BIOINFORMATICS ORIGINAL PAPER Vol. 27 no. 19 2011, pages 2738–2745
doi:10.1093/bioinformatics/btr464

*Data and text mining*

Advance Access publication August 8, 2011

# A biclustering algorithm for extracting bit-patterns from binary datasets

Domingo S. Rodriguez-Baena[1,*], Antonio J. Perez-Pulido[2] and Jesus S. Aguilar-Ruiz[1]

[1]School of Engineering and [2]Centro Andaluz de Biologia del Desarrollo (CABD), Pablo de Olavide University, Seville, Spain

Associate Editor: Martin Bishop

**ABSTRACT**

**Motivation:** Binary datasets represent a compact and simple way to store data about the relationships between a group of objects and their possible properties. In the last few years, different biclustering algorithms have been specially developed to be applied to binary datasets. Several approaches based on matrix factorization, suffix trees or divide-and-conquer techniques have been proposed to extract useful biclusters from binary data, and these approaches provide information about the distribution of patterns and intrinsic correlations.

**Results:** A novel approach to extracting biclusters from binary datasets, *BiBit*, is introduced here. The results obtained from different experiments with synthetic data reveal the excellent performance and the robustness of *BiBit* to density and size of input data. Also, *BiBit* is applied to a central nervous system embryonic tumor gene expression dataset to test the quality of the results. A novel gene expression preprocessing methodology, based on expression level layers, and the selective search performed by *BiBit*, based on a very fast bit-pattern processing technique, provide very satisfactory results in quality and computational cost. The power of biclustering in finding genes involved simultaneously in different cancer processes is also shown. Finally, a comparison with *Bimax*, one of the most cited binary biclustering algorithms, shows that *BiBit* is faster while providing essentially the same results.

**Availability:** The source and binary codes, the datasets used in the experiments and the results can be found at: http://www.upo.es/eps/bigs/BiBit.html

**Contact:** dsrodbae@upo.es

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

Received on May 11, 2011; revised on July 31, 2011; accepted on August 2, 2011

## 1 INTRODUCTION

Binary datasets represent a compact and simple way to store data about the relationships between a group of objects and their possible properties. This type of data is present in many research fields, including data mining (Alqadah *et al*., 2010; Colantonio *et al*., 2010; Sun *et al*., 2008), text mining (Mimaroglu *et al*., 2007), bioinformatics (Figueroa *et al*., 2004; Perco *et al*., 2005; Shmulevich *et al*., 2002), engineering (Haibing *et al*., 2008) or paleontology (Puolamaki *et al*., 2006), among others. What the values 0 and 1 stand for depends on the context. For example, when working with gene and protein features, if gene $r$ encodes a protein that belongs to a protein of class $c$, then $\langle r, c\rangle$ is equal to 1; otherwise, it is equal to 0 (Koyuturk *et al*., 2004). Commonly, the binary values 1 and 0 mean that under experimental condition $c$, gene $r$ is either expressed or not, respectively (Prelic *et al*., 2006; Zhang *et al*., 2010). In some transcription regulation datasets, a matrix element is 1 if the transcription factor associated with the motif group putatively binds the upstream region of a certain gene (Van Uitert *et al*., 2008).

Clustering is one of the most popular techniques used to identify the distribution of patterns and intrinsic correlations in large datasets (Kerr *et al*., 2008). However, the impossibility of discovering local patterns by this technique has led to the popularization of biclustering algorithms, which are able to analyze all dataset dimensions simultaneously and, consequently, extract local patterns that provide a better understanding of the underlying biological phenomena.

Different biclustering algorithms have been developed for their use with binary datasets. For example, in Koyuturk *et al*. (2004), biclusters are defined as submatrices dense enough with 1's to be considered statistically significant. Based on the same concept, the work of Van Uitert *et al*. (2008) adapts a score function to effectively deal with sparse binary datasets. However, these changes lead to a more complicated user input parameters. Besides, all the elements of every generated bicluster are set to zero in the input matrix, introducing noise.

Matrix factorization can be applied to reduce the dimensionality of data, yielding a representation of conditions as linear combinations of a reduced set of $k$-factors. Based on this idea, a Binary Matrix Factorization (*BMF*) algorithm is presented in Zhang *et al*. (2010). The objective is to decompose the input binary matrix, $X$, into two binary matrices, $W$ and $H$, such that $X \approx WH$. Determining the suitable number of factors k is, however, the main hurdle with these type of approaches (Brunet *et al*., 2004; Carmona *et al*., 2006).

Suffix trees have also been used to extract biclusters from binary matrices. Thus, the *e-BiMotif* algorithm (Gonsalves *et al*., 2010) combines sequence alignment with an adaptation of the e-CCC-Biclustering algorithm (Madeira *et al*., 2009) to further group sequences with similar motif structures. To find all the maximal contiguous column biclusters, the *e-BiMotif* algorithm generates a suffix tree T from a binary matrix where rows and columns represent candidate motifs and sequences, respectively. Then, T is used to spell all valid biclusters taking into account an error threshold in the structure of the motifs.

*To whom correspondence should be addressed.

*Bimax* (Prelic *et al*., 2006) is one of most popular biclustering algorithms (Bhattacharya *et al*., 2009; DiMaggio *et al*., 2008; Harpaz *et al*., 2011; Serin *et al*., 2011) and finds all the inclusive maximal biclusters such that all their elements are 1. However, this method is hindered by the large number of results produced and the large execution times. In some situations, this approach produces such a great number of results that it is unable to store and process all of them. *Bimax* uses a simple divide-and-conquer approach that provides a very fast response. Nevertheless, as with all divide-and-conquer algorithms, performance can dramatically worsen if the number of recursive calls is large enough.

In this work, a novel alternative to extract biclusters from binary datasets is presented: the Bit-Pattern Biclustering Algorithm, (*BiBit*). The bit-pattern processing technique used and the selective search performed makes this algorithm very fast, reaching very high processing speeds. In addition, *BiBit* was found not to be affected by shape or density of the input binary matrices. Synthetic and real datasets have been used in several tests to assess the performance and efficiency of *BiBit*, and we found that, while our approach is significantly faster than *Bimax*, it can provide essentially the same results.

