Proceedings of the Federated Conference on Computer Science and Information Systems pp. 1375–1378
DOI: 10.15439/2016F129
ACSIS, Vol. 8. ISSN 2300-5963

QtBiVis: a software toolbox for visual analysis of biclustering experiment

Artur Pańszczyk
AGH University of Science and Technology,
Dep. of Automatics and Bioengineering,
al. Mickiewicza 30,
30-059 Kraków
Email: panszczyk.artur@gmail.com

Patryk Orzechowski
AGH University of Science and Technology,
Dep. of Automatics and Bioengineering,
al. Mickiewicza 30,
30-059 Kraków,
Email: patrick@agh.edu.pl

***Abstract*—In this article we introduce QtBiVis - a novel software intended for the comparative analysis of biclustering results. This modular tool has been efficiently implemented in C++ with Qt framework GUI. It may be successfully used for coverage analysis of the results of biclustering as well filtering or sorting biclusters by Gene Ontology (GO) identifiers or bicluster enrichment values. It may also be useful for parameter studies of biclustering algorithms. In future releases we plan to add different modules for visualizing and comparing different GO terms and biclusters.**

I. Introduction

MICROARRAY technology has become a subject of multiple biological experiments since its theoretical foundations in 1980's and first application in 1995 [1]. As labeled nucleic acids immobilized on a solid surface proved to be capable of monitoring the expression levels of nucleic acids molecules, the technology has been predominately used for measuring in parallel multiple gene expression patterns. It has also gained wide scope of application in disease diagnostics, drug discovery and comparative genomics.

The result of microarray experiment after background adjustment, normalization and summarization at the probe level is structured into a data matrix of real numbers, in which each value corresponds typically to a gene expression level under the specified condition. The whole microarray data may contain up to tens of thousands of gene expressions. Biclustering algorithms have been applied to identify groups of genes that show resemblance under particular subsection of conditions. Multiple biclustering methods have been developed so far [2], [3], [4]. Different metrics have been adapted to measure gene expression level [5], [6]. The collection of the most recognizable biclustering approaches applied to GDS datasets is included in Eren et. al. [7].

A. Methods of visualization of biclusters

The most popular way to visualize a single bicluster uses a heatmap, in which cells are colored based on the gene expression level. This perspective has been implemented in multiple software tools among which the most popular are BiVoc [8], BiVisu [9], BicAT [10] and its extension BicAT-Plus [11], BicOverlapper [12], [13], Furby [14], Bicluster Viewer [15] or BiGGEsTS [16]. A single bicluster is usually resorted and drawn in the upper left corner with its rows and columns rearranged.

The second popular visualization method, which is very useful for gene expression profile comparison, uses parallel coordinates, in which conditions are vizualized on horizontal axis. Gene profiles are represented by lines, which join corresponding values of corresponding conditions. This perspective is popularly used by multiple software tools (including BiVisu, BicAT, BicOverlapper or Bicluster Viewer).

A widespread visualization method for multiple biclusters uses heatmaps or heatmaps with dendograms. For example hierarchical biclustering is represented by a tree and a heatmap, in which rows are reordered to fit the recognized bicluster. This perspective is offered for example by BiGGesTS.

There are also more sophisticated visualization perspectives, which are provided by some of the available tools. For example BicOverlapper offers a transcription regulatory networks view, wordcloud or bubble map perspective. Furby presents graphically the attracting force between each pair of the biclusters. This resembles a class diagram, which is popular in relational databases. The number of rows and columns shared between each pair of biclusters are reflected by the transparency of the interconnecting line.

B. Motivation

The major motivation for QtBiVis is to provide a fast and reliable software, which would be able to support the researchers in parameter study of the biclustering methods. We believe that the analysis of biclustering experiments based on the inspection of biclusters' coverage may provide a novel insight for assessment of biclustering methods capabilities. It may also become helpful in determining the optimal setting of parameters for each algorithm.

