Biclustering using Biological Context Information

Willi Gierke
Hasso Plattner Institute
Potsdam, Germany
willi.gierke@student.hpi.de

***Abstract*—Recently, RNA sequencing has become the crucial technique to obtain gene expression data. Biclustering has emerged as a powerful tool to find genes with similar expression patterns under specific conditions in this data. However, most approaches are limited as they only rely on patterns that can be found in the data itself. This prevents them from uncovering biological processes that do not manifest in the chosen dataset.**

We propose an algorithm that extends biclustering by including additional biological context information. The presented algorithm is evaluated on a public cancer data set, yielding promising results by finding new patterns and suggesting encouraging possibilities for further research.

***Keywords*-biclustering; pathway**

I. Introduction

With recent advances in sequencing technology, RNA sequencing has become the tool of choice to obtain gene expression data. In comparison to the traditional microarray technology, it yields a higher specificity and sensitivity and it supports detecting weakly expressed genes as well (Zhao et al., 2014). This allows to also study weakly expressed genes whose functions can then be inferred. The underlying assumption is that genes sharing similar expression patterns also contribute to similar biological processes.

An approach to group genes into collections of akin patterns is clustering. The corresponding algorithms make it possible to discover disease-specific genes. However, studies have shown that a gene contributes to ten different biological processes on average, which cannot be reflected by traditional clustering approaches as they assign each gene to a single cluster only. Therefore, single clustering algorithms are not suitable to capture this property. In contrast, biclustering approaches assign genes and samples to multiple clusters. Their main advantage is that they support finding overlapping clusters, a characteristic that reflects the behavior of genes very well. They also simultaneously assign both genes and samples to clusters. This allows interpreting genes of a cluster as contributors to biological processes that are shared across a specific set of samples only. However, most biclustering approaches are limited as they only rely on patterns that can be found in the data itself. Hidden biological processes that are not represented by the sampled data can not be discovered. Moreover, the algorithms can find structures that are not of interest. Due to their strong dependence on the given data set, they can also detect nonexistent patterns that simply result from noise or by accident. Including biological context information can support the algorithm in finding patterns that contribute to the same processes. By adjusting the added context information, we can regularize the algorithm to find structures that are of interest only. The results of such analyses can highly improve the plausibility of the found patterns since they no more only depend on statistical properties of the expression data alone.

We model the problem of finding biclusters as a graph problem while incorporating third-party data from the biological database STRING by adding corresponding edges and searching for maximal cliques.

In this work, we propose an approach that extends the biclustering algorithm *BiMax* by including additional biological context information. The remainder of this paper is structured as follows. Section II describes related literature discussing the targeted problem. In Section III we present the proposed algorithm. Next, Section IV evaluates our approach on a public cancer data set. Section V discusses advantages and limitations of the algorithm. Lastly, Section VI concludes this paper.

II. Related Work

Ponzoni et al. (2014) demonstrate how to combine gene expression data with pathway information to infer association rules between the pathways. For each pathway, they generate a matrix containing expression data of the genes that are involved in the pathway. Lastly, they reduce the matrices using principal component analysis and apply association rule mining to detect relationships between the pathways. While this approach successfully demonstrates how to combine context information with gene expression data, it does not aim at improving biclustering of the mentioned data. Dutta et al. (2012) present an approach that uses both gene expression data and information from neighboring pathways to detect dependencies and relations between pathways. Liu et al. (2012) use gene expression data, protein-protein interactions and cellular pathways to infer dysregulated pathways.

Adding third-party data has also shown great improvements in tasks for single-mode clustering. As presented by Hindle et al. (2011), incorporating meta information of

	G_1	...	G_n
S_1	0.2546451	...	-0.390651
...	...	...	...
S_m	-1.294704	...	0.044671

⇒

	G_1	...	G_n
S_1	1	...	0
...	...	...	...
S_m	0	...	1

Table I
BINARIZATION OF A GENE EXPRESSION MATRIX WITH m SAMPLES AND n GENES USING A THRESHOLD OF $\tau_e = 0$.

videos enabled the clustering algorithm to greatly improve the quality of video clusters.

Prelić et al. (2006) present *Bimax*, a divide-and-conquer algorithm that expects a binary expression matrix. It recursively divides the input matrix into possibly overlapping submatrices until all inclusion-maximal biclusters are found. As shown by Voggenreiter et al. (2012), *Bimax* can be reformulated as a graph problem. In this setting, a bipartite graph is built between a set of samples and a set of genes. A connection between two nodes of the different sets is drawn if the gene is significantly expressed in the corresponding sample. Finding all inclusion-maximal biclusters is then equal to finding all maximal bicliques in the bipartite graph. This objective can for example be achieved by using the Bron-Kerbosch algorithm (Bron and Kerbosch, 1973).