## 2 METHODS

The methodology designed to extract biclusters from binary datasets is composed of two phases: *encoding* and *searching* (Fig. 1).

The only input parameters needed are the binary input matrix, $B$, the minimum number of rows, *mnr*, and the minimum number of columns, *mnc*, allowed in the final biclusters. The effect of *mnr* and *mnc* in the results will be analyzed in Section 3.1.1.

### 2.1 Definitions

DEFINITION 1.

*An input matrix is defined as a triplet* $B=(R,C,\ell)$*, where* $R$ *and* $C$ *are two finite sets referred to as* the set of rows *and* the set of columns*, respectively, and* $\ell : R \times C \longrightarrow \{0,1\}$ *is the binary* level *function. We will denote the binary value* $\ell(r,c)$ *by* $\langle r,c \rangle$.

The binary matrix $B=(R,C,\ell)$, with $N=|R|$ and $M=|C|$, can be decomposed into $N$ sets of $M$ bits: $B=\{r_1,r_2,\ldots,r_N\}$, with $ri=\{b_{i1},b_{i2},\ldots,b_{iM}\}$, being $b_{ij}=\{0,1\}$. These groups of bit sets can be equipped by a Boolean algebra $\beta(\wedge$ (AND),$\vee$ (OR), $'$ (NOT)) defined for the binary function $\ell$.

DEFINITION 2.

*A bit-pattern is a bicluster composed of the pair of non-empty sets* $(I,J)$*, with* $I \subseteq R$ *and* $J \subseteq C$*. A set of columns* $J=\{c_1,c_2,\ldots,c_k\}$ *is called a* ***pattern*** *if for every* $c_i \in J$ *and for every pair of rows* $r,r' \in I$*, then* $\langle r,c_i \rangle \wedge \langle r',c_i \rangle = 1$.

DEFINITION 3.

*The bit-pattern* $(I,J)$ *is called a maximal bicluster if and only if it is a bicluster and it is not entirely contained in any other bicluster.*

Notice that the objective of this work is not to extract all maximal biclusters in a dataset. To define the subset of maximal biclusters that will be generated by *BiBit*, let us consider $L=\{(r_1,r_2),(r_1,r_3),\ldots,(r_N,r_{N-1})\}$ as the set of all possible pairs of rows, with $|L|=\frac{N!}{2!*(N-2)!}$. If every pair $(r_i,r_j)$ is considered a potential seed from which a bit-pattern can be generated, the value $|L|$ is the maximum number of maximal biclusters that can be extracted from matrix $B$. That is, every pair of rows creates a *pattern*, $\rho$, that will be used to form a bit-pattern by adding new rows that suit $\rho$.

In short, the biclusters will be maximal submatrices created from a pattern obtained by the application of the boolean operator $\wedge$ to a seed pair of rows. In addition, the maximum number of maximal biclusters that can be extracted from a binary matrix is limited by the number of possible pairs of rows.

### 2.2 Encoding

In this phase, the dataset is transformed into an integer-coded matrix, which will reduce the computational cost of the next phase. As shown in Figure 1, every row, $r_i$, is divided into bit words of a certain size (set to 4 in this example). Every bit word is translated into its integer representation. As a result, the column dimension, and consequently the number of operations required in the second phase, will be reduced drastically.

### 2.3 Searching

The generation of biclusters is illustrated in Algorithm 1. An example of the process is depicted in Figure 1. The row pair composed of $r_1$ and $r_2$ creates the *pattern* $\rho_{12}=\{9,0,6\}$ as a result of applying the AND boolean operator ($\wedge$) (line 2). The result is the *pattern* $J=\{c_1,c_4,c_{10},c_{11}\}$. This set will form part of a new potential bicluster if $|J| \geq mnc$ (set to 3 in the example) and if it is the first time it is observed. A new potential bicluster, $B_{12}=\{I,J\}=\{\{\{r_1,r_2\},\{c_1,c_4,c_{10},c_{11}\}\}$ is created (lines 3 and 4). Next, the remaining rows are processed. If the result of applying the AND boolean operator between every remainder row and the *pattern* matches that *pattern*, the aforementioned row can be added to the bicluster (lines 5–9). In the example of Figure 1, rows $r_3$ and $r_4$ are compatible with the pattern $\rho_{12}$. The final row, $r_5$, contains at least a 0 in one of the positions in pattern $\rho_{12}$ (in column $c_4$). In this case, the row is rejected. A new bicluster $Bic_{12}=\{I,J\}=\{\{r_1,r_2,r_3,r_4\},\{c_1,c_4,c_{10},c_{11}\}\}$ is generated, verifying previously that $|I| \geq$ the *mnr* parameter.

In spite of the algorithm complexity being $O(N^2M')$, with $N$ as the number of rows and $M'$ as the number of columns after the encoding step, the most demanding computational operations, that is, the comparison operations between rows, are performed at the bit level (the most elemental operation for which the computer processors are designed). This optimization makes our algorithm extremely fast.