The second justification of QtBiVis emergence are limitations of the available software. As the majority of the tools has been implemented in Java, existing tools encounter different performance issues during analyzis of hundreds or thousands of biclusters. This limits analysis of complex experiments in which thousands of biclusters need to be visualized and compared simultaneously. QtBiVis has been implemented in C++ with Qt used for graphical user interface.

978-83-60810-90-3/$25.00

II. METHODS

In this article we present QtBiVis, an open-source toolbox, which implementation may be found at github.com/Archi0/QtBiVis. In this section we present an overview of QtBiVis and detailed information about its design.

A. Main features of QtBiVis

The QtBiVis tool, which is herein presented, has been implemented in C++ programming language with Qt 5.5 framework used for graphical interface design. A Dynamic_bitset from Boost C++ library has been used for performing bitwise operations. The main features of QtBiVis include:

- loading main data set which contains microarray data,
- loading files with biclusters with Gene Ontology ID (GO ID) and p-values,
- filtering biclusters by a specific GO ID,
- calculations, plotting and saving results of the degree of coverage of the bicluster environment,
- displaying information about a single bicluster with its values, labels, level of coverage, GO IDs and p-values,
- plotting statistics of the bicluster based on average of values in columns and standard deviations of the values in columns in examined cluster,
- calculating, plotting and saving information about the relation between number of occurrences of value in different biclusters and size of the bicluster,
- drawing a heatmap based on the number of occurrences of a given value in the loaded clusters.

B. Overview of application

The main workflow of QtBiVis includes loading a microarray dataset and definitions of series of biclusters from multiple biclustering experiments. Information about loaded biclusters is shown in the table on the right-hand side of the main window (see Fig. 1). Each row of the table contains a Gene Ontology ID, p-values (before and after multiple test correction) and the identifiers of rows and columns of the bicluster. Filtering is applied for biclusters after entering a value in "GO Filter" text area.

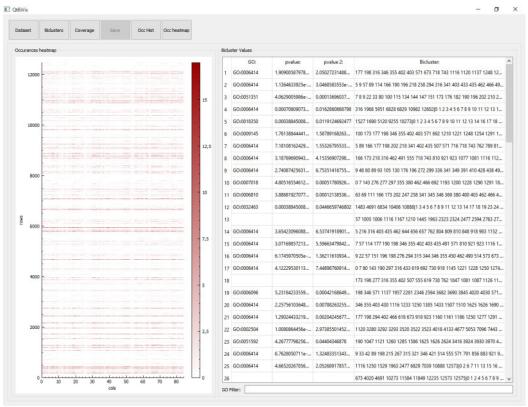

Fig. 1. The main window of QtBiVis.

The application provides access to a standard overview of a single bicluster (i.e. its expression values, labels of rows and columns and parallel coordinates plot), as well as tools for visual comparison of relations between multiple biclusters. This includes the statistics of coverage and the frequency of occurrences of expression values presented in form of heatmap and histogram.

C. Analysis of a single bicluster

A single bicluster analysis overview, which is demonstrated in Fig. 2, contains biclusters values with rows and columns names, gene ontology ID's with p-values before and after correction and neighborhood of the selected bicluster with percentage values.

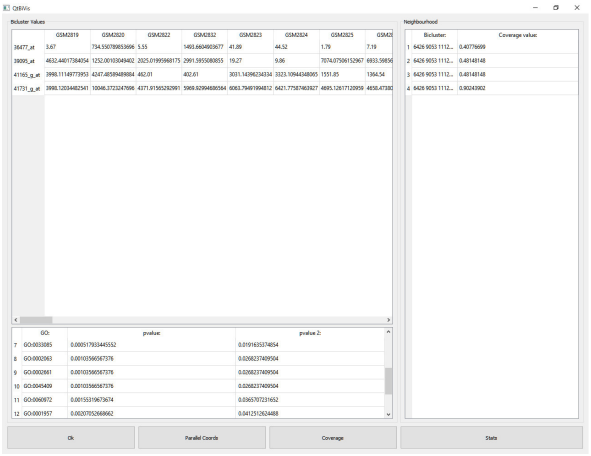

Fig. 2. A single bicluster overview.

