Noname manuscript No.
(will be inserted by the editor)

Discovering pan-correlation patterns from time course data sets by efficient mining algorithms

Qian Liu, · Shameek Ghosh, · Jinyan Li · Limsoon Wong · Kotagiri Ramamohanarao

Received: date / Accepted: date

Abstract Time-course correlation patterns can be positive or negative, and time-lagged with gaps. Mining all these **correlation patterns** help to gain broad insights on variable dependencies. Here, we prove that diverse types of correlation patterns can be represented by a generalized form of positive correlation patterns. We prove a correspondence between positive correlation patterns and sequential patterns, and present an efficient single-scan algorithm for mining the correlations. Evaluations on synthetic time course data sets, and yeast cell cycle gene expression data sets indicate that: (i) the algorithm has linear time increment in terms of increasing number of variables; (ii) negative correlation patterns are abundant in real-world data sets; and (iii) correlation patterns with time lags and gaps are abundant. Existing methods have only discovered incomplete forms of many of these patterns, and have missed some important patterns completely.

Keywords pan-correlation pattern · time-course data · positive correlation patterns · negative correlation patterns · time-lagged positive correlation patterns · time-lagged negative correlation patterns

Q. Liu, S. Ghosh
Advanced Analytics Institute, University of Technology Sydney, Broadway, NSW 2007, Australia

J. Li
Advanced Analytics Institute, University of Technology Sydney, Broadway, NSW 2007, Australia
Tel.: +61 2 9514 9264
E-mail: Jinyan.Li@uts.edu.au

L. Wong
School of Computing, National University of Singapore, 13 Computing Drive, Singapore

K. Ramamohanarao
Department of Computing and Information Systems, The University of Melbourne, Victoria 3010, Australia

1 Introduction

In real-world applications like finance and health-care, a correlation pattern describes a tightly correlated trend of data changes between two time-course variables. Correlations can be positive or negative. A positive correlation pattern indicates data movements in same directions between a set of variables, whereas a negative correlation pattern moves in opposite directions. Time-course variables have also time-dependent interactions i.e after some time delay. This time-lagged influence between variables is called time-lagged correlation. Thus, four types of correlation patterns exist: basic positive and negative correlation patterns, and time-lagged positive and negative correlation patterns. Also, noise can interrupt the continuity of a correlation, leading to gaps in the correlation. Figures 1(a), (b), (c) and (d) describe the correlations further.

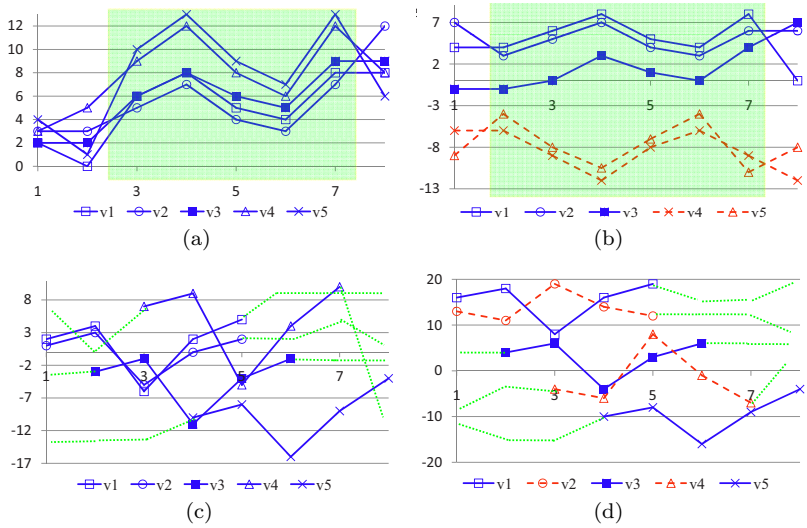


Fig. 1 Examples of basic positive and negative correlation patterns, time-lagged positive and negative correlation patterns. (a) A basic positive correlation pattern. (b) A basic negative correlation pattern. (c) A time-lagged positive correlation pattern. (d) A time-lagged negative correlation pattern. For these negative patterns, one group of variables is drawn using dash lines, and the other group using solid lines. X-axis represents time points. In (a) and (b), the areas in green denotes those time points used in the patterns, while in (c) and (d), the lines in green represent those time points not occurring in the patterns.

This work introduces "pan-correlation patterns", to maximize the sequence of data movements in one pattern. A pan-correlation pattern consists of a maximized sub-list V_0 of all variables, where all the listed variables are associated with a segment of time points having the same length, such that V_0 can be

divided into two not necessarily mutually-exclusive lists of variables V_1 and V_2, satisfying: (i) every pair of variables within V_1 are positively correlated, or time-lag positively correlated, or time-lag positively correlated with gaps; (ii) every pair of variables within V_2 are positively correlated, or time-lag positively correlated, or time-lag positively correlated with gaps; and (iii) every pair of variables between V_1 and V_2 are negatively correlated, or time-lag negatively correlated, or time-lag negatively correlated with gaps. V_1 or V_2 can be empty—in this case, a pan-correlation pattern is simplified as a positive pan-correlation pattern. Thus, a maximal pan-correlation pattern is a set of all possible variables that share highly correlated movement trends, tolerating time delay and noise.

2 Related Work

Mining all significant pan-correlation patterns is complex. Existing algorithms separately detect time-lagged, gap-containing, or the basic subspace correlation patterns. The union of these separate results is sub-optimal to mining significant subspace pan-correlation patterns. Biclustering algorithms have been proposed to detect positive subspace correlation patterns represented in the form of constant-value, shifting and/or positively-scaling biclusters. They may be able to detect a special sub-type of pan-correlation patterns, for example, positive correlation patterns [6] [3] or negative correlation patterns [18], [8] or both positive and negative correlations by [4], [19] or time-lagged positive correlation patterns [2], [5]. Some algorithms have also been proposed to determine time-lagged biclusters in time-course data [9,17,12,15].

We introduce an efficient algorithm for mining all significant pan-correlation patterns. First, we prove that all the different types of correlation patterns can be represented by a generalized form of positive correlation patterns. Second, the time course data set is transformed into a sequential data set containing sequences of "up", "down", and "no-change", for three movement trends of variables. Using this commonly-used idea [13,8,11], the pan-correlation mining problem is converted into a sequential pattern mining problem. For representing negative correlation patterns through the generalized form of positive correlation patterns we employ an opposite-mirror copy [10] of the original sequential data set. Third, we modify the sequential pattern mining algorithm to efficiently prune redundant patterns. Our pan-correlation mining algorithm is tested on synthetic time course data sets and four microarray gene expression time course data sets.

3 Problem formulation and Definitions

Let V be a set of N_V variables $v_1, v_2, \ldots, v_{N_V}$. Let T be a set of N_T consecutive time points $t_1, t_2, \ldots, t_{N_T}$. Here, t_j and t_{j+1} in N_T are two ordered consecutive time points with $t_j \prec t_{j+1}$, indicating that t_j precedes t_{j+1}. Let $m_{i,j}$ denote

the value of variable v_i at time point t_j. A time course data set is then defined by the data matrix $M = [m_{i,j}]_{N_V \times N_T}$.

Definition 1 *A positive correlation pattern p is a pair comprising a subset V_0 of variables in V and a continuous segment T_p of time points in T such that, for every pair of consecutive time points from t_j to t_{j+1} in T_p, the values of all variables in V_0 decrease or increase simultaneously. A positive correlation p is written as $p = \langle V_0, T_p \rangle$.*

Definition 2 (Cf. [8]) *A negative correlation pattern n is a triplet comprising two non-overlapping subsets V_1 and V_2 of variables in V and a continuous segment T_n of time points in T such that, for every pair of consecutive time points from t_j to t_{j+1} in T_n, the values of all variables in V_1 decrease while the values of all variables in V_2 increase, and vice versa. A negative correlation n is written as $n = \langle (V_1, V_2), T_n \rangle$.*

These two definitions describe a synchronized pace of value change without time delay. In fact, some variables in the data matrix M may have influence on others, but the effect may not take place immediately (i.e., after some time delay).