III. ALGORITHM

In this section we elaborate on the algorithm steps. First, we present the preprocessing steps that are necessary. Next, we explain how the bipartite graph is built based on the input. We then elaborate on in which way context information is added. Lastly, we show how biclusters can be found.

A. Preprocessing

First, logarithmic scaling is applied since the magnitudes of the gene expression levels are not of interest. It is only desirable to compare whether a gene is over- or underregulated regarding various samples. Therefore, binarizing the matrix with respect to an expression threshold τ_e is in accordance with this objective. τ_e can be set to the mean or median of each gene expression level over all samples. This is also necessary because the algorithm that finds biclusters only works on undirected, unweighted graphs. Since the gene expression matrix functions as the graph's adjacency matrix, the matrix is supposed to be binary. In this matrix a cell is set to one if the corresponding gene is higher expressed in the sample than in one half of the remaining samples. Table I shows the binarization of a gene expression matrix when setting $\tau_e = 0$.

B. Graph Building

Based on this matrix a bipartite graph is built. A bipartite graph is a triple $G = (V, U, E)$ where V and U are disjoint sets of vertices. E is the set of edges that connects vertices from V with vertices from U. In this instance, V corresponds to the samples and U to the genes. An edge between a sample and a gene is drawn if the corresponding cell in the matrix equals one. Formally, let A be the given gene expression matrix. A_{ij} corresponds to the expression of gene j in sample i. It holds that:

$$A_{ij} > \tau_e \iff (i, j) \in E \tag{1}$$

In order to incorporate biological context information, an edge is drawn between two nodes of the set of genes if both genes are known to interact with each other. This context information can be the certainty of a third-party data source that two genes interact with each other. This is for example the case if genes contribute so the same pathways. The edge is drawn if the confidence score of the third-party source is above a threshold τ_i. Let $s_\alpha(x, y)$ be a score function that returns the confidence that the genes x and y interact with each other based on data from α. Furthermore, let $g_1, g_2 \in U$. It holds that:

$$s_\alpha(g_1, g_2) > \tau_i \iff (g_1, g_2) \in E \tag{2}$$

By choosing the external data source and the interaction criterion α, the algorithm can be pushed to focus on expression patterns of the desired domain. In Subsection III-D we elaborate on how to define the score using data from the biological database STRING (Szklarczyk et al., 2015).

C. Finding Biclusters

Based on the resulting graph, we are identifying biclusters by finding the maximal cliques in the graph. A clique is a set of vertices such that each vertex is directly connected to each other of the clique. Thus, the subgraph that is induced by a clique is fully-connected. A maximal clique is a clique that can not be extended by more nodes. The Bron-Kerbosch (Bron and Kerbosch, 1973) algorithm finds maximal cliques by recursively checking whether a subgraph builds a clique and whether adding adjacent nodes would generate a larger clique. As shown by Tomita et al. (2006), any n-vertex graph has at most $3^{n/3}$ maximal cliques which matches the worst-case runtime of the algorithm which is $O(3^{n/3})$.

D. Implementation

We use the Python programming language (Rossum, 1995) to implement the presented algorithm. In order to simplify data management and data analysis, we use the Pandas (McKinney, 2011) library which offers efficient data structures that accelerate processing big numeric tables. Using the machine learning library scikit-learn (Pedregosa et al., 2011) we remove genes that have a low variance across the samples as a preprocessing step. To incorporate biological context information, we use the *interactionsList*-API-endpoint[1] of the STRING database. For a given gene identifier, the endpoint returns JSON data including genes

[1] http://string-db.org/api/json/interactionsList

Score Name	Explanation
nscore	neighborhood score (computed from the inter-gene nucleotide count)
fscore	fusion score (derived from fused proteins in other species)
pscore	cooccurence score of the phyletic profile (derived from similar absence/presence patterns of genes)
hscore	homology score, (the degree of homology of the interactors)
ascore	coexpression score (derived from similar pattern of mRNA expression measured by DNA arrays and similar technologies)
escore	experimental score (derived from experimental data, e.g., affinity chromatography)
dscore	database score (derived from curated data of various databases)
tscore	textmining score (derived from the co-occurrence of gene/protein names in abstracts)
score	combination of the aforementioned scores

