Fall 2017

Spatiotemporal subspace feature tracking by mining discriminatory characteristics

Richard D. Appiah
Louisiana Tech University

SPATIOTEMPORAL SUBSPACE FEATURE TRACKING BY MINING DISCRIMINATORY CHARACTERISTICS

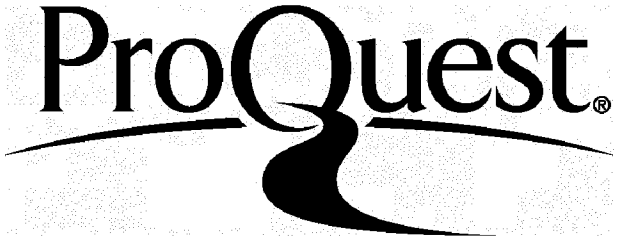
by

Richard Darko Appiah, B. Sc., M. Sc., M. S.

A Dissertation Presented in Partial Fulfillment
of the Requirements of the Degree
Doctor of Philosophy

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

November 2017



LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

AUGUST 2, 2017
Date

We hereby recommend that the dissertation prepared under our supervision by

Richard Darko Appiah, M. S.

Entitled **Spatiotemporal Subspace Feature Tracking by Mining Discriminatory Characteristics**

be accepted in partial fulfillment of the requirements for the Degree of

Doctor of Philosophy in Computational Analysis and Modeling

Supervisor of Dissertation Research

Head of Department
Computational Analysis and Modeling
Department

Recommendation concurred in:

Advisory Committee

Approved:
Director of Graduate Studies

Dean of the College

Approved:
Dean of the Graduate School

GS Form 13
(8/10)

ABSTRACT

Recent advancements in data collection technologies have made it possible to collect heterogeneous data at complex levels of abstraction, and at an alarming pace and volume. Data mining, and most recently data science seek to discover hidden patterns and insights from these data by employing a variety of knowledge discovery techniques. At the core of these techniques is the selection and use of features, variables or properties upon which the data were acquired to facilitate effective data modeling. Selecting relevant features in data modeling is critical to ensure an overall model accuracy and optimal predictive performance of future effects. The problem of relevant feature selection becomes compounded when the relevance of previously selected features cannot be guaranteed due to changes in the underlying dataset. This dissertation proposes an algorithm based on the statistical Plaid Model for the discovery of high quality biclusters from which sets of features and their corresponding relevance scores are tracked in datasets that undergo changes with time.

Initially, the algorithm employs an enhanced Plaid Model that integrates multiple results from the traditional Plaid Model to generate a list of statistically significant biclusters. This is achieved through the recursive use of combined set operations and statistical inferential tests to guide the generation of persistent set of biclusters of high quality in goodness scores. Next, the sets of features that define these biclusters are selected and marked for tracking based on their discriminatory powers exerted on the host biclusters at different time instances. As the dataset changes with time, the originally

discovered biclusters also change together with the previously established discriminatory tendencies of the respective sets of features per biclusters. These changes in discriminatory powers among the sets of features that define the host biclusters are then modeled for tracking as the underlying dataset changes with time.

The proposed technique was tested on simulated spatiotemporal phenomena in a real microarray gene expression dataset. The results indicate that the algorithm was able to generate and track subsets of features successfully through their relevance based discriminatory characteristics over a span of time instances, as the underlying dataset underwent changes.

APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Dissertation. It is understood that "proper request" consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Dissertation. Further, any portions of the Dissertation used in books, papers, and other works must be appropriately referenced to this Dissertation.

Finally, the author of this Dissertation reserves the right to publish freely, in the literature, at any time, any or all portions of this Dissertation.

Author RICHARD DARKO APPIAH

Date AUGUST 2, 2017

DEDICATION

To the curious minds and long suffering personalities of my family and friends, I hereby dedicate this work for their collective patience, hopes and prayers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

Thanks to the Lord for bringing me this far in my pursuit of advanced knowledge. My sincere gratitude goes to Dr. Sumeet Dua for his invaluable academic guidance and financial support towards my doctoral research. I gratefully acknowledge the support from his research grants that made the completion of this dissertation a possibility. I also acknowledge the support by the University of Ghana faculty Ph. D. research grant that supplemented this dissertation in part. I am especially indebted to Dr. Pradeep Chowriappa for his selfless advice and shared research insights to make sure this work sees the light of day. To the doctoral research committee members, Dr. Katie Evans, Dr. Jinko Kanno and Dr. Jean Gourd, I express my heartfelt gratitude for their time and dedication in reviewing and critiquing my work to ensure the deserved quality. A special thank you goes to Dr. Weizhong Dai, the coordinator of the computational analysis and modeling doctoral program at Louisiana Tech University, my friends and colleagues who in diverse ways contributed their time, knowledge and resources to assist me in this academic journey.

Finally to my family, Emelia my mother, Joyce my wife and our children Jedidiah, Darda, Ethan, and Calcol, I say the Lord bless you and keep you for your patience, love, dedication, support, and prayers throughout this rather intriguing journey with me. Thank you.

CHAPTER 1

INTRODUCTION

Years of advancements in the use of data-driven information retrieval systems have necessitated the need for data analytics experts to acquire advanced knowledge in datasets, data modeling methodologies and the overall underlying market-oriented business objectives. These advancements in the age of big data have spawned relatively new disciplines such as machine learning, data mining and, quite recently, data science. At the core of this data-driven information acquisition revolution is data with such characteristic attributes as sparseness, evolving size and dimensionality. These attributes have motivated intensive research and algorithm development to handle different complex problems that arise in domains that rely on effective ways of turning these available data into useful knowledge.

In order to unravel useful but mostly hidden insights from the massive amount of data collected by organizations and devices around the globe, researchers in the past three decades have proposed and implemented a plethora of algorithms to aid in making sense of the ever increasing amount of data. Major goals for most of these algorithms range from classification and clustering to complex predictive models that require huge amounts of data from several different sources and formats. Generally, these algorithms are formulated based on various characteristic features or attributes that are collectively used to obtain data on the phenomena under investigation. These attributes are usually

subjected to relevance analysis to establish their weighted inclusion in any potential algorithmic models for knowledge discovery in the datasets under investigation.

Spatiotemporal datasets are a class of datasets that have both spatial and temporal dimensions. Temporal dimension allows for features that define the associated spatial dataset to be investigated and modeled over time to learn their differential effects as the dataset changes in size and spatial orientation. Domains that generate and analyze datasets with temporal dimension include biomedical data analytics, geographical information systems, urban and traffic planning systems, communication systems, multimedia systems, behavioral pattern analytics, wireless sensors and video data analytics, and collaborative filtering for marketing [1, 2]. This dissertation aims at designing and implementing algorithms to model for tracking the discriminatory effects of data features or attributes as the related dataset undergoes spatiotemporal modifications.

1.1 Data Mining

Data mining is considered to be an interdisciplinary subject, and hence, several different working definitions exist in the literature. From a working definition standpoint, data mining is defined as the analysis of observational datasets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner [3]. Functionally, data mining is also defined as the process of discovering interesting patterns and knowledge from large amounts of data [4, 5]. In practice, data mining is considered to be an essential phase in a broader context of knowledge discovery from data (KDD), a term that originated from artificial intelligence (AI) research [3, 4, 6]. **Figure 1-1**shows the various stages involved in the KDD process

for knowledge discovery. They include data selection, data preprocessing, data transformation, data mining, patterns evaluation, and knowledge discovery.

1. **Data Selection**: This involves the retrieval of records usually from existing data warehouse or data center, to form a target dataset to be considered for further processing in the knowledge discovery cycle. This might involve selecting subsets of data attributes or features and record samples that are deemed relevant for efficient knowledge discovery.
2. **Data Preprocessing**: This is the process of data cleaning to remove noisy data containing errors or outliers and inconsistent records. It might also include data integration where multiple data sources are combined to form a single improved dataset that enhances efficient data mining [4, 7].
3. **Data Transformation**: This step involves transforming and consolidating data into forms appropriate for specific data mining tasks. Activities here include data normalization, data discretization, feature construction and data smoothing.
4. **Data Mining**: This is where intelligent data modeling techniques are applied to extract hidden data patterns from the target dataset.
5. **Evaluation**: Extracted patterns are analyzed at this phase to identify truly interesting patterns to represent the knowledge discovered from the underlying dataset. This eventually leads to knowledge presentation where visualization and knowledge representation techniques are used to present mined knowledge to users of the system [4].

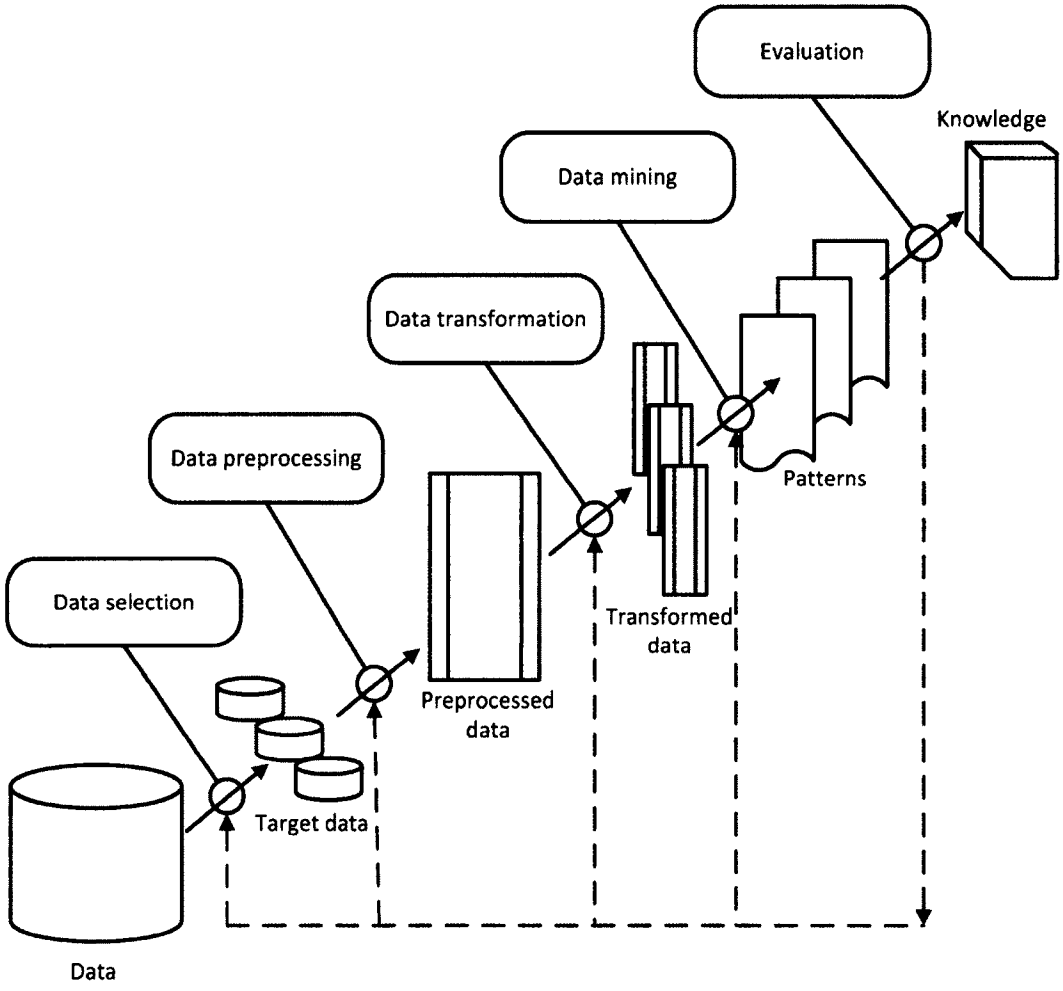


Figure 1-1: The KDD process.

1.2 Feature Selection

Datasets originating from areas such as internet text documents processing, gene expression array analysis, and combinatorial chemistry are characterized by high-dimensional attributes, variables or features that run into hundreds of thousands. This poses a challenge to most machine learning algorithms in terms of model accuracy and efficiency.

Due to advanced information and location-aware technologies, data are generated and stored at an incredible pace, volume and variety from diverse sources like social media, weather records, gene expression datasets, and various forms of customer management datasets. Other spatiotemporal data sources include global positioning system (GPS) tracking data of vehicles and animals, credit card transaction history, and records of residential address changes of individuals [8].

Data mining and data science seek to discover hidden patterns and insights from these available data by employing a variety of knowledge discovery techniques. At the core of these knowledge discovery techniques is the use of features, variables or properties upon which the data were collected. The difficulty in selecting the right set of features for pattern recognition mostly depends on the specific problem formulation and the underlying dataset [9], and most feature selection methods base their decision on the degree of feature relevance [10]. However, feature relevance which describes the discriminatory power of a given feature tends to fluctuate in datasets that undergo structural changes with time by either dropping from or adding to existing records. Hence, effective tracking of feature relevance in datasets that change with time is paramount for accurate and reliable knowledge discovery undertakings that rely on them.

Feature selection has become an active area of research with the objectives of improving the prediction performance of model predictors, providing faster and more cost-effective predictors, and providing a better understanding of the underlying process that generated the datasets [10]. Some potential benefits of feature selection include facilitating data visualization and data understanding, reduction in measurements, storage

requirements, training and utilization times, and controlling the curse of dimensionality to improve prediction performance.

Features can either be selected as individuals based on their ranking scores, or as a subset of candidate features based on their ability to achieve optimal performance together. The works of Guyon and Elisseeff [10] and Kohavi and John [11] highlight the different criteria for feature selection, as summarized in the ensuing subsections. The following notations are used: Let $\{X_k, Y_k\}$ with $k = 1, \ldots, m$ be a set of m examples consisting of n input features $x_{k,i}$ with $i = 1, \ldots, n$ and one output variable y_k.

1.2.1 Correlation Based Feature Selection

Under this scheme, a feature X_i is selected if its Pearson correlation coefficient, $R(i)$ with the output variable Y, given by **Eq. 1-1** and estimated by **Eq. 1-2**, is the highest, where *var* and *cov* are the respective variance among the $x_{k,i}$ and the covariance between X_i and Y; $\bar{x_i}$ and $\bar{y}$ are the input and output averages over the index k, respectively:

$$R(i) = \frac{cov(X_i, Y)}{\sqrt{var(X_i)var(Y)}}. \qquad \textbf{Eq. 1-1}$$

$$R(i) = \frac{\sum_{k=1}^{m}(x_{k,i} - \bar{x_i})(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{m}(x_{k,i} - \bar{x_i})^2 \sum_{k=1}^{m}(y_k - \bar{y})^2}}. \qquad \textbf{Eq. 1-2}$$

1.2.2 Single Feature Classifiers

This involves the ranking and selection of features for the construction of regression models according to the goodness of linear fit of individual features.

Individual features are substituted in the regression model and the feature with the highest coefficient of determination, designated by $R(i)^2$ based on either **Eq. 1-1**or **Eq. 1-2** is selected.

1.2.3 Information Theoretic Ranking Criteria

This approach uses the empirical estimates of the mutual information between each feature and the target variable. This criterion estimates the dependency $I(i)$ between the density of feature x_i and the density of the target y. $I(i)$ is computed by **Eq. 1-3**, where $p(x_i)$ and $p(y)$ are the probability densities of x_i and y, respectively, and $p(x_i, y)$ is their joint density:

$$I(i) = \int\int p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)} dx\, dy. \qquad \textbf{Eq. 1-3}$$

1.2.4 Feature Subset Selection

In practice, there are situations where features exhibit weaker discriminatory abilities individually, but put together by some established criteria, they tend to demonstrate excellent predictive power, as opposed to ranking them by their individual predictive powers [10, 11]. Many machine learning algorithms are also faced with performance degradation in terms of prediction accuracy when challenged with many features some of which are not necessary for predicting the desired output. Hence, there is the need to define an optimal feature subset with respect to the underlying induction algorithm, taking into account its heuristics, biases, and tradeoffs. Three methods exist in the literature for selecting a group or subset of features for the purpose of improving the predictive power of a particular algorithm [12, 13]. They are wrappers, filters and embedded methods.

1. **Wrappers for Feature Subset Selection**: In its general formulation, wrapper methods employ the prediction performance of a given learning machine to assess the relative usefulness of subsets of the features [11, 14]. For effective implementation of wrappers, one needs to define: (i) how to exhaustively search the space of all possible feature subsets; (ii) how to quantify the prediction performance of the learning machine to guide the search and halt it; and (iii) the most appropriate predictor to use [10]. Although the complexity of the wrapper problem formulation is known to be NP-hard where exhaustive search becomes quickly intractable computationally [15], a range of heuristic search strategies are used in practice, including best-first, branch-and-bound, simulated annealing and genetic algorithms [10]. Algorithms that implement wrappers include decision trees, naïve Bayes, least-square linear predictors, and support vector machines. Wrappers generally employ either forward feature selection or backward feature elimination. In forward selection mode, features are progressively incorporated into larger subsets, and in backward elimination mode, the algorithm starts with the set of all available features and progressively eliminates the least relevant ones to come up with the optimal feature set.
2. **Filters for Feature Subset Selection**: Filter methods are mostly used for preprocessing to reduce space dimensionality and overcome overfitting. They serve as linear predictors whose outcome form a set of selected features to train more complex, usually non-linear predictors. Filters are considered to be

faster, and those based on mutual information criteria provide a generic selection of features that are not tuned for specific learning machines.

3. **Embedded Methods for Feature Selection**: These methods are usually learning machine specific, and perform feature selection in the process of algorithm training. Embedded methods tend to implement the feature selection process as an integral part of a classifier at the training phase where the selection is done based on the performance of the classifier [12]. They make better use of features data available by not needing to split the training data into training and validation sets. As a result, solutions are reached faster by avoiding retraining a predictor from scratch for every feature subset investigated.

1.3 Clustering for Feature Selection

Clustering is the process of partitioning a set of data objects or observations into subsets known as clusters. The objects in a cluster exhibit high intra-cluster similarity and data objects between different clusters exhibit low inter-cluster similarity [4, 16, 17, 18]. In the literature, clustering technique is mostly used for feature construction from existing features where a group of features that define a given cluster is replaced by what is known as the cluster centroid. The centroid can be defined in several different ways, such as the mean or median of the feature scores located within a given cluster. The overall cluster quality Q is measured by the within-cluster variation, which is the sum of squared errors between all features within a cluster and the centroid of that cluster. Let $C_1, C_2 \cdots, C_k$ be a set of k clusters such that $C_i \cap C_j = \emptyset$ for $1 \leq i, j \leq k$, $f \in C_i$ be a feature in cluster C_i, and d_i be the centroid of the cluster C_i, then the Euclidean distance

$dist(f, d_i)$ defines the distance between any feature $f \in C_i$ and the centroid d_i; the overall cluster quality Q is computed by **Eq. 1-4**:

$$Q = \sum_{i=1}^{k} \sum_{f \in C_i} \left(dist(f, d_i)\right)^2 . \qquad \textbf{Eq. 1-4}$$

1.4 Dissertation Organization

The remainder of the dissertation is divided into five chapters. Chapter 2 outlines existing works related to the problem domains of feature selection and object tracking. Chapter 3 details the basics of spatiotemporal feature selection and tracking based on feature discriminatory characteristics. The chapter presents notations, formal definitions and the main algorithm for tracking sets of spatiotemporal features based on relevance in a changing dataset. Chapter 4 presents the use of a proposed enhanced biclustering algorithm in this dissertation for the discovery of high quality biclusters used for feature selection, and its application on real gene expression dataset. Chapter 5 illustrates how sets of selected features are tracked over time in a changing dataset to demonstrate differing feature discriminatory powers as the underlying dataset changes. Chapter 6 concludes the dissertation with a brief on future research directions.

CHAPTER 2

RELATED WORKS

Recent advancements in data collection technologies have made it possible to collect data at complex levels of abstraction to facilitate the presentation, analysis and tracking of objects and events in spatiotemporal domains. Of immense research interest is the tracking of features or variables in spatiotemporal domains, which typifies the problem of path-finding of objects across dimensions in space and time. Different approaches exist in the literature, ranging from general spatiotemporal object tracking to the detection and tracking of rare events in space and time. The rest of the chapter outlines existing works on object tracking and techniques for selecting relevant features that aid in the tracking process.

2.1 Object Tracking

Most existing algorithms for spatiotemporal object tracking work on two major assumptions, which are 1) unchanged spatial configuration over time, and 2) object's identity remains unchanged as its location and content change. Under these assumptions, the work of Yilmaz *et al.* [19] categorizes object tracking into three groups, namely, point tracking, kernel tracking and silhouette tracking.

1. **Point Tracking**: This uses deterministic and statistical models where points are utilized to represent objects to be detected for tracking in consecutive frames. The

association of the points is based on the object's previous state that might include information on its position and motion. Algorithms in this group include the modifying greedy exchange (MGE) algorithm [20], greedy optimal assignment (GOA) tracker algorithm [21], and iterated Kalman filters for nonlinear object tracking [22].

2. **Kernel Tracking:** Kernel refers to the appearance and shape of an object, and this approach is based on template and density appearance models. By this approach, objects are tracked by computing the motion of the kernel in consecutive frames. Sample algorithms in this group include the mean-shift algorithm [23], and the layering algorithm [24].

Silhouette Tracking: Silhouette tracking involves the estimation of the object region in each frame of the image being tracked. This tracking method performs either shape matching or contour evolution. This is achieved through the use of information encoded within the object region in the form of appearance density and shape models. Sample algorithms include the state space models [25], variation methods [26], and heuristic methods [27].

The work of Wang *et al.* [28] demonstrates the tracking of words as features in multiple connected documents by employing a nonparametric Bayesian model for topics modeling. Here, observations of words are treated as tracking objects on trajectories of documents. The algorithm accomplishes its goal by partial use of available information generated with Gibbs sampling of the established semantic regions. In order to track groups of features, vectors of features that define a set of images being tracked are initially transformed to a higher dimensional space. These are then categorized into usual

and unusual events by the abnormality detection algorithm based on nearest neighbor discovery, proposed by Breitenstein *et al.* [29]. A relatively robust subspace feature tracking algorithm has been implemented in [30] which is based on a robust *l-norm* objective function. The objective function estimates and tracks non-stationary subspaces involving streaming data vectors corrupted with outliers. This is done on the condition that the subset of features to be tracked must always be orthonormal.

Density based rare events detection approaches also exist for tracking features in spatiotemporal datasets where the features might undergo rare or subtle changes with time. Binary space-time descriptors of data streams to map the data vectors of an object to a higher-dimensional feature density estimates to cluster events into frequently observed and rare is implemented in [31]. Lima de Carvalho *et al.* [32] proposed an online tracking of multiple objects using a model called the Wilkie, Stonham and Aleksander's Recognition Device (WiSARD). WiSARD works with binary datasets and uses the weightless neural networks model whose neurons store either '0' or '1' in the random access memory (RAM) to indicate the presence or otherwise an input pattern during classification for tracking.

There are algorithms designed to handle specific problems encountered in object tracking to enhance the overall tracking accuracy. Oversampling technique in signal processing is employed in the work of Pernici and Del Bimbo [33] to build a robust discriminative objects classifier. Object tracking is based on nonparametric algorithm and transitive matching property to handle tracking updates on objects under occlusion, where the physical appearance of the tracked object might undergo changes. This method relies on the oversampling of local features and potentially suffers from local minima and

maxima problems. The work of Zhu *et al.* [34] achieves consistent multi-scale object representation through the use of correlation filters for tracking. This is achieved through a kernel of multi-scale correlation filter and failure detection based on adaptive learning.

By this, model accuracy and efficiency is maintained in situations of scale variation and model drifting where existing models cannot accurately track the target objects due to changes in their spatial and structural configurations. As objects move across different spatial dimensions and configurations, the existing associations and correlations among its inherent features undergo changes, too. This is particularly problematic in visual feature tracking and constitutes a problem known as tracking drift. Tracking drift is experienced in a model when there is inconsistency in the target object representation in different scenarios and at different times, thereby introducing significant accumulated errors into the model. This problem has been solved by the sparsity-induced subspace learning technique proposed by Sui *et al.* [35].

This algorithm utilizes useful temporally acquired mutual relations among the observed features for effective subspace representation in visual tracking. However, the over-reliance of this approach on previously established mutual information whose relevance tend to fluctuate with time makes this technique less attractive. The model drift problem has also been addressed by the algorithm proposed by Liu *et al.* [36] where the authors introduced a technique that uses multiple weaker classifiers that are selected based on their performances over individual instance significance estimates learned over time. Although innovative, the use of instance-specific weaker algorithms to handle different aspects of the same visual tracking problem exposes this method to potential local optimization problems.

One area of active research in object tracking is real-time object tracking. Contrary to the tracking of objects whose shape and appearance remain unchanged, real-time object tracking require different representation schemes and models for effective tracking. Recently, researchers have proposed the use of different algorithms to handle different aspects of the same problem of real-time object tracking. The work of Moujtahid *et al.* [37] employs independent heterogeneous algorithms trained on different sets of features corresponding to different aspects of an object to track the object. By this approach, the model switches algorithms based on the object's spatiotemporal appearance, and eventually integrates the results from all the participating algorithms.