Definition 3 A time-lagged positive correlation pattern kp is a list L of h distinct pairs $\{(v_{x_1}, T_p^1)$, ..., $(v_{x_h}, T_p^h)\}$, such that: (i) $V_0 = \{v_{x_1}, \ldots, v_{x_h}\}$ is a list of not necessarily distinct variables of V; (ii) $T_K^0 = \{T_p^1, \ldots, T_p^h\}$ is a list of continuous time segments of the same length in T; and (iii) for every $1 \leq r < |T_p^1|$ and for every $v_{x_i} \in V_0$, the value of v_{x_i} increases (decreases) from the rth time point in T_p^i to the $(r+1)$th time point in T_p^i if and only if for all other $v_{x_j} \in V_0$, the value of v_{x_j} increases (decreases) from the rth time point in T_p^j to the $(r+1)$th time point in T_p^j. For convenience, a time-lagged positive correlation pattern kp can be written as $kp = \langle V_0, T_K^0 \rangle$.

Definition 4 *A time-lagged negative correlation pattern kn is a pair of distinct lists $\{(v_{x_1}, T_p^1), \ldots, (v_{x_h}, T_p^h)\}$ and $\{(v_{y_1}, T_q^1), \ldots, (v_{y_g}, T_q^g)\}$, such that: (i) $V_1 = \{v_{x_1}, \ldots, v_{x_h}\}$ and $V_2 = \{v_{y_1}, \ldots, v_{y_g}\}$ are two possibly overlapping lists of not necessarily distinct variables of V; (ii) $T_K^1 = \{T_p^1, \ldots, T_p^h\}$ and $T_K^2 = \{T_q^1, \ldots, T_q^g\}$ are two lists of h and g continuous time segments of the same length in T; (iii) for every $1 \leq r < |T_p^1|$ and for every $v_{x_i} \in V_1$, the value of v_{x_i} increases (decreases) from the rth time point in T_p^i to the $(r+1)$th time point in T_p^i if and only if for all other $v_{x_j} \in V_1$, the value of v_{x_j} increases (decreases) from the rth time point in T_p^j to the $(r+1)$th time point in T_p^j; (iv) for every $1 \leq r < |T_p^1|$ and for every $v_{y_i} \in V_2$, the value of v_{y_i} increases (decreases) from the rth time point in T_q^i to the $(r+1)$th time point in T_q^i if and only if for all other $v_{y_j} \in V_2$, the value of v_{y_j} increases (decreases) from the rth time point in T_q^j to the $(r+1)$th time point in T_q^j; and (v) for every $1 \leq r < |T_p^1|$, for every $v_{x_i} \in V_1$, and for every $v_{y_j} \in V_2$, the value of v_{x_i} increases (decreases) from the rth time point in T_p^i to the $(r+1)$th time point in T_p^i if and only if the value of v_{y_j} decreases (increases) from the rth time*

point in T_q^j to the $(r+1)$th time point in T_q^j. For convenience, a time-lagged negative correlation kn is written as $kn = \langle (V_1, V_2), (T_K^1, T_K^2) \rangle$.

A time segment can be extended into a discontinuous time segment to tolerate some small amount of noise. For example, $T_p = [1, 2, 3, 4, 7, 8, 9, 10]$ is a discontinuous time segment containing a gap of length 2 between 4 and 7. The first 4 time points of T_p are continuous from 1 to 4, and the next 4 time points are continuous from 7 to 10. The pattern $p = \{(v, T_p = [1, 2, 3, 4, 7, 8, 9, 10]), (v', T'_p = [1, 2, 3, 4, 5, 6, 7])\}$ is defined as a positive correlation pattern with gaps if the changes of the values of v for any two consecutive time points of $[1, 2, 3, 4]$ are in the same direction as the changes of the values of v' for $[1, 2, 3, 4]$, and the changes of the values of v for any two consecutive time points of $[7, 8, 9, 10]$ are in the same direction as the changes of the values of v' for $[4, 5, 6, 7]$. The data movement trends between the time points 4 and 7 in v are not considered due to the gap.

Next, we introduce the definitions for (time-lagged) positive/negative correlation patterns that contain gaps. A pair of consecutive time points t_i and t_{i+1} is denoted as $tpp_{(i,i+1)}$. In this work, all time-point pairs are pairs of consecutive time points. Let $Tpp = \{tpp_{(i_j, i_j+1)} \mid j = 1, 2, \ldots, h\}$ be an ordered list of h time-point pairs, where $t_{i_j} \prec t_{i_{j+1}}$. Tpp is continuous if and only if for every $1 \leq k \leq h$, $i_k + 1 = i_{k+1}$. Otherwise, Tpp is discontinuous and contains gaps. A continuous Tpp corresponds to a continuous time segment. For example, $\{tpp_{(1,2)}, tpp_{(2,3)}, tpp_{(3,4)}\}$ corresponds to time segment $\{t_1, t_2, t_3, t_4\}$. A discontinuous Tpp may also corresponds to a continuous time segment. For example, $\{tpp_{(1,2)}, tpp_{(3,4)}\}$ correspond to time segment $\{t_1, t_2, t_3, t_4\}$. So, a time segment alone is not sufficient to define the data movements on the time-point pairs and the movement gaps.

Definition 5 A positive pan-correlation pattern is a time-lagged positive correlation pattern with gaps. That is, it is a list of h distinct pairs $\{(v_{x_1}, Tpp_p^1), \ldots, (v_{x_h}, Tpp_p^h)\}$ such that: (i) $\mathcal{V} = \{v_{x_1}, \ldots, v_{x_h}\}$ is a list of not necessarily distinct variables in V; (ii) $\mathcal{TPP} = \{Tpp_p^1, \ldots, Tpp_p^h\}$ is a list of time-point-pair lists of the same length and possibly containing gaps; and (iii) for every $1 \leq r < |Tpp_p^1|$ and for every $v_{x_i} \in \mathcal{V}$, the value of v_{x_i} increases (decreases) at the rth time-point pair in Tpp_p^i if and only if for all other $v_{x_j} \in \mathcal{V}$, the value of v_{x_j} increases (decreases) at the rth time-point pair in Tpp_p^j. For convenience, a positive pan-correlation pattern $\mathcal{C}$ is written as $\mathcal{C} = \langle \mathcal{V}, \mathcal{TPP} \rangle$.

Every continuous time segment $T*$ in the definitions from Definition 1 to Definition 4 can be converted into a continuous Tpp. Thus all correlation patterns by these definitions can be rewritten by using time-point-pair list Tpp to replace time segment $T*$.

Definition 6 A partial order $\sqsubseteq_p$ is defined on positive pan-correlation patterns as follows. Let $\mathcal{C} = \langle \mathcal{V} = \{v_{x_1}, \ldots, v_{x_h}\}, \mathcal{TPP} = \{Tpp_p^1, \ldots, Tpp_p^h\} \rangle$ and $\mathcal{C}' = \langle \mathcal{V}' = \{v_{y_1}, \ldots, v_{y_g}\}, \mathcal{TPP}' = \{Tpp_q^1, \ldots, Tpp_q^g\} \rangle$ be two positive pan-correlation patterns. We say $\mathcal{C} \sqsubseteq_p \mathcal{C}'$ if and only if for each variable $v_{x_*} \in \mathcal{V}$

and any of $Tpp_p^* \in \mathcal{TPP}$ which are associated with v_{x_*}, there is $v_{y_j} \in \mathcal{V}'$ and its $Tpp_q^j \in \mathcal{TPP}'$, such that $v_{x_*} = v_{y_j}$ and $Tpp_p^* \subseteq Tpp_q^j$. The space of positive pan-correlation patterns under this partial order is denoted by $\mathbb{CP}$.

