BiGGEsTS
Integrated environment for biclustering analysis of time series gene expression data


Joana P. Gonçalves∗1,2,3, Sara C. Madeira1,2,3, Arlindo L. Oliveira1,2

1Knowledge Discovery and Bioinformatics (KDBIO) group, INESC-ID, Rua Alves Redol, Apartado 13069, 1000-029 Lisboa, Portugal
2Instituto Superior Técnico, Technical University of Lisbon, Av. Rovisco Pais, 1049-001 Lisboa, Portugal
3University of Beira Interior, Rua Marquês d’Ávila e Bolama, 6201-001 Covilhã, Portugal

Email: Joana P. Gonçalves∗- jpg@kdbio.inesc-id.pt; Sara C. Madeira - smadeira@di.ubi.pt; Arlindo L. Oliveira - aml@inesc-id.pt;

∗Corresponding author

Abstract

Background: The ability to monitor changes in expression patterns over time, and to observe the emergence of coherent temporal responses using expression time series, is critical to advance our understanding of complex biological processes. Biclustering has been recognized as an effective method for discovering local temporal expression patterns and unraveling potential regulatory mechanisms. The general biclustering problem is NP-hard. In the case of time series this problem is tractable, and efficient algorithms can be used. However, there is still a need for specialized applications able to take advantage of the temporal properties inherent to expression time series, both from a computational and a biological perspective.

Methods: BiGGEsTS makes available state of the art biclustering algorithms for analyzing expression time series. Gene Ontology (GO) annotations are used to assess the biological relevance of the biclusters. Methods for preprocessing expression time series and post-processing results are also included. The analysis is additionally supported by a visualization module capable of displaying informative representations of the data, including heatmaps, dendrograms, expression charts and graphs of enriched GO terms.

Results and Conclusions: BiGGEsTS is a free open source graphical software tool for revealing local coexpression of genes in specific intervals of time, while integrating meaningful information on gene annotations. It is freely available at: http://kdbio.inesc-id.pt/software/biggests. We present a case study on the discovery of transcriptional regulatory modules in the response of Saccharomyces cerevisiae to heat stress.
Background

Extracting relevant biological information from expression data provides important insights into the relations between genes participating in biological processes. This information can be used to identify co-regulated genes corresponding to transcriptional regulatory modules, thus contributing to the challenging goal of regulatory network inference.

Processing expression data is time and resource consuming. In this context, the development of novel computational algorithms and tools for expression data analysis is primarily focused the efficiency and robustness. Clustering techniques have been extensively applied to both dimensions of expression matrices separately, focusing on either gene or sample expression patterns. However, many patterns are common to a subset of genes only in a specific subset of experimental conditions. In fact, our general understanding of cellular processes leads us to expect subsets of genes to be coexpressed only in certain experimental conditions, but to behave almost independently in other. These local expression patterns can only be discovered using biclustering techniques [1, 2], which may be the key for uncovering many regulatory mechanisms that are not apparent otherwise [3]. Although the majority of the biclustering formulations are NP-hard [1], the biclustering problem becomes tractable when expression levels are measured over time, restricting the analysis to biclusters with consecutive time points [4–7].

BiGGEsTS (BiclusterinG Gene Expression Time Series) is a free and open source graphical application using state of the art biclustering algorithms specifically developed for analyzing gene expression time series. The current version integrates the methods proposed by Zhang et al. [5] and Madeira and Oliveira [6,7]. An alternative approach from Ji and Tan [5] was not included due to complexity issues [6]. The integration of additional algorithms pursuing similar goals is straightforward. In addition, BiGGEsTS offers well known preprocessing techniques to filter genes, treat missing values and to smooth, normalize, and discretize expression data. A visualization module supports the analysis of both data and results. Graphical representations include colored matrices (heatmaps), expression and pattern charts, and dendrograms. Biclusters can be studied using Gene Ontology (GO) annotations. BiGGEsTS is also able to generate ontology graphs representing enriched GO terms, and filter and/or sort biclusters according to several numerical and statistical criteria.
Related tools

Several applications are available for the analysis of gene expression data using clustering [8–15], biclustering [16,17] or both [18,19] approaches. For clustering we highlight Genesis [8], which implements hierarchical clustering, k-means, self organizing maps (SOMs), principal component analysis and support vector machines, together with filtering and normalization methods.