Table II
SCORE NAMES RETURNED BY THE STRING API AND ASSOCIATED EXPLANATION

that interact with the specified gene with a given confidence. We first have to parse the JSON to in-memory data structures to access the desired information. For each interaction partner, a list of scores specifies how certain the service is that both genes interact with each other based on various data. A description of the returned scores according to the STRING API[2] is given in Table II. Listing 1 shows Python code that obtains interaction partners of the SFTPB gene with corresponding scores from the STRING service. Note that the endpoint expects an Ensembl Protein ID.

```
import requests
response = requests.get(”http://string-db.org/api/”\
”json/interactionsList?identifiers=ENSP00000377409”)
print(response.text)
```

Listing 1. Request all interaction partners of SFTPB from STRING

```
[{”ncbiTaxonId”:”9606”
  ”preferredName_A”:”SFTPB”,
  ”preferredName_B”:”CTSH”,
  ”stringId_A”:”ENSP00000377409”,
  ”stringId_B”:”ENSP00000220166”,
  ”ascore”:0.099,
  ”dscore”:0.9,
  ”escore”:0.36,
  ”fscore”:0,
  ”nscore”:0,
  ”pscore”:0,
  ”score”:0.953,
  ”tscore”:0.284},...]
```

Listing 2. Returned Sample Interaction Partners of the SFTPB Gene

Listing 2 shows the beginning of the response that is received by the code. The entry describes the certainty of the service that SFTPB interacts with CTSH based on various scores. To find biclusters in the end based on the built graph, we use an existing implementation of the Bron-Kerbosch algorithm from Kang (2014). Algorithm 1 shows a

```
   Data: Weighted Gene Expression Matrix $A$
   Result: List of Samples and Genes in Detected
           Biclusters
1  Initialization: InteractionScores = Map();
2  $\alpha$ = One of $\{nscore, fscore, pscore, hscore, ...\}$;
3  for Gene $g$ in $A$ do
4      Save interaction partners of $g$ from STRING
         endpoint based on $\alpha$ in InteractionScores
5  end
6  for Cell $c$ in $A$ do
7      if $c > \tau_e$ then
8          c = 1
9      else
10         c = 0
11     end
12 end
13 Build a bipartite graph $G$ based on $A$;
14 for Gene $g1$, Gene $g2$ in $A$ do
15     if InteractionScores[g1][g2] $> \tau_i$ then
16         Draw edge between $g1$ and $g2$ in $G$
17     end
18 end
19 BiclusterList = BronKerbosch(G);
20 return BiclusterList
21
```

Algorithm 1: An overview of our presented approach in pseudo-code

high-level description of our approach. After binarizing the matrix and building a bipartite graph, context information is added by drawing an edge between genes in line 15 based on the interaction scores obtained from STRING. We can apply the Bron-Kerbosch algorithm in line 19 without any modifications to the algorithm itself.

IV. EVALUATION

This section presents the evaluation of our approach on a big cancer dataset. We elaborate on metrics that can be used for evaluation, how we set up the experiments and in which way the results matched our expectations.

A. Criteria

Eren et al. (2013) propose to use both internal and external criteria for the evaluation of biclusters. Internal approaches rely on intrinsic properties of the found clusters. Pontes et al. (2015) present a comprehensive collection of metrics that are based on the correlation within the clusters. Measures like the Dice coefficient and the F_1 score evaluate the accordance of the detected bicluster with the expected bicluster. However, a ground truth is often needed to evaluate clusters based on statistical properties. Since

[2] http://version10.string-db.org/help/api/

we are interested in the biological relevance of the found clusters, this class of methods is not applicable. External approaches in turn use third-party data sources to evaluate the biclustering results. For example, Prelić et al. (2006) and Oghabian et al. (2014) used the number of gene ontology (GO) terms that have been enriched by the biclusters. Li et al. (2012) used the GO terms in order to calculate a p-value which expresses the probability that the genes in the clusters significantly enrich the terms. They also introduced a protein-protein interaction score (PPI) which is based on external data from the STRING database. The database is based on further information sources such as KEGG (Kanehisa and Goto, 2000) and Reactome (Croft et al., 2014). It offers an interface that returns the probability that two given proteins interact with each other based on several measurements. These measurements include experiments, co-expression analyses, text mining results based on publications on PubMed (Canese and Weis, 2013) and many more. Li et al. evaluated biclusters then based on the ratio of proteins in the cluster that interact with each other according to the STRING database.