Definition 7 A negative pan-correlation pattern is a time-lagged negative correlation pattern with gaps. That is, it is a pair of distinct lists $\{(v_{x_1}, Tpp_p^1), \ldots, (v_{x_h}, Tpp_p^h)\}$ and $\{(v_{y_1}, Tpp_q^1), \ldots, (v_{y_g}, Tpp_q^g)\}$, such that: (i) $\mathcal{V}_1 = \{v_{x_1}, \ldots, v_{x_h}\}$ and $\mathcal{V}_2 = \{v_{y_1}, \ldots, v_{y_g}\}$ are two possibly overlapping lists of not necessarily distinct variables in V; (ii) $\mathcal{TPP}_K^1 = \{Tpp_p^1, \ldots, Tpp_p^h\}$ and $\mathcal{TPP}_K^2 = \{Tpp_q^1, \ldots, Tpp_q^g\}$ are two lists of time-point-pair lists all with the same length and possibly containing different gaps; (iii) for every $1 \leq r < |Tpp_p^1|$ and for every $v_{x_i} \in \mathcal{V}_1$, the value of v_{x_i} increases (decreases) at the rth time-point pair in Tpp_p^i if and only if for all other $v_{x_j} \in \mathcal{V}_1$, the value of v_{x_j} increases (decreases) at the rth time-point pair in Tpp_p^j; (iv) for every $1 \leq r < |Tpp_p^1|$ and for every $v_{y_i} \in \mathcal{V}_2$, the value of v_{y_i} increases (decreases) at the rth time-point pair in Tpp_q^i if and only if for all other $v_{y_j} \in \mathcal{V}_2$, the value of v_{y_j} increases (decreases) at the rth time-point pair in Tpp_q^j; (v) for every $1 \leq r < |Tpp_p^1|$, for every $v_{x_i} \in \mathcal{V}_1$, and for every $v_{y_j} \in \mathcal{V}_2$, the value of v_{x_i} increases (decreases) at the rth time-point pair in Tpp_p^i if and only if the value of v_{y_j} decreases (increases) at the rth time-point pair in Tpp_q^j. For convenience, a negative pan-correlation pattern $\mathcal{C}$ is written as $\mathcal{C} = \langle(\mathcal{V}_1, \mathcal{V}_2), (\mathcal{TPP}_K^1, \mathcal{TPP}_K^2)\rangle$.

Definition 8 A partial order $\sqsubseteq_n$ is defined on negative pan-correlation patterns as follows. Let $\mathcal{C} = \langle(\mathcal{V}_1, \mathcal{V}_2), (\mathcal{TPP}_K^1, \mathcal{TPP}_K^2)\rangle$ and $\mathcal{C}' = \langle(\mathcal{V}_{1'}, \mathcal{V}_{2'}), (\mathcal{TPP}_K^{1'}, \mathcal{TPP}_K^{2'})\rangle$ be two negative pan-correlation patterns. We say $\mathcal{C} \sqsubseteq_n \mathcal{C}'$ if and only if $\langle \mathcal{V}_1, \mathcal{TPP}_K^1\rangle \sqsubseteq_p \langle \mathcal{V}_{1'}, \mathcal{TPP}_K^{1'}\rangle$ and $\langle \mathcal{V}_2, \mathcal{TPP}_K^2\rangle \sqsubseteq_p \langle \mathcal{V}_{2'}, \mathcal{TPP}_K^{2'}\rangle$. The space of negative pan-correlation patterns under this partial order is denoted by $\mathbb{CN}$.

Definition 9 A partial order $\sqsubseteq$ is defined on the combined collection of positive and negative pan-correlation patterns as follows. Let $\mathcal{C} = \langle \mathcal{V}, \mathcal{TPP}\rangle$ be a positive pan-correlation pattern. Let $\mathcal{C}' = \langle(\mathcal{V}_1,$
$\mathcal{V}_2), (\mathcal{TPP}_K^1, \mathcal{TPP}_K^2)\rangle$ be a negative pan-correlation pattern. We say $\mathcal{C} \sqsubseteq_{pn} \mathcal{C}'$ if and only if $\langle \mathcal{V}, \mathcal{TPP}\rangle \sqsubseteq_p \langle \mathcal{V}_1, \mathcal{TPP}_K^1\rangle$ or $\langle \mathcal{V}, \mathcal{TPP}\rangle \sqsubseteq_p \langle \mathcal{V}_2, \mathcal{TPP}_K^2\rangle$. Then for any two positive or negative pan-correlation patterns $\mathcal{C}_1$ and $\mathcal{C}_2$, we say $\mathcal{C}_1 \sqsubseteq \mathcal{C}_2$ if and only if $\mathcal{C}_1 \sqsubseteq_p \mathcal{C}_2$ when both patterns are in $\mathbb{CP}$, or $\mathcal{C}_1 \sqsubseteq_n \mathcal{C}_2$ when both patterns are in $\mathbb{CN}$, or $\mathcal{C}_1 \sqsubseteq_{pn} \mathcal{C}_2$ when $\mathcal{C}_1$ is in $\mathbb{CP}$ and $\mathcal{C}_2$ is in $\mathbb{CN}$. This combined partially-ordered space of patterns is denoted by $\mathbb{C}$. We also write $\sqsubseteq$ instead of $\sqsubseteq_p$, $\sqsubseteq_n$, and $\sqsubseteq_{pn}$ when the context is clear or the distinction is unimportant.

There are a huge number of positive and negative pan-correlation patterns in the data matrix M. However, we are only interested in those patterns that are maximal with respect to the partial ordering in the respective spaces. These patterns are called closed patterns and more specifically $\mathbb{C}$-, $\mathbb{CP}$-, and $\mathbb{CN}$-closed patterns. The following relationships between the various types of pan-correlation patterns can be easily proved.

Proposition 1 *Let* $\mathcal{C} = \langle(\mathcal{V}_1, \mathcal{V}_2), (\mathcal{TPP}^1_K, \mathcal{TPP}^2_K)\rangle$, $\mathcal{C}_1 = \langle \mathcal{V}_1, \mathcal{TPP}^1_K\rangle$, *and* $\mathcal{C}_2 = \langle \mathcal{V}_2, \mathcal{TPP}^2_K\rangle$. *Then*

- $\mathcal{C}$ *is in* $\mathbb{CN}$ *implies both* $\mathcal{C}_1$ *and* $\mathcal{C}_2$ *are in* $\mathbb{CP}$.
- $\mathcal{C}$ *is in* $\mathbb{CN}$ *if, and only if,* $\mathcal{C}' = \langle(\mathcal{V}_2, \mathcal{V}_1), (\mathcal{TPP}^2_K, \mathcal{TPP}^1_K)\rangle$ *is in* $\mathbb{CN}$.
- $\mathcal{C}$ *is closed in* $\mathbb{C}$ *if, and only if, it is closed in* $\mathbb{CN}$.
- $\mathcal{C}'_1 = \langle(\mathcal{V}_1, \{\}), (\mathcal{TPP}^1_K, \{\})\rangle$ *is closed in* $\mathbb{CN}$ *implies* $\mathcal{C}_1$ *is closed in* $\mathbb{CP}$.
- $\mathcal{C}$ *is closed in* $\mathbb{C}$ *implies for* $i \in \{1, 2\}$, *for every (closed) pattern* $\mathcal{C}' = \langle \mathcal{V}', \mathcal{TPP}'\rangle$ *in* $\mathbb{CP}$ *where* $\mathcal{C}_i \sqsubseteq_p \mathcal{C}'$, *it is the case that* $\mathcal{V}_i = \mathcal{V}'$ *(Note that* $\mathcal{C}_i = \mathcal{C}'$ *does not hold).*

The second point of Proposition 1 implies some degree of redundancy, as the two patterns $\mathcal{C}$ and $\mathcal{C}'$ capture the same correlation information. We will deal with this redundancy later in Section 4.