Expander [18] and BicAT [19] offer both clustering (hierarchical and k-means) and biclustering techniques. Expander also performs clustering using SOMs and CLICK [18]. Regarding biclustering, Expander uses SAMBA [20], and BicAT integrates the Cheng and Church approach [2], the Iterative Signature algorithm (ISA) [21], the Order-preserving Submatrix method (OPSM) [3], xMotif [22] and BiMax [19]. Both tools include preprocessing methods such as filtering, normalization, log transformation and discretization. Expander further evaluates the biological relevance of clusters/biclusters by computing the functional enrichment of GO terms and retrieves information on promoter signals.

Few applications actually address the problem of analyzing time series and they typically apply clustering [11–14]. CAGED [11] and GQL [12] model expression profiles using Markov chains. CAGED applies agglomerative clustering to group genes with similar expression profiles, while GQL combines the individual Markov models into a mixture model. STEM [13] uses a greedy clustering algorithm. TimeClust [14] offers hierarchical clustering and SOMs, together with Bayesian and temporal abstraction approaches. To our knowledge, only PAGE [17] provides a biclustering algorithm specifically designed for expression time series, which is a modified version of the approach of Ji and Tan [4]. Most of these applications miss essential preprocessing steps useful to clean and prepare data for analysis.

Methods

This section describes the functionalities of BiGGEsTS from a biology/medical researcher’s perspective, providing further insight into the underlying methods. The graphical user interface (GUI) includes a set of tabs and three panels (Figure 1). Main functionalities are directly selected in the tabs and guide the researcher through a complete process of analysis: 1) input and preprocessing of expression data, 2) identification of coexpressed genes in specific subsets of time points using biclustering (biclusters), 3) post-processing of biclusters in order to rank the results, 4) usage of exploratory analysis tools (visualization options, GO annotations and functional enrichment of GO terms).
Figure 1: **Input and preprocessing modules** - This figure shows: (a) The main window of BiGGEsTS and its input module for loading time series gene expression data. The graphical user interface (GUI) includes a set of tabs, for functionality selection, and three panels: a top-left panel displaying the dataset tree, where expression matrices and biclusters are organized; a bottom-left panel displaying a box with information about the selected node in the dataset tree; and a right panel, whose content depends on both the selected node and functionality tab. The navigation on the dataset tree, as well as on the tabs, is intuitive and straightforward. A session can be saved anytime to keep record of data and results. Saved sessions can be loaded later enabling researchers to recover previous stages of their analysis. The input of expression time series is performed using a standard text file. The file contains the elements of the gene expression matrix delimited by a specific character (usually tab), together with additional information about the data, including the organism, and the row and column specifying the names of the time points and the genes, respectively. When the names of the genes used in the biological experiments and their corresponding symbol approved by the Human Genome Organization (HUGO) Gene Nomenclature Committee (HGNC) differ, the researcher may want to provide an additional file. This input file is optional, since it is only required for retrieving the gene annotations and assessing the biological relevance of the biclusters. (b) The preprocessing module for filtering genes, filling missing values, normalizing, smoothing and/or discretizing gene expression data. Available preprocessing techniques are described in the Quickstart Guide.
Input and preprocessing of time series gene expression data

The input of expression time series is straightforward (Figure 1(a)) and is usually followed by a set of preprocessing steps (Figure 1(b)). These handle occasional and systematic errors, reduce noise, and prepare data to be analyzed.

Occasional errors may occur when measuring the abundance of mRNA in cells, leading to missing values, not always supported by biclustering algorithms. This can be addressed by filtering all genes with missing values, thus eliminating all rows with at least one missing value, and may be regarded as a good strategy to reduce noise. However, when analyzing a small number of genes, removing some of them can lead to a significant reduction in the dimension of the dataset, potentially compromising further analysis. The tradeoff between the elimination of missing values and the dimension of the dataset is usually mitigated by establishing an upper bound for the percentage of missing values allowed per gene. Genes with percentages higher than a user-defined threshold are filtered. The remaining missing values must be filled.