2.2 Feature Relevance

Different definitions of feature relevance exist in the literature, and they are based on a set of assumptions that are designed to reflect the nature and characteristics of the target datasets [11]. Almuallim and Dietterich [38] assume an all-Boolean feature set with no noise, and propose that for a feature X_i in the feature vector space $X = \{X_1, X_2, \cdots, X_n\}$ to be relevant to a concept C, X_i must appear in every Boolean formula that represents C, and irrelevant otherwise. Gennari *et al.* [39] assume datasets with multi-valued features in the presence of noise, and define features to be relevant if their values systematically vary with categorical membership. Formally, a feature X_i is relevant iff there exists some elements x_i and y for which $P(X_i = x_i) > 0$ such that **Eq. 2-1** holds:

$$P(Y = y \mid X_i = x_i) \neq P(Y = y). \qquad \textbf{Eq. 2-1}$$

Under the definition given by **Eq. 2-1**, X_i is relevant if knowing its current value changes the estimate for the class label Y to indicate the conditional dependency of Y on X_i. To

account for the relevance of all the features in the parity concept where every datapoint in the given dataset is equally probable, Kohavi and John [11] modify the definition given by **Eq. 2-1** as follows: Let $F_i = \{X_1, \cdots, X_{i-1}, X_{i+1}, \cdots, X_n\}$ be the set of all features except X_i, and f_i denote possible values of all features in F_i, then X_i is relevant iff there exists some x_i, y and f_i for which $P(X_i = x_i) > 0$ such that **Eq. 2-2** hold:

$$P(Y = y, F_i = f_i \mid X_i = x_i) \neq P(Y = y, F_i = f_i). \qquad \textbf{Eq. 2-2}$$

A modified version of **Eq. 2-2** is the case defined by **Eq. 2-3** where X_i is relevant if the probability of some class labeled Y, given all other features except X_i, can change when we eliminate knowledge about the value of X_i. That is, X_i is relevant iff there exists some x_i, y and f_i for which $P(X_i = x_i, F_i = f_i) > 0$ such that **Eq. 2-3** holds:

$$P(Y = y \mid X_i = x_i, F_i = f_i) \neq P(Y = y \mid F_i = f_i). \qquad \textbf{Eq. 2-3}$$

2.2.1 Degree of Feature Relevance

Feature relevance estimates are categorized into degrees of relevance to reflect whether the removal or otherwise of any given features results in a measurable change in the underlying model's prediction accuracy. In order to define optimal probabilistic classifiers, feature relevance scores are divided into two degrees, namely, *weak relevance* and *strong relevance* [11, 18]. A feature X is weakly relevant if there exists a subset of features, F such that the performance of a given model on F is worse than the performance on $F \cup \{X\}$. A feature X is strongly relevant if its exclusive removal from a given feature set results in a noticeable performance degradation of an optimal classifier. Any feature that is neither strongly nor weakly relevant is classified to be irrelevant.

2.2.2 Formal Definition: Degree of Feature Relevance

A given feature X_i is said to be weakly relevant iff it does not qualify to be strongly relevant, and for a given set of features F_i, there exists a subset of F_i, F_i' for which there exists some elements x_i, y and f_i' with $P(X_i = x_i,\ F_i' = f_i') > 0$ such that **Eq. 2-4** holds:

$$P(Y = y \mid X_i = x_i, F_i' = f_i') \neq P(Y = y \mid F_i' = f_i') \cdot \qquad \textbf{Eq. 2-4}$$

A feature X_i belonging to the feature set F_i is said to be strongly relevant iff there exists some elements x_i, y and f_i with $P(X_i = x_i, F_i = f_i) > 0$ such that **Eq. 2-5** holds:

$$P(Y = y \mid X_i = x_i, F_i = f_i) \neq P(Y = y \mid F_i = f_i). \qquad \textbf{Eq. 2-5}$$

2.3 Biclustering for Feature Selection

In cluster analysis of datasets to identify functionally related patterns, the entire feature set is considered in deciding datapoints membership of a cluster, and datapoints are only allowed to belong to a single cluster [40]. However, datapoints may not necessarily portray the desired pattern within all the attributes under consideration but might be evident only under a subset of attributes. Similarly, a given datapoint might express significant phenomena under different subsets of attributes [40, 41]. Given a data matrix $\mathbb{A}$ with a set of rows X and a set of columns Y such that the element a_{ij} represents the relation between row i and column j, a bicluster, on the contrary, is defined as a subset of X that exhibits similar behavior across a subset of Y, and vice versa [42, 43, 44]. As with cluster analysis, bicluster quality assessment is based on the sum of squared errors over all features and datapoints within the bicluster. This work proposes and explores effective biclustering techniques from which the set of features that define a

bicluster could be selected and marked for tracking over time, as the underlying dataset changes.

Several biclustering algorithms exist, and comprehensive surveys of the most widely used biclustering techniques [40, 42, 45, 46, 47] highlight the Plaid Model (PM) [48, 49, 45], statistical algorithmic method for bicluster analysis (SAMBA) [50], Cheng and Church (CC) [45, 51], flexible overlapped biclustering (FLOC) [52], order-preserving submatrices (OPSM) [53, 54], iterative signature algorithm (ISA) [55], and Spectral [56]. Others include BiMax [57], xMOTIFs [58], Bayesian biclustering (BBC) [59], combinatorial algorithm for expression and sequence-based cluster extraction (COALESCE) [60], correlated pattern biclusters (CPB) [61], qualitative biclustering (QUBIC) [62], and factor analysis for bicluster acquisition (FABIA) [45, 63].

The PM generates randomly observable data values to fit the given dataset such that the underlying parameters of the target model are iteratively estimated to minimize the mean squared error (MSE) between the true and the fitted datasets [45, 48, 49]. SAMBA uses graph formalism to identify statistically significant biclusters by discovering the equivalent maximum weighted subgraphs. CC minimizes a fitness function in a greedy approach based on the mean squared residue (MSR) associated with the discovery of biclusters with least variances [45, 51]. FLOC uses the bicluster definition by CC to unravel a set of overlapping biclusters [52], based on probabilistic assignments followed by an iterative process to improve the biclusters. This is achieved by the addition or removal of one row or column at a time to determine the action that best improves the average MSR. OPSM discovers biclusters by a greedy approach that ensures the generation of order-preserving submatrices with column-wise linear order to

realize row values of the identified submatrices either increasing or decreasing linearly [45, 53, 54]. ISA finds a bicluster using a seed bicluster made up of randomly selected rows and columns that is continuously updated until convergence in a greedy fashion [45, 55].

Both Spectral and BiMax discover checkerboard patterned biclusters. Spectral employs a technique based on singular value decomposition to find biclusters with least variance relative to a stated threshold [45, 56]. It achieves this by reformulating the biclustering problem as a sequential search for individual bicluster sequences based on the message passing optimization algorithm of the generalized distributive law family, known as the max-sum algorithm. BiMax recursively employs divide and conquer algorithm on a binary dataset to find upregulated biclusters. xMOTIFs uses greedy approach to find a bicluster in a discretized dataset with same values on the rows in a nondeterministic fashion [45, 54, 58]. BBC generates a form of the PM with Bayesian properties based on Gibbs sampling techniques and with the restriction that no two biclusters share the same data elements by ensuring only row-wise or column-wise overlaps, and not both [45, 59].

COALESCE initiates the bicluster discovery process with a row pair that are correlated, followed by a series of iterative updates on both rows and columns until convergence [45, 60]. CPB [45, 61] uses a greedy approach and relies on high row-wise Pearson correlation coefficient to discover biclusters in such a way that rows are systematically added to an initially randomly selected row to achieve higher correlations above a set threshold. It also ensures that biclusters have rows and columns with least MSE.

QUBIC uses a deterministic approach to reformulate the biclustering problem to that of the subgraph discovery task in bipartite graphs, which results in the discovery of biclusters exhibiting column-wise constant values in a discretized dataset [45, 62]. FABIA discovers biclusters by fitting a model to the data where the set of rows and columns per bicluster are treated as sparse vector sets and the concerned bicluster being the outer product of these vector sets, with an additional factor to account for any potential noise [45, 63].

The work of Denitto and Bicego [64] reformulates the biclustering problem as a sequential search for individual bicluster sequences based on the message passing optimization algorithm of the distributive law family, known as the max-sum algorithm.

2.4 Conclusion

This chapter highlights existing research in the area of object tracking that are related to this dissertation. Following a brief introduction, the chapter discusses the three major categories of object tracking in the literature: point tracking, kernel tracking and silhouette tracking. This was followed by detailed summaries and weaknesses, where applicable, of existing models and algorithmic implementations for tracking objects under varieties of tracking conditions. Next, the chapter presents existing works on biclustering, and discusses formal ways of estimating feature relevance in datasets.

CHAPTER 3

SPATIOTEMPORAL FEATURE TRACKING

Research in subspace discovery and biclustering predominantly involves the development and use of algorithmic techniques to identify biclusters based on their association with subspaces in high dimensional data structures. Spatiotemporal subspace biclustering employs specialized biclustering techniques that incorporate an additional dimension of time to the biclusters. This makes it possible for time-dependent biclustering criteria to be established adaptively on a temporal basis. A set of features that defines a given bicluster, and their collective set of relevance scores constitute the biclustering criteria of the bicluster at any time. In this chapter, we outline the formal notations, definitions, and the main problem formulation relating to the analysis and discovery of reliable biclustering criteria for the purpose of feature relevance tracking.

The rest of the chapter is organized as follows. Section 3.1 presents formal notations used in formulating the subspace feature tracking problem and the subsequent algorithmic presentations. Section 3.2 discusses and presents some formal definitions pertaining to spatiotemporal subspace feature tracking, and Section 3.3 outlines the symbolic presentation of the main problem formulation.

3.1 Notations

Let $\mathbb{A}_{IJ} = (XY)$ be a real-valued data matrix that represents a dataset with a set of rows $X = \{x_1, x_2, \cdots, x_R\}$ and a set of columns $Y = \{y_1, y_2, \cdots, y_S\}$. Let a_{ij} be an element of $\mathbb{A}_{IJ}$ that corresponds to the relation between row i and column j. For the data matrix $\mathbb{A}_{IJ}$, let $\mathbb{A}_{I.}$ and $\overline{\mathbb{A}}_{I.}$ represent the sum and average of values on the I^{th} row, respectively; $\mathbb{A}_{.J}$ and $\overline{\mathbb{A}}_{.J}$ represent the sum and average of values in the J^{th} column, respectively; $\mathbb{A}_{..}$ and $\overline{\mathbb{A}}_{..}$ represent the overall sum and average of values in $\mathbb{A}_{IJ}$, respectively.

Let $\{B_{k,t}\} = \{B_{1,t}, B_{2,t}, \cdots, B_{K,t}\}$ denotes a set of K biclusters generated at time t for $t \in \mathbb{N}$ and $1 \leq t \leq T$, where T is the most recent time at which the bicluster set $\{B_{k,T}\}$ was generated, $\mathbb{N}$ is a set of natural numbers, and $1 \leq k \leq K$. For the bicluster $B_{k,t}$, $Row(B_{k,t})$ and $Col(B_{k,t})$ represent the row and column elements of $B_{k,t}$, respectively. Without reference to time, let $\{B_k\} = \{B_1, B_2, \cdots, B_K\}$ denotes a set of K biclusters.

3.2 Formal Definitions

The notations in Section 3.1 are used in this section to provide formal definitions that are utilized in the tracking of spatiotemporal subspace of features and feature relevance in this work.

Definition 3.1 (Bicluster): Given a real-valued $R \times S$ data matrix $\mathbb{A}_{IJ} = (X, Y)$, with a set of rows $X = \{x_1, x_2, \cdots, x_R\}$ and a set of columns $Y = \{y_1, y_2, \cdots, y_S\}$, a bicluster $\mathbb{A}_{ij}$ is a submatrix of $\mathbb{A}_{IJ}$ defined as the ordered pair given by **Eq. 3-1** such that $i = \{i_1, i_2, \cdots, i_M\}$ with $i \subset X$, $M \leq R$ and $j = \{j_1, j_2, \cdots, j_P\}$ with $j \subset Y$ and $P \leq S$.

$$\mathbb{A}_{ij} = (i, j). \qquad \textbf{Eq. 3-1}$$

Definition 3.2 (Subspace): Let $\mathcal{S} = \{S_1, S_2 \cdots, S_K\}$ denote a set of K subspaces. Then, given $\mathcal{C}$ as a conceptualized space that commensurate a $R \times S$ data matrix $\mathbb{A} = (X, Y)$, $\mathcal{C}$ can be partitioned into $\mathcal{S} = \{S_1, S_2 \cdots, S_K\}$ subspaces each of which is defined as the ordered pair given by **Eq. 3-2** where, respectively, $I_i = \{I_1, I_2, \cdots, I_M\}$, $1 \leq i \leq M$ with $M \leq R$ and $F_j = \{F_1, F_2, \cdots, F_P\}$, $1 \leq j \leq P$ with $P \leq S$ are the *instance* and *feature vectors* corresponding to the rows and columns of S_k:

$$S_k = (I_i, F_j). \qquad \textbf{Eq. 3-2}$$

Definition 3.3 (Conserved Biclusters): A set of K biclusters $\{B_k\} = \{B_1, B_2, \cdots, B_K\}$ with $1 \leq k \leq K$ is said to be conserved if for any given data matrix $\mathbb{A}_{IJ}$ that contains $\{B_k\}$, there exists no further biclusters that can be discovered without altering the elements of $\mathbb{A}_{IJ}$.

Definition 3.4 (Bicluster Mean): For a set of K biclusters $\{B_k\} = \{B_1, B_2, \cdots, B_K\}$, the bicluster mean, μ_k associated with each B_k is defined as the bicluster-specific effect that is exerted on the data matrix $\mathbb{A}_{ij}$, and is given by **Eq. 3-3**, **Eq. 3-4** or **Eq. 3-5**, where $M$and P are the respective number of rows and columns of $\mathbb{A}_{ij}$:

$$\mu_k = \bar{\mathbb{A}}_{...} \qquad \textbf{Eq. 3-3}$$

$$\mu_k = \frac{\mathbb{A}_{..}}{M * P}. \qquad \textbf{Eq. 3-4}$$

$$\mu_k = \frac{\sum_{i=1}^{M} \sum_{j=1}^{P} \mathbb{A}_{ij}}{M * P}. \qquad \textbf{Eq. 3-5}$$

Definition 3.5 (Row or Instance Effect): For a data matrix $\mathbb{A}_{ij}$ that constitutes the bicluster B_k, an instance effect α_i is defined as the row-specific effect exerted on B_k by

the i^{th} row in B_k, and is given by **Eq. 3-6** where $\bar{\mathbb{A}}_{i.}$ is the mean value of the i^{th} row in B_k with bicluster mean μ_k:

$$\alpha_i = \bar{\mathbb{A}}_{i.} - \mu_k. \qquad \textbf{Eq. 3-6}$$

Definition 3.6 (Column or Feature Effect): The feature effect β_j, given by **Eq. 3-7**, is the column-specific effect exerted on bicluster B_k of data matrix $\mathbb{A}_{ij}$ by the j^{th} column of B_k with bicluster mean μ_k, and column mean $\bar{\mathbb{A}}_{.j}$:

$$\beta_j = \bar{\mathbb{A}}_{.j} - \mu_k. \qquad \textbf{Eq. 3-7}$$

Definition 3.7 (Instance Relevance Score): For a set of instance effects $\{\alpha_i\} = \{\alpha_1, \alpha_2, \cdots, \alpha_M\}$ pertaining to the bicluster B_k, the instance relevance score P_{α_i} for each α_i, given by **Eq. 3-8**, measures the discriminatory power exerted by the instance i on B_k.

$$P_{\alpha_i} = \frac{\alpha_i}{\sum_{i=1}^{M} |\alpha_i|}. \qquad \textbf{Eq. 3-8}$$

Definition 3.8 (Feature Relevance Score): For a set of feature effects $\{\beta_j\} = \{\beta_1, \beta_2, \cdots, \beta_P\}$ associated with the bicluster B_k, the feature relevance score P_{β_j} for each β_j, given by **Eq. 3-9**, measures the discriminatory power exerted by the feature j in B_k:

$$P_{\beta_j} = \frac{\beta_j}{\sum_{j=1}^{P} |\beta_j|}. \qquad \textbf{Eq. 3-9}$$

Definition 3.9 (Affinity Matrix): This is the transpose of a vector whose elements are the relevance scores with respect to either the instances or features of a given bicluster B_k. The instance affinity matrix $AM_i(B_k)$ and the feature affinity matrix $AM_j(B_k)$ that correspond to the bicluster B_k are given by **Eq. 3-10** and **Eq. 3-11**, respectively:

$$AM_i(B_k) = \left(P_{\alpha_1}, P_{\alpha_2}, \cdots, P_{\alpha_M}\right)^T. \quad \textbf{Eq. 3-10}$$

$$AM_j(B_k) = \left(P_{\beta_1}, P_{\beta_2}, \cdots, P_{\beta_P}\right)^T. \quad \textbf{Eq. 3-11}$$

Definition 3.10 (Bicluster Purity and Efficiency): Let $B_{k,i}$ be a target bicluster and $B_{l,i+1}$ a retrieved bicluster to be compared with $B_{k,i}$. The purity of $B_{l,i+1}$, $Purity(B_{l,i+1})$ measures its compositional closeness to $B_{k,i}$, and it is given by **Eq. 3-12** or **Eq. 3-13**. The efficiency of $B_{l,i+1}$, $Efficiency(B_{l,i+1})$ measures how comprehensive the biclustering criteria of $B_{l,i+1}$ truly replicates the target bicluster $B_{k,i}$, and it is given by **Eq. 3-14** or **Eq. 3-15**:

$$Purity(B_{l,i+1}) = \frac{|B_{k,i} \cap B_{l,i+1}|}{|B_{k,i}|}. \quad \textbf{Eq. 3-12}$$

$$Purity(B_{l,i+1}) = \frac{|Row(B_{k,i}) \cap Row(B_{l,i+1})| \times |Col(B_{k,i}) \cap Col(B_{l,i+1})|}{|Row(B_{k,i})| \times |Col(B_{k,i})|}. \quad \textbf{Eq. 3-13}$$

$$Efficiency(B_{l,i+1}) = \frac{|B_{k,i} \cap B_{l,i+1}|}{|B_{l,i+1}|}. \quad \textbf{Eq. 3-14}$$

$$Efficiency(B_{l,i+1}) = \frac{|Row(B_{k,i}) \cap Row(B_{l,i+1})| \times |Col(B_{k,i}) \cap Col(B_{l,i+1})|}{|Row(B_{l,i+1})| \times |Col(B_{l,i+1})|}. \quad \textbf{Eq. 3-15}$$

Definition 3.11 (Bicluster Specificity and Sensitivity): Given $B_{k,i}$ and $B_{l,i+1}$ as the target and discovered biclusters, respectively, the specificity of $B_{l,i+1}$, $Specificity(B_{l,i+1})$ measures the proportion of datapoints in $B_{k,i}$ that has been successfully retrieved by $B_{l,i+1}$, and the sensitivity of $B_{l,i+1}$, $Sensitivity(B_{l,i+1})$ measures the proportion of datapoints in $B_{l,i+1}$ that are also in $B_{k,i}$. Computationally, $Specificity(B_{l,i+1})$ and $Sensitivity(B_{l,i+1})$ are defined by **Eq. 3-13** and **Eq. 3-15**, respectively.

Definition 3.12 (F-value): Given $B_{k,i}$ and $B_{l,i+1}$ as the target and retrieved biclusters, respectively, the F-value associated with the biclustering criterion of $B_{l,i+1}$, $F-value(B_{l,i+1})$ measures with equal weighting of sensitivity and specificity, an overall bicluster quality that represents the harmonic mean of the sensitivity and specificity which cannot be factorized into marginal components, and is given by **Eq. 3-16**:

$$F-value(B_{l,i+1}) = \frac{2|Row(B_{k,i}) \cap Row(B_{l,i+1})| \times |Col(B_{k,i}) \cap Col(B_{l,i+1})|}{|B_{k,i}| \times |B_{l,i+1}|}. \quad \textbf{Eq. 3-16}$$

Definition 3.13 (Jaccard index): For the two biclusters $B_{k,i}$ and $B_{l,i+1}$, the Jaccard index, $Jac(B_{k,i}, B_{l,i+1})$ given by **Eq. 3-17** or **Eq. 3-18**, measures the equality or otherwise of the two biclusters by computing the fraction of row-column combinations in both biclusters from all row-column combinations in at least one bicluster:

$$Jac(B_{k,i}, B_{l,i+1}) = \frac{|B_{k,i} \cap B_{l,i+1}|}{|B_{k,i} \cup B_{l,i+1}|}. \quad \textbf{Eq. 3-17}$$

$$Jac(B_{k,i}, B_{l,i+1}) = \frac{|Row(B_{k,i}) \cap Row(B_{l,i+1})| \times |Col(B_{k,i}) \cap Col(B_{l,i+1})|}{|Row(B_{k,i}) \cup Row(B_{l,i+1})| \times |Col(B_{k,i}) \cup Col(B_{l,i+1})|}. \quad \textbf{Eq. 3-18}$$

Definition 3.14 ($z-score$ Normalization): An entry X_{ij} in a data matrix is normalized with respect to the column j to Z_{ij} by either **Eq. 3-19** or **Eq. 3-21**. In **Eq. 3-19**, σ_j is the standard deviation of the column j as given by **Eq. 3-20**; and in **Eq. 3-21**, s_j is the mean absolute deviation associated with the column j, given by **Eq. 3-22**. In both **Eq. 3-20** and **Eq. 3-22**, n is the total number of rows in the underlying data matrix:

$$Z_{ij} = \frac{X_{ij} - \bar{X}_{.j}}{\sigma_j}. \quad \textbf{Eq. 3-19}$$

$$\sigma_j = \sqrt{\frac{1}{n}\sum_i^n \left(X_{ij} - \bar{X}_{.j}\right)^2}. \quad \textbf{Eq. 3-20}$$

$$Z_{ij} = \frac{X_{ij} - \bar{X}_{.j}}{s_j}. \quad \textbf{Eq. 3-21}$$

$$s_j = \frac{1}{n}\left[\left|X_{1j} - \bar{X}_{.j}\right| + \left|X_{2j} - \bar{X}_{.j}\right| + \cdots + \left|X_{nj} - \bar{X}_{.j}\right|\right]. \quad \textbf{Eq. 3-22}$$

Definition 3.15 (min-max Normalization): This performs a linear transformation on the original data matrix entries X_{ij}. Let the column j of a given data matrix have min_j and max_j as the respective minimum and maximum values, then the min-max normalization process maps a value X_{ij} of the column j to X'_{ij} in a new range $[min'_j, max'_j]$ by computing **Eq. 3-23** [4, 18]:

$$X'_{ij} = \frac{X_{ij} - min_j}{max_j - min_j} \cdot \left[max'_j - min'_j\right] + min'_j. \quad \textbf{Eq. 3-23}$$

3.3 Biclustering with the Plaid Model

For a data matrix $\mathbb{A}_{IJ}$ with a set of rows X and a set of columns Y such that the element a_{ij} represents the relation between row i and column j, a bicluster is defined as a subset of X with similar behavior across a subset of Y, and vice versa [42, 43]. Biclustering algorithm has the goal of discovering a set of biclusters $\{B_k\}$, such that each B_k satisfies localized properties [42]. The Plaid Model (PM) [65, 48, 49, 66] is a statistical biclustering model that fits each data entry a_{ij} of $\mathbb{A}_{IJ}$ with **Eq. 3-24**,

$$a_{ij} = \theta_{ij0} + \sum_{k=1}^{K} \rho_{ik}\lambda_{jk}\theta_{ijk} + \epsilon_{ij}. \quad \textbf{Eq. 3-24}$$

where k is the bicluster index such that $1 \leq k \leq K$, K is the number of biclusters in $\mathbb{A}_{IJ}$, θ_{ij0} models the background bicluster which contains the entirety of $\mathbb{A}_{IJ}$, θ_{ijk} models the

bicluster k, ρ_{ik} and λ_{jk} are the respective row-wise and column-wise bicluster membership parameters, and ϵ_{ij} is the residual error associated with the model.