## 3 RESULTS

The aim of this section is to assess the performance of the *BiBit* algorithm, and it is divided into two parts. In the first part, synthetic datasets are used to evaluate the behavior of the algorithm with respect to sensitivity and to compare the algorithm with *Bimax* in terms of the number of results generated, the execution time and the quality of results. A novel cumulative binarization method, with an increasing density of 1's, is presented in Section 3.2. One of the objectives in our experiments with synthetic datasets is therefore to

**Algorithm 1** Bit-patterns biclustering algorithm

**Input:** $B$: Encoded binary matrix
*mnr*: Minimum number of rows allowed
*mnc*: Minimum number of columns allowed
**Output:** $X$: List of final biclusters

1. **for** every rows pair $(r_i,r_j)$ **do**
2. $\rho_{ij}=r_i \wedge r_j$
3. **if** $\rho_{ij}$ is new and $|\rho_{ij}| \geq mnc$ **then**
4. $Bic_{ij}=\{I,J\}=\{\{r_i,r_j\},\{\rho_{ij}\}\}$
5. **for** Every remainder row $r_q$ **do**
6. **if** $r_q \wedge \rho_{ij}=\rho_{ij}$ **then**
7. Add $r_q$ to $I$
8. **end if**
9. **end for**
10. **if** $|I| \geq mnr$ **then**
11. Add $Bic_{ij}$ to $X$
12. **end if**
13. **end if**
14. **end for**

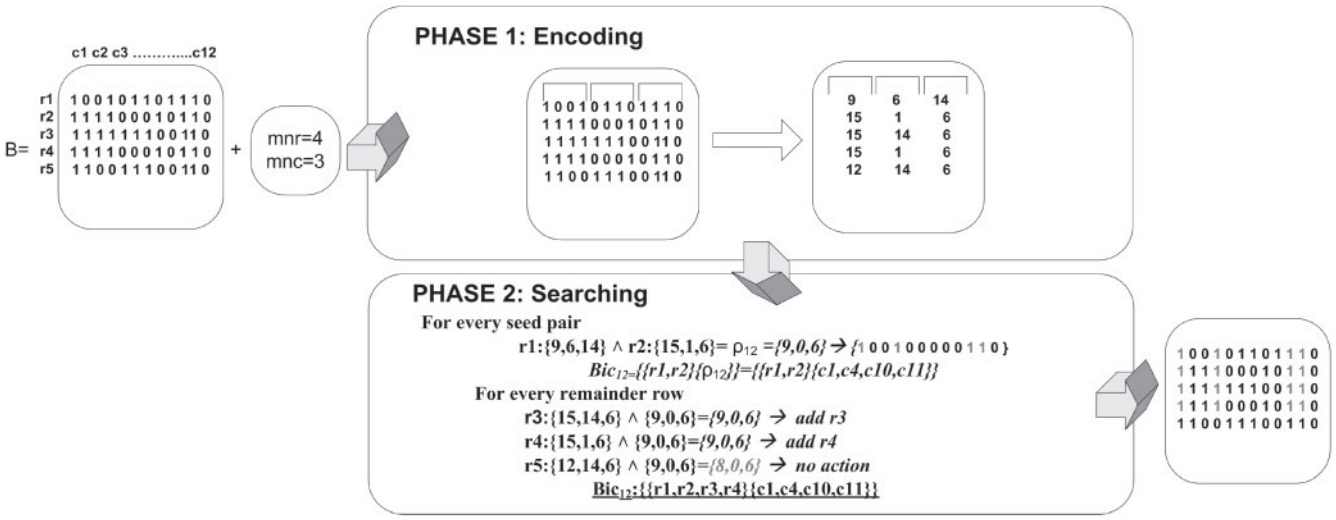


**Fig. 1.** A schematic representation of the methodology presented in this work and a brief and simple example. The binary matrix $B$ and the user parameters, defined as *the minimum number of rows* (*mnr*) and *the minimum number of columns* (*mnc*) allowed in a final bicluster, are the inputs of the method. The extraction process of bit–patterns is divided into two sequential phases. In the first phase, the encoding manages the input database reduction. Using a novel encoding process, columns are divided into bit words of a certain length (4 in this particular case). Every bit word is translated independently into its integer representation. Thus, the column dimension, and consequently the number of operations needed in the second phase, will be reduced drastically. The second phase is related with the biclustering generation process. Every pair of rows generates a *pattern*, $\rho$. If the number of elements of $\rho$ is not below the input parameter *mnc* and that *pattern* has not been processed previously, an initial bicluster is created. Later, rows are added to that bicluster. At the end of the process, if the number of rows of the final bicluster is equal to or greater than the parameter *mnr*, that bicluster is considered a valid result.

analyze the performance of *BiBit* with different densities of the input matrix. The objective of the second part is to use prior biological knowledge to evaluate the usefulness of *BiBit* algorithm in relation to real data. To achieve this, a central nervous system embryonic tumor gene expression dataset (Pomeroy *et al*., 2002) was used.

### 3.1 Synthetic datasets

*3.1.1 Performance test* The purpose of this test is to check the behavior of *BiBit* with different user parameter values, as well as comparing the performance of *BiBit* and *Bimax* algorithms. *BiBit* is not influenced by the shape or the density of 1's of the input binary matrices. Square binary matrices of different sizes and density of 1's were used. Twenty different sizes in the range $50\times50$ to $1000\times1000$ were used, with an increment of 50 in both dimensions. For each size, 10 matrices were generated with a density of 1's that varies from 10% to 100%, or from sparse to dense databases, increasing 10% in each case. The number of 1's in each dataset is uniformly distributed (see Supplementary Material). The following notation will be used to mention the features of each matrix: *NxM_P%*, with $N$ as the number of rows, $M$ as the number of columns and $P$ as the percentage of 1's included in the matrix.

First, we analyzed the effect of input parameters. The minimum number of rows (*mnr*) and the minimum number of columns (*mnc*) required for a bicluster are very simple parameters to use. Their objective is to modify the search space to increase or reduce the number of results. The *BiBit* algorithm was applied to a *500x500_10%* binary matrix with different user input parameter values: in one case, *mnr* is fixed to 2, and *mnc* varies from 2 to 10 ($mnc=\{2,4,6,8,10\}$), and in the other case, $mnc=2$ and $mnr=\{2,4,6,8,10\}$. The tests performed show how the number of biclusters decreases with an increase in the restrictions imposed by the input parameters. This trend is more pronounced in the case of the *mnr* parameter. Biclusters are first created from a pair of rows, so it is to be expected that the maximum number of biclusters is reached when *mnr* parameter is equal to 2. With greater values, the number of biclusters decreased in a sharp way. The faster way to reduce the number of results is therefore to increase the *mnr* parameter. However, if reduction of execution time is required, the *mnc* parameter is the most important.