Systematic errors, on the other hand, affect every measurement action and are associated with the differences between the experimental settings of each trial. Sources of this kind of errors include the different incorporation efficiency of dyes, and the different scanning and processing parameters of distinct experiments. Normalization is a widely used technique, which attempts to compensate for these systematical differences between time points and highlight the similarities and differences in the expression profiles. Additionally, a smoothing algorithm acts as a low-pass filter, attenuating the effect of outliers.

Depending on the biclustering algorithm, it may be necessary to discretize data, reducing the range of expression values to an adequate set of discrete values.

Biclustering

Three biclustering algorithms are available: CCC-Biclustering [6], $e$-CCC-Biclustering [7] and CC-TSB [5] (Figure 2(a)). The first two process discretized matrices, while the third uses real-valued data.

CCC-Biclustering uses a generalized suffix tree to identify, in time linear on the size of the expression matrix, all maximal biclusters with contiguous columns that exhibit coherent expression evolutions over time. In a CCC-Bicluster, all genes have exactly the same discretized expression pattern.

$e$-CCC-Biclustering extracts all maximal CCC-Biclusters with approximate expression patterns in time polynomial on the size of the expression matrix. The expression patterns in an $e$-CCC-Bicluster may vary from one gene to another, as long as the number of errors between each pattern and the pattern profile does not exceed a predefined value. Two kinds of errors are supported: general and restricted. General errors
identify measurement errors and allow symbols to be substituted by any other symbol in the discretization alphabet. Restricted errors identify discretization errors and only consider as valid the substitutions of symbols by a predefined number of neighbors in the discretization alphabet.

Both CCC-Biclustering and e-CCC-Biclustering are provided with three additional extensions that identify biclusters with shifted/scaled, anti-correlated and time-lagged patterns [23]. Sometimes, distinct genes exhibit similar expression evolutions at different expression levels, thus not reflecting a similar pattern after discretization. This problem is addressed by identifying biclusters with shifted patterns. Anti-correlation allows genes with opposite expression patterns, in a set of consecutive time points, to be included in the same bicluster. The time-lagged approach identifies genes that exhibit similar expression patterns starting at different time points, enabling the identification of activation/inhibition delays.

CC-TSB is an adaptation of the biclustering algorithm proposed by Cheng and Church [2]. This heuristic approach uses the mean squared residue (MSR) as merit function and iteratively alternates the addition/removal of genes/time points, forcing the MSR to reduce. The addition/removal of time points is restricted to discover only biclusters with contiguous columns.

**Post-processing**

Applying biclustering to expression data often yields a large number of biclusters. Since analyzing all is usually prohibitive, post-processing techniques are performed in order to rank biclusters according to their relevance. Several methods are available to filter and sort biclusters based on numerical and statistical criteria (Figure 2(b)).
Figure 2: **Biclustering and post-processing modules** - This figure shows: (a) biclustering and (b) post-processing modules. The biclustering module is used to select the biclustering algorithm to be applied to the expression matrix. Additional extensions enabling shifted, anti-correlated and time-lagged patterns are available in CCC-Biclustering and e-CCC-Biclustering. Different types of errors are supported in e-CCC-Biclustering. The post-processing module enables the researcher to select and apply filtering and sorting techniques to groups of biclusters. Biclusters can be filtered by setting a threshold for the number of genes and/or conditions, size, average column variance, average row variance, mean-squared residue score, and overlapping percentage of genes and/or conditions. It is also possible to remove biclusters with constant or statistically non significant patterns. Biclusters may additionally be sorted using their best functional enrichment p-value, statistical significance of expression pattern, average column or row variance, mean-squared residue score and a number of other measures available for selection. Details on biclustering and post-processing techniques are described in the Quickstart Guide.
**BiGGEsTS: BiclusterinG Gene Expression Time Series**