The parameters ρ_{ik} and λ_{jk} are binary assignments with values $\{0, 1\}$, defined for $k \geq 1$. An iterative process is used to estimate the model parameters when searching for the next bicluster such that for the r^{th} iteration, let $\mathbb{A}_{IJ}^{*(r-1)}$ be the residual matrix of $\mathbb{A}_{IJ}$ corresponding to the $(r-1)^{th}$ iteration, then $\hat{\rho}_{ik}^{r}$ and $\hat{\lambda}_{jk}^{r}$ are the estimates of ρ_{ik} and λ_{jk}, given by **Eq. 3-25** and **Eq. 3-26**, respectively.

$$\hat{\rho}^{r}{}_{ik} = \begin{cases} 1, if\ \sum_j \left[A_{IJ}^{*(r-1)} - \hat{\lambda}_{jk}^{(r-1)}\left(\hat{\mu}_k^r + \hat{\alpha}_{ik}^r + \hat{\beta}_{jk}^r\right)\right]^2 < \sum_j \left(A_{IJ}^{*(r-1)}\right)^2 \\ 0, \qquad otherwise \end{cases}. \quad \textbf{Eq. 3-25}$$

$$\hat{\lambda}^{r}{}_{jk} = \begin{cases} 1, if\ \sum_i \left[A_{IJ}^{*(r-1)} - \hat{\rho}_{ik}^{r-1}\left(\hat{\mu}_k^r + \hat{\alpha}_{ik}^r + \hat{\beta}_{jk}^r\right)\right]^2 < \sum_i \left(A_{IJ}^{*(r-1)}\right)^2 \\ 0, \qquad otherwise \end{cases}. \quad \textbf{Eq. 3-26}$$

The rest of the model parameters estimates $\hat{\mu}_k^r$, $\hat{\alpha}_{ik}^r$, $\hat{\beta}_{jk}^r$ and $\hat{\theta}_{ijk}^r$ are given by **Eq. 3-27**, **Eq. 3-28**, **Eq. 3-29** and **Eq. 3-30**, respectively, for bicluster k with M rows and P columns:

$$\hat{\mu}_k^r = \frac{\sum_{i=1}^{M}\sum_{j=1}^{P}\mathbb{A}_{ij}}{M*P}. \quad \textbf{Eq. 3-27}$$

$$\hat{\alpha}_{ik}^r = \overline{\mathbb{A}}_{i.} - \hat{\mu}_k^r. \quad \textbf{Eq. 3-28}$$

$$\hat{\beta}_{jk}^r = \overline{\mathbb{A}}_{.j} - \hat{\mu}_k^r. \quad \textbf{Eq. 3-29}$$

$$\hat{\theta}_{ijk}^r = \hat{\mu}_k^r + \hat{\alpha}_{ik}^r + \hat{\beta}_{jk}^r \quad \textbf{Eq. 3-30}$$

3.4 Problem Formulation

In this work, the columns of a bicluster B_k are conceptualized as a subspace of features for spatiotemporal subspace feature tracking based on feature relevance. The problem constitutes the establishment and tracking of biclustering criteria that define different biclusters $B_{k,t}$ at different times $t = i, (i + 1), \cdots, T$. At $t = i$, an initial set of biclusters $\{B_{k,t}\}$, referred to as the base or core biclusters, are generated with the PM from the given dataset which can be conceptualized as the parent space. Next is the establishment of biclustering criteria for each of the biclusters in the base bicluster set $\{B_{k,t}\} = \{B_{1,t}, B_{2,t}, \cdots, B_{K,t}\}$. The goal is to track changes in this initially established biclustering criteria over time as the underlying dataset undergo changes. This is achieved by computing and tracking the discriminatory powers of features that define the individual biclusters. This is done temporally such that at any time t, a corresponding affinity matrix is constructed based on the computed discriminatory powers of individual features that uniquely define the concerned bicluster $B_{k,t}$. Potentially, the spatiotemporal changes in the underlying dataset could alter the structural composition and the corresponding biclustering criteria of existing biclusters. Such changes might result in the formation of new biclusters, the splitting, merging, or disappearance of existing ones.

3.4.1 Problem Statement

Given that a real-valued $R \times S$ data matrix $\mathbb{A}^{(t)}$ at time t with a set of rows $X = \{x_r\}$ such that $\{x_r\} = \{x_1, x_2, \cdots, x_R\}$ with $1 \leq r \leq R$ and a set of columns Y= $\{y_s\}$ such that $\{y_s\} = \{y_1, y_2, \cdots, y_S\}$ with $1 \leq s \leq S$, contains the set of K biclusters $\{B_{k,t}\} = \{B_{1,t}, B_{2,t}, \cdots, B_{K,t}\}$ with $1 \leq k \leq K$, where each $B_{k,t}$ has a set of rows $i = \{x'_1, x'_2, \cdots, x'_M\}$ and a set of columns $j = \{y'_1, y'_2, \cdots, y'_P\}$ such that $M < R$ and P< S,

the goal is to compute for feature subspace tracking, the biclustering criteria associated with the set of initially discovered biclusters $\{B_{k,t}\}$ and the subsequently modified bicluster sets at times $(t+1), (t+2), \cdots, T$ as $\mathbb{A}^{(t)}$ undergoes spatiotemporal changes to become $\mathbb{A}^{(t+1)}, \mathbb{A}^{(t+2)}, \cdots, \mathbb{A}^{(T)}$.

3.5 The Proposed Model for Spatiotemporal Subspace Feature Tracking

Figure 3-1 shows the proposed model for the identification and subsequent tracking of spatiotemporal feature subspaces from a given data matrix that represents a parent space of features.

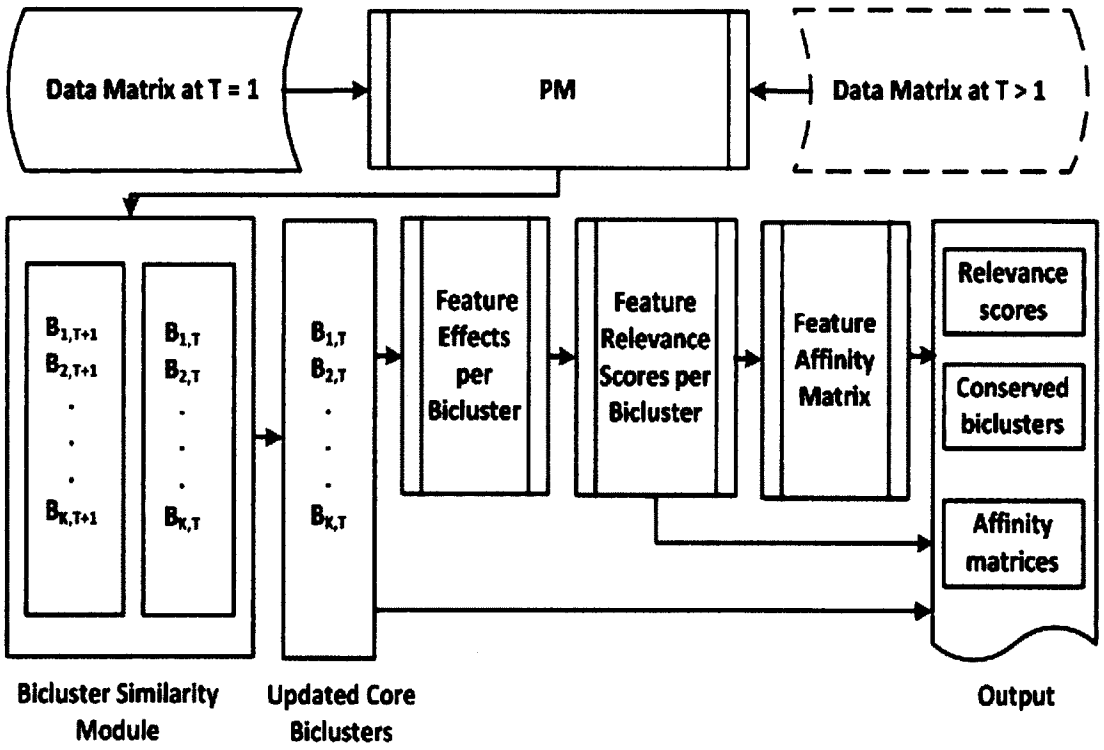


Figure 3-1: The proposed feature subspace discovery and tracking model.

In the proposed model, $B_{K,T}$ represents bicluster K generated at time instance T. The model gives a high level presentation of the algorithm outlined in **Figure 3-2**.

3.5.1 Major Phases of the Proposed Model

1. **Core Biclusters Generation**: Initially, the PM is used to fit the available data matrix at time $T = 1$, with the purpose of generating a core set of biclusters, $B_{k,T} = (B_{1,T}, B_{2,T}, \cdots, B_{K,T})$ to form the initial subspaces whose features are to be marked for tracking based on their discriminatory influences exerted on the containing biclusters.
2. **Bicluster Similarity Module**: At time $(T + 1)$, a new set of biclusters that correspond to the current state of the underlying changing dataset is generated and compared with those generated at previous time T. Different scenarios could materialize, such that (a) previously generated biclusters could remain unchanged or conserved, (b) an existing bicluster might disappear due to its feature members' weakened discriminatory powers, and (c) some existing biclusters could gain more features but continue to maintain their previously established biclustering criteria.
3. **Updated Core Biclusters:** The outcome of the bicluster similarity module is used to update the current core bicluster set to obtain an updated core bicluster set from which such measures as feature effects from **Eq. 3-7**, relevance scores from **Eq. 3-9**, and affinity matrices from **Eq. 3-11**are computed.

3.5.2 The Algorithm for Spatiotemporal Subspace Feature Tracking

Figure 3-2 outlines the main algorithm for the discovery and tracking of subsets of features that form the biclusters within a given dataset.

Algorithm: Feature Subspace Discovery and Tracking
Input: Real-valued R× S data matrix $\mathbb{A}$, biclusters merging threshold, δ
Output: Conserved biclusters, instance and feature relevance scores, affinity matrices

1. At time t, generate the **core biclusters** $\{B_{k,t}\} = \{B_{1,t}, B_{2,t}, \cdots, B_{K,t}\}$ such that each $B_{k,t} = \mathbb{A}_{ij}, 1 \leq i \leq M, 1 \leq j \leq P$, and $1 \leq k \leq K$ using the PM.
2. **for** $k = 1{:}K$ **do** // loop through the set of biclusters
   Compute the bicluster mean, $\mu_k = \bar{\mathbb{A}}_{..}$
   for j= $1{:}P$ **do**
   Feature effect due to column j, $\beta_j = \bar{\mathbb{A}}_{.j} - \mu_k$
   end for
   for $j = 1{:}P$ **do**
   Feature Relevance Score: $P_{\beta_j} = \frac{\beta_j}{\sum_{j=1}^{P} |\beta_j|}$
   end for
   Feature Affinity Matrix, $AM_j(B_{k,t}) = (P_{\beta_1}, P_{\beta_2}, \cdots, P_{\beta_P})^T$
   end for
3. At time instance $t + 1$, generate the bicluster set $\{B_{l,t+1}\} = \{B_{1,t+1}, B_{2,t+1}, \cdots, B_{L,t+1}\}$ that corresponds to the dataset at $(t + 1)$.
4. **for** $k = 1{:}K$ **do** // loop through the current set of biclusters
   for $l = 1{:}L$ **do**
   if $(B_{k,t} \cap B_{l,t+1} \neq \emptyset)$ **and** $(Jac(B_{k,t}, B_{l,t+1}) \geq \delta)$ **then**
   $B_{k,t}$ is conserved
   merge $(B_{k,t}, B_{l,t+1})$ // perform row and column updates
   else if $(B_{k,t} \cap B_{l,t+1} \neq \emptyset)$ **and** $(Jac(B_{k,t}, B_{l,t+1}) \leq \delta)$ **then**
   $B_{k,t}$ is conserved; and a new bicluster $B_{l,t+1}$ is discovered
   end if
   if $(B_{k,t} \cap B_{l,t+1} = \emptyset)$ **then**
   $\{B_{k,t}\}$ have disappeared; new set of biclusters $\{B_{l,t+1}\}$ are discovered
   end if
   end for
   end for
5. Update the core bicluster set to a new core $\{B_{k,t}\}$ such that $1 \leq k \leq K + L$
6. Repeat **Steps 2 – 5** for subsequent times $t + 2, t + 3, \cdots, T$
7. At time $t = T$, list:
   i. All conserved and newly discovered biclusters
   ii. All feature relevance scores, P_{β_j}
   iii. The feature affinity matrix, $AM_j(B_{k,t})$
8. Stop

Figure 3-2: Algorithm for tracking subspace of features in a real-valued data matrix.

3.6 Conclusion

This chapter introduces the various notations and symbols used to present formal definitions and computational relations utilized in this dissertation. It presents an outline of biclustering with the statistical Plaid Model, followed by the problem formulation and problem statement of the dissertation. Next, the proposed model and detailed algorithmic steps are presented.

CHAPTER 4

PERSISTENT BICLUSTERS FOR FEATURE TRACKING

Given a real-valued data matrix described by a set of attributes called features in this work, a biclustering algorithm determines submatrices of the original matrix where subsets of rows exhibit a correlated pattern over subsets of columns [67]. This chapter proposes the use of biclustering as a means of feature subsets selection from a vector of features upon which the data were collected. A group of features that defines a bicluster tend to offer common local feature relevance and local feature correlation [68]. In order to track these features accurately, it is challenging to select an optimal set of features at the beginning of the tracking process to ensure reliability and optimal performance of the tracking algorithm. Many biclustering algorithms exist in the literature and the PM is one of the most widely used techniques. This chapter outlines the use of an enhanced PM (EPM) for the generation of persistent and reliable subsets of features whose relevance scores can be tracked in a spatiotemporal dataset.

4.1 Research Motivation

The work by Lazzeroni and Owen [49] first proposed the use of the PM as a biclustering technique for the analysis of gene expression dataset. The technique discovers biclusters in a given numeric data matrix by treating its elements as a sum of terms called layers or biclusters that are used to fit a linear function to describe the

elements of the underlying dataset [69]. However, the nondeterministic nature of the solutions by the original PM problem formulation leads to situations where the number of discovered biclusters and the overall biclusters quality is not guaranteed. The outputs associated with different executions of the model fluctuate erratically for the same dataset. As it is with other existing biclustering algorithms, the PM is either based on generative or greedy algorithms that do not offer guarantees of the inclusiveness and completeness of biclustering solutions [67]. Hence, there is the need for an algorithmic technique that ensures the generation of a set of biclusters that can be considered persistent or conserved and reflects the true nature and number of biclusters contained in the underlying dataset.

4.2 Problem Statement

Given a real-valued $R \times S$ data matrix $\mathbb{A}_{IJ} = (X, Y)$, with a set of rows $X = \{x_1, x_2, \cdots, x_R\}$ and a set of columns $Y = \{y_1, y_2, \cdots, y_S\}$, the goal is to discover a set of K conserved biclusters $\{B_k\} = \{B_1, B_2, \cdots, B_K\}$ with $1 \leq k \leq K$ by imposing a convergence technique on the nondeterministic outputs of multiple iterations of the PM until convergence on $\{B_k\}$.

4.3 Methodology and Materials

Most biclustering algorithms currently in use avoid prohibitive exhaustive search and rely on heuristics to explore the solution space due to the NP-hard nature of the biclustering problem formulation [42, 70, 71, 72, 73]. The proposed EPM takes advantage of this by using, as the input, a set of nondeterministic outputs of the PM to generate a conserved list of biclusters that are more coherent and statistically significant.

4.3.1 The Proposed Model

Figure 4-1 shows the proposed EPM.

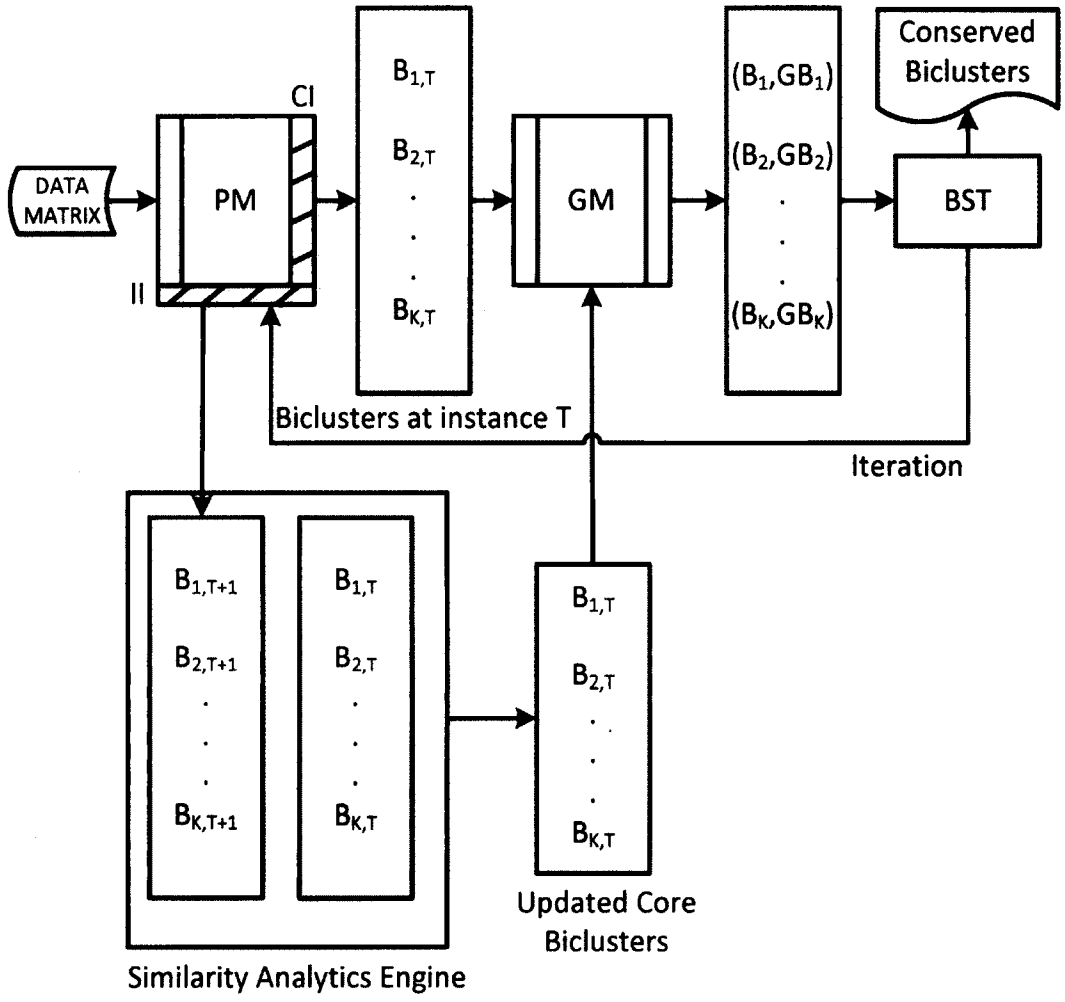


Figure 4-1: The EPM for conserved biclusters discovery.

The following abbreviations (in the format abbreviation: meaning) are used in the model. CI: Core Interface, II: Iterative Interface, GM: Goodness Measure, B_K: Bicluster K, $B_{K,T}$: Bicluster K generated at instance T, GB_K: Goodness measure for bicluster B_K, and BST: Bicluster Significance Test.

4.3.2 Phases of the EPM

1. **Core Biclusters Generation**: The initial phase of the EPM involves the generation of a set of biclusters, $B_{k,T} = (B_{1,T}, B_{2,T}, \cdots, B_{K,T})$, known as the core biclusters. This is realized via the core interface (CI) of the model in **Figure 4-1**, whereby the PM is run on the given data matrix to generate the set of K biclusters, $B_{K,T}$ at the initial instance T.
2. **Bicluster Goodness Measure**: The bicluster's goodness measure which represents the coherence or quality of each discovered bicluster is computed based on the differential co-expression score proposed by Chia and Karuturi [2] to generate a vector of goodness scores, $GB_k = (GB_1, GB_2, \cdots, GB_K)$, to be utilized in a statistical significance analysis of the discovered biclusters in step 3. Each GB_k is computed by **Eq. 4-1** where $T_h(k)$ quantifies the T-type co-expression in bicluster k to indicate strong rows only effect in group h, $B_h(k)$ quantifies the B-type co-expression in bicluster k to indicate strong columns only effect, and a is a small fudge effect factor to offset large ratios based on very small co-expression in both groups of biclusters such that $0 < a \ll 1$. As proposed by the authors, the higher positive the bicluster goodness score the better:

$$GB_k = \log\left(\frac{max\{T_1(k) + a,\ B_1(k) + a\}}{max\{T_2(k) + a,\ B_2(k) + a\}}\right). \qquad \textbf{Eq. 4-1}$$

3. **Bicluster Significance Test (BST)**: In this step, we perform a statistical significance test on the vector GB_k generated in step 2. We assume the set of

goodness scores represented by the vector GB_k to be a sample of independently distributed random variables drawn from a normally distributed population with mean μ and standard deviation σ i.e. $N(\mu, \sigma^2)$, and test the following null hypothesis H_0 for the given data matrix $\mathbb{A}$:

H_0: The mean goodness score of the current list of GB_k, $\mu_{GB_k} = \mu$.

H_1: $\mu_{GB_k} \neq \mu$.

Under the null hypothesis H_0, the test statistic t given by **Eq. 4-2** has a t-distribution with $(n-1)$ degrees of freedom where n is the size of the sample with sample mean $\bar{x}$ and standard deviation s:

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}}. \qquad \textbf{Eq. 4-2}$$

The EPM maximizes the power, $\mathcal{P} = (1-\beta)$ with $\mathcal{P} \in [0,1]$ associated with the test hypotheses, where β is the Type II error of the test which measures the probability of incorrectly retaining a false null hypothesis H_0. The *t.test* and *power.t.test* functions in core R [74] were used to obtain the t-statistic and $\mathcal{P}$. These functions assume a universe, from which a sample is drawn, to be distributed normally with $N(0,1)$. Thus, as the distribution of GB_k approaches the standard normal distribution over time, the power of the test $\mathcal{P}$ approaches 1. $\mathcal{P}$ is utilized in an objective function to control the iteration process of the algorithm. The acceptance of H_0 leads to a conclusion that $GB_k$is a true representation of the current core biclusters in$\mathbb{A}$; otherwise, the selection of H_1 invokes a series of iterations to improve the richness of the

previously generated elements in GB_k. This process eventually leads to the modification of the goodness scores vector to guide the generation of conserved biclusters as outlined in the iteration process phase in step 4.

4. **The Iteration Process**: Each time H_0 in step 3 is rejected, the PM module in the EPM is re-run via the iterative interface (II), and the outcome used to improve the set of biclusters generated at instance T (it is customary here to refer to the previously generated biclusters as those generated at instance T, and the most current list as those generated at instance T+1) as follows:

   i. A new set of biclusters $B_{k,T+1} = \left(B_{1,T+1}, B_{2,T+1}, \cdots, B_{K,T+1}\right)$ is generated at instance (T+1) to be compared with those generated at T through an in-depth similarity analysis via the similarity analytics engine of the model. The EPM employs a systematic set of operations and bicluster comparisons based on the Jaccard index, Jac [49, 45, 75], defined by **Eq. 3-17** or **Eq. 3-18** to modify either existing biclusters in the list from instance T or append to it a set of newly discovered biclusters at instance (T+1), which were not revealed at T.

   ii. Modification of existing biclusters is enforced based on Jac computed between the existing and any of the newly generated biclusters. For extensive comparisons at different levels of similarity between biclusters, this work employs an objective function based on biclusters merging threshold, δ written as EPM (δ) or EPM @δ.

   Computationally, δ signifies the degree of overlap between any two given biclusters with values in [0, 1] where a value of 0 means no

overlap and that of 1 means 100% overlap between two biclusters. We reported results for $\delta = \{0.90, 0.95, 0.99\}$ to indicate the merging of any two biclusters that produced Jac signifying 90%, 95% and 99% similarities between them.

iii. Goodness measures for the biclusters in the currently updated core list are then obtained, followed by the BST as in steps 2 and 3, respectively.

iv. The iterative process is repeated until a desired vector of goodness measures is obtained, and the process ends with an output of conserved biclusters.

5. **Termination Criteria**: At the end of every iteration, the quality in terms of goodness scores of the most recent bicluster set forming the core is used to assess its closeness to the desired conserved list inherent in the given data matrix. To achieve this, the model utilizes the Type II error associated with the BST module in such a configuration that the error tends to approach zero as the optimal bicluster set is realized. Let B_{core} and $B_{conserved}$ be the respective set of core and the desired conserved biclusters in the data matrix $\mathbb{A}$ at any given point in the discovery process. If β and $\mathcal{P}$ are the respective Type II error and the power under the null hypothesis in step 3, then either **Eq. 4-3** or **Eq. 4-4** ensures the termination of the EPM algorithm as the conserved biclusters in $\mathbb{A}$ are realized:

$$\lim_{\beta \to 0} B_{core} \to B_{conserved}. \qquad \textbf{Eq. 4-3}$$

$$\lim_{\mathcal{P} \to 1} B_{core} \to B_{conserved}. \qquad \textbf{Eq. 4-4}$$

Figure 4-2 illustrates the behavior of $\mathcal{P}$ based on **Eq. 4-4** during the iterative process leading to unveiling the final list of biclusters contained in the given dataset.

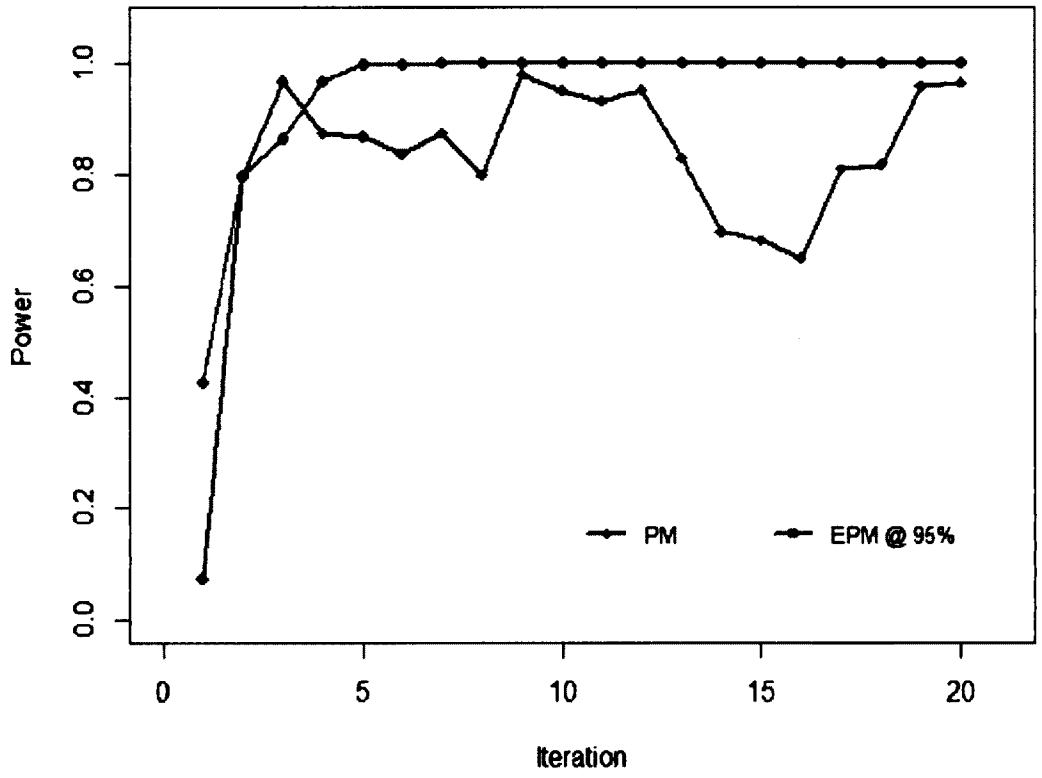


Figure 4-2: Illustrating the convergence technique of the EPM. Both the EPM and the PM were run on the same dataset. The EPM terminated after 20 iterations, and the PM was independently run 20 times.