This perspective offers access to three different analysis tools. The first one, called "Parallel Coords", provides access to profile analysis using parallel coordinates plot (see Fig. 3).

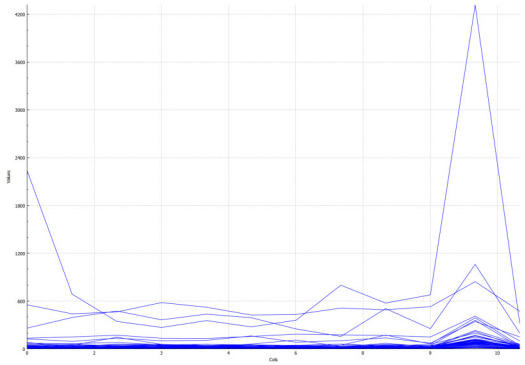

Fig. 3. Gene profile analysis of a single bicluster.

The second one, named "Coverage", displays a histogram, which presents the level of bicluster overlap with respect to other biclusters (see Fig. 4). A histogram presents on horizontal axis a degree of coverage (i.e. a percentage of shared area with the bicluster) and on vertical axis - a number

of occurrences (i.e. number of biclusters that share the same area).

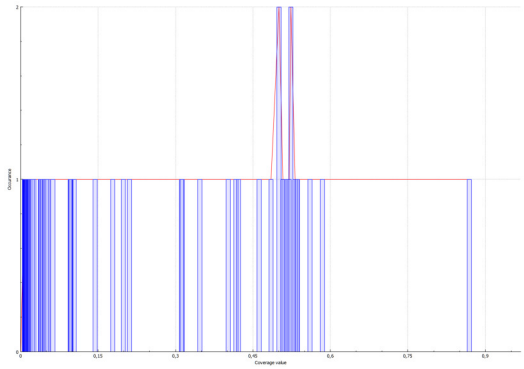

Fig. 4. Coverage analysis of a single bicluster.

The third one, "Stats", shows means and standard deviations of particular columns of a bicluster. In current version, for each column the plot displays mean and standard deviations as dots, whilst the average standard deviation of all columns is represented by a line (see Fig. 5).

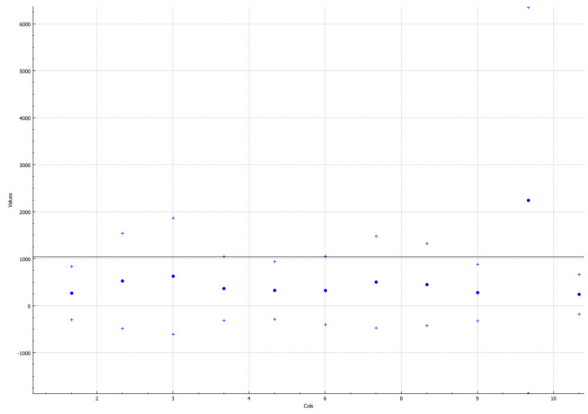

Fig. 5. Statistics of mean and standard deviation of the whole bicluster and its values in columns.

D. Coverage statistics

Another statistics, which is provided by QtBiVis, is analysis of the degree of bicluster intersection. Degree of overlap for two biclusters (A and B) is determined by Jaccard index (1), which takes into account the number of intersecting biclusters' elements with respect to the total area occupied by both biclusters.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

Clicking on the "Coverage" button on main window calculates the percentage coverage for every loaded bicluster, which is presented on a histogram (see Fig. 6). The button "Save" stores the results in a selected file.

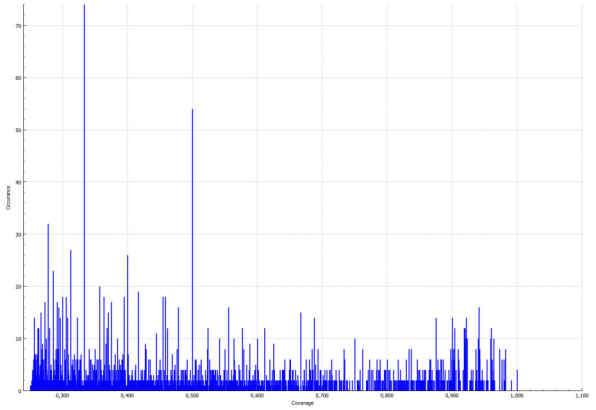

Fig. 6. A histogram presenting the degree of coverage of multiple biclusters with each other.