**Biological analysis**

Biclustering groups genes and conditions based on relations inferred from data, relying strictly on computational methods. Researchers are usually interested in analyzing the results looking for statistically significant biological phenomena. This significance can be assessed using Ontologizer’s term-for-term analysis [24], which computes the functional enrichment of the genes in the biclusters by identifying the overrepresented GO terms. In a first step, GO annotations are extracted requiring two distinct files (downloadable from the GO repository if not available) containing the complete ontology and organism-dependent annotations. Term-for-term analysis is then applied to compute a p-value for each GO term. Such p-value is calculated with respect to the null hypothesis that the inclusion of genes follows a hypergeometric distribution, and measures the statistical significance of each term by computing the ratio of the frequencies of annotated genes in the bicluster and in the complete dataset. The Bonferroni correction for multiple testing is applied. The lower the p-value, the more significant the term is. According to standard statistical practice, terms with a corrected p-value lower than 0.05 and 0.01 are considered significant and highly significant, respectively.

**Visualization**

A visualization module provides different graphical representations of expression time series, enhancing their most important features: tables of values, colors and symbols; dendrograms; expression charts; pattern charts; tables of GO terms and functional enrichment; and graphs of enriched terms.

**Expression matrices and heatmaps**

Expression matrices are displayed as tables of values (Figure 3(a)). Tables of colors are commonly known as heatmaps. They resemble tables of values, although each cell is given a different color according to the expression value it contains (Figure 3(b)). Tables of symbols are heatmaps for expression matrices with discrete values. Each cell is given a different color depending on the discretization symbol (Figure 3(c)).

Expression tables share additional functionalities. They can be sorted by the values of a given column by clicking on the corresponding table header, provide access to the GO terms that annotate each gene by clicking on its row, and be exported as PNG or JPEG image files by clicking the right button of the mouse over the table and selecting the “Export as image” menu item. GO terms annotating a given gene are displayed in a popup (Figure 3(d)). Since this information is automatically parsed from the GO files (using Ontologizer), no annotations are shown if the GO files are not available. In the latter case, GO files can be downloaded from the GO repository when an Internet connection is available.
Dendrograms

Dendrograms are branching tree-like diagrams used for representing similarity relationships between the genes/time points in the expression data (Figure 3 (e)). The similarity hierarchy is obtained using agglomerative hierarchical clustering to group genes and/or time points. At each step, the cluster pairwise similarity is used to decide which clusters to merge. For single element clusters, this similarity is computed using a distance measure (Euclidean or cityblock) or a correlation coefficient (uncentered, Pearson’s, absolute uncentered, absolute Pearson’s, Spearman’s or Kendall’s correlation). Otherwise, a single-linkage, complete-linkage, average-linkage or centroid-linkage approach is used.

Java TreeView [25] is used to interpret hierarchical clustering results and display the dendrograms.

Expression and pattern charts

Expression and pattern charts show the evolution of the expression level of the genes in the biclusters over time, using the corresponding submatrices. Expression charts are obtained using real-valued matrices, while pattern charts are generated from discretized matrices (Figure 4).

Expression charts can be displayed using either the subset of time points in the bicluster, or all the time points in the dataset. The latter are particularly suited for highlighting the coherent behavior of the genes in the bicluster time points as opposed to the uncorrelated behavior in the remaining time points.

Both expression and pattern charts provide a context menu with extra functionalities, including displaying and modifying chart properties, saving chart to an image file, printing, and zooming in or out.
Figure 3: Expression matrix, heatmaps, GO annotations, and dendrograms - This figure shows: (a) tables of values, (b) tables of colors, (c) tables of symbols, (d) list of GO terms annotating a gene, and (e) dendrograms. In the tables of values, the names of the experimental conditions appear in the first row and usually correspond to consecutive instants in time. The first column displays the names of the genes. Each remaining cell in the table contains the expression value of a given gene in a specific condition. In the tables of colors, cells with high expression values are, by default, colored red, while the ones with low expression values are given a green color. Cells holding the mean value are colored black. The intensity of the color is set according to the actual expression value of each cell, thus generating a scale of reds and greens for all possible expression values. Cells with no expression value, that is, a missing value, are given a yellow color. Tables of symbols resemble tables of colors and are computed using a discretized version of the expression matrix. The GO terms listed as annotations of a given gene correspond to the most specific GO terms (before applying the true path rule). Dendrograms are visualized using Java TreeView [25] in a separate window. They are displayed together with the expression matrix and enable the researcher to individually select clusters, which are then displayed in a separate panel. The researcher may further change the settings of the dendrogram, search for genes or conditions within the data, compare with other hierarchical clustering results and export both the dendrogram and the gene expression matrix as vector (PS) or raster (PNG, PPM, JPG) image files.
Figure 4: **Expression and pattern charts** - This figure shows examples of expression and pattern charts of (a) CCC-Biclusters, (b) CCC-Biclusters with anti-correlated patterns, and (c) CCC-Biclusters with time-lagged patterns. In expression charts, expression values can be normalized on the fly by checking the “Normalize to mean 0 and std 1” checkbox.
Go terms and functional enrichment