It shows how the EPM achieves this, compared with scores obtained by individual runs of the PM. Here, the power scores by the EPM converge smoothly to 1 as the inherent biclusters are revealed after 20 iterations, while 20 individual runs of the PM portray a rather erratic trend for the power scores.

4.3.3 The EPM Algorithm

Figure 4-3 outlines the main steps of the proposed EPM algorithm. The algorithm takes as input, a real data matrix $\mathbb{A}$ and outputs a set of conserved biclusters.

Algorithm: Conserved Biclusters Discovery
Input: Real-valued R × S data matrix $\mathbb{A}$, merging threshold δ
Output: A list, L of conserved biclusters inherent in $\mathbb{A}$

1. At T, generate $B_{k,T} = (B_{1,T}, B_{2,T}, \cdots, B_{K,T})$: each $B_{k,T} = \mathbb{A}_{ij}$, $1 \leq i \leq M$ and $1 \leq j \leq P$, using the PM.
2. **for** $k = 1{:}K$ **do**
   Compute $GB_k = (GB_1, GB_2, \cdots, GB_K)$
   end for
3. Perform BST on GB_k // H_0versus H_1
   Compute $\beta, \mathcal{P}$
   if $((H_0 \leftarrow$ TRUE) **AND** $(\beta == 0))$ **then**
   $\{L \leftarrow B_{k,T}$
   go to 5$\}$ **else**
   Generate $B_{l,T+1} = (B_{1,T+1}, B_{2,T+1}, \cdots, B_{P,T+1})$ at $T + 1$
   end if
4. **while** $(\beta\ != 0)$ **do**
   for k = 1: K **do**
   for $l = 1{:}$ P **do**
   if $(Jac(B_{k,T}, B_{l,T+1}) \geq \delta)$ **then**
   merge $(B_{k,T}, B_{l,T+1})$ //row or column updates
   if $(Jac(B_{k,T}, B_{l,T+1}) == 0)$ **AND** $(GB_l \geq GB_k)$ **then**
   $B_{k,T} \leftarrow$ **append** $(B_{k,T}, B_{l,T+1})$
   end for
   end for
   go to 2
   end while
5. STOP

Figure 4-3: Core steps of the EPM algorithm.

4.3.4 Comparison with other Biclustering Algorithms

To ascertain and validate the performance of the proposed EPM algorithm, four state-of-the-art non plaid and one plaid biclustering algorithms were compared with the EPM. The non-plaid algorithms are BiMax, CC, xMOTIFs, Spectral, and the plaid

algorithm is the PM. The biclust package in R [76] was used to run all five competing algorithms considered, and the EPM was entirely implemented in R.

4.3.5 Parameter Settings

Parameters of the different algorithms considered in this work were set to either the default values recommended by their authors, or specific values were chosen to suite the dataset distribution under consideration. Here, we outline those specific parameters among the default settings that were changed to enhance individual algorithm performances. For Spectral, the *normalization* and *numberofEigenvalues* parameters were respectively set to bistochastization and 7. We set the maximum accepted score, *delta* and the scaling factor, *alpha* parameters of the CC algorithm to 0.02 and 1, respectively. In particular, a smaller value of 0.02 was chosen for *delta* to ensure the detection of more refined patterns in the dataset, as recommended by the authors, Cheng and Church [51]. The *max.layer* parameter of the PM indicating the maximum number of layers to include in the model was set to 100. All parameters were kept at default values for BiMax and xMOTIFs. BiMax only works with binary data, and all datasets to it were converted with the *binarize* function from the biclust package in R, with the median score as the *threshold* parameter value. xMOTIFs requires discrete data input and all datasets to it were converted with the *discretize* function from the biclust package in R, with equally spaced interval from minimum to maximum values. Three different values of 0.90, 0.95, and 0.99 were chosen for the merging threshold parameter, δ of the EPM, to indicate the degree of overlap between different biclusters.

4.3.6 Artificial Dataset Generation

Artificial datasets with implanted biclusters were used to assess the performance of the proposed EPM. In the literature, researchers have generated synthetic datasets with either a single data model [67], or multiple data models [45, 46, 75]. A Gaussian-based single data model that favors no specific algorithm was used in this work to generate a set of five datasets, each with a different number of hidden biclusters to be discovered by the proposed algorithm, EPM. The Gaussian distribution represented by $N(\mu, \sigma^2)$ where μ is the mean and σ is the standard deviation was used for the synthetic data generation, with parameter settings as detailed in **Table 4-1**.

Table 4-1: Outline of the five synthetic datasets specifying the number of hidden biclusters, standard deviation of each bicluster and the size of each dataset.

Dataset	Number of Implanted Biclusters	Bicluster Standard Deviation	Size (Rows X Cols)
1	2	{0.2, 0.4}	100 X 20
2	4	{0.2, 0.4, 0.6, 0.8 }	200 X 30
3	8	{0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6}	500 X 60
4	10	{0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0}	1000 X 100
5	15	{0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0}	2000 X 160

Each dataset was generated by the following steps: (1) A data matrix, $\mathbb{A}$ for the background layer was generated with the standard normal distribution, $N(0,1)$ (2) Pre-defined biclusters were created with the distributions $N(10, \sigma^*)$, where $\sigma^* = \{\sigma_1^2, \sigma_2^2, \cdots, \sigma_k^2\}$ to introduce different noise levels to the K biclusters per dataset, and (3) The pre-defined biclusters in step (2) were then implanted in $\mathbb{A}$ without allowing overlaps.

4.3.7 Real Dataset

The Saccharomyces cerevisiae gene expression dataset, originally generated by Eisen *et al.* [74, 76, 77] was used to assess the performance of the proposed EPM. It is a microarray data matrix with information about the expression levels of 6,221 yeast genes over 80 conditions. Missing values in the original dataset were imputed using k-nearest neighbor averaged with the *impute.knn* function from the impute library in core R [74], with default k value of 10, resulting in 10-nearest neighbors averaged.

4.3.8 Evaluation Techniques on Synthetic Dataset

Biclusters generated by the proposed EPM on the synthetic datasets were validated following the protocol proposed by Eren *et al.* [45], adopted from the work of Prelic *et al.* [57]. Given two sets of biclusters, B_1 and B_2, the method calculates what is called a set score, $S(B_1, B_2)$ which compares the two sets by assigning higher scores to similar bicluster pairs and lower scores to dissimilar pairs, based on the Jaccard coefficient $s(b_1, b_2) \in [0,1]$ with $b_1 \in B_1$ and $b_2 \in B_2$ defined by **Eq. 4-5**, where $|b_1 \cap b_2|$ and $|b_1 \cup b_2|$ are the respective bicluster data points intersection and union between b_1 and b_2:

$$s(b_1, b_2) = \frac{|b_1 \cap b_2|}{|b_1 \cup b_2|}. \qquad \textbf{Eq. 4-5}$$

Let $B_1$and B_2 respectively represent the ground truth of the expected bicluster set implanted in the matrix $\mathbb{A}$ and the set discovered by the algorithm, then the Recovery, $S(B_1, B_2)$ and Relevance, $S(B_2, B_1)$ scores are obtained by **Eq. 4-6** and **Eq. 4-7**, respectively:

$$S(B_1, B_2) = \frac{1}{|B_1|}\sum_{b_1 \in B_1} \max_{b_2 \in B_2} s(b_1, b_2). \quad \textbf{Eq. 4-6}$$

$$S(B_2, B_1) = \frac{1}{|B_2|}\sum_{b_2 \in B_2} \max_{b_1 \in B_1} s(b_1, b_2). \quad \textbf{Eq. 4-7}$$

The recovery score measures the percentage of the ground truth B_1 that was discovered by the proposed algorithm and it is maximized for $B_1 \subseteq B_2$. The relevance score measures the percentage of the discovered biclusters B_2 that overlaps with the ground truth, B_1, and it is maximized for $B_2 \subseteq B_1$. All the algorithms considered and compared with the proposed EPM were evaluated for biclusters quality based on the goodness score procedure developed by Chia and Karuturi [2], given by **Eq. 4-1**.

4.3.9 Evaluation Techniques on Real Gene Expression Dataset

Validation of biclusters discovered by the EPM from real gene expression dataset was done using both internal and external evaluation protocols. In this dissertation, the evaluation protocols by Eren *et al.* [45] and Oghabian *et al.* [46] were followed where internal bicluster validation involved the use of algorithmic and dataset properties, while external validation involved the use of other external sources of information to establish the quality of biclusters generated. Internally, biclusters generated from the gene expression dataset were evaluated by measuring their goodness based on the differential co-expression scoring function suggested by Chia and Karuturi [2], defined by **Eq. 4-1** and available in the R package, biclust by Kaiser *et al.* [76]. With this protocol, a stronger positive goodness score indicates a bicluster's superiority. Externally, biclusters from the real gene expression dataset were evaluated by carrying out enrichment analysis to calculate the Gene Ontology (GO) term enrichments for the genes per bicluster.

GO enrichment was done using the Web-based Gene Set Analysis Toolkit (WebGestalt) by Wang *et al.* [78], and reported on all the three categories of Biological Processes, Molecular Functions and Cellular Components at three significant levels (0.05, 0.02 and 0.01) of analysis. Following the protocol used by both Sun *et al.* [75] and Eren *et al.* [45], the first phase of the enrichment analysis involved using the list of genes within a bicluster as input for a hypergeometric test with the Entrez Gene identifiers list [79] as the gene universe to generate the initial raw p-values. A second phase to adjust the raw p-values via multiple significance test correction using the Hochberg and Benjamini [80] correction method was performed to obtain adjusted p-values. A bicluster is considered to be enriched if the adjusted p-value of at least one GO term is smaller than the significance level under consideration.

4.4 Results and Discussions

This section presents the experimental and performance assessments of the proposed EPM and the other competing algorithms mentioned earlier on the five synthetic datasets and the real Saccharomyces cerevisiae gene expression dataset. Results on the synthetic datasets are reported first, followed by the performance assessment on the real gene expression dataset.

4.4.1 Synthetic Datasets

The proposed EPM algorithm and the others considered in this work were evaluated on the artificial datasets outlined in **Table 4-1**. The experiments were repeated five times on each dataset and the average results were reported as follows:

4.4.1.1 *Number and Scalability Experiments*

The EPM and the other competing algorithms were assessed and compared on their ability to accurately discover increasing number of implanted biclusters in the underlying datasets. The average numbers of biclusters discovered are shown in **Table 4-2** and **Figure 4-4**.

Table 4-2: Number of biclusters discovered by the individual algorithms on the synthetic datasets. BM: BiMax, xMs: xMOTIFs, Sp: Spectral.

Number of Implanted Biclusters	BM	xMs	Sp	CC	PM	EPM δ = 0.90	EPM δ = 0.95	EPM δ = 0.99
2	89	6	601	30	13	6	5	4
4	100	37	593	53	4	10	11	11
8	100	86	554	100	10	12	16	16
10	43	93	111	100	12	15	14	15
15	100	0	719	100	18	16	16	17

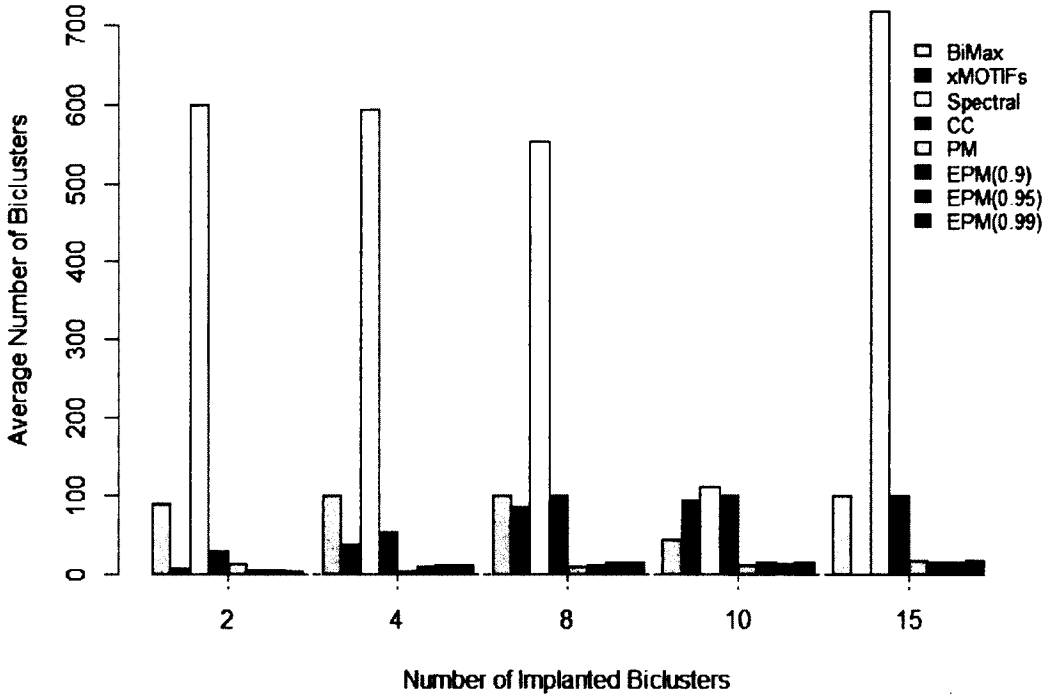


Figure 4-4: The number of biclusters discovered by different algorithms.

Scalability in term of each algorithm's ability to accurately discover hidden biclusters as the number of biclusters and data size increase were measured as the recovery and relevance scores of the different algorithms. The respective average recovery and relevance scores are shown in **Table 4-3** and **Table 4-4**, with **Figure 4-5** and **Figure 4-6** showing the corresponding bar charts.

Table 4-3: Recovery scores by the different algorithms. BM: BiMax, xMs: xMOTIFs, Sp: Spectral.

Number of Implanted Biclusters	BM	xMs	Sp	CC	PM	EPM δ = 0.90	EPM δ = 0.95	EPM δ = 0.99
2	1.00	0.05	0.05	0.05	1.00	1.00	1.00	1.00
4	0.83	0.30	0.02	0.02	0.82	0.98	0.98	0.96
8	1.00	0.01	0.01	0.01	0.85	0.95	0.96	0.96
10	1.00	0.00	0.01	0.01	0.78	0.92	0.93	0.92
15	0.16	0.00	0.13	0.00	0.33	0.34	0.39	0.35

Table 4-4: Relevance scores by the different algorithms. BM: BiMax, xMs: xMOTIFs, Sp: Spectral.

Number of Biclusters	BM	xMs	Sp	CC	PM	EPM δ = 0.90	EPM δ = 0.95	EPM δ = 0.99
2	0.47	0.05	0.05	0.05	1.00	0.97	0.99	1.00
4	0.33	0.30	0.02	0.02	0.82	0.94	0.95	0.96
8	0.52	0.00	0.01	0.01	0.79	0.82	0.84	0.76
10	0.35	0.00	0.01	0.01	0.68	0.77	0.76	0.65
15	0.15	0.00	0.06	0.00	0.49	0.45	0.45	0.49

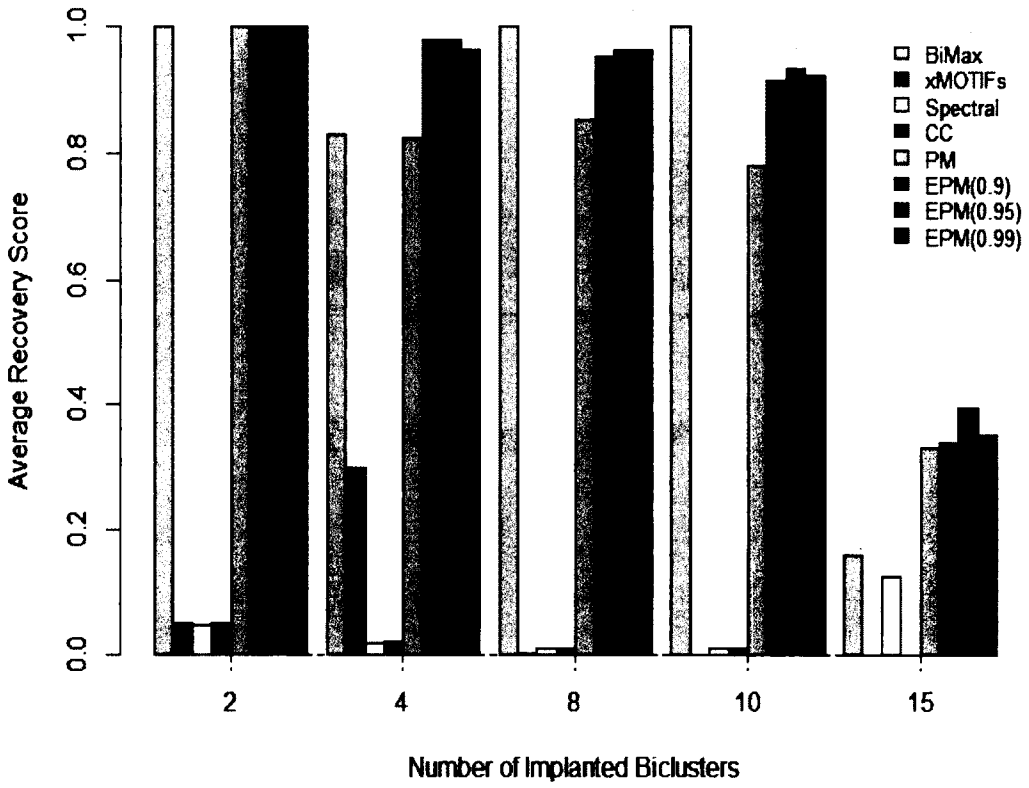


Figure 4-5: A chart showing the mean recovery scores by the different algorithms.

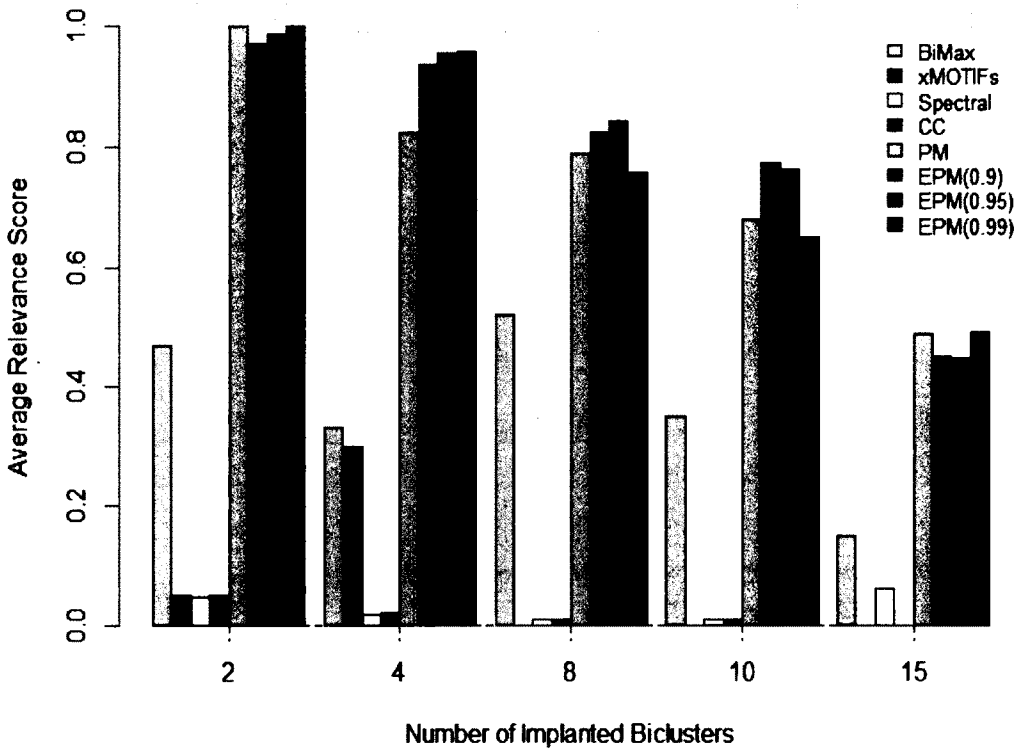


Figure 4-6: A chart showing the mean relevance scores by the different algorithms.

For the datasets with 2, 4, 8 and 10 implanted biclusters, BiMax, PM and the EPM performed best in discovering the hidden biclusters. With the exception of the case with 2 hidden biclusters, the EPM outperformed the PM in recovery scores. All the algorithms dropped in recovery rates when the number of implanted biclusters was 15, with the EPM scoring the best rate of 0.39 when the merging threshold $\delta = 0.95$. Generally, with the exception of BiMax, similar trends were observed for the relevance scores with the EPM and PM outperforming the rest. BiMax's poor relevance scores could be attributed to the relatively large number of biclusters found, most of which are differentially different from the implanted biclusters.

4.4.1.2 *Bicluster Quality Experiment*

The average bicluster goodness scores for each algorithm on the five synthetic datasets are shown in **Table 4-5** and **Figure 4-7**.

Table 4-5: Bicluster goodness scores reported by the different algorithms on the five synthetic datasets considered.

	Goodness Scores				
	Number of Implanted Biclusters				
Algorithm	**2**	**4**	**8**	**10**	**15**
BiMax	4.15	3.77	4.54	4.59	3.87
xMOTIFs	-0.90	-0.72	-0.09	0.00	0.00
Spectral	0.10	0.10	0.06	0.02	0.52
CC	0.53	0.80	0.56	0.52	0.34
PM	0.68	0.59	0.51	0.40	0.30
EPM $\delta = 0.90$	0.68	0.65	0.51	0.42	0.30
EPM $\delta = 0.95$	0.68	0.66	0.53	0.42	0.29
EPM $\delta = 0.99$	0.68	0.66	0.53	0.41	0.36

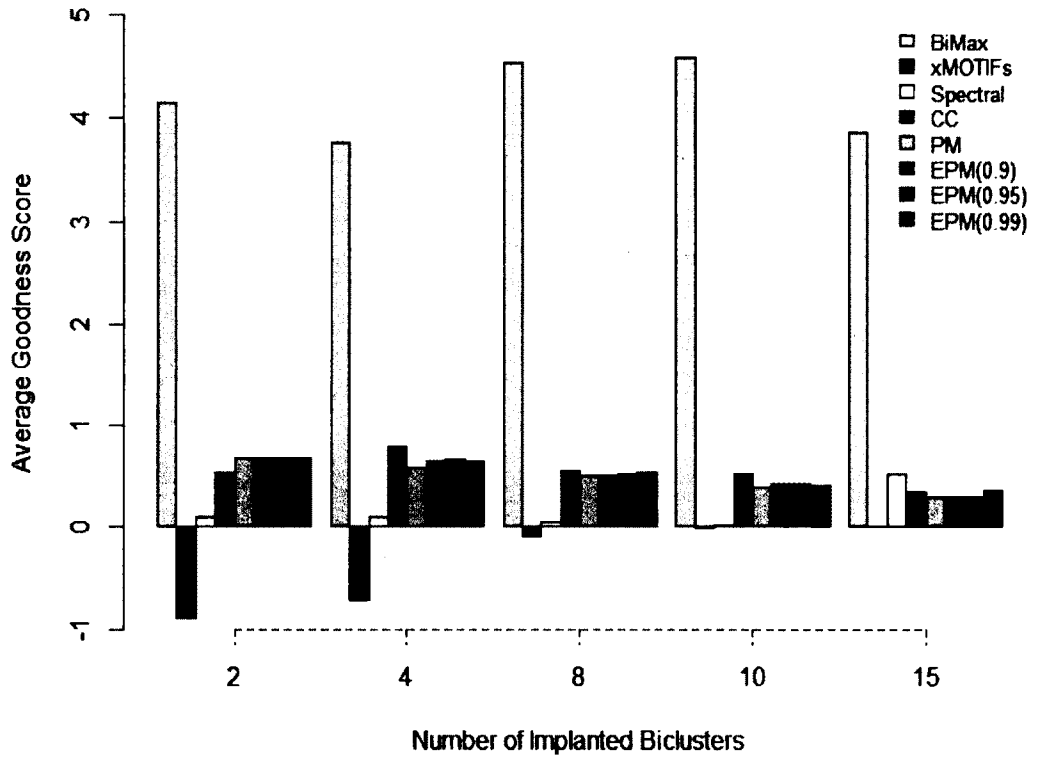


Figure 4-7: A chart showing the biclusters goodness scores reported by each algorithm on the five synthetic datasets.

The EPM with $\delta = 0.99$ gave better goodness scores than the PM in all the cases except with 2 implanted biclusters where the scores were equal. BiMax had the best goodness scores across all the cases of the artificial datasets. xMOTIFs scored worse across all the datasets considered.

4.4.1.3 *Runtime Experiment*

This section presents an assessment of the benchmark used to ensure that the proposed EPM executes within a reasonable amount of time. The central processing unit (CPU) execution times of the EPM were compared with those of the PM and the other

algorithms on the five artificial datasets considered in this work. The results are shown in **Table 4-6**, **Figure 4-8** and **Figure 4-9**.

Table 4-6: Algorithms CPU execution times in seconds (s).