3.1 Unified representation of all correlation patterns

Let $V*$ be a set of variables v_1*, v_2*, ..., $v_{N_V}*$. Let $m_{i,j}* = -m_{i,j}$ denote the value of variable v_i* at time point t_j, and this value of v_i* at time point t_j is the negation of the value of v_i at time point t_j. A negated time course data set is then defined by the data matrix $M* = [m_{i,j}*]_{N_V \times N_T}$. It is also called a mirror-copy of M. Clearly, whenever the value of v_i increases (decreases) from time point t_j to time point t_{j+1}, the value of v_i* decreases (increases) from time point t_j to time point t_{j+1}. I.e., the value of v_i* moves in the opposite direction of v_i. Let M' be the matrix obtained by adding the negated data matrix $M*$ to the original data matrix M (details are given in Section 4.1). The lemma below follows from this observation and can be easily proved.

Lemma 1 $\mathcal{C} = \langle(\mathcal{V}_1 = \{v_{x_1}, \ldots, v_{x_h}\}, \mathcal{V}_2 = \{v_{y_1}, \ldots, v_{y_g}\}),$ $(\mathcal{TPP}^1_K = \{Tpp^1_p, \ldots, Tpp^h_p\}, \mathcal{TPP}^2_K = \{Tpp^1_q, \ldots, Tpp^g_q\})\rangle$ *is in* $\mathbb{CN}$ *in the data matrix* M *if, and only if,* $\mathcal{C}* = \langle \mathcal{V} = \{v_{x_1}, \ldots, v_{x_h}, v_{y_1}*, \ldots, v_{y_g}*\},$ $\mathcal{TPP} = \{Tpp^1_p, \ldots, Tpp^h_p, Tpp^1_q, \ldots, Tpp^g_q\}\rangle$ *is in* $\mathbb{CP}$ *in the data matrix* M'.

Based on the equivalence above, for $\mathcal{C}$ in $\mathbb{CN}$ with regard to M, we write $\mathcal{C}*$ for its counterpart in $\mathbb{CP}$ with regard to M'. Every closed $\mathbb{CP}$ pattern in the data matrix M' is in a one-to-one correspondence with a closed $\mathbb{CN}$ pattern (also a closed $\mathbb{C}$ pattern) in the data matrix M.

Theorem 1 $\mathcal{C} = \langle(\mathcal{V}_1 = \{v_{x_1}, \ldots, v_{x_h}\}, \mathcal{V}_2 = \{v_{y_1}, \ldots, v_{y_g}\}), (\mathcal{TPP}^1_K = \{Tpp^1_p, \ldots, Tpp^h_p\}, \mathcal{TPP}^2_K = \{Tpp^1_q, \ldots, Tpp^g_q\})\rangle$ *is closed in* $\mathbb{C}$ *in the data matrix* M *if, and only if,* $\mathcal{C}*$ *is closed in* $\mathbb{CP}$ *in the data matrix* M'. *Thus,* $\mathbb{C}$*-closed patterns in* M *are in one-to-one correspondence with* $\mathbb{CP}$*-closed patterns in* M'.

It follows from Theorem 1 and the second-last bullet of Proposition 1 that every $\mathbb{CP}$-closed pattern, say $\langle \mathcal{V}, \mathcal{TPP}\rangle$, in the data matrix M' involving no negated variables, corresponds to the $\mathbb{C}$-closed pattern $\langle(\mathcal{V}, \{\}), (\mathcal{TPP}, \{\})\rangle$ in

the data matrix M, which is also a closed $\mathbb{CP}$-pattern in the data matrix M. But note—and this is subtle, cf. the last point of Proposition 1—that not every $\mathbb{CP}$-closed pattern $\mathcal{C}' = \langle \mathcal{V}', \mathcal{TPP}' \rangle$ in the data matrix M, which implies $\langle (\mathcal{V}', \{\}), (\mathcal{TPP}', \{\}) \rangle$, is necessarily $\mathbb{C}$-closed in the data matrix M (and is thus also not necessarily closed in $\mathbb{CP}$ in the data matrix M').

4 Mining algorithms

Given a time-course data set $M = [m_{i,j}]_{N_V \times N_T}$, let $s_{i,j}$ be the value movement of the variable v_i between time point t_j and t_{j+1} $(= t_j + 1)$. Specifically, $s_{i,j}$ is U (up) if $m_{i,j+1} \geq m_{i,j} + \delta_i$, and is D (down) if $m_{i,j+1} \leq m_{i,j} - \delta_i$, and is O otherwise. Let $R_i = \{s_{i,1}, s_{i,2}, \cdots, s_{i,N_T-1}\}$ be the sequence of all value movements of $v_i \in V$. Let $S = [s_{i,j}]_{N_V \times (N_T-1)}$ be a sequential transaction data set which is easily transformed from M. S has the same variables V as M does, but each variable in S has $N_T - 1$ sequential value movements. In the transformation, δ_i is used to define the scale of the variable v_i's value movement in M.In this work, the δ_i for $v_i \in V$ is set as twenty percents of the absolute difference between the second maximum value of $m_{i,j}$ and the second minimum value of $m_{i,j}$, $1 \leq j \leq N_T$. The maximum value and the minimum value are discarded to avoid some outlier values of v_i in M.

We view S as a set of sequential transactions. And each row R_i in S corresponds to a sequential transaction and is viewed a sequence of value movements (U, D, and O). Given any variable $v_i \in V$ and any ordered set of time-point pairs $Tpp = \{tpp_{(i_j, i_j+1)} \mid j = 1, 2, \ldots, h\}$. Let $f(v_i, Tpp)$ be the list $\{s_{i,i_1}, \ldots, s_{i,i_h}\}$. Thus, $f(v_i, Tpp)$ gives the value movements of v_i during Tpp. We write $f'(v_i, Tpp)$ to denote the list obtained by flipping every U to D and every D to U in $f(v_i, Tpp)$. In S, a sequential pattern is a list of value movements (U, D, and O). A sequential pattern $sp = \{s_1, \ldots, s_h\}$ is said to occur in a sequential transaction R_i if there is a list of time-point pairs $Tpp = \{tpp_{(i_j, i_j+1)} \mid j = 1, 2, \ldots, h\}$, such that $f(v_i, Tpp) = sp$. That is, the value movements specified in the pattern sp occur in the transaction R_i in the same order as they appear in sp, possibly separated by other value movements. We write $supp(sp, S)$ to denote the support of the sequential pattern sp in S. I.e., $supp(sp, S) = \{R_i \in S \mid sp \text{ occurs in } R_i\}$.

The space of all sequential patterns occurring in S is denoted by $\mathbb{SP}$. A closed sequential pattern in $\mathbb{SP}$ is defined below, which is similar to those in previous works [16].

Definition 10 Let sp and sp' be two sequential patterns. We say $sp \leq sp'$ in $\mathbb{SP}$ if, and only if, sp is a subsequence of sp' or is identical to sp', and $supp(sp, S) = supp(sp', S)$. The closed patterns of $\mathbb{SP}$ are the maximal patterns in $\mathbb{SP}$ according to this partial order.

It is obvious that $f(v_{x_i}, Tpp^i) = f(v_{x_j}, Tpp^j)$ for $1 \leq i, j \leq h$, for any pattern $C = \langle \mathcal{V} = \{v_{x_1}, ..., v_{x_h}\}, \mathcal{TPP} = \{Tpp^1, \ldots, Tpp^h\} \rangle$ in $\mathbb{CP}$ in M. The

following easily-proved property connects the closed patterns in $\mathbb{SP}$ of S and those in $\mathbb{CP}$ of M.

Proposition 2 *For every $\mathbb{SP}$-closed pattern sp in S, there is a unique $\mathbb{CP}$-closed pattern $\mathcal{C} = \langle \mathcal{V} = \{v_{x_1}, \ldots, v_{x_h}\}, \mathcal{TPP} = \{Tpp^1, \ldots, Tpp^h\}\rangle$ in M, such that $sp = f(v_{x_i}, Tpp^i)$ for $1 \leq i \leq h$. And for every $\mathbb{CP}$-closed pattern $\mathcal{C} = \langle \mathcal{V} = \{v_{x_1}, \ldots, v_{x_h}\}, \mathcal{TPP} = \{Tpp^1, \ldots, Tpp^h\}\rangle$ in M, there is a $\mathbb{SP}$-closed pattern sp in S, such that $sp = f(v_{x_i}, Tpp^i)$ for $1 \leq i \leq h$. Thus, $\mathbb{SP}$-closed patterns in S are in one-to-one correspondence with $\mathbb{CP}$-closed patterns of M.*

4.1 Opposite mirror copy of S

In $S = [s_{i,j}]_{N_V \times (N_T - 1)}$, a positive correlation pattern is denoted by one sequence of value movements, while a negative correlation pattern is denoted by two sequences of value movements whose value movements are opposite to each other at every position, U vs. D, and D vs. U. To make available in S the unified formulation of positive and negative correlation patterns, an opposite mirror copy of each transaction in S is created and added into S. This data management technique was similarly used by [10] for mining biclusters.