The GO terms that annotate the genes in the dataset can be displayed in a table, where each row corresponds to a GO term and contains: the GO term ID, the term name, and the total number of genes annotated with it. In the case of a bicluster (Figure 5(a)), each row in the table further includes the number of genes in the bicluster annotated with the term, the \( p \)-value computed using the term-for-term analysis, and its Bonferroni corrected value. Enriched terms, given a threshold (default value is 0.01), are highlighted. Additionally, the list of genes annotated with each term is displayed in a popup window by clicking the corresponding row using the left button of the mouse.

Graphs of enriched terms

Term-for-term results can be used to generate tree structured graphs highlighting the enriched terms in each of the three GO ontologies (Figure 5(b)). Graphs of enriched terms are generated using Ontologizer [24], which outputs the structure of the graph into a text file. Graphviz [26] is used to convert the text file into an SVG file describing the same graphical structure using the XML standard. Finally, the Batik SVG Toolkit [27] is used to interpret the SVG file and display the corresponding image. The graph with the enriched terms can be zoomed in or out and saved as a raster (PNG) or vector (SVG) image file.
Figure 5: Term-for-term analysis and graph of enriched GO terms - This figure shows: (a) A summary of the results of the term-for-term analysis applied to a given bicluster. The list of genes annotated with GO term highlighted in blue is displayed at left. The GO terms highlighted in green correspond to highly significant terms. (b) A graph displaying the distribution of the biological terms in the ontology of GO terms. Enriched terms are colored in purple, yellow or green whether they specialize from cellular component, molecular function or biological process, respectively. The intensity of the color depends on the Bonferroni corrected \( p \)-value of the corresponding term: the lower the \( p \)-value, the more intense the color of its node. Arrows define specialization relations: each arrow goes from a more general term to its specialization(s).
Results and Conclusions

BiGGEsTS is a software for analyzing gene expression time series using biclustering. It was designed to comply with the broad specifications of a software tool, essentially focused on user-friendliness, platform independence, modularity, reusability and efficiency.

We present a case study (Appendix A), describing how to use the software to discover transcriptional regulatory modules in a dataset containing the response of *Saccharomyces cerevisiae* to heat stress, and reproducing the results published in [6].

Availability and requirements

- Project name: BiGGEsTS - BiclusterinG Gene Expression Time Series
- Project home page: http://kdbio.inesc-id.pt/software/biggests/
- Operating systems: Platform independent
- Programming language: Java
- Other requirements: Java 1.5 or higher, 1024 MB of RAM, Graphviz (in OSs other than Windows and Mac OS)
- License: GNU GPL version 3 or higher

Authors’ contributions

SCM implemented the techniques for preprocessing gene expression data, the biclustering algorithms and the post-processing approaches. JPG and SCM designed the software. SCM supervised the implementation of the software. JPG implemented the software and integrated the preprocessing, biclustering, and post-processing methods. Additionally, JPG produced the documentation, created the website, and wrote the first draft of the manuscript. All authors worked together in the final version of the manuscript. All authors approved the final manuscript.