Number of Biclusters	BiMax	xMOTIFs	Spectral	CC	PM	EPM δ = 0.90	EPM δ = 0.95	EPM δ = 0.99
2	0.14	0.14	4.65	0.53	0.25	0.26	0.21	0.24
4	0.20	0.50	11.69	1.41	0.33	0.09	0.24	0.20
8	0.28	1.79	22.41	4.97	1.36	1.89	1.70	1.88
10	0.24	4.53	45.55	10.45	5.91	7.95	8.45	10.02
15	1.82	0.00	106.08	15.63	17.56	18.71	21.33	16.89

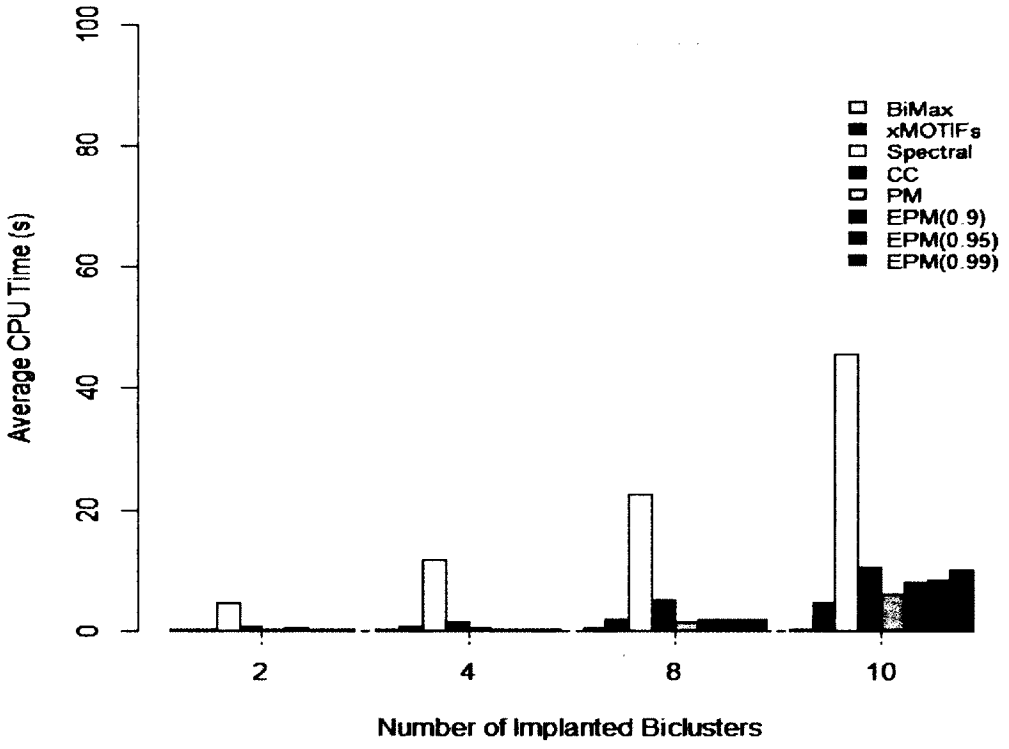


Figure 4-8: CPU execution times in seconds (s), reported by the different algorithms on the four artificial datasets with 2, 4, 8 and 10 implanted biclusters.

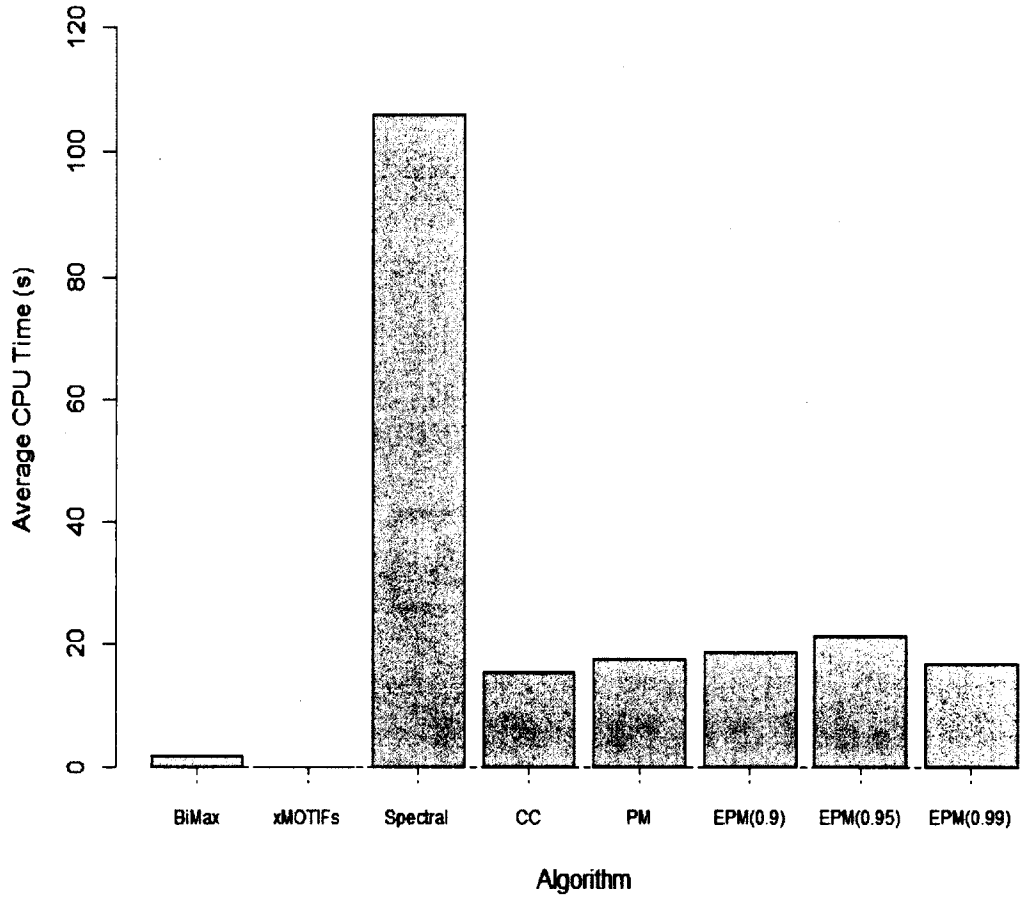


Figure 4-9: CPU execution times in seconds (s), reported by the different algorithms on the artificial dataset with 15 implanted biclusters.

In general, all the algorithms including the proposed EPM portrayed linear execution times with potential exponential growth as the number of implanted biclusters and data sizes grow for all five cases considered. Spectral was the slowest algorithm among the set considered.

4.4.1.4 *Memory Usage Experiment*

The mean random access memory (RAM) size in megabytes (MB) used by the EPM, along with the other competing algorithms on the synthetic datasets are shown in **Table 4-7**, **Figure 4-10**, **Figure 4-11** and **Figure 4-12**.

Table 4-7: Memory usage by the EPM and the other competing algorithms in MB.

Number of Biclusters	**BiMax**	**xMOTIFS**	**Spectral**	**CC**	**PM**	**EPM δ = 0.90**	**EPM δ = 0.95**	**EPM δ = 0.99**
2	516.13	530.31	506.37	617.09	591.47	579.72	570.09	524.18
4	533.74	603.24	547.93	573.60	519.25	496.40	592.13	628.61
8	546.84	546.37	538.93	611.90	413.44	456.26	438.55	433.73
10	543.50	635.01	545.58	464.91	422.89	438.83	432.25	377.27
15	576.53	-	545.97	432.00	426.77	408.60	439.46	457.18

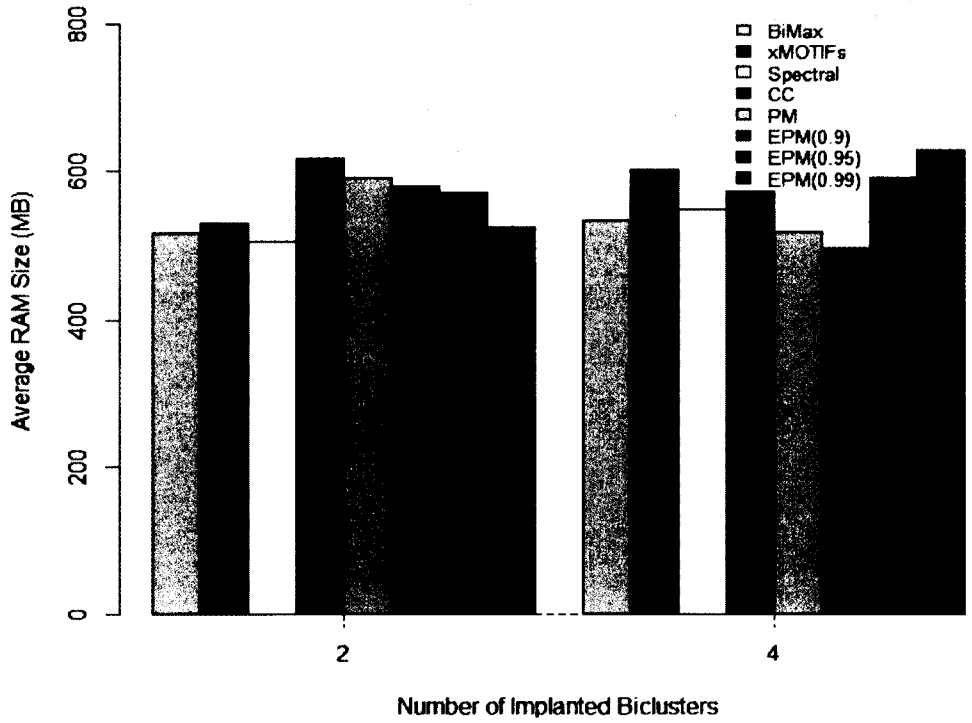


Figure 4-10: A chart showing the amount of memory utilized by the different algorithms on the synthetic datasets with 2 and 4 implanted biclusters.

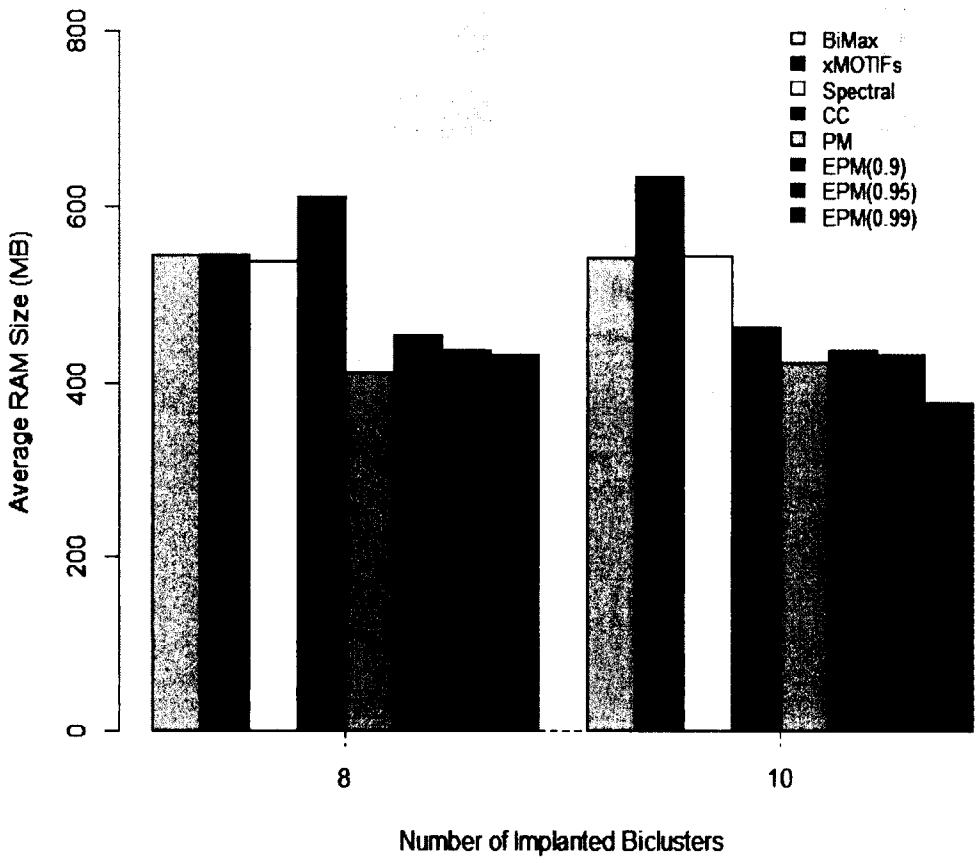


Figure 4-11: A chart showing the amount of memory utilized by the different algorithms on the synthetic datasets with 8 and 10 implanted biclusters.

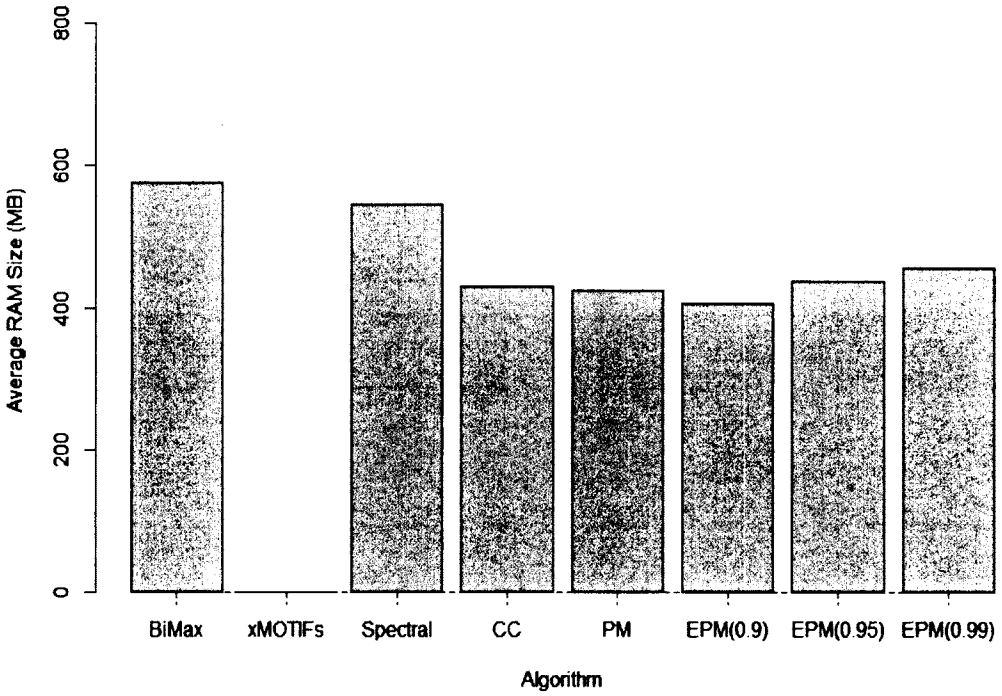


Figure 4-12: The amount of memory utilized by the different algorithms on the synthetic datasets with 15 implanted biclusters.

The EPM with $\delta = 0.90$ showed decreasing memory usage as data size and the number of implanted biclusters increased. On the whole, the EPM with $\delta = 0.99$ and 10 implanted biclusters recorded the least memory usage of 377.27 MB while xMOTIFs recorded the most memory usage of 635.01 MB for the same number of implanted biclusters. With 15 implanted biclusters, xMOTIFs could not execute on the dataset, and the corresponding memory usage is shown with a dash (-) in **Table 4-7**.

4.4.2 Real Gene Expression Dataset

The EPM was evaluated and compared with the other five algorithms on the *Saccharomyces cerevisiae* gene expression dataset described earlier in the chapter.

Benchmarks considered include the number of biclusters discovered, algorithm execution time, memory (RAM) usage, bicluster quality in terms of goodness measure and GO term enrichment analysis. **Table 4-8** summarizes the results obtained for the number of biclusters found, execution times and memory usage across all the algorithms considered, and **Figure 4-13**, **Figure 4-14** and **Figure 4-15** show the corresponding bar charts.

Table 4-8: Algorithm performance scores using the real gene expression dataset. It shows the number of biclusters found, CPU execution times and the size of RAM used.

	Performance Measures		
Algorithm	**Number of Biclusters**	**CPU Time (Seconds)**	**Memory Usage (MB)**
BiMax	100	11.50	558.82
xMOTIFs	46	22.80	526.74
Spectral	41	171.49	2722.25
CC	100	29.56	2447.46
PM	21	75.17	570.02
EPM $\delta = 0.90$	81	45.28	565.27
EPM $\delta = 0.95$	116	34.92	560.77
EPM $\delta = 0.99$	101	86.14	542.04

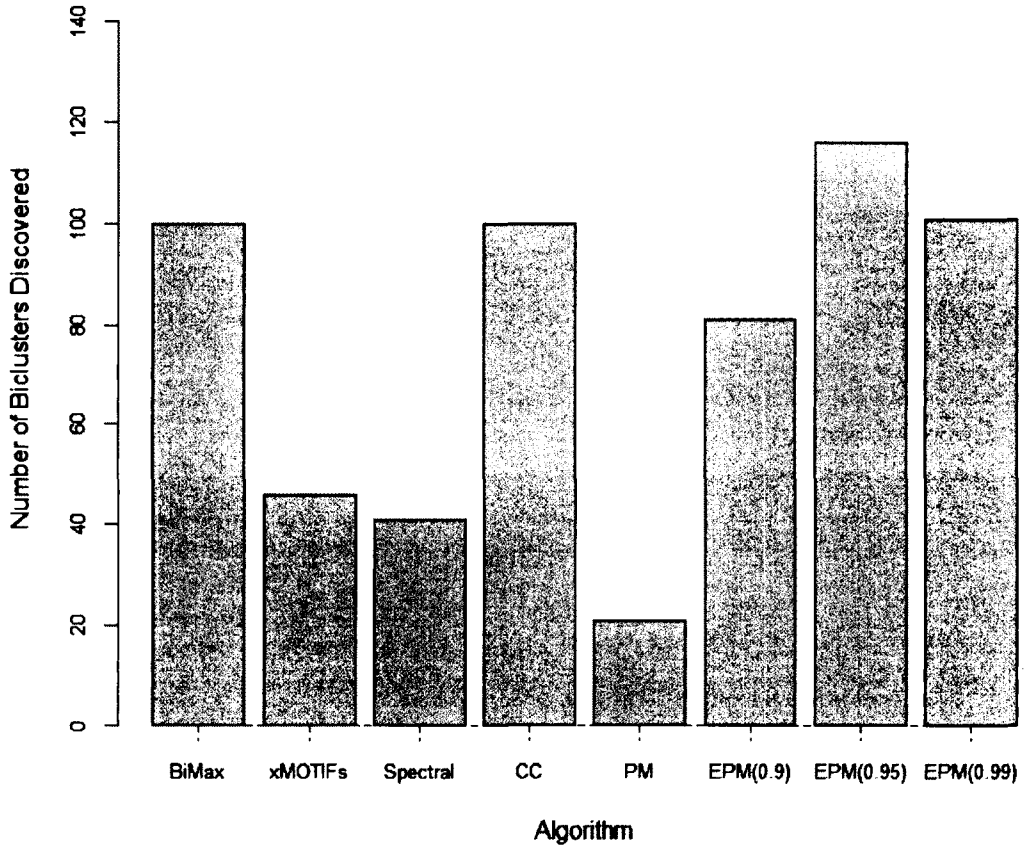


Figure 4-13: The number of biclusters discovered from the real gene expression dataset.

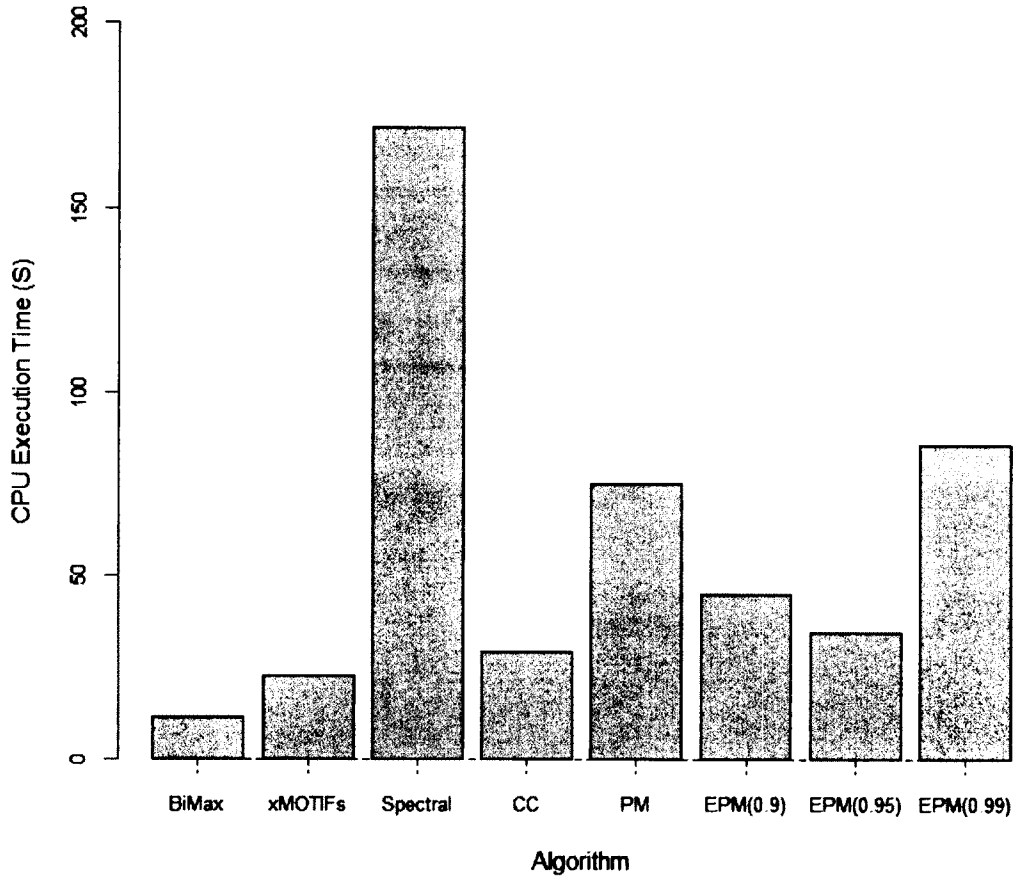


Figure 4-14: A chart showing the times taken by the individual algorithms to complete the biclustering task from the real gene expression dataset.

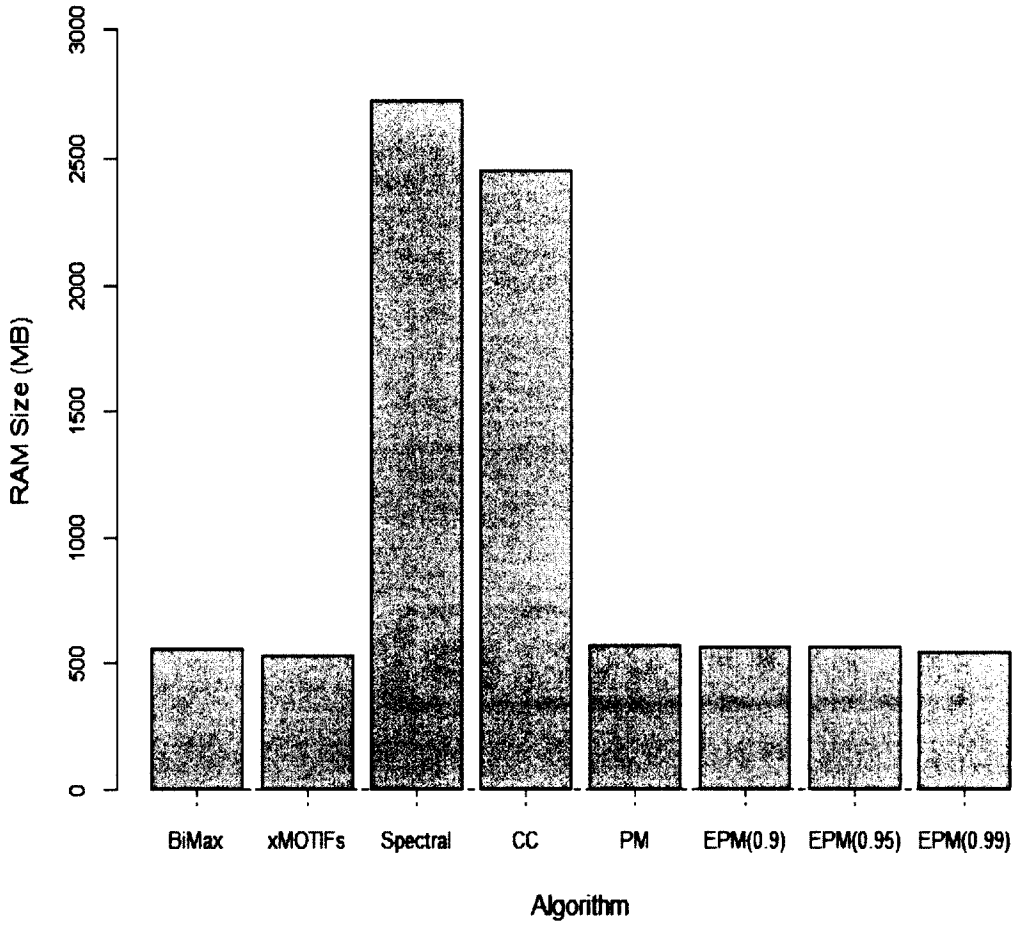


Figure 4-15: The amount of memory utilized by each algorithm in discovering biclusters from the real gene expression dataset.

Compared with the PM, the EPM with $\delta = 0.95$ discovered the most biclusters of 116 used lesser execution time of 34.92 seconds and memory size of 542.04 MB. Spectral was both the slowest and worse algorithm in memory usage.