Given the value movements of v_i in S, i.e., $R_i = \{s_{i,1}, s_{i,2}, \ldots, s_{i,N_T-1}\}$, let its opposite mirror copy be $R{*}_i = \{s{*}_{i,1}, s{*}_{i,2}, \ldots, s{*}_{i,N_T-1}\}$ where $s{*}_{i,j}$ is up if $s_{i,j}$ is down, $s{*}_{i,j}$ is down if $s_{i,j}$ is up, and otherwise $s{*}_{i,j} = s_{i,j}$. The opposite mirror copy of all transactions in S are added into S. The new transaction data set is denoted by $S' = [s'_{i,j}]_{2N_V \times (N_T-1)}$, where all R_is of v_is are indexed from 0 to $2N_V$-2 with step 2 in S', and all $R{*}_i$s are indexed from 1 to $2N_V$-1 with step 2. This index strategy is used later. S' is also the sequential transaction data set derived from M'. Then, the crucial theorem below follows immediately from Theorem 1 and Proposition 2.

Theorem 2 *$\mathbb{SP}$-closed patterns in S' are in one-to-one correspondence with $\mathbb{C}$-closed patterns in M.*

4.2 Mine frequent closed sequential value movements in S'

All $\mathbb{SP}$-closed patterns in S' can be detected using efficient algorithms of mining closed sequential patterns. After that, given a $\mathbb{SP}$-closed pattern in S', by Theorem 2, there is a corresponding $\mathbb{CP}$-closed pattern in M', i.e., a $\mathbb{C}$-closed pattern in M. Then, all pan-correlation patterns can be easily obtained from these frequent closed sequential value movements by restoring the time-point pair information and the transaction id information: given a $\mathbb{SP}$-closed pattern sp and its $supp(sp, S)$ with $\{v_{x_1}, ..., v_{x_h}, v_{y_1}{*}, ..., v_{y_g}{*}\}$, the variables from V of M' are grouped in one set while those from $V{*}$ are grouped in another set, indicating the negative correlation between the two sets; then, the time-point pair information associated with sp is detected by matching sp with each variable $v_{x_i} \in supp(sp, S)$ where there might be multiple matches in v_{x_i}, indicating multiple occurrence of sp in v_{x_i}.

4.3 Opposite mirror copy causes redundancy in patterns

In M', every pan-correlation pattern has a mirror image that carries the same information. For example, a negative correlation pattern $\mathcal{C} = \langle(\mathcal{V}_1, \mathcal{V}_2), (\mathcal{TPP}_1, \mathcal{TPP}_2)\rangle$ in M can be represented by $\mathcal{C}* = \langle \mathcal{V}_1 \cup \mathcal{V}_2*, \mathcal{TPP}_1 \cup \mathcal{TPP}_2\rangle$ or $\mathcal{C}*' = \langle \mathcal{V}_1 * \cup \mathcal{V}_2, \mathcal{TPP}_1 \cup \mathcal{TPP}_2\rangle$ in M'. Here, $\mathcal{V}_1*$ is the negation of $\mathcal{V}_1$ and $\mathcal{V}_2*$ is the negation of $\mathcal{V}_2$. Correspondingly in S' from M', $sp = f(v_{x_i}, Tpp^i)$ for $v_{x_i} \in \mathcal{V}_1 \cup \mathcal{V}_2*$ and $sp' = f(v_{y_j}, Tpp^j)$ for $v_{y_j} \in \mathcal{V}_1 * \cup \mathcal{V}_2$, and $sp \neq sp'$. Thus, $\mathcal{C}$ is mined twice in terms of sp or sp' in S'. And once one of sp and sp' is known, there is no need to mine the other because the other can be produced according to the flip relationship between their components. Thus, sp and sp' are redundant. It is easily proved that a closed $\mathcal{C}$ correlation pattern in M is always detected twice in terms of sp and sp' in S'.

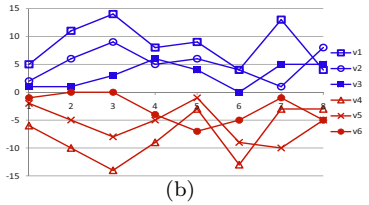
Fortunately, each pair of redundant patterns has some unique property below. Without loss of generality, let a pair of redundant patterns $\mathcal{C}* = \langle(v_{x_1}, Tpp_p^1), \ldots, (v_{x_h}, Tpp_p^h), (v_{y_1}*, Tpp_q^1), \ldots, (v_{y_g}*, Tpp_q^g)\rangle$ and $\mathcal{C}*' = \langle(v_{x_1}*, Tpp_p^1), \ldots, (v_{x_h}*, Tpp_p^h), (v_{y_1}, Tpp_q^1), \ldots, (v_{y_g}, Tpp_q^g)\rangle$ on M' both capture the same information as $\mathcal{C} = \langle(\mathcal{V}_1 = \{v_{x_1}, \ldots, v_{x_h}\}, \mathcal{V}_2 = \{v_{y_1}, \ldots, v_{y_g}\}), (\mathcal{TPP}_K^1 = \{Tpp_p^1, \ldots, Tpp_p^h\}, \mathcal{TPP}_K^2 = \{Tpp_q^1, \ldots, Tpp_q^g\})\rangle$ in M. Here $v*$ is the negation of v. Then, we rewrite $\mathcal{C}* = \{(v_{z_1}, Tpp^1), \ldots, (v_{z_{h+g}}, Tpp^{h+g})\}$ and $\mathcal{C}*' = \{(v'_{w_1}, Tpp^{1'}), \ldots, (v'_{w_{h+g}}, Tpp^{h+g'})\}$. In $\mathcal{C}*$ and $\mathcal{C}*'$, assume that all pairs of (v_{z_*}, Tpp^*) are ordered first according to the transaction indexes of v_{z_*} and then according to the time-point pairs in Tpp^*. After that, it is easily proved that $v_{z_1} = v_{w_1}*$, or $z_1 = w_1$ and $Tpp^1 \leq Tpp^{1'}$, or vice versa.

Thus to avoid producing redundant $\mathbb{SP}$-closed patterns in S', we must modify the algorithm for mining sequential value movements. We apply two constraints below to prune the redundant patterns. (i) On a sub-dataset $S'_s \subseteq S'$ with the ascending order of the indexes of all transactions on S' (Please refer to Section 4.1 for the detail of the indexes), assume R_{x_j} is the first transaction on S'_s , i.e., the transaction with the minimum transaction index. If R_{x_j} is produced from a $v_i* \in V*$, all sequential patterns on S'_s are redundant and thus the search of new sequential patterns on S'_s should be pruned. (ii) Otherwise, given a frequent value movement e (i.e. a value movement U, D or O) on S'_s, let $R_{x_{min1}}$ be the transaction with the minimum id where e occurs, and pos_1 be the first occurrence position of e in $R_{x_{min1}}$; let $R_{x_{min2}}$ be the transaction with the second minimum id where e occurs, and pos_2 be the first occurrence position of e in $R_{x_{min2}}$. If $R_{x_{min1}}$ is produced from $v_i \in V$ and $R_{x_{min2}}$ is produced from $v_i* \in V*$ and $pos_1 > pos_2$, the search in the branch of frequent sequential patterns adding e is redundant and should be pruned. The lemma below is easily proved.