III. Implementation

QtBiVis has been designed as a modular application. Each module is responsible for providing a different perspective. Several coding optimizations have been used to reduce the computation time of certain calculations. For example row and column of a bicluster are represented in application by bits. If the row or column belongs to a bicluster it is set to '1', otherwise it remains '0'. Bitsets are used for computations of their intersections or unions.

A. Application design

The class diagram of the application is presented in Fig. 7. The main components of the application have been presented hereafter.

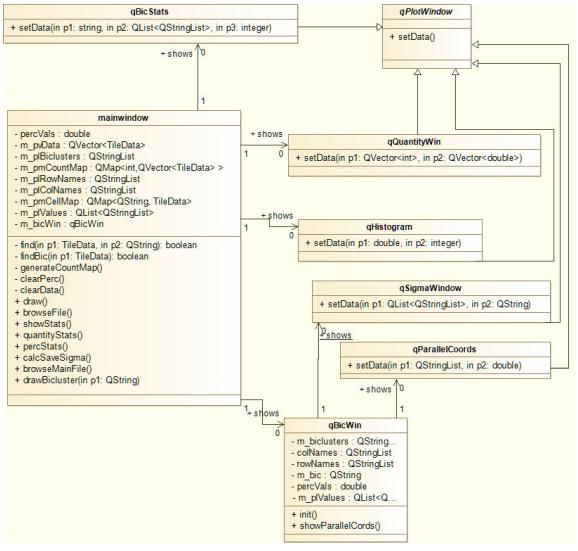

Fig. 7. Class Diagram of QtBiVis.

1) MainWindow: The MainWindow module is responsible for loading microarray data as well as loading, displaying and filtering the biclusters.

2) QBicWin: This module is used for displaying statistics about a single bicluster: its values and rows and columns labels. The module also presents the neighboring biclusters by showing the percentage of coverage. Different modules for single bicluster analysis may be triggered from here (i.e. qBicStats, qParallelCoords and qSigmaStats).

3) QBicStats: This module calculates of degree of coverage for the examined biclusters with the rest of biclusters.

4) QHistogram: The QHistogram module is responsible for calculation and presenting a histogram based on degree of coverage for every detected bicluster.

5) QSigmaStats: QSigmaStats is another module, which calculates and shows common statistics of values in bicluster, such as an average of values, a standard deviation and an average of standard deviation of values in each column of bicluster.

6) QParallelCoords: The QParallelCoords module is responsible for vizualization of a single bicluster with parallel coordinates perspective.

7) QQuantityWin: QQuantityWin is used for displaying the histogram based on the number of occurrences of values in microarray data.

IV. Results

For demonstration purposes eight GDS datasets have been taken, which have been previously used by Eren et al. [7]. As the original dataset haven't been provided by the authors in supplementary materials, we downloaded their copies from Gene Omnibus and tried to follow the authors' preprocessing procedure (i.e. missing value imputation, validation and Benjamini-Hochberg multiple value correction [17]). Unfortunately, we didn't manage to obtain similar results to the authors. The reason for this is that Eren et al. specify neither the method of missing values imputation, nor the method of gene universe creation (i.e. algorithm used for filtering features from an ExpressionSet, which exhibit a little variation or where GO or Entrez Gene identifiers are missing). Thus, we decided to parse the original file with their biclustering results, which have been publicly available. The file has been split into separate folders corresponding to each dataset and divided into the separate files for each biclustering method respectively.

The detailed analysis of the results of coverage of each biclustering method remains out of scope of this paper and has been mentioned only to demonstrate the usage of the QtBiVis.

V. Conclusions & future work

The QtBiVis tool performs extraordinarily fast for analysis of multiple biclusters and their coverage. Its capabilities include filtering by a specific GO term across all detected biclusters and sorting by p-values. Based on this, we hope to assess if the relation between the uniqueness of the detected biclusters and their biological significance exists.