We then applied the *BiBit* and *Bimax* algorithms to 200 binary matrices in order to verify the results with each method. In both algorithms, the parameters that limit the minimum number of rows and columns of the results are fixed to 2, the least restrictive option. Both approaches have been developed in the Java programming language, and only the bicluster generation process was taken into account when measuring execution times.

The *BiBit* algorithm took ~19 min and 89 s to process the 200 matrices. *Bimax* could not end the test due to excessive memory consumption and extremely large execution times (it only finished with success in 12 cases). To compare both approaches, only the first five matrices with a size of $100\times100$ (*100x100_D%*, with $D\in\{10,20,30,40,50\}$) could be used. Figure 2 depicts three graphs comparing the number of biclusters generated (Fig. 2a), the execution time (Fig. 2b) and a speed measure (Fig. 2c). Concerning the number of biclusters, *BiBit* generates a number of results that ranges between 947 (for a density of 10%) and 4950 (densities from 40% to 90%). As can be observed in Figure 2a, *Bimax* obtained larger number of biclusters, from 989 to 9 246 446 biclusters. Note again the great difference of values in Figure 2b, with execution times that ranges from 47 (for a density of 10%) to 63 ms (for the rest of densities) in the case of *BiBit*, and values that ranges from

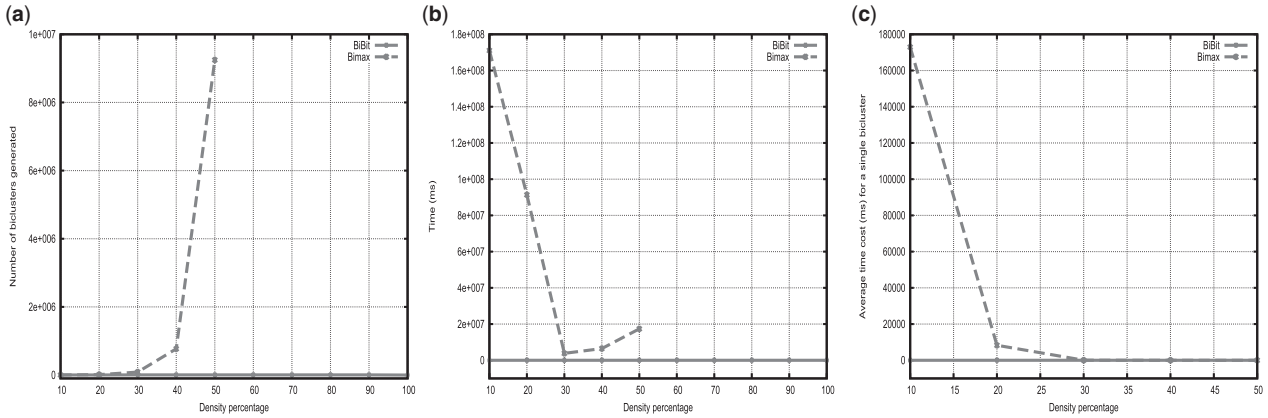


**Fig. 2.** Performance comparison between *BiBit* and *Bimax*. From left to right **(a)** Number of biclusters obtained when varying the density of 1's. **(b)** Execution time (in milliseconds). **(c)** Average time per bicluster.

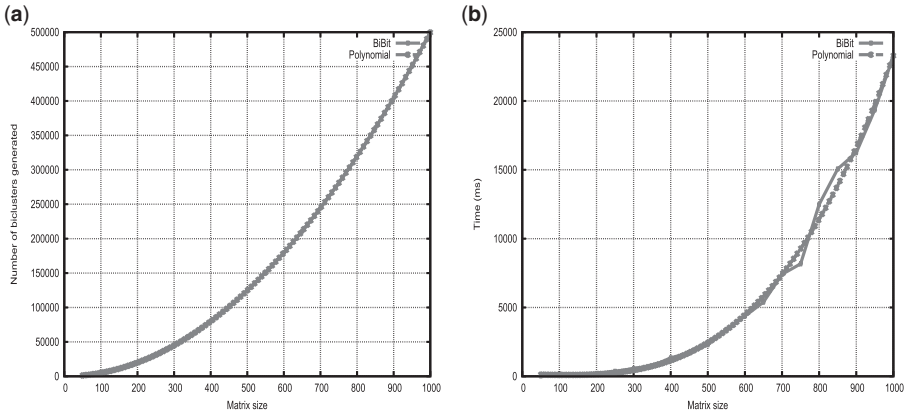


**Fig. 3.** Impact of the matrix size on, from left to right, (**a**) the number of biclusters generated, and (**b**) the execution time, for matrices with a density equal to 50%.

3 847 704 to 171 051 625 ms for *Bimax*. In order to compare more effectively these two approaches, we measured the execution times per bicluster instead ($\frac{\text{Time}}{|\text{Biclusters}|}$), as shown in Figure 2c. *BiBit* took at least 0.01 ms/bicluster and at most 0.04 ms/bicluster, while *Bimax* ranged from 1.88 ms to 172.9 s per bicluster. The large difference in execution times in the 10% density matrix is remarkable. Both approaches find a similar number of results (927 by *BiBit* and 989 by *Bimax*) but while *BiBit* takes only 47 ms, *Bimax* takes 47.5 h. This is explained by an increase in the resources needed by the recursive process included in the *Bimax* algorithm under certain circumstances. In divide-and-conquer techniques, a problem is divided into a certain number of subproblems. In some cases, as with *Bimax*, the size of these subproblems is data dependent, and the algorithm complexity could be negatively affected when the workload of the subproblems is unbalanced. *Bimax* uses the columns with 1's of a certain row as a template to perform the matrix splitting. In this particular case, the sparseness of the matrix could lead to an unbalanced division. This fact, together with the square shape of the matrix, could lead to a worst-case running time complexity of $O(n^3\beta)$, with $\beta$ representing the total number of all inclusion-maximal biclusters in the input matrix (Supplementary Material in Prelic *et al.*). As Figure 2b clearly shows, *Bimax* execution times decrease as the density increases up to 30%, after which times increase due to the large number of biclusters generated with higher densities (Fig. 2a).