Lemma 2 *Our pruning strategy can guarantee that the closed sequential patterns detected are complete and non-redundant in S'.*

ID	1	2	3	4	5	6	7	8
v1	5	11	14	~~8~~	9	4	13	~~4~~
v2	2	6	9	~~5~~	~~6~~	4	1	8
v3	~~1~~	1	3	6	4	0	5	~~5~~
v4	-6	-10	-14	-9	-3	-13	~~-3~~	~~-3~~
v5	-2	-5	-8	-5	-1	-9	~~-10~~	~~-5~~
v6	~~-1~~	~~0~~	0	-4	-7	-5	-1	-5

(a)



(b)

ID		2−1 a	3−2 b	4−3 c	5−4 d	6−5 e	7−6 f	8−7 g
		The discritized data set						
v1	0	U	U	D	~~Ø~~	D	U	~~D~~
v2	2	U	U	D	~~Ø~~	~~Ø~~	D	U
v3	4	~~Ø~~	U	U	D	D	U	~~Ø~~
v4	6	D	D	U	U	D	~~U~~	~~Ø~~
v5	8	D	D	U	U	D	~~Ø~~	~~U~~
v6	10	~~Ø~~	~~Ø~~	D	D	U	U	D
		The opposite discritized data set						
v1	1	D	D	U	~~Ø~~	U	D	U
v2	3	D	D	U	~~Ø~~	~~U~~	U	D
v3	5	~~Ø~~	D	D	U	U	D	~~Ø~~
v4	7	U	U	D	D	U	~~D~~	~~Ø~~
v5	9	U	U	D	D	U	~~Ø~~	~~D~~
v6	11	~~Ø~~	~~Ø~~	U	U	D	D	U

(c)

ID		a	b	c	d	e	f	g
v1	0	U	U	D	~~Ø~~	D	U	~~D~~
	1	D	D	U	~~Ø~~	U	D	~~U~~
v2	2	U	U	D	~~Ø~~	~~D~~	D	U
	3	D	D	U	~~Ø~~	~~U~~	U	D
v3	4	~~Ø~~	U	U	D	D	U	~~Ø~~
	5	~~Ø~~	D	D	U	U	D	~~Ø~~
v4	6	D	D	U	U	D	~~U~~	~~Ø~~
	7	U	U	D	D	U	~~D~~	~~Ø~~
v5	8	D	D	U	U	D	~~Ø~~	~~U~~
	9	U	U	D	D	U	~~Ø~~	~~D~~
v6	10	~~Ø~~	~~Ø~~	D	D	U	U	D
	11	~~Ø~~	~~Ø~~	U	U	D	D	U

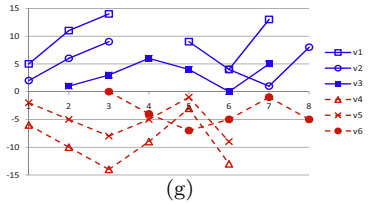
(d)

Pattern 1	U,U,D,D,U
Occurrence 1	0,2,4,7,9,11
Pattern 2	D,D,U,U,D
Occurrence 2	1,3,5,6,8,10

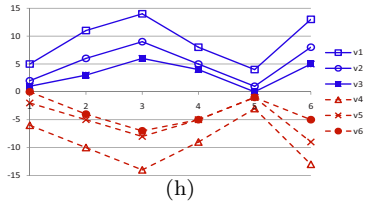
(e)

Pattern 1	U,U,D,D,U	
Occurrence 1	group 1	v1,v2,v3
	group 2	v4,v5,v6

(f)



(g)



(h)

Fig. 2 An illustrative example of our algorithm. (a) An example of time-course data set M. (b) The plot of the example data set. (c) The discretized data set. (d) The combined data set using the opposite mirror copy strategy. (e) The negative pan-correlation patterns. (f) The pattern matching in the original data. (g) The plot of the pattern with gaps and lagged time points. (h) The plot of the pattern merging gaps and ignoring lagged time points (for visualization only). The strike-through ~~numbers~~, ~~U~~, ~~Ø~~ and ~~D~~ indicate those values and value movements not in the detected patterns in (e). From (c) to (f), U indicates Up-changed, O no change, while D Down-changed.

4.4 Parameter setting

Three parameters, min_V, min_{TPP} and max_O, are used to prune trivial correlation patterns. In a given pan-correlation pattern $\mathcal{C} = \langle \mathcal{V}, \mathcal{TPP} \rangle$, min_V is the

minimum size of $\mathcal{V}$, min_{TPP} is the minimum size of $Tpp \in \mathcal{TPP}$, and max_O is the maximum number of O contained.

An illustrative example. Figure 2 illustrates how our algorithm works. A time-course data set M has six variables and eight time points. M is shown in Figure 2(a) and visualized in Figure 2(b). Figure 2(b) does not easily show a very nice pan-correlation between the six variables. But our algorithm can discover a good negative correlation pattern among the six variables.

By our algorithm, M is firstly discretized to obtain a sequential data S in the first part of Figure 2(c). Then the opposite mirror copy of all sequences in S, as shown in the second part of Figure 2(c), is constructed using the strategy in Section 4.1. All sequences in Figure 2(c) comprise S' in Figure 2(d). With $min_V = 5$ and $min_{TPP} = 5$, two pan-correlation patterns are available in S', as shown in Figure 2(e). It can be clearly seen from Figure 2(e) that these two pan-correlation patterns are the same in the original data M, which can be represented in Figure 2(f). Our algorithm can prune the redundancy and only outputs this pattern (visualized in Figure 2(g)). If the gaps are merged and the time points lagged are ignored (for visualization only), this correlation pattern is shown in Figure 2(h).

5 Performance Evaluation and Application

Our algorithm was tested on both synthetic time-course data sets and real-world time-course data sets of biomedical gene expression.

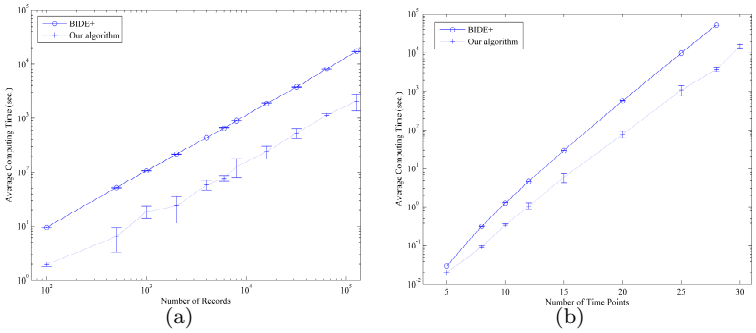


Fig. 3 The assessment on the synthetic data. Both min_{TPP} and min_V are set to 2, and max_O to the number of time points. (a) The computing time (sec.) when the number of variables increases. (b) The computing time (sec.) when the number of time points increases.

Two series of synthetic data sets are used. The first series of data sets have the same number of time points but have an increasing number of variables. The second series of data sets have the same number of variables but have an increasing number of time points. The values in these data sets are randomly chosen from $\{-150, -148, -146, \ldots, 150\}$. The efficiency of BIDE+

without our pruning strategies is also evaluated on the mirror-copy datasets of the synthetic data. This performance is used for the comparison to show the contribution of our algorithm.

Our algorithm was applied to the first series of data sets to see its scalability when the variable size increases. We set the number of time points as $N_T = 20$, and increase N_V from 100 to 500, 1,000, 2,000, 4,000, 6,000, 8,000, 16,000, 32,000, 64,000, and to 128,000. The data at each N_V are randomly produced three times to avoid some randomization effect. The average computing time costs are shown in Figure 3(a). It can be seen that the computing time cost by our algorithm increases very slowly. It has approximately linear increment of time complexity with increasing N_V. In particular when $N_V = 128,000$, the average computing time is about 30 minutes. This indicates that our algorithm is efficient to mine all pan-correlation patterns on those data sets with very large number of variables or transactions, such as time-course gene expression data set where hundreds of thousands of genes are detectable at the same time. Figure 3(a) also shows that BIDE+ without our pruning strategies is more than nine times slower than our algorithm when $N_V = 128,000$.