B. Experiments

We applied the presented algorithm to a cancer dataset which is offered by The Cancer Genome Atlas project (Cancer Genome Atlas Research Network et al., 2013). It consists of 3,190 expression samples, each containing the expression levels of 55,572 genes. The data was obtained by sequencing tissue of eight tumor types (GBM, HNSC, KIRC, LAML, LUAD, SARC, THCA, UCEC) and has already been normalized using logarithmic scaling. To reduce the high dimensionality of the data set, we first applied variance-based feature selection. The idea is that genes that clearly distinguish classes from each other should highly vary across the dataset. In contrast, genes with constant expression levels are unqualified to differentiate biological processes and clusters yield very little information and can be removed. Therefore, we only kept the top 20 genes that have the highest variance in the dataset. We chose $\tau_e = 0$ and $\tau_i = 0.5$ for the thresholds that influence how the graph is built. For $s_\alpha(x, y)$ we use the combined score returned by the STRING API endpoint between two genes x and y.

C. Results

The algorithm finds 59 biclusters in the data set with the mentioned settings. Figure 1 shows the size distribution of the found biclusters. It visualizes that the number of samples in a bicluster follows a power-law distribution. Thus, it can be desirable to discard biclusters that have too few samples. The relationship between the genes in the identified biclusters is visualized in Figure 2. In the graph, an edge is drawn if two genes occur in the same found bicluster. Note that the resulting graph consists of two connected components. The lower subgraph is densely connected and visualizes various connections between members of the keratin family. This family of proteins forms the structural framework for cells that build e.g. hair. Interestingly, keratin 14 and keratin 5 were found to interact with each other. This emphasizes that the algorithm can infer biological processes since e.g. both genes partner in order to build the cytoskeleton of epithelial cells. We refer again to Table II for the scores S that are provided by the STRING API. In order to evaluate which of the scores are most useful, we ran the algorithm multiple times by setting each time $s_\alpha(x, y)$ to a different $s \in S$. To evaluate the cluster results, we calculated how many of the genes in found biclusters also interact with each other based on all scores $s \in S$. The resulting precision matrix can be found in Figure 3. It is worth noting that the algorithm produced the best results when incorporating the coexpression score *ascore*, the textmining score *tscore* and the combined score *score*. We assume that the underlying information used to compute the scores is the most reliable and helpful to compute the biclusters. The fusion score *fscore*, neighborhood score *nscore* and phyletic

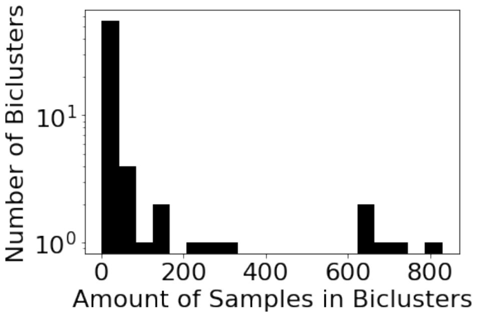

Figure 1. The histogram shows the number of biclusters with a certain amount of included samples.

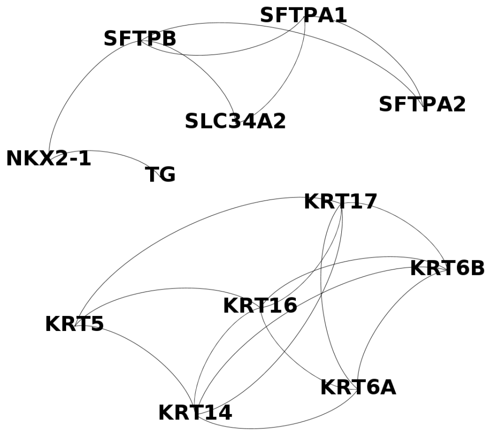

Figure 2. The graph visualizes the interactions between clustered genes in the evaluation dataset. It shows

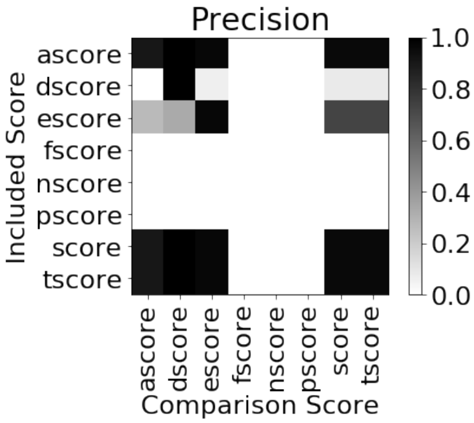

Figure 3. The matrix shows the precision of the algorithm based on one score compared to an other. For explanations of the scores refer to Table II.

score *pscore* did not yield a high precision since they were very small for the gene combinations of the preprocessed gene expression matrix. This means that the STRING service was very unsure about whether two genes interact based on the respective information. The reason might be that it is difficult to infer whether two genes interact with each other solely based on e.g. their nucleotide count. The results show that while the algorithm is able to capture biological processes, its quality highly depends on the third-party data that is incorporated.