Finally, an analysis of the performance of the *BiBit* algorithm during the processing of the 200 binary matrices is presented. The largest number of biclusters generated, 499 500, was reached while processing the matrix with a size of $1000\times1000$ using different densities. A time of 40 735 ms, or $\sim$40 s, was the largest execution time measured and corresponds to the binary matrix of $1000\times1000\_20\%$. Figure 3 presents the behavior of *BiBit* with density fixed at 50% for different matrix sizes. As it can be observed, the number of biclusters found and the execution times are approximately cubic in the size of the input matrix.

*3.1.2 Match score test* We analyzed the accuracy of *BiBit* and *Bimax* to find specific biclusters using the Match Score, which is a commonly used measure to assess biclustering performance (Prelic *et al.*, 2006) (see Supplementary Material). The *average bicluster relevance* reflects to what extent the generated biclusters represent 'true' biclusters, i.e. biclusters that have been introduced in the dataset. The *average module recovery* quantifies how well each of the true biclusters was recovered. These two measures are

normalized so that they take a value of 1 if the set of generated biclusters is equal to the true set of biclusters, and a value of 0 if both sets are disjoint.

We developed two different tests. The first test measures the accuracy of *BiBit* and *Bimax* when biclusters of different degrees of overlap are introduced artificially created (see Supplementary Material for further details).

The second test tries to measure the performance of both algorithms when matrices with different densities of 1's are used. Binary matrices of different sizes ($50\times50$, $100\times100$, $200\times200$) with densities of 1's that range from 5% to 50%, with increments of 5%, and with non-overlapped biclusters of different sizes randomly inserted were used (see details in Supplementary material). Only the bicluster relevance results for the $100\times100$ matrix are shown in Figure 4. Results obtained for the different matrix sizes are very similar, but there is a clear relationship between density and match score results. *BiBit* localizes the correct biclusters more often if the matrix is sparse, that is, its accuracy decreases as the density increases. However, *BiBit* results are always better than those of *Bimax*, as depicted in Figure 4.

In conclusion, *BiBit* and *Bimax* are equally accurate when considering different degrees of overlap in the true biclusters. However, when considering matrices of varying densities, *BiBit* always performs better than *Bimax* and particularly so with sparse datasets.

### 3.2 Experimental dataset

To investigate the usefulness of the *BiBit* algorithm, a central nervous system (CNS) embryonic tumor gene expression dataset (Pomeroy *et al*., 2002) was analyzed. In this dataset, a classification system based on DNA microarray gene expression data was developed. The aim of our analysis was to find biclusters that isolate the samples of every type of CNS tumor to study the biological relevance of their genes. The dataset selected for this experimental test, matrix **A1**, is composed of 40 tumor samples [including 10 medulloblastomas, 10 malignant gliomas, 10 atypical teratoid/rhabdoid tumors (AT/RT, 5 brain, 3 renal and 2 extrarenal), 4 normal cerebellums and 6 supratentorial primitive neuroectodermal tumors (PNETs)] analyzed on Affymetrix HuGeneFL. The result is a $7129\times40$ matrix of integers. In Figure 5, a schematic representation of the analysis methodology is shown. The process is composed of three different phases.

We started by preprocessing the dataset as follows. The integer matrix was first standardized, generating a real value matrix with a mean of 0 and a variance of 1. This was followed by a discretization

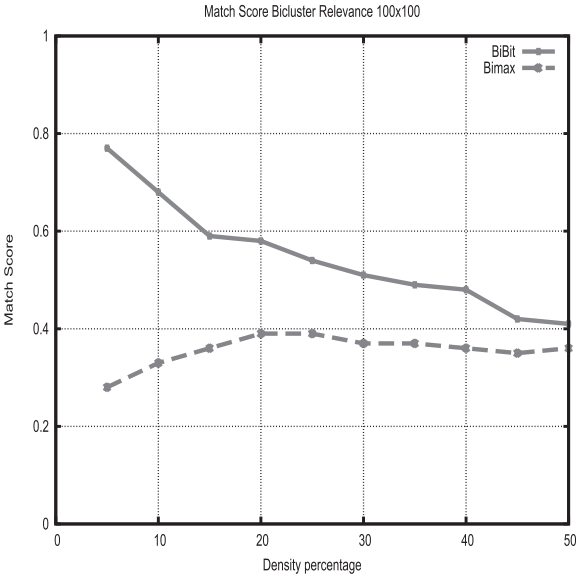


**Fig. 4.** Average bicluster relevance as measured by the match score test for a matrix of size $100\times100$ with different densities of 1's and randomly inserted biclusters.

