Semantic Biclustering: a New Way to Analyze and Interpret Gene Expression Data

Jiří Kléma, František Malinka, and Filip Železný

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science
Karlovo náměstí 13, 121 35 Prague 2, Czech Republic
{klema,malinfr1,zelezny}@fel.cvut.cz

Abstract. We motivate and define the task of semantic biclustering. In an input gene expression matrix, the task is to discover homogeneous biclusters allowing joint characterization of the contained elements in terms of knowledge pertaining to both the rows (e.g. genes) and the columns (e.g. situations). We propose two approaches to solve the task, based on adaptations of current biclustering, enrichment, and rule and tree learning methods. We compare the approaches in experiments with Drosophila ovary gene expression data. Our findings indicate that both the proposed methods induce compact bicluster sets whose description is applicable to unseen data. The bicluster enrichment method achieves the best performance in terms of the area under the ROC curve, at the price of employing a large number of ontology terms to describe the discovered bicluster.

1 Introduction

The objective of *biclustering* (or *block-clustering, co-clustering*) [19] is to find submatrices of a data matrix such that these submatrices exhibit an interesting pattern in their contained values; for example their values are all equal whereas the values in the containing matrix are non-constant. Biclustering has found significant applications in bioinformatics [13] and specifically in the context of gene expression data analysis [9, 18]. In the latter domain, biclustering can reveal special expression patterns of gene subsets in sample subsets. Numerous variations of biclustering have been considered, depending on whether the target set of biclusters should cover the entire original matrix, whether the biclusters may overlap mutually, etc.

By *semantic clustering* we refer to conventional clustering with a subsequent step in which the resulting clusters are described in terms of prior domain knowledge. A typical case of semantic clustering in gene expression analysis is clustering of genes with respect to their expression, followed by *enrichment* analysis where the clusters are characterized by Gene ontology terms overrepresented in them. In [10] the authors blend these two phases in that they directly cluster genes according to their functional similarities. The term semantic clustering

was introduced on analogical principles in the software engineering domain [11]. It can be also viewed as an unsupervised counterpart of the *subgroup discovery* method [24]. The semantic descriptions provide an obvious value for interpretation of analysis results, as opposed to plain enumeration of cluster elements.

Here we explore a novel analytic technique termed *semantic biclustering*, combining the two concepts above. In particular, we aim at discovering biclusters satisfying the usual biclustering desiderata, and also allowing joint characterization of the contained elements in terms of knowledge pertaining to both the rows (e.g. genes) and the columns (e.g. situations). This task is motivated by the frequent availability of formal ontologies relevant to both of the dimensions, as is the case of the publicly available *Dresden ovary table* dataset [1, 8]. Informally, we want to be able to discover biclusters described automatically e.g. as *"sugar metabolism genes in early developmental stages"* whenever such genes exhibit uniform expression in the said stages (situations). In [17], the authors present a closely related approach, where the knowledge pertaining to both the matrix dimensions is directly applied to define constraints to filter biclusters (the authors use the more general term patterns). The user is provided only with the interpretable biclusters whose description is compact.

Besides the novel problem formulation stated above, our contributions described below include the proposal of two adaptations of existing computational methods towards the objective of semantic biclustering and their comparative evaluation on the mentioned publicly available dataset [8]. As usual in unsupervised data analysis, the way to validate the methods statistically is not fully obvious. Thus our proposed validation protocol represents a contribution on its own right.

2 Two Candidate Approaches

In this initial study, we explore two approaches based on established techniques which can be adjusted to fulfill the objective of semantic biclustering. Firstly, we consider the natural workflow in which a biclustering algorithm is used first and its results are subjected to enrichment analysis on both dimensions of the produced submatrices. Secondly, we propose an approach based on the classical symbolic machine-learning techniques known as decision rule and tree learning.

In what follows, we assume for simplicity that the subject of semantic biclustering is a matrix $\mathbb{A}$ with dimensions $m \times n$ and binary elements $a_{i,j} \in \{0, 1\}$, indicating the detected (1) or absent (0) expression of gene i in situation j. Furthermore, we assume that each of the m genes (n situations, respectively) is assigned a set of gene-ontology (situation-specific ontology) terms drawn from a set of g ontology (s situation) terms.