Both our algorithm and BIDE+ without our pruning strategies were also applied to the second series of synthetic data sets to examine its scalability when the size of time points increases. We keep the number of variables always as $N_V = 5000$ and randomly produce data sets with N_T varying from 5 to 8, 10, 12, 15, 20, 25, 28, and to 30. The data sets of each N_T are also randomly produced three times to avoid the randomization effect. The average computing time costs are shown in Figure 3(b). The computing costs increase exponentially when the number of time points N_T increases. This means that the current algorithm cannot handle well for data sets of large N_T. It is one of our future works to overcome this problem. Again, Figure 3(b) suggests that BIDE+ without our pruning strategies is more than 14 times slower than our algorithm when $N_T = 28$, and BIDE+ without our pruning strategies cannot finish after 24 hours when $N_T = 30$ (its computing time for $N_T = 30$ is thus not shown.). In conclusion, our algorithm is much faster than sequential pattern mining algorithms to detect pan-correlation patterns.

Our algorithm was also evaluated on four real-life microarray gene expression data sets: *alpha, cdc15, elu* [14], and *cdc28* [1]. All of them are time-course gene expression data related to Yeast cell cycle. *elu, cdc28, alpha* and *cdc15* involve 14, 17, 18 and 24 time points, respectively. The four data sets have 5,114 common genes each with less than 3 missing values. Our algorithm is able to detect significant pan-correlation patterns efficiently with less than 7 minutes.

At the min_{TPP} level of 70% of N_T (i.e., spanning at least 10, 12, 13 and 17 time-point pairs in *elu, cdc28, alpha* and *cdc15* respectively), our algorithm detects 1,934 $\mathbb{C}$ pan-correlation patterns in *elu*, 5,942 in *cdc28*, 13,693 in *alpha* and 139,811 in *cdc15*. This filtering results in 588, 2,392, 3,191 and 9,501 non-overlapping $\mathbb{C}$ correlation patterns in *elu, cdc28, alpha* and *cdc15*, respectively.

We examine the correlation coefficient of the variables in our pan-correlation patterns to demonstrate that highly correlated patterns cannot be observed,

if the time lagging effect is not considered. Given a pan-correlation pattern $\mathcal{C} = \langle \mathcal{V}, \mathcal{TPP} \rangle$, its Pearson's correlation coefficient PCC is calculated by Equation 1.

$$PCC = \frac{\sum_{v_{x_i} \in \mathcal{V}, v_{x_j} \in \mathcal{V}, x_i \neq x_j} abs(p(v_{x_i}, v_{x_j}))}{(\|\mathcal{V}\| \times (\|\mathcal{V}\| - 1))} \quad (1)$$

where $abs(*)$ returns the absolute value of $*$, $p(v_{x_i}, v_{x_j})$ is the Pearson's correlation coefficient between the value movements of two variables v_{x_i} and v_{x_j} on all time points in the original time-course data, and $\|\mathcal{V}\|$ is the number of unique variables in $\mathcal{V}$.

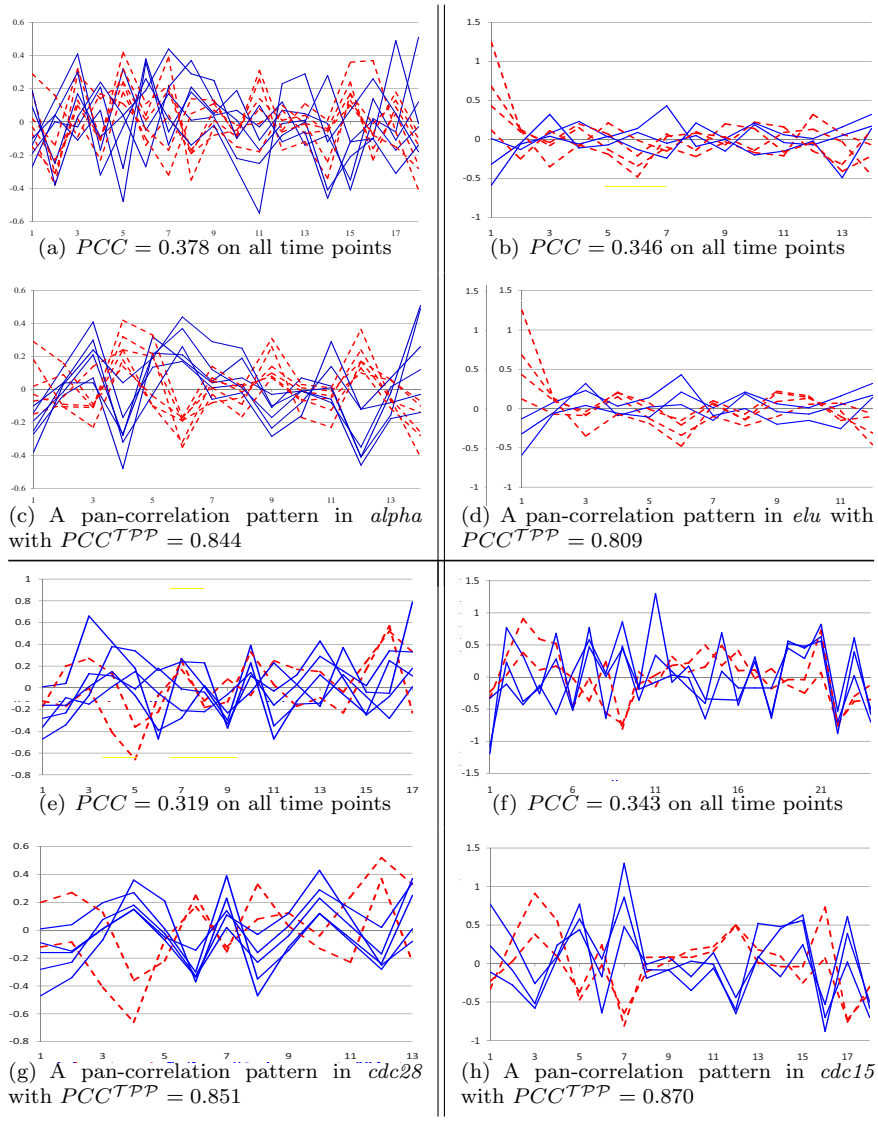
In comparison, we also calculate PCC only on $\mathcal{TPP}$, and call it $PCC^{\mathcal{TPP}}$. $PCC^{\mathcal{TPP}}$ is calculated also by Equation 1 except that $p(v_{x_i}, v_{x_j})$ is computed only on those time-point pairs involving in $\mathcal{TPP}$. When PCC or $PCC^{\mathcal{TPP}}$ is 1, it means that all the variables in $\mathcal{V}$ are correlated ideally with each other. When PCC or $PCC^{\mathcal{TPP}}$ is 0, there is completely no correlation for the variables. PCC and $PCC^{\mathcal{TPP}}$ are compared to signify particularly that time-lagged correlation patterns can have strong correlations. The results are shown in Table 1. It is observed that the variables in our $\mathbb{C}$ pan-correlation patterns are highly correlated with each other, having an average $PCC^{\mathcal{TPP}} > 0.82$ across the four datasets. However, their correlation on all time-point pairs without consideration of time lagging effect is very low with an average $PCC < 0.35$ across the four datasets, implying that significant correlation patterns are overlooked.

Table 1 PCC and $PCC^{\mathcal{TPP}}$ on four time-course gene expression data.

Dataset		min[a]	mean[a]	std[a]	max[a]
elu	PCC	0.191	0.294	0.026	0.450
	$PCC^{\mathcal{TPP}}$	0.719	0.832	0.022	0.923
cdc28	PCC	0.069	0.264	0.036	0.483
	$PCC^{\mathcal{TPP}}$	0.657	0.827	0.028	0.919
alpha	PCC	0.133	0.299	0.048	0.565
	$PCC^{\mathcal{TPP}}$	0.685	0.832	0.029	0.936
cdc15	PCC	0.122	0.347	0.083	0.799
	$PCC^{\mathcal{TPP}}$	0.620	0.826	0.034	0.933

[a]: The minimum, mean, standard deviation and maximum PCC or $PCC^{\mathcal{TPP}}$ of all pan-correlation patterns in each data set.