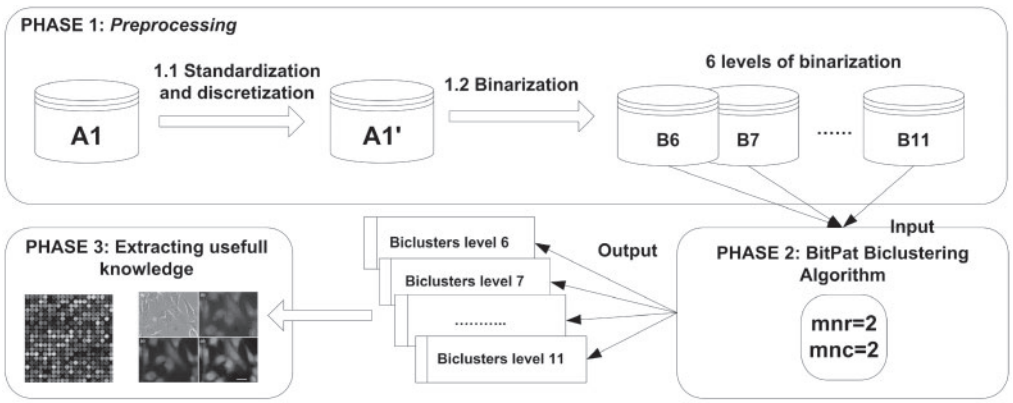


**Fig. 5.** A schematic summary of the tumor cancer dataset analysis process. In the first phase, the dataset is preprocessed in two different steps. In the first step, the data are standardized (mean 0 and variance 1) and then discretized into 12 different levels (with values from 0 to 11). Each value corresponds to a gene expression value range. Second, a binarization process is applied to the discretized data. The output will be composed of different binary datasets, each belonging to a level i, with $6\leq i\leq11$ (only those levels in which genes are activated). In a binary dataset with level $i$, values equal to 1 will be those that, in the discretized dataset, are equal or greater than $i$. The rest of the values will be equal to 0. In phase two, the *BiBit* algorithm is then applied to every binary dataset, generating a group of final biclusters in each case. In this experiment, the parameters *mnr* and *mnc* are equal to 2. Finally, in Phase 3, the results are analyzed to extract useful knowledge.

step in order to establish different levels of gene expression values. Specifically, the range $[-3.0, 3.0]$ was divided into 12 levels, with a distance of 0.5 between each level. Each real value was converted into a discrete value $d$, with $d \in \{0, 1, 2, \ldots, 11\}$, according to its gene expression level. Values equal to or lower than $-3.0$ were considered to be at level 0, and values $\geq 3.0$ were considered to be at level 11. Levels from 6 to 11 corresponded to expressed genes, and levels from 0 to 5 to non-expressed genes. Then, a new dataset named **A1'** was created. The same data preprocessing actions were performed in (Pomeroy *et al*., 2002). In this article, the standardization process was applied before using the self-organizing map (Kohonen *et al*., 1997) clustering technique. In addition, the same gene expression ranges were used to graphically represent the behavior of the genes with respect to the different tumor types. The second preprocessing step consists in obtaining a binary matrix that can be directly used by *BiBit*. This is the aim of the second step in the preprocessing. As was mentioned in Section 1, in gene expression datasets, binary values of 1 and 0 under an experimental condition $c$ mean that a gene $r$ is expressed or not, respectively. For example, in the work of Prelic *et al*. (2006), a discretization threshold was set to $\underline{e} + (\bar{e} - \underline{e})/2$, with $\underline{e}$ and $\bar{e}$ as the minimum and maximum expression values in the data matrix, respectively. In this work, a novel method was used to transform the data into binary values: for each different gene expression level previously observed, a new binary matrix is created. That is, for each level $i$, a new matrix $B_i$ was generated, in which $B_i(x, y) = 1$ if $A1'(x, y) \geq i$; $B_i(x, y) = 0$ otherwise. This new cumulative binarization method gives us more information in the sense that not only the activation of genes is taken into account. In addition, the minimum level of gene expression in which this activation takes place is known and can be used to extract new and useful conclusions. As only expressed genes are of interest for this study, at the end of the preprocessing phase, six binary matrices, that corresponded with gene expression levels from 6 to 11, were generated, with a density of 1's that increases as the gene expression level decreases. This density ranges from 2.22%, for level 11, up to 23%, for level 6.

The objective of the second phase was to extract biclusters that included the maximum number of columns (samples) and only from the same type of tumor. This type of results was called *tumor biclusters*. To achieve this objective, the *BiBit* algorithm was applied to the six aforementioned binary matrices, using the following values for the input parameters: $mnr = 2$ and $mnc = 2$. The minimum number of columns (samples) was set to 2, because this was the smaller number of samples that referred to a certain type of tumor, the AT/RT extra renal tumor. The minimum number of rows (genes) is set to 2 to extract the maximum number of possible results. Because of the encoding preprocessing step of the *BiBit* method, the original dimension of the input matrix, $7129 \times 40$, is drastically reduced, in the column dimension, to $7129 \times 3$. A summary of the performance of *BiBit* is included in Table 1. In this table, information about the number of biclusters and tumor biclusters obtained and the execution time is presented for every gene expression level binary matrix. As can be observed here, the bicluster processing for the six matrices took only $\sim$2.5 min.

In Phase 3, the most relevant results were analyzed to extract useful knowledge. For every type of tumor, a representative bicluster was selected. The criteria used for these selections were the following: the maximum percentage of tumor samples included and the minimum number of genes. This last constraint was taken into account with the aim of looking for a group of genes as specialized as possible. Consequently, it is important to find biclusters belonging to a gene expression level matrix that are as high as possible. The result of the selection process is summarized in Table 2. For every type of tumor, the following information about the selected tumor biclusters is included: the number of genes, the number of samples, the percentage of coverage concerning the samples of the tumor and the binary matrix level in which it was found. A noteworthy detail is that the seven biclusters share an important number of genes. However, they are mainly the positive controls of the transcriptomic experiment (as was expected) or are from ribosomal proteins commonly expressed in different tumors. Because of this, in every bicluster, the common genes (included in 70% of the rest of the biclusters), are rejected. Thus, the value shown in the second column of Table 2 is the number of genes after this filtering. The *BiBit* algorithm was able to find biclusters composed of groups of genes fully related to a certain tumor type. In four of the seven cases, biclusters with a 100% coverage were discovered. The mean percentage coverage is very high, up to 87%. It is worth highlighting the case of the glioblastoma bicluster, with all the 10 samples included. Another important detail is that all the results were extracted from high gene expression level binary matrices (the mean level is $\sim$10), implying that genes involved in these carcinogenic conditions are in general highly expressed.

