing factor is the subset selection over the columns introduced by the cardinality constraint. Moreover, this is compounded in that our algorithmic approaches solve one or more integer programs for $O(mn)$ column and row combinations.

In the future, it should be further explored why, at least for the two data sets explored, the number of GOPSMs seems to peak around $\gamma = 5$, and further why no GOPSMs were recovered in either data set with $\gamma \geq 9$. This may have to do with the strength of the upper bound computed in (13). Another productive avenue for future research may be to exploit the natural structure of the mixed-integer programs. For both the MAXGOPSM$^h_\gamma$ and the MINGOPSM$^h_\gamma$ formulations, solving over row variables (continuous) is easy once column variables (binary) are fixed, which suggests a decomposition approach such as Benders. There may be potential for such an approach to solve problems with larger column dimensions.

7 Appendix: Additional Algorithmic and Computational Details

Tables 4 and 5 below identify, for a given column level γ, the corresponding minimum number of rows ρ^α_γ necessary for a GOPSM to meet the statistical significance threshold for level α as motivated in [1]. These values are computed via (13), and are used in the construction of constraints (14) and (15). We detail these right-hand side values for both the Cooper promoter (Table 4) and the Spellman yeast (Table 5) data sets.

Significance	GOPSMs with γ Columns													
α	3	4	5	6	7	8	9	10	11	12	13	14	15	16
10^{-25}	390	161	70	36	21	14	11	8	7	6	5	4	4	3
10^{-50}	448	205	98	54	34	24	18	14	11	10	8	7	6	5
10^{-75}	493	242	122	70	45	32	24	19	16	13	11	10	9	8
10^{-100}	530	274	145	85	56	40	31	25	20	17	15	13	11	10
10^{-125}	562	303	165	100	67	48	37	30	24	21	18	16	14	12
10^{-150}	591	330	185	114	77	56	43	35	29	24	21	18	16	14

Table 4 Minimum number of rows required for statistical significance level α on (730×16) Cooper Promoter data set [5].

References

1. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: The order-preserving submatrix problem. Journal of Computational Biology **10**(3-4), 373–384 (2003)

Significance	GOPSMs with γ Columns															
α	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
10^{-25}	339	144	65	34	21	14	11	8	7	6	5	4	4	3	3	3
10^{-50}	392	185	91	51	33	23	17	14	11	10	8	7	6	6	5	5
10^{-75}	432	219	114	67	44	31	24	19	16	13	11	10	9	8	7	6
10^{-100}	466	249	136	82	55	40	30	24	20	17	15	13	11	10	9	8
10^{-125}	495	276	155	96	65	47	37	29	24	21	18	16	14	12	11	10
10^{-150}	520	301	174	109	75	55	43	34	28	24	21	18	16	14	13	12

Table 5 Minimum number of rows required for statistical significance level α on (612×18) Cooper Promoter data set [16].

2. Busygin, S., Prokopyev, O., Pardalos, P.M.: Biclustering in data mining. Computers & Operations Research **35**(9), 2964–2987 (2008)
3. Causton, H.C., Ren, B., Koh, S.S., Harbison, C.T., Kanin, E., Jennings, E.G., Lee, T.I., True, H.L., Lander, E.S., Young, R.A.: Remodeling of yeast genome expression in response to environmental changes. Molecular Biology of the Cell **12**(2), 323–337 (2001)
4. Chui, C.K., Kao, B., Yip, K.Y., Lee, S.D.: Mining order-preserving submatrices from data with repeated measurements. In: The 8th IEEE International Conference on Data Mining (ICDM), pp. 133–142. IEEE (2008)
5. Cooper, S.J., Trinklein, N.D., Anton, E.D., Nguyen, L., Myers, R.M.: Comprehensive analysis of transcriptional promoter structure and function in 1% of the human genome. Genome Research **16**(1), 1–10 (2006)
6. Fang, Q., Ng, W., Feng, J., Li, Y.: Mining bucket order-preserving submatrices in gene expression data. IEEE Transactions on Knowledge and Data Engineering **24**(12), 2218–2231 (2012)
7. Fang, Q., Ng, W., Feng, J., Li, Y.: Mining order-preserving submatrices from probabilistic matrices. ACM Transactions on Database Systems **39**(1), 1–43 (2014)
8. Gao, B.J., Griffith, O.L., Ester, M., Jones, S.J.: Discovering significant OPSM subspace clusters in massive gene expression data. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 922–928. ACM New York, NY, Philadelphia, PA (2006)
9. Gao, B.J., Griffith, O.L., Ester, M., Xiong, H., Zhao, Q., Jones, S.J.: On the deep order-preserving submatrix problem: A best effort approach. IEEE Transactions on Knowledge and Data Engineering **24**(2), 309–325 (2012)
10. Griffith, O.L., Gao, B.J., Bilenky, M., Prychyna, Y., Ester, M., Jones, S.J.: KiWi: A scalable subspace clustering algorithm for gene expression analysis. In: Proceedings of the 3rd International Conference on Bioinformatics and Biomedical Engineering (iCBBE), pp. 1–9. IEEE (2009)
11. Hochbaum, D.S., Levin, A.: Approximation algorithms for a minimization variant of the order-preserving submatrices and for biclustering problems. ACM Transactions on Algorithms **9**(2), 1–12 (2013)
12. Humrich, J., Gartner, T., Garriga, G.C.: A fixed parameter tractable integer program for finding the maximum order preserving submatrix. In: The 11th Interna-

tional Conference on Data Mining (ICDM), pp. 1098–1103. IEEE (2011)
13. IBM: IBM ILOG CPLEX 12.5.1 User's Manual. IBM ILOG CPLEX Division, Incline Village, NV (2015)
14. King, J.Y., Ferrara, R., Tabibiazar, R., Spin, J.M., Chen, M.M., Kuchinsky, A., Vailaya, A., Kincaid, R., Tsalenko, A., Deng, D.X.F., et al.: Pathway analysis of coronary atherosclerosis. Physiological Genomics **23**(1), 103–118 (2005)
15. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE Transactions on Computational Biology and Bioinformatics **1**(1), 24–45 (2004)
16. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Molecular Biology of the Cell **9**(12), 3273–3297 (1998)
17. Trapp, A.C., Prokopyev, O.A.: Solving the order-preserving submatrix problem via integer programming. INFORMS Journal on Computing **22**(3), 387–400 (2010)
18. Yip, K.Y., Kao, B., Zhu, X., Chui, C.K., Lee, S.D., Cheung, D.W.: Mining order-preserving submatrices from data with repeated measurements. IEEE Transactions on Knowledge and Data Engineering **25**(7), 1587–1600 (2013)
19. Zhang, M., Wang, W., Liu, J.: Mining approximate order preserving clusters in the presence of noise. In: IEEE 24^{th} International Conference on Data Engineering (ICDE), pp. 160–168. IEEE (2008)