2.1 Bicluster Enrichment Analysis

The enrichment approach to semantic biclustering stems from the binary expression matrix $\mathbb{A}$. In the first step, it searches for a set of biclusters that best

describe the input data. The goal is to find a small set of biclusters that cover as many 1s as possible and as few 0s as possible. In other words, we search for the most concise biset-based description that minimizes occurrence of false positives and false negatives. The bicluster semantics is disregarded for the moment. In the field of biclustering it is a well-known problem that can be tackled with approximate pattern matching [12, 14, 22], non-negative matrix decomposition [25, 26], bipartite graph partitioning [7] or heuristic algorithms [5, 15, 20]. In our approach, we employed the well-known PANDA+ tool [12] to find the initial concise description of the input data. The main parameter is the level of accepted noise which may be used to balance between the size of the description (the number of biclusters and their size) and the quality of the description (the amount of false predictions). $\mathbb{A}$ has to be transformed into the FIMI sparse format [2] before calling PANDA+.

Algorithm 1: Bi-directional enrichment.

input : $\mathbb{A}^{m \times n}$, $a_{i,j} \in \{0, 1, NA\}$; `// NAs for testing fields`
`1` GO; L; `// gene and location ontology`
output: Π^S; `// the matrix of gene and location p-values`

`2` `/* Get list of biclusters, i.e., bi-sets of gene/location indices */`
`3` $A \leftarrow$ `convertToSparseFIMIFormat(`$\mathbb{A}$`);`
`4` $\Pi \leftarrow$ `PANDA+(`A`);`
`5` `/* Get actual genes and locations, e.g., from` $\mathbb{A}$ `row/column names */`
`6` $\mathcal{M} \leftarrow$ `getAllGeneNames(`$\mathbb{A}$`); // all genes in` $\mathbb{A}$
`7` $\mathcal{G} \leftarrow$ `getAllGoTerms(`GO`,`$\mathcal{M}$`); // GO transitive closure wrt` $\mathcal{M}$
`8` $\mathcal{N} \leftarrow$ `getAllLocationNames(`$\mathbb{A}$`); // all locations in` $\mathbb{A}$
`9` $\mathcal{L} \leftarrow$ `getAllLocationTerms(`L`,`$\mathcal{N}$`); // L transitive closure wrt` $\mathcal{N}$
`10` $g \leftarrow |\mathcal{G}|$; $s \leftarrow |\mathcal{L}|$; $\Pi^S \leftarrow 0^{k \times (g+s)}$;
`11` `/* Annotate the individual biclusters */`
`12` **for** $k \leftarrow 1$ **to** $|\Pi|$ **do**
`13` | **for** $i \leftarrow 1$ **to** g **do**
`14` | | $\Pi^S_{k,i} \leftarrow$ `enrichmentGet(`$\Pi_{k,genes}$`,`$\mathcal{G}_i$`,`$\mathcal{M}$`,`GO`)`
`15` | **end**
`16` | **for** $j \leftarrow 1$ **to** s **do**
`17` | | $\Pi^S_{k,g+j} \leftarrow$ `enrichmentGet(`$\Pi_{k,locs}$`,`$\mathcal{L}_j$`,`$\mathcal{N}$`,`L`)`
`18` | **end**
`19` **end**

In the second step, the biclusters get annotated in terms of prior domain knowledge, i.e., their semantics is revealed. In our case, we use gene ontology (GO) terms [3, 6] to annotate the individual genes and the dedicated Drosophila location ontology terms [1] to annotate the stages. Each non-trivial bicluster (comprehending more than 1 gene and 1 stage) gets annotated by all the terms (GO and situation ontology, respectively) whose enrichment exceeds the predefined statistical significance threshold. In order to avoid this hyperparame-

ter in our workflow, we propose to set the threshold automatically within the permutation-based test, that compares the bicluster enrichment scores with the scores reached in permuted gene expression matrix. The significance threshold is set to guarantee that the false discovery rate for annotation terms in real biclusters remains small. The individual terms get scored proportionally to their statistical significance, i.e., each of the biclusters is described by a sparse real vector of the term scores $s_1, s_2, \ldots, s_g, s_{g+1}, \ldots, s_{g+s}$. The score s_i is positive iff the i-th term is enriched in the given bicluster, it is 0 otherwise. We employed the topGO Bioconductor package [4] to find the GO terms and Fisher test to reveal the location ontology terms enriched in the individual biclusters.