Genes included in a bicluster are expected to be involved in similar biological processes. Because of this, a gene enrichment analysis was performed. To convert Affymetrix HuGeneFL ids into ensemble

**Table 1.** Performance information of *BiBit* during binary matrix processing

| Level | No. of biclusters | No. of tumour bicluster | Time (s) |
|---|---|---|---|
| 11 | 12192 | 278 | 1.19 |
| 10 | 16859 | 388 | 1.52 |
| 9 | 25682 | 438 | 2.14 |
| 8 | 49246 | 619 | 4.29 |
| 7 | 135129 | 791 | 13.29 |
| 6 | 891373 | 1148 | 135.70 |

The first column shows the level of the binary matrix. The next two columns indicate the number of biclusters generated, but the third one refers only to those bicluster with all of their samples belonging to only one type of tumor (called tumor biclusters). The last column presents the execution time in seconds.

**Table 2.** Features of the selected *tumor biclusters*

| Tumour type | NumGenes[a] | NumSamples | Coverage (%) | Level |
|---|---|---|---|---|
| Brain | 101 | 4 | 80 | 10 |
| Extra Renal | 78 | 2 | 100 | 11 |
| Renal | 168 | 3 | 100 | 9 |
| Glioblastoma | 24 | 10 | 100 | 9 |
| Medulloblastoma | 87 | 8 | 80 | 8 |
| Normal C. | 31 | 4 | 100 | 11 |
| PNET | 64 | 3 | 50 | 9 |

[a]This is the number of genes after eliminating the common ones.
The first column is the type of tumor that every bicluster represents. The second and third columns are its number of genes and samples, respectively. The fourth column shows the percentage of tumor samples included in every bicluster. Finally, the last column presents the binary matrix level from which the bicluster was extracted.

**Table 3.** For each tumor type, this table includes some of the GO attributes found in the bicluster enrichment analysis

| Tumour type | *P*-value | GO attribute |
|---|---|---|
| Brain | 5.33E-14 | MHC class I protein complex |
| | 1.87E-08 | Cell killing |
| | 5.10E-06 | Immune response |
| Extra Renal | 7.94E-32 | Ribonucleoprotein complex |
| | 1.06E-13 | Ribosomal small subunit biogenesis |
| | 2.61E-06 | Antigen processing and presentation |
| Renal | 1.04E-12 | U4 snRNA binding |
| | 6.50E-07 | Ribosomal large subunit biogenesis |
| | 2.60E-05 | Microtubule-based movement |
| Glioblastoma | 4.39E-18 | Natural killer cell mediated immunity |
| | 5.18E-15 | Spindle assembly |
| | 7.34E-15 | MHC protein binding |
| Medulloblastoma | 3.60E-13 | Ribosomal small subunit biogenesis |
| | 2.04E-12 | Ribonucleoprotein complex biogenesis |
| | 2.34E-06 | Ribosomal large subunit biogenesis |
| N. Cerebellum | 3.30E-05 | Fructose-bisphosphate aldolase activity |
| | 4.46E-05 | Homophilic cell adhesion |
| | 4.68E-03 | Neuron projection morphogenesis |
| PNET | 9.38E-14 | Leukocyte mediated cytotoxicity |
| | 1.60E-09 | Large ribosomal subunit |
| | 2.50E-05 | Negative regulation of RNA splicing |

The first column refers to a bicluster associated with a certain type of tumor. The second column is the *P*-value, and the GO attribute is in the last column (see details at http://www.upo.es/eps/bigs/BiBit_datasets.html).

gene ids, the BioMart Ensemble data integration system (Smedley *et al*., 2009) was used. Finally, the web application FuncAssociate (Berriz *et al*., 2009) was used to discover gene ontology (Consortium *et al*., 2006) functional attributes enriched in our sets of genes. Some of the results of this analysis are presented in Table 3. This table includes, for every bicluster representing a type of tumor, an example composed of three biological processes enriched in their genes, along with the *P*-value. All the biclusters contain a high degree of enrichment and almost all of them present annotations related to cancer.

The work of Pomeroy *et al*. (2002) included an unsupervised study of the intrinsic structure of the medulloblastoma data, in which the genes that were most highly correlated were primarily ribosomal protein-encoding genes. In our medulloblastoma bicluster, the significant annotations were mainly related to ribosomal proteins, as was expected, even though the tumors with poor prognoses appear with these annotations. However, the significant annotations linked to malignant gliomas were enriched in related processes belonging to natural killer (NK) cells. These glioma tumors are known to express ligands of activating NK receptors (Castriconi *et al*., 2009), and the enrichment result is in accordance with this outcome. In addition, the spindle assembly notation is related to the cellular division process inherent to cancer. Finally, Pomeroy *et al.* provide the top 10 gene markers per tumor class. As proof of the usefulness of the biclustering techniques, the *BiBit* approach confirms this classification for some of these genes but detects that in some cases a gene is associated with more than one class. For example, gene D76435 was classified as medulloblastoma but is also included in our normal cerebellum bicluster. Yokota proved that this gene was highly expressed in the nuclei of the cerebellar granule cell lineage but was also detected in medulloblastoma (26/29 cases), and was not present in any other tumors examined in this work (Yokota *et al*., 1996). Gene X86809 provides another example of association with malignant glioblastomas but detected in the normal cerebellum bicluster as well. In spite of being a ubiquitous gene, it presents higher expression levels in brain tissues, including the cerebellum (Estelles *et al*., 1996).

*Bimax* algorithm has been applied to the CNS gene expression dataset (see the Supplementary material for further details) as well. The conclusion of the comparative between *BiBit* and *Bimax* is that *BiBit* produces similar results to Bimax, but requiring considerable less computation time.