V. Discussion

The presented approach is able to find possibly overlapping biclusters in gene expression data while incorporating biological context information. This makes it possible to focus the search for biclusters on processes of interest only. However, one major limitation is that the given gene expression matrix needs to be binarized so the algorithm can build an unweighted, undirected graph from that. This makes it necessary to discard information about the exact expression levels of genes. As described in Section III, the matrix could for example be binarized by setting all cells to one that have a higher value than τ_e. This would limit the algorithm to only find biclusters of over-expressed genes. It might also be desirable to find biclusters of over- and under-expressed genes at the same time. For this case, we suggest to vary the criterion when a cell is set to one among different genes. An other limitation is that the algorithm returns multiple biclusters of various sizes. We would expect it to return the biclusters clipped and sorted by some evaluation criterion. This metric, like the Silhouette coefficient (Rousseeuw, 1987) for single-mode clustering, should measure the coherence of a bicluster compared to the distance to the nearest neighboring bicluster. The approach should therefore return biclusters of a given quality only.

VI. Conclusion

In this work, we proposed an algorithm that finds overlapping biclusters in gene expression data while incorporating biological context information. We showed that the detected biclusters correspond to distinct biological processes. This enables the user to focus the bicluster detection to certain biological domains only by adjusting the third-party information that is used. Due to the dependence on third-party information, the quality of the findings is directly influenced by the quality of the third-party data. We thus conclude that advances in obtaining and processing biological data will also increase the performance of our algorithm.

References

Coen Bron and Joep Kerbosch. Algorithm 457: Finding All Cliques of an Undirected Graph. *Commun. ACM*, 16(9): 575–577, 1973.

Cancer Genome Atlas Research Network et al. The Cancer Genome Atlas Pan-Cancer analysis project. *Nat Genet*, 45(10):1113–1120, 2013.

Kathi Canese and Sarah Weis. PubMed: The Bibliographic Database. *The NCBI Handbook*, 2013.

David Croft et al. The Reactome pathway knowledgebase. *Nucleic Acids Research*, 42:472–477, 2014.

Bhaskar Dutta et al. PathNet: a Tool for Pathway Analysis Using Topological Information. *Source Code for Biology and Medicine*, 7(1):10, 2012.

Kemal Eren et al. A Comparative Analysis of Biclustering Algorithms for Gene Expression Data. *Briefings in Bioinformatics*, 14(3):279–292, 2013.

Alex Hindle et al. Clustering Web video search results based on integration of multiple features. *World Wide Web*, 15: 53–73, 2011.

Minoru Kanehisa and Susumu Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

Cheol Kang. Performance comparison of three BronKerbosch algorithm implementations that find all maximal cliques in a graph. https://github.com/cornchz/Bron-Kerbosch, 2014. Accessed: 2018-03-21.

Li Li et al. A Comparison and Evaluation of Five Biclustering Algorithms by Quantifying Goodness of Biclusters for Gene Expression Data. *BioData Mining*, 5(1):8, 2012.

Ke-Qin Liu et al. Identifying dysregulated pathways in cancers from pathway interaction networks. *BMC Bioinformatics*, 13(1):126, 2012.

Wes McKinney. pandas: a foundational python library for data analysis and statistics. 2011.

Ali Oghabian et al. Biclustering Methods: Biological Relevance and Application in Gene Expression Analysis. 2014.

F. Pedregosa et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Beatriz Pontes, Ral Girldez, and Jess S. Aguilar-Ruiz. Quality Measures for Gene Expression Biclusters. *PLOS ONE*, 10(3):1–24, 2015.

Ignacio Ponzoni et al. Pathway Network Inference from Gene Expression Data. *BMC Systems Biology*, 8(2):S7, 2014.

Amela Prelić et al. A Systematic Comparison and Evaluation of Biclustering Methods for Gene Expression Data. *Bioinformatics*, 22(9):1122–1129, 2006.

Guido Rossum. Python Reference Manual. Technical report, Amsterdam, The Netherlands, The Netherlands, 1995.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65, 1987.

Damian Szklarczyk et al. STRING v10: Protein-Protein interaction networks, integrated over the tree of life. *Nucleic Acids Research*, 43:447–452, 2015.

Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28 – 42, 2006. Computing and Combinatorics.

Oliver Voggenreiter, Stefan Bleuler, and Wilhelm Gruissem. Exact Biclustering Algorithm for the Analysis of Large Gene Expression Data Sets. 13, 2012.

Shanrong Zhao et al. Comparison of RNA-Seq and Microarray in Transcriptome Profiling of Activated T Cells. *PLOS ONE*, 9(1):1–13, 2014.