The approach to semantic biclustering could as well be referred to as *bi-directional enrichment*. The procedure pseudocode is in Algorithm 1.

2.2 Rule and Tree Learning

The alternative approach is based on a reduction of the problem to a classification-learning problem. This entails a transformation of the original data matrix $\mathbb{A}$ into an auxiliary binary matrix $\mathbb{B}$ of dimensions $(m \cdot n) \times (g + s + 1)$. Matrix $\mathbb{A}$ is unrolled into $\mathbb{B}$ so that each row of $\mathbb{B}$ corresponds to one element $a_{i,j}$ of $\mathbb{A}$ and has the form

$$t_1, t_2, \ldots t_g, t_{g+1}, t_{g+2}, \ldots t_{g+s}, \mathit{expression} \tag{1}$$

where the first g numbers are binary indicators of gene-ontology terms (acquiring value 1 iff the corresponding term is associated with gene i), the subsequent s numbers are analogical indicators of situation ontology-terms for situation j, and the last number is the expression indicator for gene i and situation j and thus equals $a_{i,j}$. The transformation details are shown in Algorithm 2.

The next step is to learn a classification model to predict *expression* from $t_1, \ldots t_{g+s}$. To this end, $\mathbb{B}$ represents the training data with individual rows such as (1) corresponding to learning examples with the last element being the class indicator. The model is seeked in the form of a list of conjunctive decision rules [16], each of which acquires the form

$$\wedge_{k \in G} t_k \wedge_{k \in S} t_{k+g} \rightarrow \mathit{expression} \tag{2}$$

where the rule conditions $G \subseteq [1; g], S \subseteq [1; s]$ are learned selections of gene and situation ontology terms. The rule stipulates that a gene annotated with all the gene-ontology terms found in G is likely to be expressed in situations annotated with all the situation-ontology terms in S. If no rule in the learned rule set predicts expression, the rule set defaults to the no-expression prediction.

Consider the set $P = G \times S$ containing all the gene-situation pairs (i, j) satisfying conditions of rule (2). It is easy to see that P forms a submatrix of $\mathbb{A}$, i.e. there exists a permutation of A's rows and columns making P a rectangular section of $\mathbb{A}$. Indeed, G identifies a set of rows and S identifies a set of columns. The conjunction in (2) is satisfied exactly by the genes in the intersection of G

Algorithm 2: Unrolling $\mathbb{A}$ into $\mathbb{B}$.

input : $\mathbb{A}^{m \times n}$, $a_{i,j} \in \{0,1\}$
output: $\mathbb{B}^{m \cdot n \times g+s+1}$, $b_{i,j} \in \{0,1\}$

```
/* Genes are represented by a set of FBgn identifiers                    */
M ← getAllGeneNames(A); // all genes in A
G ← getAllGoTerms(M); // GO transitive closure wrt M
g ← |G|;
for i ← 1 to m do
    ∀x ∈ {1, ..., g + s + 1} : T_x ← 0; // initialization
    for j ← 1 to g do
        if term G_j is associated with gene M_i then T_j ← 1;
    end
    for k ← 1 to s do // where s is a set of situation terms
        associations ← find all associations in a set of situations for s_k;
        for ∀assoc ∈ associations do
            T_{g+assoc_i} ← 1; // where assoc_i is an index of situation term
            assoc
        end
        T_{g+s+1} ← a_{i,k}; // add expression indicator
        B_{i,*} ← T;
    end
end
B ← filterGoTerms(B, Θ); // due to a given threshold Θ;
```

and S, which is thus a rectangle.[1] Therefore each rule such as (2) identifies a bicluster in $\mathbb{A}$.