We also plan to use the software for parameter study of biclustering algorithms. By analyzing the results obtained from a series of biclusters, each run with different input parameters, QtBiVis may allow us to support analysis of the most commonly detected biclusters by a specific algorithm. Thus, we hope to assess how the input parameters affect the stability of the results.

The current version of the algorithm doesn't take full advantage of the enrichment level of specific biclusters. In future releases of the software we plan to add different statistics, which would present the impact of the biclustering uniqueness on biological significance. This may be performed for example by visualizing only those biclusters or GO terms, which significance is higher than the adjustable threshold.

Acknowledgments

This research was funded by the Polish National Science Center (NCN), grant No. 2013/11/N/ST6/03204. This research was supported in part by PL-Grid Infrastructure.

References

[1] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary dna microarray," *Science*, vol. 270, no. 5235, pp. 467–470, 1995.

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

[3] 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, p. e90801, 2014.

[4] B. Pontes, R. Giráldez, and J. S. Aguilar-Ruiz, "Biclustering on expression data: A review," *Journal of biomedical informatics*, vol. 57, pp. 163–180, 2015.

[5] P. Orzechowski, "Proximity measures and results validation in biclustering—A survey," in *Artificial Intelligence and Soft Computing* (L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, and J. M. Zurada, eds.), vol. 7895 of *Lecture Notes in Computer Science*, pp. 206–217, Springer Berlin Heidelberg, 2013.

[6] B. Pontes, R. Girldez, and J. S. Aguilar-Ruiz, "Quality measures for gene expression biclusters," *PloS one*, vol. 10, no. 3, p. e0115497, 2015.

[7] K. Eren, M. Deveci, O. Küçüktunç, and Ü. Çatalyürek, "A comparative analysis of biclustering algorithms for gene expression data," *Briefings in Bioinformatics*, 2012.

[8] G. A. Grothaus, A. Mufti, and T. Murali, "Automatic layout and visualization of biclusters," *Algorithms for Molecular Biology*, vol. 1, no. 1, p. 15, 2006.

[9] K.-O. Cheng, N.-F. Law, W.-C. Siu, and T. Lau, "Bivisu: software tool for bicluster detection and visualization," *Bioinformatics*, vol. 23, no. 17, pp. 2342–2344, 2007.

[10] S. Barkow, S. Bleuler, A. Prelić, P. Zimmermann, and E. Zitzler, "Bicat: a biclustering analysis toolbox," *Bioinformatics*, vol. 22, no. 10, pp. 1282–1283, 2006.

[11] F. M. Al-Akwaa, M. H. Ali, and V. M. Kadah, "Bicat_plus: An automatic comparative tool for bi/clustering of gene expression data obtained using microarrays," in *Radio Science Conference, 2009. NRSC 2009. National*, pp. 1–8, IEEE, 2009.

[12] R. Santamaría, R. Therón, and L. Quintales, "Bicoverlapper: a tool for bicluster visualization," *Bioinformatics*, vol. 24, no. 9, pp. 1212–1213, 2008.

[13] R. Santamaría, R. Therón, and L. Quintales, "Bicoverlapper 2.0: visual analysis for gene expression," *Bioinformatics*, p. btu120, 2014.

[14] M. Streit, S. Gratzl, M. Gillhofer, A. Mayr, A. Mitterecker, and S. Hochreiter, "Furby: fuzzy force-directed bicluster visualization," *BMC bioinformatics*, vol. 15, no. Suppl 6, p. S4, 2014.

[15] J. Heinrich, R. Seifert, M. Burch, and D. Weiskopf, "Bicluster viewer: a visualization tool for analyzing gene expression data," in *Advances in Visual Computing*, pp. 641–652, Springer, 2011.

[16] J. P. Gonçalves, S. C. Madeira, and A. L. Oliveira, "Biggests: integrated environment for biclustering analysis of time series gene expression data," *BMC research notes*, vol. 2, no. 1, p. 124, 2009.

[17] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 289–300, 1995.