Acknowledgements

This work was partially supported by projects ARN - Algorithms for the Identification of Genetic Regulatory Networks, PTDC/EIA/67722/2006, and Dyablo - Models for the Dynamic Behavior of Biological Networks,
PTDC/EIA/71587/2006, funded by FCT, Fundação para a Ciência e a Tecnologia. The work of JPG was partially supported by FCT grant SFRH/BD/36586/2007.

References


27. Batik - Java SVG Toolkit [http://xmlgraphics.apache.org/batik/].


APPENDIX A

Discovering transcriptional regulatory modules using BiGGEsTS: a case study

Dataset

Throughout the case study we analyze the **Yeast Stress** dataset, used by Madeira et al. [6], and directly downloadable from the CCC-Biclustering website [28]. This dataset, derived from the experiment identified as “heat shock 2” in the original group of datasets from Gasch et al. [29], concerns the response of 6142 *Saccharomyces cerevisiae* genes to heat shock, and comprises five different time points along the first hour of exposure to 37 °C (0, 5, 15, 30 and 60 minutes).

**Loading the dataset and performing a preliminary analysis**

Load the file **yeast_stress.txt**, containing the **Yeast Stress** dataset, using the “Loading” tab and the default options. The corresponding table of colors is now displayed in the “Analyzing” tab (Figure 6). The dataset is identified by the node “Dataset 0” in the tree shown in top left panel. The bottom left panel contains information about the dataset: a cDNA expression matrix from *Saccharomyces cerevisiae* with 6142 genes and 5 time points, and contains missing values (colored yellow in the table of colors).

![Figure 6: Table of colors.](image-url)
Perform a preliminary analysis of the dataset, not only by inspecting the table of colors, but also by analyzing the table of values (Figure 7), the dendrogram (Figure 8), and the table with the list of GO terms annotating the genes in the dataset, together with the set of genes annotated with a specific term of interest (Figure 9).

Figure 7: Table of values.
Figure 8: Dendrogram.
Preprocessing the dataset

Preprocess “Dataset 0” as shown in Figure 10(a). As in [6], the genes with missing values are removed, and the gene profiles are normalized to zero mean and standard deviation. Since the biclustering algorithm used in this case study, CCC-Biclustering, processes discretized matrices, the dataset is discretized using a three symbol alphabet \{D, N, U\}, where “D”, “N” and “U” stand for down-regulation, no-change and up-regulation, respectively. The technique identified as “variation between time points (3 symbols)”, originally proposed by Ji and Tan [4], is used with the corresponding threshold set to 1.

Figure 10(b) shows the table of symbols, representing the discretized version of the expression matrix in “Dataset 0”, “Preprocessed Matrices”, identified as “Preprocessed 0”. In this table, the colors green, black, and red, identify the symbols “D”, “N” and “U”, respectively. Note that a node representing this discretized matrix, named “Discretized 0”, was created in the tree on the top left panel.
Figure 10: Preprocessing.
**Biclustering yeast stress time series**

Apply CCC-Biclustering (without extensions) to the discretized matrix “Discretized 0”, as shown in Figure 11(a). This algorithm identifies 167 biclusters (Figure 11(b)). These biclusters can be visualized by expanding the node identified as “Biclusters Group 0” in the tree on the top left panel, which represents the group of biclusters discovered by CCC-Biclustering when applied to “Discretized 0”. Note that the biclusters are listed according to the order they are extracted by the algorithm. No quality criterion has been applied yet. Still, we can analyze their matrices of values, colors and symbols, and inspect their expression profiles using expression and pattern charts. These charts can be displayed for the group of biclusters or individual biclusters. We can also analyze GO annotations and functional enrichment results. However, we postpone these analyses until we perform the post-processing steps described in [6].
Figure 11: Biclustering.
Post-processing biclustering results

Post-processing techniques are performed in order to rank biclusters according to their relevance. In [6], the authors were interested in biclusters with high statistical significance. As such, in a first post-processing step, the set of 167 biclusters discovered using CCC-Biclustering were filtered using their \( p \)-values, computed as described in [6]. All biclusters which were not highly significant at the 1\% level after applying the Bonferroni correction for multiple testing were discarded. The remaining biclusters were then sorted in ascending order of their \( p \)-values.