Moreover, a rule set optimized for classification accuracy on training data such as (1) will produce those biclusters of $\mathbb{A}$ which contain a high number of elements with value 1. Indeed, perfect training-set accuracy is achieved if and only if the biclusters represented by the rules in the rule-set collectively cover all the 1-elements and no 0-element in $\mathbb{A}$.

Summarizing the two observations, the learned rule set represents a set of biclusters of $\mathbb{A}$, each of which is homogeneous in that it collects positive indicators of expression. Furthermore, each such bicluster is characterized by the ontology terms G and situation terms S found in the corresponding rule such as (2). Thus, the procedure described indeed conveys the semantic biclustering task.

In addition, we propose an alternative to the described workflow, in which the rule-set learner is replaced by a *decision tree* learner [16]. Each vertex in a learned tree corresponds to one ontology term, and the test represented by the vertex determines whether the term is among the annotation of the classified pair of gene and situation. Since all the attributes (including the class attribute) of the

[1] Note that this property essentially follows from the propositional-logic form of the rule and would not hold true for the more general *relational* rules considered in [24].

training data (1) are binary, also the learned tree is binary. Hence, there is exactly one path in the tree with only positive branches; i.e. branches corresponding to satisfying the condition in the source vertex. This unique path can be rewritten as a single decision rule in the form (2) and thus represents a single semantic bicluster. The main reason for exploring this alternative is that decision trees are often claimed to exhibit performance superior to that of decision rule sets.

In our implementation of this approach, we used the JRip and J48 algorithms from the WEKA machine-learning software [21] to learn the rule-sets and decision trees, respectively.

3 Experimental Evaluation

Both biclustering and enrichment analysis are unsupervised data mining methods and the exact way to validate their performance is not obvious. For example, perfectly homogeneous biclusters can usually be found at the cost of their very small size. The size and homogeneity should thus be traded-off but their relative importance would have to be set apriori. Similarly, the discovered semantic annotations may either represent genuine characteristics of the biclusters, or the included terms may be enriched just by chance. Distinguishing apart these two effects through a statistical test involves distributional assumptions which we cannot guarantee.

We propose to solve the latter dilemma by measuring the quality of semantic biclusters from the point of view of *predictive classification*. This assumes that the available data is split randomly into a training partition and a testing partition. Semantic biclusters are found through the earlier described two approaches on the training split. Each found bicluster collects genes and situations such that the genes tend to be expressed in the situations. A legitimate interpretation of the cluster's semantic annotations is that other genes and situations not found in the training set, but complying with the annotations should also exhibit expression. We thus employ the set of discovered biclusters as a predictive model of expression of each combination of a gene and a situation on the testing set. We then measure the area under the ROC curve (AUROC) achieved on the test set and argue that this quantity is an unbiased and justified measure of quality of the discovered set of biclusters. The former ROC curve applications to biclustering outcomes evaluated to what extent this outcome conforms to prior biological knowledge and matches the outcome of other biclustering algorithms [23]. In here, we evaluate generalization ability of the resulting set of biclusters rather than their agreement with a gold standard which is often not available or difficult to be defined.

The implementation of this validation protocol is straightforward for the rule and tree learning approach (Section 2.2) as the semantic biclusters come in the form of predictive classifiers. For the bicluster enrichment approach (Section 2.1), we propose to interpret the biclusters as classifiers in the following way. Each testing combination of a gene and a situation is classified as positive iff the

corresponding semantic description $t_1, t_2, \ldots, t_g, t_{g+1}, \ldots, t_{g+s}$ is matched by the semantic description of any of the produced biclusters.

For a single bicluster, the matching is expressed as a scalar product of its score vector with the binary term indicator-vector of the classified entry, and this is done separately for the gene and situation ontology. The score vector of a bicluster originates as follows. The term scores 0 if the term enrichment does not reach the predefined statistical significance threshold. If the significance p-value is below the threshold, the term scores $-log_{10}(pval)$. In summary, the gene expression entry is classified as positive iff

$$\langle s_1, s_2, \ldots, s_g \rangle \cdot \langle t_1, t_2, \ldots, t_g \rangle \geq \theta_G$$

and at the same time

$$\langle s_{g+1}, \ldots, s_{g+s} \rangle \cdot \langle t_{g+1}, \ldots, t_{g+s} \rangle \geq \theta_S$$

where θ_G and θ_S stand for the respective minimum match thresholds. They represent the trade-off between sensitivity and specificity of the unseen data imputation procedure. The evaluated ROC curve is obtained by varying these two thresholds. The semantic biclustering validation procedure is summarized in Algorithm 3.