4.4.2.1 *Bicluster Quality Experiment*

The goodness scores indicating the quality of the top 10 biclusters discovered by each algorithm from the gene expression dataset are shown in **Table 4-9**. **Figure 4-16** gives the corresponding distribution plots for each set of goodness scores per algorithm.

Table 4-9: Goodness scores for the top 10 biclusters reported by each algorithm on the real gene expression dataset. BM: BiMax, xMs: xMOTIFs, Sp: Spectral.

	Algorithm							
Bicluster	**BM**	**xMs**	**Sp**	**CC**	**PM**	**EPM δ = 0.90**	**EPM δ = 0.95**	**EPM δ = 0.99**
1	2.56	2.11	0.76	0.39	3.79	3.90	3.89	3.95
2	2.52	0.39	0.76	0.37	3.76	3.89	3.89	3.90
3	2.48	-0.36	0.76	0.36	3.56	3.89	3.89	3.89
4	2.35	-0.84	0.76	0.35	3.50	3.89	3.89	3.89
5	2.29	-1.34	0.76	0.17	3.48	3.79	3.89	3.89
6	2.28	-1.41	0.66	0.08	3.26	3.79	3.89	3.89
7	2.26	-1.51	0.66	0.06	3.24	3.79	3.89	3.89
8	2.26	-1.80	0.66	0.06	2.73	3.77	3.89	3.89
9	2.25	-1.86	0.45	0.06	2.31	3.77	3.89	3.89
10	2.24	-2.06	0.45	0.03	2.30	3.76	3.79	3.80

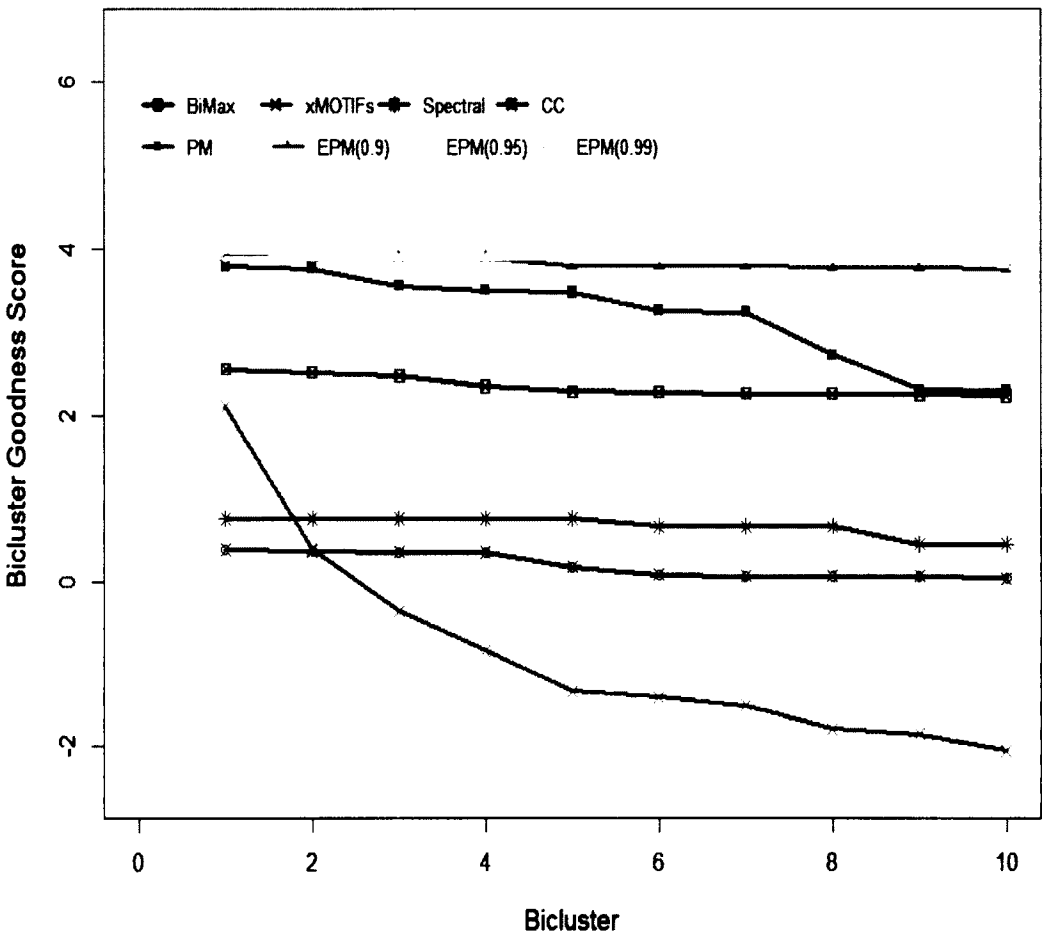


Figure 4-16: Goodness scores distribution for the top 10 biclusters by each algorithm on the gene expression dataset.

The EPM with $\delta = 0.99$ outperformed the PM and all the non-plaid algorithms in terms of goodness scores. This performance can be attributed to the EPM's ability to iteratively improve the quality of prior biclusters until convergence on the inherent biclusters. The comparatively poorer performances of the remaining approaches can be attributed to their inability to: 1) improve the already generated biclusters with substandard quality scores and 2) check plaid effects due to possible pairwise interactions

and select an adequate number of expression levels, as explained by Henriques and Madeira [67].

4.4.2.2 *GO Term Enrichment Analysis*

GO term enrichment analysis for all the three categories, namely *biological processes*, *molecular functions* and *cellular components* were reported. For the top 10 biclusters by each algorithm, the number of genes per bicluster, and the percentage of genes enriched are shown in **Table 4-10** and **Table 4-11**, respectively.

Table 4-10: The number of genes per bicluster for the top 10 biclusters by each algorithm.

	Bicluster									
Algorithm	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
BiMax	470	932	296	599	446	486	246	518	198	487
xMOTIFs	2	18	76	40	52	3598	1271	544	255	114
Spectral	7	7	7	7	7	7	7	7	7	7
CC	19	22	16	17	31	22	24	28	16	23
PM	282	493	139	206	720	2	8	60	42	486
EPM $\delta = 0.90$	479	493	493	493	367	319	319	393	305	323
EPM $\delta = 0.95$	487	487	493	493	493	493	493	493	493	365
EPM $\delta = 0.99$	536	477	489	487	493	493	493	493	494	159

Table 4-11: Percentage of genes enriched per biclusters discovered by each algorithm.

Algorithm	Bicluster									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
BiMax	94	88	96	94	93	93	95	95	93	94
xMOTIFs	100	83	91	88	94	88	91	93	91	87
Spectral	100	100	100	100	100	100	100	100	100	100
CC	84	82	75	88	87	86	75	96	88	83
PM	90	92	90	92	92	100	63	87	93	92
EPM $\delta = 0.90$	92	92	92	92	89	93	93	90	90	90
EPM $\delta = 0.95$	92	92	92	92	92	92	92	92	92	89
EPM $\delta = 0.99$	93	92	92	92	92	92	92	92	92	91

The five most enriched terms for the best bicluster per algorithm at a significance level of $\alpha = 0.05$ were reported, where RawP and AdjP represent the raw and adjusted p-values of the analysis results, respectively. **Table 4-12** and **Table 4-13** show the biological process GO term enrichment analysis results.

Table 4-12: Biological process GO terms enrichment analysis at $\alpha = 0.05$.

Algorithm	Enriched Terms (RawP/AdjP Value)
BiMax	oxidation-reduction process (5.60e-12/6.38e-09)
	small molecule metabolic process (6.82e-09/3.88e-06)
	generation of precursor metabolites and energy (4.84e-08/1.10e-05)
	small molecule biosynthetic process (3.33e-08/1.10e-05)
	single-organism biosynthetic process (4.10e-08/1.10e-05)
xMOTIFs	cell wall organization (0.0010/0.0024)
	external encapsulating structure organization (0.0010/0.0024)
	cellular cell wall organization (0.0010/0.0024)
	fungal-type cell wall organization (0.0007/0.0024)
	fungal-type cell wall organization or biogenesis (0.0009/0.0024)

Table 4-13: Biological process GO terms enrichment analysis at $\alpha = 0.05$.

Algorithm	Enriched Terms (RawP/AdjP Value)
Spectral	cytokinesis, completion of separation (1.24e-13/1.39e-11)
	cytokinetic cell separation (2.29e-12/1.28e-10)
	cytokinesis (1.53e-10/5.71e-09)
	cytokinetic process (7.16e-09/2.00e-07)
	cell division (7.95e-08/1.78e-06)
CC	None
PM	sporulation (2.35e-30/1.87e-27)
	anatomical structure formation involved in morphogenesis (1.14e-29/3.02e-27)
	sporulation resulting in formation of a cellular spore (1.10e-29/3.02e-27)
	anatomical structure morphogenesis (2.28e-28/3.63e-26)
	anatomical structure development (2.28e-28/3.63e-26)
EPM ($\delta = 0.90$)	cytoplasmic translation (3.23e-86/3.01e-83)
	translation (8.60e-31/4.00e-28)
	organic substance biosynthetic process (2.15e-25/6.67e-23)
	ribosome biogenesis (3.97e-25/9.24e-23)
	biosynthetic process (5.81e-25/1.08e-22)
EPM ($\delta = 0.95$)	cytoplasmic translation (2.82e-85/2.63e-82)
	translation (2.30e-30/1.07e-27)
	ribosome biogenesis (1.44e-24/3.36e-22)
	organic substance biosynthetic process (1.27e-24/3.36e-22)
	biosynthetic process (3.47e-24/6.48e-22)
EPM ($\delta = 0.99$)	cytoplasmic translation (1.00e-78/1.02e-75)
	translation (4.98e-24/2.53e-21)
	ribosome biogenesis (1.80e-23/6.11e-21)
	biosynthetic process (5.00e-23/1.02e-20)
	organic substance biosynthetic process (4.13e-23/1.02e-20)

Table 4-14 and **Table 4-15** show the molecular function GO terms enrichment analysis results.

Table 4-14: Molecular function GO terms enrichment analysis at $\alpha = 0.05$.

Algorithm	Enriched Terms (RawP/AdjP Value)
BiMax	oxidoreductase activity (1.22e-08/3.79e-06)
	oxidoreductase activity, acting on the CH-OH group of donors, NAD or NADP as acceptor (7.60e-08/1.18e-05)
	oxidoreductase activity, acting on CH-OH group of donors (1.46e-07/1.51e-05)
	catalytic activity (9.74e-07/7.57e-05)
	hydrogen ion transporting ATP synthase activity, rotational mechanism (7.57e-06/0.0004)
xMOTIFs	structural constituent of cell wall (0.0048/0.0144)
Spectral	hydrolase activity, hydrolyzing O-glycosyl compounds (4.07e-10/7.73e-09)
	hydrolase activity, acting on glycosyl bonds (9.20e-10/8.74e-09)
	glucosidase activity (2.51e-06/1.59e-05)
	glucan endo-1,3-beta-D-glucosidase activity (6.48e-06/3.08e-05)
	beta-glucosidase activity (2.26e-05/8.59e-05)
CC	None
PM	lysophospholipid acyltransferase activity (1.32e-05/0.0025)
	triglyceride lipase activity (8.68e-05/0.0082)
	lysophosphatidic acid acyltransferase activity (0.0003/0.0189)
	retinyl-palmitate esterase activity (0.0012/0.0459)
	chitin deacetylase activity (0.0017/0.0459)

Table 4-15: Molecular function GO terms enrichment analysis at $\alpha = 0.05$.

Algorithm	Enriched Terms (RawP/AdjP Value)
EPM ($\delta = 0.90$)	structural constituent of ribosome (2.02e-74/5.58e-72)
	structural molecule activity (4.84e-54/6.68e-52)
	rRNA binding (3.76e-12/3.46e-10)
	translation factor activity, nucleic acid binding (9.04e-09/6.24e-07)
	siderophore transporter activity (2.45e-05/0.0011)
EPM ($\delta = 0.95$)	structural constituent of ribosome (1.06e-74/2.99e-72)
	structural molecule activity (5.27e-54/7.43e-52)
	rRNA binding (5.26e-12/4.94e-10)
	translation factor activity, nucleic acid binding (1.22e-08/8.60e-07)
	siderophore transporter activity (2.63e-05/0.0012)
EPM ($\delta = 0.99$)	structural constituent of ribosome (4.92e-67/1.47e-64)
	structural molecule activity (5.36e-47/7.99e-45)
	rRNA binding (3.33e-11/3.31e-09)
	translation factor activity, nucleic acid binding (3.48e-07/2.59e-05)
	siderophore transmembrane transporter activity (3.93e-05/0.0017)

Table 4-16 and **Table 4-17** show the cellular component GO terms enrichment analysis results.

Table 4-16: Cellular component GO terms enrichment analysis at $\alpha = 0.05$.

Algorithm	Enriched Terms (RawP/AdjP Value)
BiMax	cell wall (1.05e-07/7.94e-06)
	fungal-type cell wall (3.77e-08/7.94e-06)
	external encapsulating structure (1.05e-07/7.94e-06)
	extracellular region (1.40e-06/7.95e-05)
	cytosolic small ribosomal subunit (5.49e-06/0.0002)
xMOTIFs	fungal-type cell wall (0.0002/0.0019)
	external encapsulating structure (0.0003/0.0019)
	cell wall (0.0003/0.0019)
	extracellular region (0.0002/0.0019)
	plasma membrane (0.0050/0.0260)

Table 4-17: Cellular component GO terms enrichment analysis at $\alpha = 0.05$.

Algorithm	Enriched Terms (RawP/AdjP Value)
Spectral	fungal-type cell wall (8.95e-11/9.98e-10)
	external encapsulating structure (1.21e-10/9.98e-10)
	cell wall (1.21e-10/9.98e-10)
	extracellular region (6.92e-11/9.98e-10)
	cell septum (3.24e-06/2.14e-05)
CC	intrinsic to Golgi membrane (0.0016/0.0328)
	integral to Golgi membrane (0.0016/0.0328)
PM	intracellular immature spore (4.27e-11/2.42e-09)
	prospore membrane (4.27e-11/2.42e-09)
	ascospore-type prospore (4.27e-11/2.42e-09)
	spore wall (1.13e-09/4.80e-08)
	ascospore wall (1.71e-08/5.81e-07)
EPM ($\delta = 0.90$)	cytosolic ribosome (1.84e-86/3.66e-84)
	ribosomal subunit (7.58e-75/7.54e-73)
	ribosome (1.28e-72/8.49e-71)
	cytosolic part (8.18e-68/4.07e-66)
	ribonucleoprotein complex (9.19e-60/3.66e-58)
EPM ($\delta = 0.95$)	cytosolic ribosome (1.64e-85/3.31e-83)
	ribosomal subunit (4.42e-75/4.46e-73)
	ribosome (1.60e-72/1.08e-70)
	cytosolic part (6.97e-67/3.52e-65)
	ribonucleoprotein complex (3.23e-59/1.30e-57)
EPM ($\delta = 0.99$)	cytosolic ribosome (6.43e-79/1.35e-76)
	ribosomal subunit (3.62e-66/3.80e-64)
	ribosome (3.23e-62/2.26e-60)
	cytosolic part (1.03e-60/5.41e-59)
	ribonucleoprotein complex (2.53e-50/1.06e-48)

Compared with the PM algorithm, the EPM had more genes enriched for biclusters 1, 3, 4, 5, 7, 8 and 10 out of the ten cases reported, while the PM was better in biclusters 2, 6 and 9. Although Spectral had every gene discovered enriched for the top 10 biclusters, it also recorded the least number of genes per biclusters across the board, as shown in **Table 4-10**. Similar small-sized bicluster effect on gene enrichment analysis can be observed in bicluster 1 of xMOTIFs with 2 genes and bicluster 6 of the PM also

with 2 genes where both algorithms recorded 100% genes enrichment per bicluster. Results for additional enrichment analysis at two significance levels of $\alpha = 0.02$ and $\alpha = 0.01$ are shown in the APPENDIX.

4.5 Conclusion

This chapter proposes and presents an enhanced Plaid Model technique to generate high quality biclusters in a given numerical dataset. The proposed approach aims at addressing the problem of generating and selecting the best set of biclusters hidden in a given dataset, a scenario that is difficult to achieve under the current implementation of the PM. The EPM algorithm iteratively combines and refines several outputs from the PM to generate a list of statistically significant biclusters of higher differential co-expression based goodness scores.

Extensive comparison between the EPM and five state-of-the-art biclustering algorithms on both synthetic and real gene expression datasets was conducted. The results on the real gene expression dataset indicate that the EPM outperformed the current implementation of the PM algorithm on the number and quality of biclusters discovered, execution time and the amount of memory used. The EPM also outperformed all the other four non-plaid algorithms on the real gene expression dataset in discovering more biclusters of higher quality in terms of goodness scores. All the top 10 biclusters discovered by the EPM were GO enriched at three different levels of significance. On the artificial datasets, the EPM indicated a comparable performance with the PM.

CHAPTER 5

SUBSPACE FEATURE TRACKING BASED ON RELEVANCE IN SPATIOTEMPORAL DATASETS

Many data modeling and object tracking algorithms rely on effective use of the available features upon which the data were collected. Most algorithms tend to use only a subset of features after subjecting the available set of feature to relevance analysis. Selecting the most relevant features in data modeling is critical to ensure higher accuracy in predictive analysis, model reliability and the realization of an optimal overall model performance. Relevance based feature selection becomes compounded in situations where feature relevance is not guaranteed to remain constant over time due to changes in the underlying dataset. For instance, in order to track subspaces of features that define space-time paths in a large spatiotemporal dataset, Shaw *et al.* [8] proposed a technique that identifies spatial cluster centers of individual events at different time periods and then connect them according to their temporal sequences. In this chapter, we focus on a technique that uses biclustering as a relevance-based feature selection method where the set of features constituting the biclustering criteria are selected and tracked in datasets that change with time.

5.1 Research Motivation

Relevance based feature selection is at the core of many predictive algorithms to ensure effective and accurate prediction of both current and future events. However, the

selection process becomes a challenge in situations where the effectiveness of previously selected features cannot be guaranteed due to changes in the underlying dataset. This chapter presents a proposed technique based on the enhanced Plaid Model EPM for the discovery and tracking of feature relevance scores in datasets that undergo changes with time. Initially, the algorithm discovers a set of biclusters with the EPM based on plaid assumptions, and then selects sets of features that represent the biclustering criteria as candidates whose relevance scores are computed and tracked over time.

5.1.1 Problem Statement

Given a set of P feature relevance scores $\{R_{\beta_j}\} = \{R_{\beta_1}, R_{\beta_2}, \cdots, R_{\beta_P}\}$, associated with the k^{th} bicluster $B_{k,T}$ at time T in a given data matrix $\mathbb{A}^{(T)}$, the goal is to model changes in $\{R_{\beta_j}\}$ as $\mathbb{A}^{(T)}$ undergoes spatiotemporal changes with time.

5.2 Methodology

Initially, at time $T = 1$, an exhaustive set of biclusters are generated by running the EPM on the initial dataset. The set of relevance scores per feature sets per individual biclusters that indicate their within-bicluster discriminatory powers are then computed and marked for tracking. In the subsequent phases at times $T > 1$, the EPM is run with an updated version of the initial dataset to generate a new set of biclusters that commensurate the current structural configuration of the dataset. These are then compared with the previous set of biclusters via the similarity analytics engine of the EPM algorithm, as explained under the methodology subsection of Chapter 4, to generate a newly updated list of biclusters whose feature relevance scores are then marked for tracking.

5.2.1 The Proposed Model

Figure 5-1 shows the proposed model for the generation, selection and tracking of feature relevance in a given changing dataset.

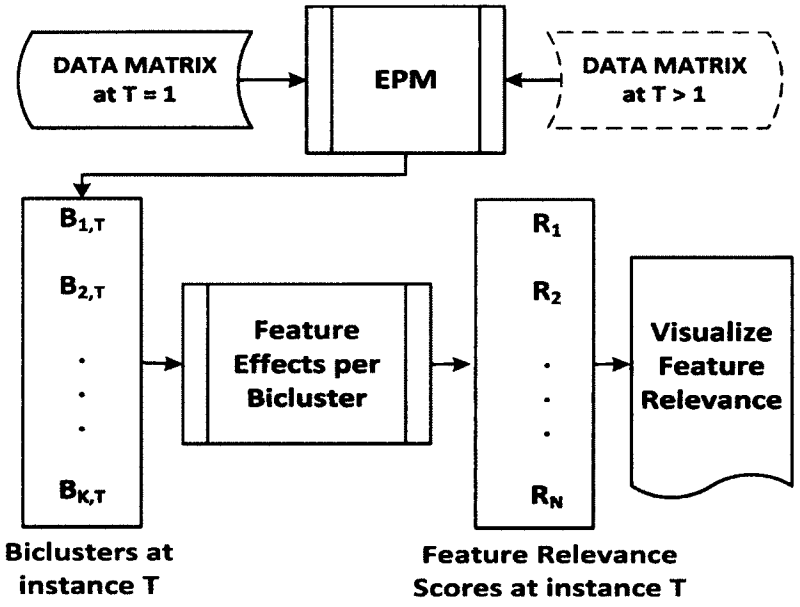


Figure 5-1: The proposed feature relevance tracking model. **T**: time instance, **EPM**: the enhanced Plaid Model, $\mathbf{B_{K,T}}$: bicluster k generated at time instance T, $\mathbf{R_N}$: relevance score for the N^{th} feature.

Figure 5-2 outlines the algorithmic steps involved in the process of generating a set of features whose relevance scores are to be tracked as the underlying dataset changes with time.

Algorithm: Feature relevance tracking
Input: Initial and subsequently updated datasets, similarity threshold, δ
Output: Feature relevance scores at time *T*

i. Initial phase:

1. At time $T = 1$, generate a set of biclusters with the EPM on the initial dataset
2. Select the set of features per biclusters to be tracked
3. For each feature *j* per bicluster *k*, compute the feature effect, β_{jk} it exerts on the bicluster
4. For each feature, compute the corresponding feature relevance per bicluster at current time T
5. Visualize the feature relevance scores at current time T

ii. Subsequent phases:

6. At time T > 1, generate a set of biclusters with the EPM on the most currently updated dataset
7. Compare the current set of biclusters generated at T with the previous set at (T – 1) and select biclusters with similarity scores greater than or equal to the similarity threshold, δ
8. Go To step 2
9. Stop

Figure 5-2: Feature relevance generation and tracking algorithm.

The initial phase of the algorithm generates the sets of features whose relevance scores are updated and tracked over time in the subsequent phases.

5.2.2 Computation of Feature Relevance Scores

The individual feature relevance scores are computed with respect to the hosting bicluster to indicate their influence in terms of the discriminatory effect they exert on the bicluster in relation to the other features in the same bicluster. Knowing the individual feature effects, as computed by **Eq. 3-29**, then for a set of feature effects $\{\beta_1, \beta_2, \ldots, \beta_p\}$ associated with any given bicluster, the corresponding individual relevance scores $R_{\beta j}$ are given by **Eq. 5-1**.

$$R_{\beta_j} = \frac{\beta_j}{\sum_{j=1}^{p} |\beta_j|}. \qquad \textbf{Eq. 5-1}$$

5.2.3 Feature Relevance Tracking

At time T, the biclustering algorithm discovers a set of biclusters, each of which is associated with a set of features with individual relevance scores. When the dataset undergoes changes at a different time instance, say (T + 1), due to either the removal of existing records or the introduction of new records, the inherent biclusters tend to undergo changes to reflect the effects introduced by any new features and/or records membership. Depending on the degree of change experienced by the current state of the dataset, a set of features that defined a bicluster previously and are still together in a current bicluster might not exert the same discriminatory effects they commanded previously. Hence, tracking a set of features that defines a common bicluster over time gives us the ability to track their relevance scores as the underlying dataset changes.

5.3 Experiment and Results

This section presents details of the dataset used to assess the performance of the proposed algorithm, parameter settings and the results obtained.

5.3.1 Dataset and Parameter Settings

The ability of the proposed algorithm to accurately and successfully identify sets of features and track their relevance over time as the underlying dataset changes was tested on the real Saccharomyces cerevisiae gene expression dataset, EisenYeast [76, 77]. The dataset is a microarray data matrix with information about levels of 6,221 genes over 80 conditions. The EPM was implemented in R [74] where the similarity threshold parameter of the proposed model δ was set to 0.95 after averaging out initial training tests of twenty repeated experiments, each at $\delta = 0.90, 0.95$ and 0.99.

5.3.2 Experiments

In order to simulate the phenomenon of a changing dataset using the gene expression dataset, the following approach was employed: initially, at time T = 1, the EPM was executed with the first 500 genes and the entire 80 features. The set of features per biclusters discovered as a result formed the initially selected sets of features whose relevance scores are to be tracked. In the next steps, the algorithm sequentially adds 300 genes at the various subsequent time instances until a total of 6,200 genes per dataset was reached at a time instance of T = 20. This allowed us to track and observe subsets of 80 features, also known as experimental conditions in the case of gene expression data analysis, as we introduce additional genes at different time instances of T = 1, 2, ..., 19, 20.

5.3.3 Results

The results of the topmost bicluster discovered in the experiment are reported in this section. All the biclusters discovered in the work were evaluated and ranked for bicluster quality based on the bicluster goodness score procedure developed by Chia and Karuturi [2], and implemented in the biclust package in R [76]. Out of the 20 time instances considered in the experiment, the proposed algorithm successfully tracked the topmost bicluster's features relevance scores of 12.