We show one pan-correlation pattern for each of the four microarray time-course data sets to partly illustrate the complexity of mining correlation patterns in Figure 4(c), 4(d), 4(g) and 4(h). From Figure 4(a), 4(b), 4(e) and 4(f), note that pan-correlation patterns are hardly visualized in the background of original data due to the gaps and lagged time points, but are clear in Figure 4(c), 4(d), 4(g) and 4(h), after the removal of gaps.



(a) $PCC = 0.378$ on all time points

(b) $PCC = 0.346$ on all time points

(c) A pan-correlation pattern in *alpha* with $PCC^{\mathcal{TPP}} = 0.844$

(d) A pan-correlation pattern in *elu* with $PCC^{\mathcal{TPP}} = 0.809$

(e) $PCC = 0.319$ on all time points

(f) $PCC = 0.343$ on all time points

(g) A pan-correlation pattern in *cdc28* with $PCC^{\mathcal{TPP}} = 0.851$

(h) A pan-correlation pattern in *cdc15* with $PCC^{\mathcal{TPP}} = 0.870$

Fig. 4 Four examples of pan-correlation patterns with two sets of variables: one set with solid blue line and the other with dashed red line. (a), (b), (e) and (f): The original time-course data of the involved variables in the four pan-correlation pattern examples on *alpha*, *elu*, *cdc28* and *cdc15* data set of Yeast cell cycle, respectively. (c), (d), (g) and (h): The corresponding pan-correlation pattern with smoothing after removing time-lagged points and gaps. Small errors may be in the pattern due to smoothing.

6 Conclusion

In this work, we proposed an efficient algorithm for mining all significant pan-correlation patterns from time-course data. Three novel ideas related to

time-course discretization, generalized representation of positive patterns, and using an opposite-mirror copy of the data for pattern mining were proposed. The yeast cell cycle dataset results showed that the method captures many significant patterns that are missed by existing algorithms.

7 Supplementary Material: Algorithms for mining pan-correlation patterns

Algorithm 1 Mining pan-correlation patterns by single scan of the data set

Require:
1) A time course data $M_{N_V \times N_T}$
2) Three parameters in $\mathbb{C}$ patterns:
(1) min_{TPP}: the minimum number of time-point pairs
(2) min_V: the minimum number of variables
(3) max_O: the maximum number of no change.

1: convert M into a sequential transaction data set S of R_i, $i \in [1, N_V]$
2: produce the opposite mirror sequences $R{*}_i$ for S, $i \in [1, N_V]$
3: merge the two sequential data sets to obtain a new sequential data set S' with all R_is indexed from 0 to $2N_V$-2 with step 2, and all $R{*}_i$s indexed from 1 to $2N_V$-1 with step 2.
4: call function mBIDE+(S', min_{TPP}, min_V, max_O, {});

References

1. Cho RJ, Campbell MJ, Winzeler EA, Steinmetz L, Conway A, Wodicka L, Wolfsberg TG, Gabrielian AE, Landsman D, Lockhart DJ, Davis RW (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. Molecular cell 2(1):65–73
2. Chuang CL, Jen CH, Chen CM, Shieh GS (2008) A pattern recognition approach to infer time-lagged genetic interactions. Bioinformatics 24(9):1183–1190
3. Getz G, Levine E, Domany E (2000) Coupled two-way clustering analysis of gene microarray data. Proceedings of the National Academy of Sciences 97(22):12,079–12,084
4. Ji L, Tan KL (2004) Mining gene expression data for positive and negative co-regulated gene clusters. Bioinformatics 20(16):2711–2718
5. Ji L, Tan KL (2005) Identifying time-lagged gene clusters using gene expression data. Bioinformatics 21(4):509–516
6. Jiang D, Pei J, Ramanathan M, Tang C, Zhang A (2004a) Mining coherent gene clusters from gene-sample-time microarray data. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, New York, NY, USA, KDD '04, pp 430–439
7. Koch K, Schonauer S, Jansen I, van den Bussche J, Burzykowski T (2007) Finding clusters of positive and negative coregulated genes in gene expression data. In: Bioinformatics and Bioengineering, 2007. BIBE 2007. Proceedings of the 7th IEEE International Conference on, pp 93–99
8. Li J, Liu Q, Zeng T (2010) Negative correlations in collaboration: concepts and algorithms. In: KDD, pp 463–472
9. Li X, Rao S, Jiang W, Li C, Xiao Y, Guo Z, Zhang Q, Wang L, Du L, Li J, Li L, Zhang T, Wang Q (2006) Discovery of time-delayed gene regulatory networks based on temporal gene expression profiling. BMC Bioinformatics 7(1):26

Algorithm 2 function mBIDE+(S'_s, min_{TPP}, min_V, max_O, prefix-of-pattern)

if prefix-of-pattern is closed, and has more than min_{TPP} value movements and more than min_V variables **then**
  output the closed sequential pattern as a closed pan-correlation pattern after restoring the time-point pair information.
end if
obtain frequent value movements in S'_s
for all each frequent value movement e **do**
  set $R_{x_{min1}}$ to the occurrence transaction of e with the minimum id
  set pos_1 to the first occurrence position of e in transaction $R_{x_{min1}}$
  set $R_{x_{min2}}$ to the occurrence transaction of e with the second minimum id
  set pos_2 to the first occurrence position of e in transaction $R_{x_{min2}}$
  if $R_{x_{min1}}$ is from $R*$
    OR ($R_{x_{min1}}$ is from R AND $R_{x_{min2}}$ is from $R*$
      AND $pos_1 > pos_2$) **then**
    it is redundant and thus pruned
  else
    if e is O and the number of O in prefix-of-pattern is $max_O - 1$ **then**
      pruned and continue;
    end if
    other checking in BIDE+ [16]
    set S'^{e}_s to $supp(e, S'_s)$
    call function mBIDE+(S'^{e}_s, min_{TPP}, min_V, max_O, prefix-of-pattern $\cup$ e);
  end if
end for

10. Madeira S, Oliveira A (2009) A polynomial time biclustering algorithm for finding approximate expression patterns in gene expression time series. Algorithms for Molecular Biology 4(1):8
11. Madeira SC, Teixeira MC, Sa-Correia I, Oliveira AL (2010) Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm. IEEE/ACM Transactions on Computational Biology and Bioinformatics 7(1):153–165
12. Parsons L, Haque E, Liu H (2004) clustering for high dimensional data: a review. SIGKDD Explor Newsl 6(1):90–105
13. Roy S, Bhattacharyya DK, Kalita JK (2013) CoBi: Pattern based co-regulated biclustering of gene expression data. Pattern Recognition Letters 34(14):1669–1678
14. Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, Eisen MB, Brown PO, Botstein D, Futcher B (1998) Comprehensive identification of cell cycleregulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Molecular Biology of the Cell 9(12):3273–3297
15. Van Mechelen I, Bock HH, De Boeck P (2004) Two-mode clustering methods: a structured overview. Statistical Methods in Medical Research 13(5):363–394
16. Wang J, Han J (2004) BIDE: efficient mining of frequent closed sequences. In: Data Engineering, 2004. Proceedings. 20th International Conference on, pp 79–90
17. Yin L, Wang G, Mao K, Zhao Y (2006) Mining time-delayed coherent patterns in time series gene expression data. In: Li X, Zaiane O, Li Zh (eds) Advanced Data Mining and Applications, Lecture Notes in Computer Science, vol 4093, Springer Berlin Heidelberg, pp 711–722
18. Zeng T, Li J (2010) Maximization of negative correlations in time-course gene expression data for enhancing understanding of molecular pathways. Nucleic Acids Research 38(1):e1
19. Zhao Y, Yu J, Wang G, Chen L, Wang B, Yu G (2008b) Maximal coregulated gene clustering. Knowledge and Data Engineering, IEEE Transactions on 20(1):83–98