Figure 12 shows how to apply these filtering and sorting steps in BiGGEsTS. The filtering technique “pattern w/columns (not significant)”, applied with the parameters Markov chain order and \( p \)-value threshold set to 1 and 0.01, respectively, correspond to computing the \( p \)-value of each bicluster using a first order Markov chain, and discarding those whose \( p \)-value does not pass the statistical significance test at the 1\% level, after Bonferroni correction for multiple testing. The sorting technique “bicluster pattern w/ columns \( p \)-value” ranks the biclusters surviving the filtering step using their \( p \)-values. The resulting group of biclusters is identified as “Post-Processed Group 0” in the tree on the top left panel.

In order to avoid the analysis of highly overlapping biclusters, the similarities between the sorted biclusters were computed using the Jaccard similarity score, and biclusters with similarity greater than 25\% were filtered, in a second post-processing step. This can easily be performed in BiGGEsTS by post-processing the “Post-Processed Group 0” using the filtering technique “overlapping genes and time points” and setting the similarity threshold to 25 (Figure 13). The resulting group of biclusters, identified as “Post-Processed Group 1” in the tree on the top left panel, corresponds to the set of biclusters analyzed in [6], and will now be subject to further analysis.

Figure 14 shows the expression profiles of these biclusters using expression and pattern charts. Two types of chronological expression patterns can be identified: transcriptional up-regulation patterns (biclusters 14, 27, 39, 48, 43, 83, 42, 79 and 92) and transcriptional down-regulation patterns (biclusters 107, 163, 151, 142, 147, 148, 159 and 99). A subset of these biclusters were analyzed in detail in [6], using GO annotations together with information about transcriptional regulations.

In this case study, the biological analysis is focused on bicluster 14, the top bicluster in the ranking, which has 1091 genes and 4 time points. Figure 15 shows its expression profile, describing a transcriptional up-regulation pattern with a short delay, between 5 and 15 minutes of exposure to heat shock, classified as “middle up regulation” in [6].
Figure 12: Post-processing step 1: filtering biclusters with Bonferroni corrected $p$-values greater than 0.01 and sorting the remaining ones in ascending order of these $p$-values.
Figure 13: Post-processing step 2: filtering biclusters with similarities above 25%. This filtering processing is applied to the biclusters resulting from step 1. These are thus sorted in ascending order of $p$-value before the overlapping filter is applied and keep their ranking after having been filtered.
Figure 14: Biclusters: Expression and pattern charts.
Figure 15: Bicluster 14: Expression chart.
**Biological analysis**

Figure 16 shows the result of the functional enrichment analysis performed for the biclusters in “Post-Processed Group 1”. Note that even when there are no significant terms associated with a given bicluster, whose expression pattern was considered as statistically significant (Figure 16), the bicluster should be further analyzed since it might as well identify biologically relevant phenomena. This was in fact shown in [6], when performing the biological analysis of this group of biclusters.

Figure 17 shows the detailed functional enrichment analysis performed for bicluster 14, together with the set of genes annotated with the GO term “response to stress”. Figure 18 displays the graph of enriched terms, highlighting the terms in the ontology “Biological Process”, which are the relevant ones when the aim of the biological analysis is the discovery of transcriptional regulatory modules. A similar GO-based analysis can be performed for the other biclusters in “Post-Processed Group 1”.

Focusing on bicluster 14, and on GO terms in the “Biological Process” ontology, the GO-based analysis reveals the occurrence of highly significant terms, including “carbohydrate metabolic process” or “energy derivation by oxidation of organic compounds”, related to energy generation, and “response to stimulus” or “response to stress”, related to the cellular response to heat shock. These terms are consistent with the induction of protein folding chaperones aiming at protecting against, and recovering from, protein unfolding with associated energetic expenses. The transcriptional induction of genes involved in alternative carbon source metabolism and respiration, in the presence of glucose, is considered a consequence of a sudden decrease in cellular ATP concentration, caused by ATP-consuming stress defense mechanisms [6,29].
Figure 16: Biclusters: Functional enrichment.
Figure 17: Bicluster 14: Functional enrichment.
Figure 18: Bicluster 14: Graph of enriched terms.