Initially, the topmost bicluster contained the six features *Sporulation_5h*, *Sporulation_7h*, *Sporulation_9h*, *Sporulation_11h*, *Sporulation_7h_(v._5h)*, *Sporulation_ndt80over*. As the dataset changed with time, four among them persisted together at nine time instances, and three persisted together at all 12 successfully tracked instances. The results of their tracked relevance scores are shown in **Table 5-1**.

Table 5-1: Relevance scores for features from the topmost bicluster per dataset at time instance T. **T**: Time instance; **#**: Number of genes; **S_5h**: Sporulation_5h; **S_7h**: Sporulation_7h; **S_9h**: Sporulation_9h; **S_11h**: Sporulation_11h.

		Feature Relevance Score			
T	**#**	**S_5h**	**S_7h**	**S_9h**	**S_11h**
1	500	0.1632	0.1820	0.2031	0.2145
2	800	0.2154	0.2327	0.2516	0.2494
4	1400	0.2250	0.2281	0.2595	0.2367
5	1700	0.2218	0.2357	0.2702	0.2466
6	2000	0.1676	0.1702	0.2050	0.1891
8	2600	0.0259	0.0186	0.2503	-
10	3200	0.2306	0.2132	0.2619	0.2533
11	3500	0.0489	0.0732	0.0276	-
14	4400	0.0735	0.2130	0.0001	-
15	4700	0.2442	0.2154	0.2642	0.2502
16	5000	0.1423	0.0017	0.2461	0.1811
17	5300	0.2470	0.2127	0.2717	0.2534

At different time instances, the feature relevance scores reflected the changing underlying datasets, and these are indicated by the changes in their scores, and hence, their varying discriminatory effects exerted on the hosting biclusters within which they are located. The results are presented with relevance scores distribution charts where the length of the individual bars in the charts indicates the relevance score of the feature concerned. **Figure 5-3** shows the results for time instances T = 1 and 2; **Figure 5-4** shows the outcome for time instances T = 4 and 5; **Figure 5-5** gives the results for time instances T = 6 and 8; **Figure 5-6** shows the results for time instances T = 10 and 11; **Figure 5-7** shows the results for time instances T = 14 and 15; and **Figure 5-8** shows the outcome for time instances T = 16 and 17.

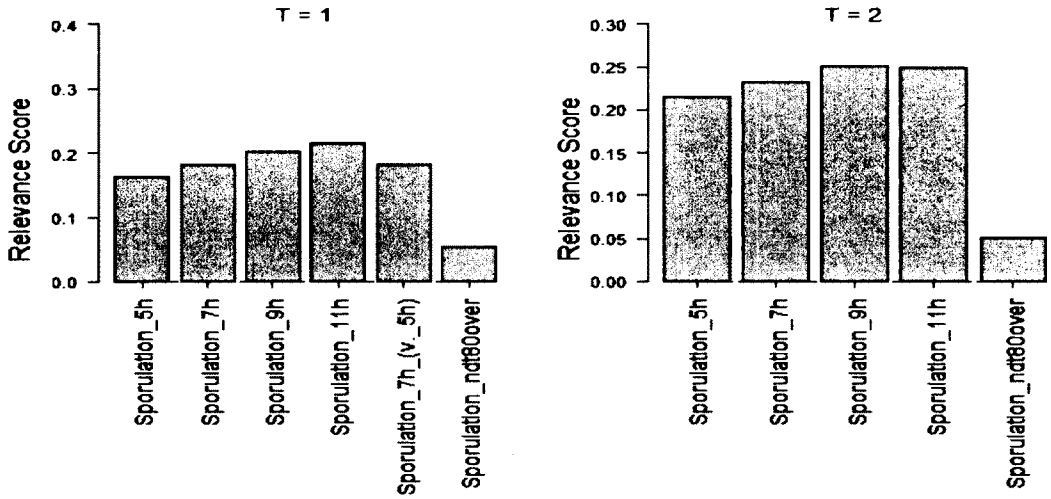


Figure 5-3: Feature relevance distribution charts for time instances T = 1, 2.

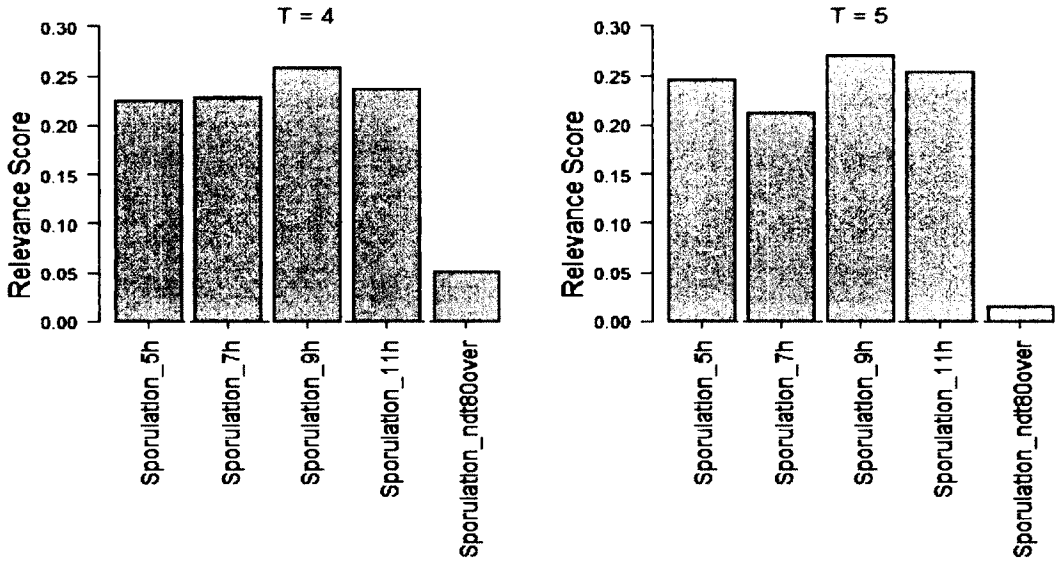


Figure 5-4: Feature relevance distribution charts for time instances T = 4, 5.

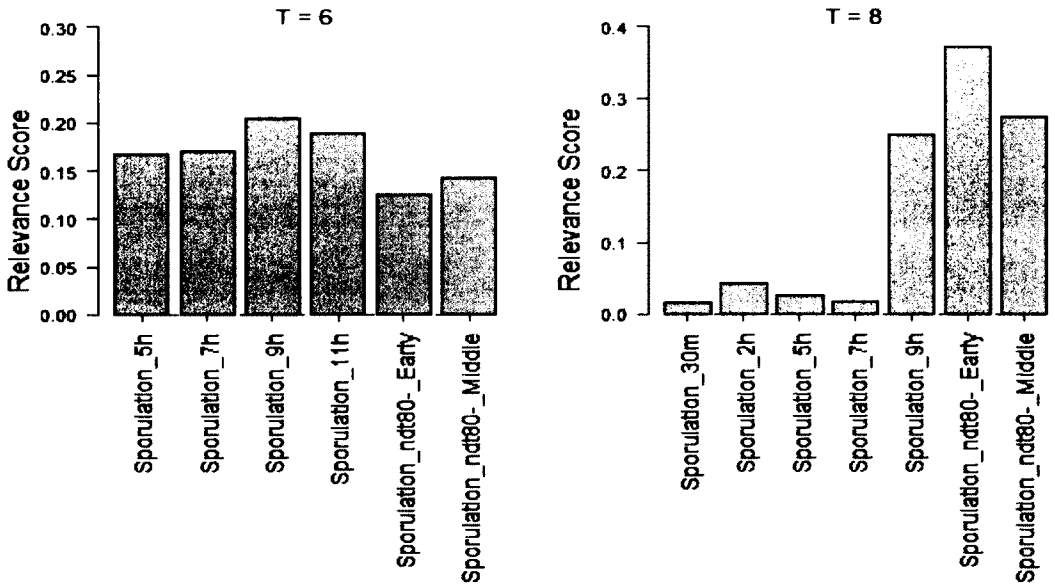


Figure 5-5: Feature relevance distribution charts for time instances T = 6, 8.

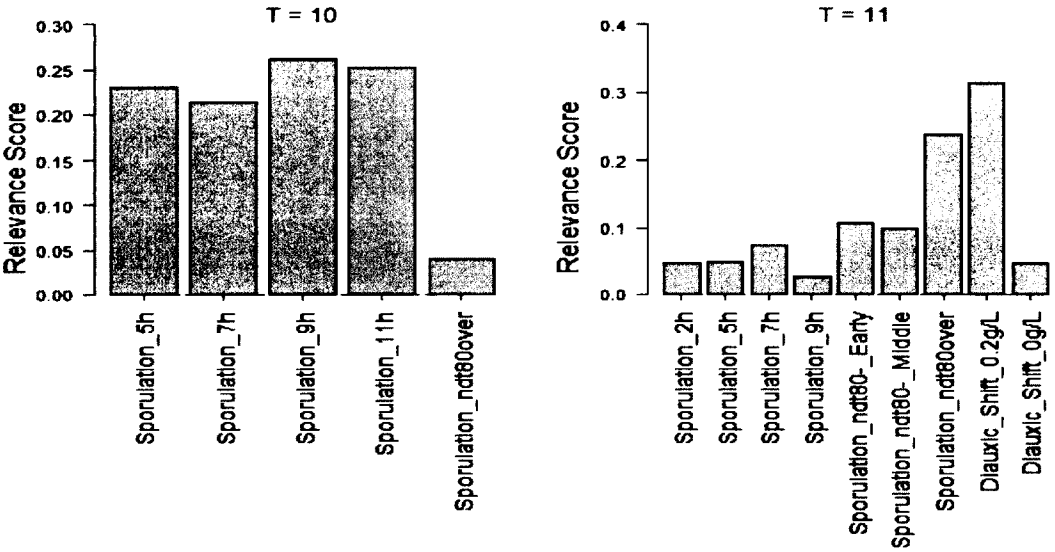


Figure 5-6: Feature relevance distribution charts for time instances T = 10, 11.

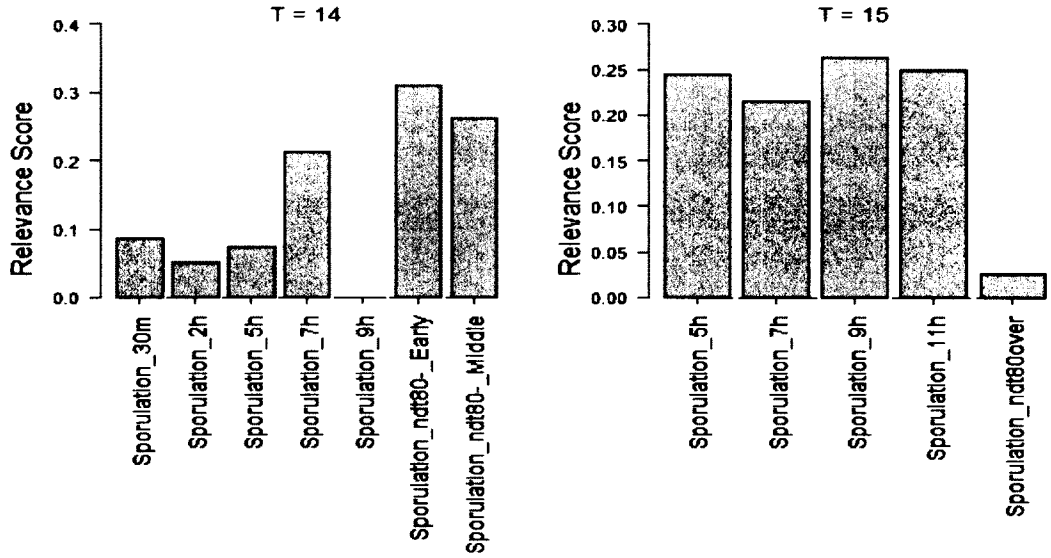


Figure 5-7: Feature relevance distribution charts for time instances T = 14, 15.

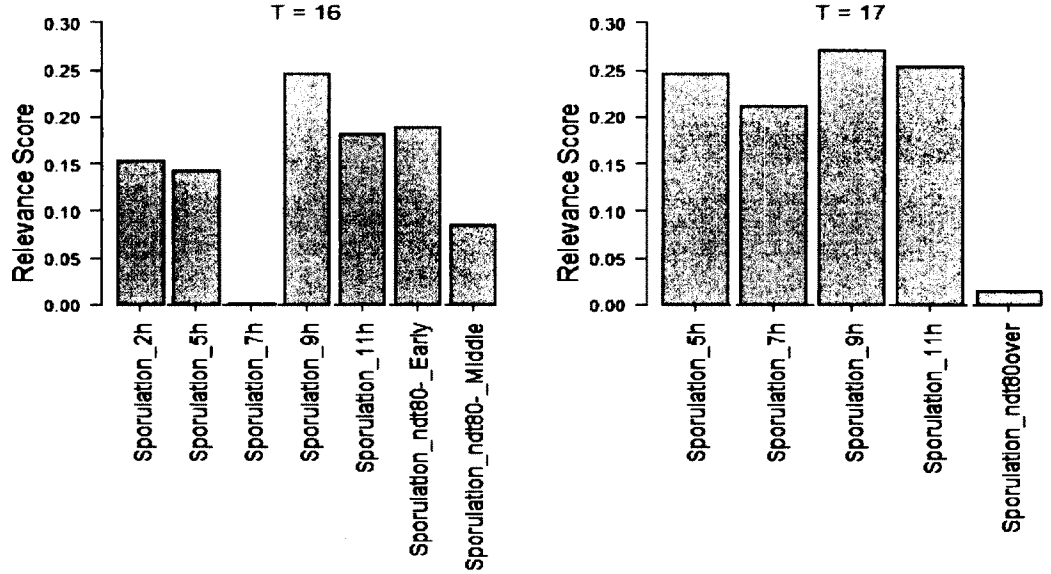


Figure 5-8: Feature relevance distribution charts for time instances T = 16, 17.

As shown in **Figure 5-9**, the behavior patterns of the three features *Sporulation_5h*, *Sporulation_7h* and *Sporulation_9h* indicate that as they transition from bicluster to bicluster due to changes in the underlying dataset, they experience fluctuations in their relevance scores.

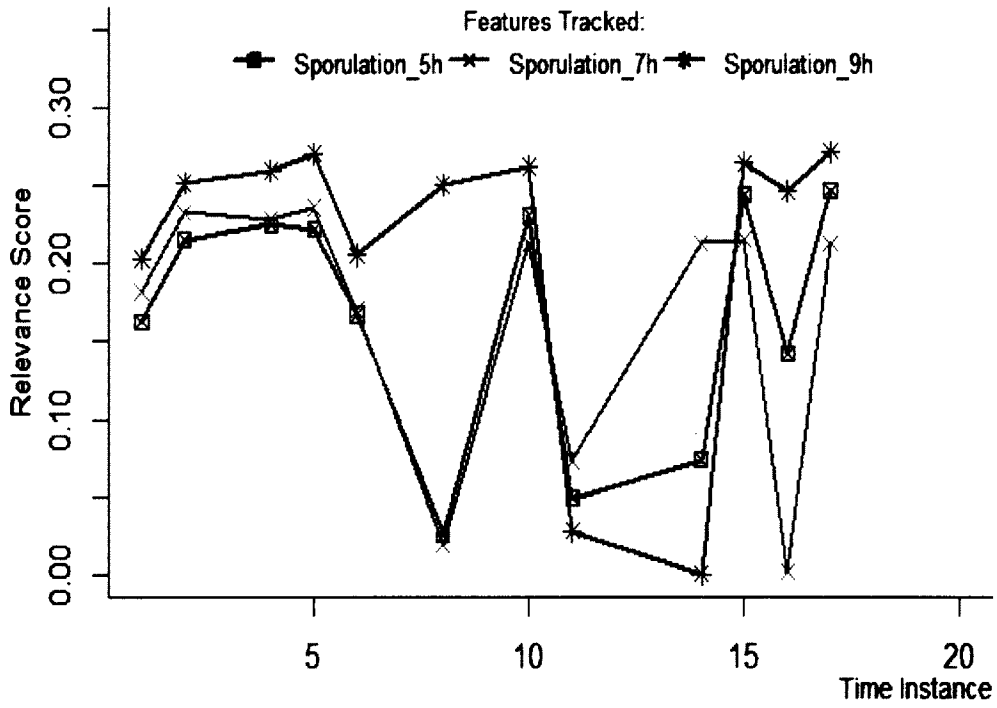


Figure 5-9: Feature relevance distribution plots showing the trend lines of feature relevance as the underlying dataset changes with time.

This is an indication that they tend to exert different discriminatory powers within those biclusters that contain them. It is also evident from the results that as the underlying dataset changes with time, an existing feature in a bicluster could lose so much discriminatory power that it might leave the bicluster entirely. This was the case with the

feature *Sporulation_7h_(v._5h)* that had an initial relevance score of 0.128 at time instance T = 1, but was never again observed in subsequent instances.

We can also observe from **Figure 5-9** that changes introduced into the underlying datasets either strengthen or weaken the discriminatory powers of those features that define the host bicluster.

5.4 Conclusion

This chapter proposes an algorithm that uses the EPM to discover high quality biclusters for the generation and selection of feature sets that can be marked for tracking. This is based on their bicluster-specific feature relevance discriminatory characteristics in datasets that undergo changes with time. This is useful in assigning accurate weights to variables that are utilized in predictive models for cluster and bicluster analysis involving spatiotemporal datasets. The algorithm was tested on real gene expression dataset, and the results indicate that it was able to successfully track subsets of features based on their relevance scores that defined those biclusters that host them over a span of time instances, as the dataset changed.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

The research presented in this dissertation is aimed at developing techniques for selecting and tracking subsets of features from spatiotemporal datasets based on their discriminatory characteristics. The problem was formulated to be the selection of subsets of features whose changing discriminatory characteristics are tracked with time in a changing dataset. The initial phase of the proposed approach involved the use of an enhanced Plaid Model technique to integrate multiple outputs from the traditional statistical Plaid Model to generate a list of statistically significant biclusters. This approach recursively combined a series of set operations and statistical inferential tests to generate biclusters of high quality in goodness scores. Following this, the sets of features that define these biclusters were selected and marked for tracking based on the discriminatory powers they exert on the host biclusters at different times as the dataset changes. Subsequently, these changes in discriminatory powers among the sets of features that define the host biclusters were modeled for tracking as the underlying dataset changed. Some specific contributions by this dissertation are presented in the following subsections.

6.1 Contribution to Bicluster Analysis

The work in this dissertation proposed the use of biclustering technique as means of selecting relevant features from a given dataset for the purpose of feature tracking in changing spatiotemporal datasets. The statistical Plaid Model (PM) was adopted as the biclustering technique for generating statistically significant biclusters whose features can be selected for tracking purposes. The challenge of using the PM, however, was the non-deterministic nature of its output where different runs of the PM on the same given dataset resulted in different sets of biclusters. This is due to the NP-complete nature of the biclustering problem formulation. Against this backdrop, this work proposed an enhanced Plaid Model (EPM) approach where the recursive use of combined set operations and statistical inferential tests were utilized to improve the quality of biclusters generated. Extensive experimental results on both synthetic and real datasets reported in the work shows the viability and effectiveness of the proposed EPM algorithm in generating reliable and more stable biclusters of higher quality. The results also show that the EPM is scalable, tractable and efficient in memory usage in discovering high quality biclusters from both synthetic and real datasets, and biologically significant biclusters from a real gene expression dataset.

6.2 Contribution to Feature Subspace Tracking

One of the core challenges in predictive modeling is feature selection for optimal performance. It becomes a model performance challenge when machine learning models are built with features whose relevance cannot be guaranteed due to changes in the underlying dataset. This work proposed a technique to track subsets of features by mining the relevance based discriminatory characteristics of sets of features in datasets that

undergo changes with time. The algorithm uses the proposed EPM to generate sets of statistically significant biclusters from which features are marked for tracking based on their discriminatory powers exerted on the host biclusters at any point in time. As the underlying dataset changes, the originally discovered biclusters also change together with the biclustering criteria which are controlled by the discriminatory tendencies of the respective sets of features per biclusters. The proposed technique was tested on real microarray gene expression dataset. The results show that it was able to track subsets of features successfully via their relevance based discriminatory characteristics over time as the dataset changed.

6.3 Future Work

The work presented in this dissertation has triggered some research ideas that could further be explored in the near future. First is the possibility of exploring the use of relevance scores of individual records in a dataset, instead of features, or an integration of both to effectively track subspaces of events that undergo spatiotemporal changes. Such a work is envisioned to generate and rely on more comprehensive information content in making decisions regarding subspace tracking. Next is the use of relevance scores of both features and records to build predictive models to forecast future subspace events.

REFERENCES

[1] P. Carmona-Saez, M. Chagoyen, A. Rodriguez, O. Trelles, J. M. Carazo and A. Pascual-Montano, "Integrated analysis of gene expression by association rules discovery," *BMC Bioinformatics,* vol. 7, no. 1, p. 54, 2006.

[2] B. K. H. Chia and R. K. M. Karuturi, "Differential co-expression framework to quantify goodness of biclusters and compare biclustering algorithms," *Algorithms for Molecular Biology,* vol. 5, no. 23, 08 07 2010.

[3] D. J. Hand, H. Mannila and P. Smyth, Principles of data mining, Cambridge: MIT press, 2001.

[4] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques, 3rd ed., Waltham, MA: Morgan Kaufmann Publishers, 2012.

[5] U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Communications of the ACM,* vol. 39, no. 11, pp. 27-34, 1996.

[6] A. Shejale and V. Gnagawane, "An implementation of efficient techniques for tree based mining in human social dynamics," in *IEEE International Conference on Data Mining and Advanced Computing (SAPIENCE)*, 2016.

[7] S. Zhang, C. Zhang and Q. Yang, "Data preparation for data mining.," *Applied Artificial Intelligence,* vol. 17, no. 5-6, pp. 375-381, 2003.

[8] S. L. Shaw, H. Yu and L. S. Bombom, "A space-time GIS approach to exploring large individual-based spatiotemporal datasets," *Transactions in GIS,* vol. 12, no. 4, pp. 425-441, 2008.

[9] V. Bolón-Canedo, N. Sánchez-Maroño and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowledge and information systems,* vol. 34, no. 3, pp. 483-519, 2013.

[10] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research,* vol. 3, no. Mar, pp. 1157-1182, 2003.

[11] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence,* vol. 97, no. 1-2, pp. 273-324, 1997.

[12] E. E. Bron, M. Smits, W. J. Niessen and S. Klein, "Feature selection based on the SVM weight vector for classification of dementia," *IEEE Journal of biomedical and health informatics,* vol. 19, no. 5, pp. 1617-1626, 2015.

[13] F. Falahati, E. Westman and A. Simmons, "Multivariate data analysis and machine learning in Alzheimer's disease with a focus on structural magnetic resonance imaging," *Journal of Alzheimer's Disease,* vol. 14, no. 3, pp. 685-708, 2014.

[14] G. H. John, R. Kohavi and K. Pfleger, "Irrelevant features and the subset selection problem," in *Eleventh international conference on machine learning*, New Brunswick, NJ, 1994.

[15] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science,* vol. 209, no. 1-2, pp. 237-260, 1998.

[16] H. B. Saber amd M. Elloumi, "A novel biclustering algorithm of binary microarray data: BiBinCons and BiBinAlter," *BioData Mining,* vol. 8, p. 38, 2015.

[17] P. H. Prathibha and C. P. Chandran, "Feature selection for Mining SNP from Leukaemia Cancer using Genetic Algorithm with BCO," in *IEEE International Conference on Data Mining and Advanced Computing (SAPIENCE)*, 2016.

[18] S. Dua and P. Chowriappa, Data Mining for Bioinformatics, 1st ed., Boca Raton, FL: CRC Press, Taylor & Francis Group, 2013.

[19] A. Yilmaz, O. Javed and M Shah, "Object Tracking: A Survey," *ACM Computing Surveys,* vol. 38, no. 4, 2006.

[20] V. Salari and I. K. Sethi, "Feature point correspondence in the presence of occlusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 12, no. 1, pp. 87-91, 1990.

[21] C. J. Veenman, M. J. Reinders and E. Backer, "Resolving motion correspondence for densely moving points," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, no. 1, pp. 54-72, 2001.

[22] T. J. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *IEEE transactions on pattern analysis and machine intelligence,* no. 1, pp. 90-99, 1986.

[23] D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking," *IEEE Transactions on pattern analysis and machine intelligence,* vol. 25, no. 5, pp. 564-577, 2003.

[24] H. Tao, H. S. Sawhney and R. Kumar, "Object tracking with bayesian estimation of dynamic layer representations," *IEEE transactions on pattern analysis and machine intelligence,* vol. 24, no. 1, pp. 75-89, 2002.

[25] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *International Journal of Computer Vision,* vol. 29, no. 1, pp. 5-28, 1998.

[26] M. Bertalmio, G. Sapiro and G. Randall, "Morphing active contours," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, no. 7, pp. 733-737, 2000.

[27] R. Ronfard, "Region-based strategies for active contour models," *International journal of computer vision,* vol. 13, no. 2, pp. 229-251, 1994.

[28] X, Wang, K. T. Ma, G. W. Ng and W. E. L. Grimson, "Trajectory analysis and semantic region modeling using a nonparametric bayesian model," in *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008.

[29] M. D. Breitenstein, H. Grabner and L. Van Gool, "Hunting nessie-real-time abnormality detection from webcams," in *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, Kyoto, Japan, 2009.

[30] J. He, L. Balzano and J. Lui, "Online robust subspace tracking from partial information," *arXiv:1109.3827,* 2011.