## 4 CONCLUSIONS

In this work, we have introduced a new biclustering algorithm, *BiBit*, designed for binary datasets. Our algorithm is based on a selective search in which the results are obtained by means of a fast bit-pattern processing technique. Synthetic and real datasets were used to perform several experiments and were compared with the *Bimax* algorithm. These experiments show that *BiBit* can obtain similar results to *Bimax* using significantly less computation time and reducing the total number of generated biclusters. BiBit was also shown to outperform Bimax when comparing their Match Scores in a synthetic dataset. Besides, the results obtained revealed an excellent and scalable performance as well as the robustness of *BiBit* to the density and the size of input data. These features allow for the application of *BiBit* to different types of biological studies such as transcription factor binding sites analysis, ontological annotations rules or RNA sequence alignment. To test the usefulness of *BiBit*, a CNS embryonic tumor gene expression dataset (Pomeroy *et al*., 2002) was analyzed. In this experiment, a new gene expression preprocessing methodology, based on expression level layers, was used. This new cumulative binarization method gives us more information as not only the activation of genes is taken into account. In addition, the minimum level of gene expression in which this activation takes place is known. The biclusters obtained classify the seven different types of tumors included in Pomeroy's dataset with 87% of coverage and contain a high enrichment degree. Finally, the power of biclustering in finding genes involved simultaneously in different cancer processes was shown.

*Funding*: Ministry of Science and Innovation project (TIN2007-68084-C02-00); Junta de Andalucia projects (P07-TIC-02611 and TIC-200) in part.

*Conflict of Interest*: none declared.

## REFERENCES

Alqadah,F. *et al.* (2010) A novel framework for detecting maximally banded matrices in binary data. *Stat. Anal. Data Min.*, **3**, 431–445.

Berriz,G.F. *et al.* (2009) Next generation software for functional trend analysis. *Bioinformatics*, **25**, 3043–3044.

Bhattacharya,A. and Rajat,D. (2009) Bi-correlation clustering algorithm for determining a set of co-regulated genes. *Bioinformatics*, **25**, 2795–2801.

Brunet,J. *et al.* (2004) Metagenes and molecular pattern discovery using matrix factorization. *Proc. Natl Acad. Sci. USA*, **101**, 4164–4169.

Carmona-Saez,P. *et al.* (2006) Biclustering of gene expression data by non-smooth non-negative matrix factorization. *BMC Bioinformatics*, **7**, 78.

Castriconi,R. *et al.* (2009) NK cells recognize and kill human glioblastoma cells with stem cell-like properties. *J. Immunol.*, **182**, 3530–3539.

Colantonio,A. *et al.* (2010) ABBA: adaptive bicluster-based approach to impute missing values in binary matrices. In *25th ACM Symposium on Applied Computing, SAC '10*. ACM New York, NY, USA, pp. 1026–1033.

Consortium,G.O. The Gene Ontology (GO) project in 2006. *Nucleic Acids Res.*, **34**, 322–326.

DiMaggio,P. *et al.* (2008) Biclustering via optimal re-ordering of data matrices in systems biology: rigorous methods and comparative studies. *BMC Bioinformatics*, **9**, 458.

Estelles,A. *et al.* (1996) The major astrocytic phosphoprotein PEA-15 is encoded by two mRNAs conserved on their full length in mouse and human. *J. Biol. Chem.*, **271**, 14800–14806.

Figueroa,A. *et al.* (2004) Clustering binary fingerprint vectors with missing values for DNA array data analysis. *J. Comput. Biol.*, **11**, 887–901

Gonsalves,J. and Madeira,S. (2010) e-BiMotif: combining sequence alignment and biclustering to unravel structured motifs. *Adv. Bioinformatics.*, **74**, 181–191.

Haibing,L. *et al.* (2008) Optimal Boolean matrix decomposition: application to role engineering. In *IEEE 24th International Conference on Data Engineering, ICDE 2008*. IEEE Computer Society Washington, DC, USA, pp. 297–306.

Harpaz,R. *et al.* (2011) Biclustering of adverse drug events in the FDA's spontaneous reporting system. *Clin. Pharmacol. Ther.*, **89**, 243–250.

Kerr,G. *et al.* (2008) Techniques for clustering gene expression data. *Comput. Biol. Med.*, **38**, 283–293.

Kohonen,T. (1997) *Self-Organizaing Maps. Series in Information Sciences*. Springer, Heidelberg, p. 30.

Koyuturk,M. *et al.* (2004) Biclustering gene-feature matrices for statistically significant dense patterns. *Comput. Syst. Bioinformatics Conf.*, 480–484.

Madeira,S. and Oliveira,A. (2009) Efficient biclustering algorithms for time series gene expression data analysis. *Lectur. Notes Comput. Sci.*, **5518**, 1013–1019.

Mimaroglu,S. and Simovici,D. (2007) Bit sequences and biclustering of text documents. In *Seventh IEEE International Conference on Data Mining Workshops*, IEEE Computer Society Washington, DC, USA, pp. 51–56.

Perco,P. *et al.* (2005) Detection of coregulation in differential gene expression profiles. *BioSystems*, **82**, 235–247.

Pomeroy,S.L. *et al.* (2002) Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, **415**, 436–442.

Prelic,A. *et al.* (2006) A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, **22**, 1122–1129.

Puolamaki,K. *et al.* (2006) Seriation in paleontological data using Markov Chain Monte Carlo Methods. *PLoS Comput. Biol.*, **2** [Epub ahead of print, doi:10.1371/journal.pcbi.0020006].

Serin,A. and Vingron,M. (2011) DeBi: discovering differentially expressed biclusters using a frequent itemset approach. *Algorithms Mol. Biol.*, **6**, 18.

Shmulevich,I. and Zhang,W. (2002) Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics*, **18**, 555–565.

Smedley,D. *et al.* (2009) BioMart - biological queries made easy. *BMC Genomics*, **10**, 22.

Sun,X. and Nobel,A. (2008) On the size and recovery of submatrices of ones in a random binary matrix. *J. Mach. Learn. Res.*, **9**, 2431–2453.

Uitert,M.v. *et al.* (2008) Biclustering sparse binary genomic data. *J. Comput. Biol.*, **15**, 1329–1345.

Yokota,N. *et al.* (1996) Predominant expression of human Zic in cerebellar granule cell lineage and medulloblastoma. *Cancer Res.*, **56**, 377–383.

Zhang,Z. *et al.* (2010) Binary matrix factorization for analyzing gene expression data. *Data Min. Knowl. Discov.*, **20**, 28–52.