[31] C. Beleznai, A.N. Belbachir and P. Roth, "Density-based rare event detection from streams of neuromorphic sensor data," in *Sixth ACM/IEEE International Conference on Distributed Smart Cameras*, Hong Kong, 2012.

[32] R. Lima de Carvalho, D. S. C. Carvalho, F. Mora-Camino, P. V. M. Lima and F. M. G. Franca, "Online tracking of multiple objects using," in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, 2014.

[33] F. Pernici, A. Del Bimbo, "Object tracking by oversampling local features," *IEEE transactions on pattern analysis and machine intelligence,* vol. 36, no. 12, pp. 2538-2551, 2012.

[34] G. Zhu, J. Wang, Y. Wu and H. Lu, "Collaborative correlation tracking," in *British Machine Vision Conference*, Swansea, 2015.

[35] Y. Sui, S. Zhang and L. Zhang, "Robust Visual Tracking via Sparsity-Induced Subspace Learning," *IEEE Transactions on Image Processing,* vol. 24, no. 12, pp. 4686-4700, 2015.

[36] J. Liu, Y. Lu and T. Zhou, "Instance significance guided multiple instance boosting for robust visual tracking," in *IEEE International Conference on Image Processing,* 2016.

[37] S. Moujtahid, S. Duffner and A. Baskurt, "Coherent Selection of Independent Trackers for Real-time Object Tracking," in *International Conference on Computer Vision Theory and Applications* , Berlin, 2015.

[38] H. Almuallim and T. G. Dietterich, "Almuallim, H. and DietteriLearning With Many Irrelevant Features," *Association for the Advancement of Artificial Intelligence (AAAI),* vol. 91, pp. 547-552, 1991.

[39] J. H. Gennari, P. Langley and D. Fisher, "Models of incremental concept formation," *Artificial intelligence,* vol. 40, no. 1-3, pp. 11-61, 1989.

[40] B. Pontes, R. Giraldez and J. S. Aguilar-Ruiz, "Biclustering on expression data: a review," *Journal of Biomedical Informatics,* vol. 57, pp. 163-180, 2015.

[41] X. Gan, A. W-C. Liew, H. Yan, "Discovering biclusters in gene expression data based on high-dimensional linear geometries," *BMC Bioinformatics,* vol. 9, no. 209, 2008.

[42] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB),* vol. 1, no. 1, pp. 24 - 45, 2004.

[43] A. Zimek, "Correlation clustering," *ACM SIGKDD Explorations Newsletter,* vol. 11, no. 1, pp. 53 - 54, 2009.

[44] J. A. Hartigan, "Direct clustering of a data matrix," *Journal of the American Statistical Association,* vol. 67, no. 337, pp. 123-129, 1972.

[45] K. Eren, M. Deveci, O. Kucuktunc and U. V. Catalyurek, "A comparative analysis of biclustering algorithms for gene expression data," *BRIEFINGS IN BIOINFORMATICS,* vol. 14, no. 3, pp. 279-292, 2013.

[46] A. Oghabian, S. Kilpinen, S. Hautaniemi and E. Czeizler, "Biclustering methods: biological relevance and application in gene expression analysis," *PloS one,* vol. 9, no. 3, 2014.

[47] V.A. Padilha and R.J. Campello, "A systematic comparative evaluation of biclustering techniques," *BMC Bioinformatics,* vol. 18, no. 1, p. 55, 2017.

[48] H. L. Turner, T. C. Bailey, W. J. Krzanowski and C. A. Hemingway, "Biclustering Models for Structured Microarray Data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB),* vol. 2, no. 4, pp. 316-329, 2005.

[49] L. Lazzeroni and A. Owen, "Plaid Models for Gene Expression Data," *Statistica Sinica,* vol. 12, pp. 62-86, 2002.

[50] A. Tanay, R. Sharan and R. Shamir, "Discovering statistically significant biclusters in gene expression data," *Bioinformatics,* vol. 18, pp. SI36-S144, 2002.

[51] Y. Cheng and G. M. Church, "Biclustering of expression data," in *The 8th International Conference on Intelligent Systems for Molecular Biology*, La Jolla, CA, 2000.

[52] J. Yang, W. Wang, H. Wang and P. Yu, "δ-Clusters: capturing subspace correlation in a large data set," *Proc. 18th IEEE International Conference on Data Engineering,* pp. 517 - 528, 2002.

[53] A. Ben-Dor, B. Chor, R. Karp and Z. Yakhini, "Discovering local structure in gene expression data: the order-preserving submatrix problem," *Journal of Computational Biology,* vol. 10, no. 3 - 4, pp. 373 - 384, 2003.

[54] R. Henriques and S. C. Madeira, "BicSPAM: flexible biclustering using sequential patterns," *BMC Bioinformatics,* vol. 15, p. 130, 2014.

[55] S. Bergmann, J. Ihmels and N. Barkai, "Iterative signature algorithm for the analysis of large-scale gene expression data," *Physical Review E, vol. 67, no. , p. , 2003. ,* vol. 67, no. 3 Pt 1, p. 031902, 2003.

[56] Y. Kluger, R. Basri, J. T. Chang and M. Gerstein, "Spectral biclustering of microarray data: coclustering genes and conditions," *Genome Research,* vol. 13, no. 4, pp. 703-716, 2003.

[57] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele and E. Zitzler, "A systematic comparison and evaluation of biclustering methods for gene expression data," *A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. Buhlmann, W. Gruissem, L. Hennig, L. Thiele and E. Zitzler, "A systematic comparison and evaluation ofBioinformatics,* vol. 22, no. 9, pp. 1122-1129, 2006.

[58] T. M. Murali and S. Kasif, "Extracting conserved gene expression motifs from gene expression data," *Proceedings of Pacific Symposium of Biocomputing,* vol. 8, pp. 77 - 88, 2003.

[59] J. Gu and J. S. Liu, "Bayesian biclustering of gene expression data," *BMC Genomics,* vol. 9, no. (Suppl 1):S4, 2008.

[60] C. Huttenhower, K. T. Mutungu, N. Indik, W. Yang, M. Schroeder, J. J. Forman, O. G. Troyanskaya and H. A. Coller, "Detailing regulatory networks through large scale data integration," *Bioinformatics,* vol. 25, no. 24, p. 3267–3274, 2009.

[61] D. Bozdag, J. D. Parvin and U. V. Catalyurek, "A biclustering method to discover co-regulated genes using diverse gene expression datasets," in *Proceedings of the 1st International Conference on Bioinformaticsand Computational Biology*, Berlin, Heidelberg, 2009.

[62] G. Li , Q. Ma, H. Tang, A. H. Paterson and Y. Xu, "QUBIC: a qualitative biclustering algorithm for analyses of gene expression data," *Nucleic Acids Research,* vol. 37, no. (15):e 101, 2009.

[63] S. Hochreiter, U. Bodenhofer, M. Heusel, A. Mayr and A. Mitterecker, "FABIA: factor analysis for bicluster acquisition," *Bioinformatics,* vol. 26, no. 12, p. 1520–1527, 2010.

[64] M. Denitto, A. Farinelli and M. Bicego, "Biclustering gene expressions using factor graphs and the max-sum algorithm," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, Buenos Aires, 2015.

[65] H. Turner, T. Bailey and W. Krzanowski, "Improved Biclustering of Microarray Data Demonstrated Through Systematic Performance Tests," *Computational Statistics & Data Analysis,* vol. 48, no. 2, pp. 235-254, 2005.

[66] G. Kerr, H. J. Ruskin, M. Crane and P. Doolan, "Techniques for clustering gene expression data," *Computers in Biology and Medicine,* vol. 38, no. 3, pp. 283 - 293, 2008.

[67] R. Henriques and S. C. Madeira, "Biclustering with flexible plaid models to unravel interactions between biological processes," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 12, no. 4, pp. 738 - 752, 2015.

[68] H.P. Kriegel, P. Kröger and A. Zimek, "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering," *ACM Transactions on Knowledge Discovery from Data (TKDD),* vol. 3, no. 1, p. 1, 2009.

[69] J. M. Kraus, G. Palm, F. Schwenker and H. A. Kestler, "Semi-Supervised Clustering in Functional Genomics," *Mathematical Analysis of Evolution, Information, and Complexity,* vol. 9, no. 1, 2009.

[70] B. Pontes, R. Girldez, J. S. Aguilar-Ruiz, "Quality Measures for Gene Expression Biclusters," *PLoS ONE,* vol. 10, no. 3, 2015.

[71] R. Peeters, "The maximum edge biclique problem is NP-complete," *Discrete Applied Mathematics,* vol. 131, no. 3, pp. 651 - 654, 2003.

[72] L. Yin and Y. Liu, "'Ensemble cuckoo search biclustering of the gene expression data," in *IEEE 15th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, Palo Alto, CA, 2016.

[73] W-H. Yang, D-Q. Dai and H. Yan, "Finding correlated biclusters from gene expression data," *IEEE Transactions on knowledge and data engineering,* vol. 23, no. 4, pp. 568 - 584, 2011.

[74] R. Core-Team, "A Language and Environment for Statistical Computing," R Foundation for Statistical Computing, Vienna, Austria, 2014.

[75] P. Sun, N. K. Speicher, R. Rottger, J. Guo and J. Baumbach, "Bi-Force: Large-scale bicluster editing and its application to gene expression data biclustering," *Nucleic Acids Research,* vol. gku201, 2014.

[76] S. Kaiser, R. Santamaria, T. Khamiakova, M. Sill, R. Theron, L. Quintales, F. Leisch and E. De Troyer, *biclust: Bicluster Algorithms. R package version 1.2.0,* 2015.

[77] M. B. Eisen, P. T. Spellman, P. O. Brown and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," in *Proc. Natl. Acad. Sci*, U.S.A., 1998.

[78] J. Wang, D. Duncan, Z. Shi and B. Zhang, "WEB-based GEne SeT AnaLysis Toolkit (WebGestalt): update 2013," *Nucleic Acids Research,* vol. 41, no. Web Server issue, pp. W77-83, 2013.

[79] S. Falcon and R. Gentleman, "Using GOstats to test gene lists for GO term association," *Bioinformatics,* vol. 23, no. 2, pp. 257-258, 2006.

[80] Y. Hochberg and Y. Benjamini, "More powerful procedures for multiple significance testing," *Statistics in medicine,* vol. 9, no. 7, pp. 811-818, 1990.

APPENDIX

SUPPLEMENTARY GO TERM ENRICHMENT ANALYSIS

The GO term enrichment analysis at significant levels of $\alpha = 0.02$ and $\alpha = 0.01$ are presented in this appendix.

Table A-1: Biological process GO terms enrichment analysis at $\alpha = 0.02$.

Algorithm	Enriched Terms (RawP/AdjP Value)
BiMax	oxidation-reduction process (5.60e-12/6.38e-09)
	small molecule metabolic process (6.82e-09/3.88e-06)
	generation of precursor metabolites and energy (4.84e-08/1.10e-05)
	single-organism biosynthetic process (4.10e-08/1.10e-05)
	small molecule biosynthetic process (3.33e-08/1.10e-05)
xMOTIFs	cell wall organization (0.0010/0.0024)
	external encapsulating structure organization (0.0010/0.0024)
	cellular cell wall organization (0.0010/0.0024)
	fungal-type cell wall organization (0.0007/0.0024)
	fungal-type cell wall organization or biogenesis (0.0009/0.0024)
Spectral	cytokinesis, completion of separation (1.24e-13/1.39e-11)
	cytokinetic cell separation (2.29e-12/1.28e-10)
	cytokinesis (1.53e-10/5.71e-09)
	cytokinetic process (7.16e-09/2.00e-07)
	cell division (7.95e-08/1.78e-06)
CC	None
PM	sporulation (2.35e-30/1.87e-27)
	sporulation resulting in formation of a cellular spore (1.10e-29/3.02e-27)
	anatomical structure formation involved in morphogenesis (1.14e-29/3.02e-27)
	anatomical structure development (2.28e-28/3.63e-26)
	anatomical structure morphogenesis (2.28e-28/3.63e-26)
	cytoplasmic translation (3.23e-86/3.01e-83)

Table A-2: Biological process GO terms enrichment analysis at $\alpha = 0.02$.

Algorithm	Enriched Terms (RawP/AdjP Value)
EPM ($\delta = 0.90$)	translation (8.60e-31/4.00e-28)
	organic substance biosynthetic process (2.15e-25/6.67e-23)
	ribosome biogenesis (3.97e-25/9.24e-23)
	biosynthetic process (5.81e-25/1.08e-22)
	cytoplasmic translation (2.82e-85/2.63e-82)
EPM ($\delta = 0.95$)	translation (2.30e-30/1.07e-27)
	ribosome biogenesis (1.44e-24/3.36e-22)
	organic substance biosynthetic process (1.27e-24/3.36e-22)
	biosynthetic process (3.47e-24/6.48e-22)
	cytoplasmic translation (1.00e-78/1.02e-75)
EPM ($\delta = 0.99$)	translation (4.98e-24/2.53e-21)
	ribosome biogenesis (1.80e-23/6.11e-21)
	biosynthetic process (5.00e-23/1.02e-20)
	organic substance biosynthetic process (4.13e-23/1.02e-20)

Table A-3: Molecular function GO terms enrichment analysis at $\alpha = 0.02$.

Algorithm	Enriched Terms (RawP/AdjP Value)
BiMax	oxidoreductase activity (1.22e-08/3.79e-06)
	oxidoreductase activity, acting on the CH-OH group of donors, NAD or NADP as acceptor (7.60e-08/1.18e-05)
	oxidoreductase activity, acting on CH-OH group of donors (1.46e-07/1.51e-05)
	catalytic activity (9.74e-07/7.57e-05)
	hydrogen ion transporting ATP synthase activity, rotational mechanism (7.57e-06/0.0004)
xMOTIFs	structural constituent of cell wall (0.0048/0.0144)
Spectral	hydrolase activity, hydrolyzing O-glycosyl compounds (4.07e-10/7.73e-09)
	hydrolase activity, acting on glycosyl bonds (9.20e-10/8.74e-09)
	glucosidase activity (2.51e-06/1.59e-05)
	glucan endo-1,3-beta-D-glucosidase activity (6.48e-06/3.08e-05)
	beta-glucosidase activity (2.26e-05/8.59e-05)

Table A-4: Molecular function GO terms enrichment analysis at $\alpha = 0.02$.

Algorithm	Enriched Terms (RawP/AdjP Value)
CC	SNAP receptor activity (4.67e-05/0.0051)
PM	lysophospholipid acyltransferase activity (1.32e-05/0.0025)
	triglyceride lipase activity (8.68e-05/0.0082)
	lysophosphatidic acid acyltransferase activity (0.0003/0.0189)
EPM ($\delta = 0.90$)	structural constituent of ribosome (2.02e-74/5.58e-72)
	structural molecule activity (4.84e-54/6.68e-52)
	rRNA binding (3.76e-12/3.46e-10)
	translation factor activity, nucleic acid binding (9.04e-09/6.24e-07)
	siderophore transmembrane transporter activity (2.45e-05/0.0011)
EPM ($\delta = 0.95$)	structural constituent of ribosome (1.06e-74/2.99e-72)
	structural molecule activity (5.27e-54/7.43e-52)
	rRNA binding (5.26e-12/4.94e-10)
	translation factor activity, nucleic acid binding (1.22e-08/8.60e-07)
	siderophore transporter activity (2.63e-05/0.0012)
EPM ($\delta = 0.99$)	structural constituent of ribosome (4.92e-67/1.47e-64)
	structural molecule activity (5.36e-47/7.99e-45)
	rRNA binding (3.33e-11/3.31e-09)
	translation factor activity, nucleic acid binding (3.48e-07/2.59e-05)
	siderophore transmembrane transporter activity (3.93e-05/0.0017)

Table A-5: Cellular component GO terms enrichment analysis at $\alpha = 0.02$.

Algorithm	Enriched Terms (RawP/AdjP Value)
BiMax	fungal-type cell wall (3.77e-08/7.94e-06)
	cell wall (1.05e-07/7.94e-06)
	external encapsulating structure (1.05e-07/7.94e-06)
	extracellular region (1.40e-06/7.95e-05)
	cytosolic small ribosomal subunit (5.49e-06/0.0002)
xMOTIFs	extracellular region (0.0002/0.0019)
	fungal-type cell wall (0.0002/0.0019)
	external encapsulating structure (0.0003/0.0019)
	cell wall (0.0003/0.0019)
Spectral	cell wall (1.21e-10/9.98e-10)
	fungal-type cell wall (8.95e-11/9.98e-10)
	external encapsulating structure (1.21e-10/9.98e-10)
	extracellular region (6.92e-11/9.98e-10)
	cell septum (3.24e-06/2.14e-05)
CC	SNARE complex (5.29e-05/0.0056)

Table A-6: Cellular component GO terms enrichment analysis at $\alpha = 0.02$.

Algorithm	Enriched Terms (RawP/AdjP Value)
PM	intracellular immature spore (4.27e-11/2.42e-09)
	prospore membrane (4.27e-11/2.42e-09)
	ascospore-type prospore (4.27e-11/2.42e-09)
	spore wall (1.13e-09/4.80e-08)
	ascospore wall (1.71e-08/5.81e-07)
EPM ($\delta = 0.90$)	cytosolic ribosome (1.84e-86/3.66e-84)
	ribosomal subunit (7.58e-75/7.54e-73)
	ribosome (1.28e-72/8.49e-71)
	cytosolic part (8.18e-68/4.07e-66)
	ribonucleoprotein complex (9.19e-60/3.66e-58)
EPM ($\delta = 0.95$)	cytosolic ribosome (1.64e-85/3.31e-83)
	ribosomal subunit (4.42e-75/4.46e-73)
	ribosome (1.60e-72/1.08e-70)
	cytosolic part (6.97e-67/3.52e-65)
	ribonucleoprotein complex (3.23e-59/1.30e-57)
EPM ($\delta = 0.99$)	cytosolic ribosome (6.43e-79/1.35e-76)
	ribosomal subunit (3.62e-66/3.80e-64)
	ribosome (3.23e-62/2.26e-60)
	cytosolic part (1.03e-60/5.41e-59)
	ribonucleoprotein complex (2.53e-50/1.06e-48)

Table A-7: Biological process GO terms enrichment analysis at $\alpha = 0.01$.

Algorithm	Enriched Terms (RawP/AdjP Value)
BiMax	oxidation-reduction process (5.60e-12/6.38e-09)
	small molecule metabolic process (6.82e-09/3.88e-06)
	single-organism biosynthetic process (4.10e-08/1.10e-05)
	generation of precursor metabolites and energy (4.84e-08/1.10e-05)
	small molecule biosynthetic process (3.33e-08/1.10e-05)
xMOTIFs	cell wall organization (0.0010/0.0024)
	external encapsulating structure organization (0.0010/0.0024)
	cellular cell wall organization (0.0010/0.0024)
	fungal-type cell wall organization (0.0007/0.0024)
	fungal-type cell wall organization or biogenesis (0.0009/0.0024)
Spectral	cytokinesis, completion of separation (1.24e-13/1.39e-11)
	cytokinetic cell separation (2.29e-12/1.28e-10)
	cytokinesis (1.53e-10/5.71e-09)
	cytokinetic process (7.16e-09/2.00e-07)
	cell division (7.95e-08/1.78e-06)
CC	None

Table A-8: Biological process GO terms enrichment analysis at $\alpha = 0.01$.

Algorithm	Enriched Terms (RawP/AdjP Value)
PM	sporulation (2.35e-30/1.87e-27)
	sporulation resulting in formation of a cellular spore (1.10e-29/3.02e-27)
	anatomical structure formation involved in morphogenesis (1.14e-29/3.02e-27)
	anatomical structure development (2.28e-28/3.63e-26)
	anatomical structure morphogenesis (2.28e-28/3.63e-26)
EPM ($\delta = 0.90$)	cytoplasmic translation (3.23e-86/3.01e-83)
	translation (8.60e-31/4.00e-28)
	organic substance biosynthetic process (2.15e-25/6.67e-23)
	ribosome biogenesis (3.97e-25/9.24e-23)
	biosynthetic process (5.81e-25/1.08e-22)
EPM ($\delta = 0.95$)	cytoplasmic translation (2.82e-85/2.63e-82)
	translation (2.30e-30/1.07e-27)
	organic substance biosynthetic process (1.27e-24/3.36e-22)
	ribosome biogenesis (1.44e-24/3.36e-22)
	biosynthetic process (3.47e-24/6.48e-22)
EPM ($\delta = 0.99$)	cytoplasmic translation (1.00e-78/1.02e-75)
	translation (4.98e-24/2.53e-21)
	ribosome biogenesis (1.80e-23/6.11e-21)
	biosynthetic process (5.00e-23/1.02e-20)
	organic substance biosynthetic process (4.13e-23/1.02e-20)

Table A-9: Molecular function GO terms enrichment analysis at $\alpha = 0.01$.

Algorithm	Enriched Terms (RawP/AdjP Value)
BiMax	oxidoreductase activity (1.22e-08/3.79e-06)
	oxidoreductase activity, acting on the CH-OH group of donors, NAD or NADP as acceptor (7.60e-08/1.18e-05)
	oxidoreductase activity, acting on CH-OH group of donors (1.46e-07/1.51e-05)
	catalytic activity (9.74e-07/7.57e-05)
	hydrogen ion transporting ATP synthase activity, rotational mechanism (7.57e-06/0.0004)
xMOTIFs	None
Spectral	hydrolase activity, hydrolyzing O-glycosyl compounds (4.07e-10/7.73e-09)
	hydrolase activity, acting on glycosyl bonds (9.20e-10/8.74e-09)
	glucosidase activity (2.51e-06/1.59e-05)
	glucan endo-1,3-beta-D-glucosidase activity (6.48e-06/3.08e-05)
	beta-glucosidase activity (2.26e-05/8.59e-05)
CC	SNAP receptor activity (4.67e-05/0.0051)

Table A-10: Molecular function GO terms enrichment analysis at $\alpha = 0.01$.

Algorithm	Enriched Terms (RawP/AdjP Value)
PM	lysophospholipid acyltransferase activity (1.32e-05/0.0025)
	triglyceride lipase activity (8.68e-05/0.0082)
EPM ($\delta = 0.90$)	structural constituent of ribosome (2.02e-74/5.58e-72)
	structural molecule activity (4.84e-54/6.68e-52)
	rRNA binding (3.76e-12/3.46e-10)
	translation factor activity, nucleic acid binding (9.04e-09/6.24e-07)
	siderophore transmembrane transporter activity (2.45e-05/0.0011)
EPM ($\delta = 0.95$)	structural constituent of ribosome (1.06e-74/2.99e-72)
	structural molecule activity (5.27e-54/7.43e-52)
	rRNA binding (5.26e-12/4.94e-10)
	translation factor activity, nucleic acid binding (1.22e-08/8.60e-07)
	siderophore transmembrane transporter activity (2.63e-05/0.0012)
EPM ($\delta = 0.99$)	structural constituent of ribosome (4.92e-67/1.47e-64)
	structural molecule activity (5.36e-47/7.99e-45)
	rRNA binding (3.33e-11/3.31e-09)
	translation factor activity, nucleic acid binding (3.48e-07/2.59e-05)
	carbon-carbon lyase activity (3.76e-05/0.0017)

Table A-11: Cellular component GO terms enrichment analysis at $\alpha = 0.01$.

Algorithm	Enriched Terms (RawP/AdjP Value)
BiMax	cell wall (1.05e-07/7.94e-06)
	external encapsulating structure (1.05e-07/7.94e-06)
	fungal-type cell wall (3.77e-08/7.94e-06)
	extracellular region (1.40e-06/7.95e-05)
	cytosolic small ribosomal subunit (5.49e-06/0.0002)
xMOTIFs	extracellular region (0.0002/0.0019)
	fungal-type cell wall (0.0002/0.0019)
	external encapsulating structure (0.0003/0.0019)
	cell wall (0.0003/0.0019)
Spectral	cell wall (1.21e-10/9.98e-10)
	fungal-type cell wall (8.95e-11/9.98e-10)
	external encapsulating structure (1.21e-10/9.98e-10)
	extracellular region (6.92e-11/9.98e-10)
	cell septum (3.24e-06/2.14e-05)
CC	SNARE complex (5.29e-05/0.0056)

Table A-12: Cellular component GO terms enrichment analysis at $\alpha = 0.01$.

Algorithm	Enriched Terms (RawP/AdjP Value)
PM	ascospore-type prospore (4.27e-11/2.42e-09)
	intracellular immature spore (4.27e-11/2.42e-09)
	prospore membrane (4.27e-11/2.42e-09)
	spore wall (1.13e-09/4.80e-08)
	ascospore wall (1.71e-08/5.81e-07)
EPM ($\delta = 0.90$)	cytosolic ribosome (1.84e-86/3.66e-84)
	ribosomal subunit (7.58e-75/7.54e-73)
	ribosome (1.28e-72/8.49e-71)
	cytosolic part (8.18e-68/4.07e-66)
	ribonucleoprotein complex (9.19e-60/3.66e-58)
EPM ($\delta = 0.95$)	cytosolic ribosome (1.64e-85/3.31e-83)
	ribosomal subunit (4.42e-75/4.46e-73)
	ribosome (1.60e-72/1.08e-70)
	cytosolic part (6.97e-67/3.52e-65)
	ribonucleoprotein complex (3.23e-59/1.30e-57)
EPM ($\delta = 0.99$)	cytosolic ribosome (6.43e-79/1.35e-76)
	ribosomal subunit (3.62e-66/3.80e-64)
	ribosome (3.23e-62/2.26e-60)
	cytosolic part (1.03e-60/5.41e-59)
	ribonucleoprotein complex (2.53e-50/1.06e-48)