The proposed validation scenario conforms to the validation frameworks as usual in machine learning. One exception from that is that the stage of splitting data (the $\mathbb{A}$ matrix) into the training and testing sets needs to be different. In particular the training set needs to form a submatrix, i.e. a rectangular section of $\mathbb{A}$, because a matrix is the assumed kind of input of the semantic biclustering methods. We thus proceed by selecting a random rectangle within $\mathbb{A}$ covering 70% of its elements and representing the training set, while all other elements fall in the testing set. The latter need not be rectangular as follows from the proposed validation principle.

4 Results

We conducted our experiments on the Dresden ovary table [1]. The table captures the distribution of different mRNA molecules in various cell types involved in oocyte production in the ovary of female Drosophila melanogaster flies. The table authors believe [8] that the resource can be used to gain insight into specific genetic features that control the distribution of mRNAs and this insight may be instrumental for cracking the RNA localization code and understanding how it affects the activity of proteins in cells. In this problem, the dedicated situation ontology (available from the same source) describes Drosophila ovary segments and their developmental stages. The ontology is in fact a location term hierarchy that binds the locations available in the Dresden ovary table by the relations part_of and develops_from. Thus, the hierarchy deals with 100 terms. The gene ontology was used in its standard available form [3, 4], there were 8,407 GO terms available altogether. After minor data cleansing, the expression matrix has

Algorithm 3: Predictive evauation of bi-directional enrichment.

input : Π^S; $\mathbb{A}^{m \times n}$, $a_{i,j} \in \{0, 1, \mathrm{NA}\}$; // NAs for training fields
GO; L; // gene and location ontology
parameters: $\theta_G; \theta_S$; // gene and location term score thresholds
p_{perm}; // p-val permutation threshold
output : $\mathbb{P}^{\mathrm{m} \times \mathrm{n}}$, $p_{i,j} \in \{0, 1, \mathrm{NA}\}$ // the predicted expressions

/* Initialize predicted expressions, zeroes or NAs only */
$\mathbb{P} \leftarrow \mathbb{A}$; $\mathbb{P}[\mathbb{P} == \mathbb{1}] \leftarrow 0$;
/* Get GO term indication vectors for all genes */
$\mathcal{M} \leftarrow$ getAllGeneNames($\mathbb{A}$); // all genes in $\mathbb{A}$
$\mathbb{T}_{\mathcal{M}} \leftarrow$ getTermsForGenes(GO,$\mathcal{M}$); // a binary m×g incidence matrix
/* Get location term indication vectors for all stages */
$\mathcal{N} \leftarrow$ getAllLocationNames($\mathbb{A}$); // all locations in $\mathbb{A}$
$\mathbb{T}_{\mathcal{N}} \leftarrow$ getTermsForStages(L,$\mathcal{N}$); // a binary n×s incidence matrix
/* Apply the individual biclusters */
for $k \leftarrow 1$ **to** $|\Pi^S|$ **do**
  /* turn p-values into scores, apply the permutation threshold */
  for $i \leftarrow 1$ **to** $g + s$ **do**
    if $\Pi^S_{k,i} < p_{perm}$ **then** $\Pi^S_{k,i} = -log_{10}(\Pi^S_{k,i})$;
    else $\Pi^S_{k,i} = 0$;
  end
  /* Search for the genes and stages covered by the bicluster, use them to fill in $\mathbb{P}$ */
  $\mathbb{P}[\mathbb{T}_{\mathcal{M}} \Pi^S_{k,1 \ldots g} > \theta_G, \mathbb{T}_{\mathcal{N}} \Pi^S_{k,g+1 \ldots g+s} > \theta_S] \leftarrow 1$
end

6,510 rows (genes) and 100 columns (situations). For the rule and tree learning approach, this matrix thus unrolls into 651,000 learning examples with 47.5% positive data instances.

The bicluster enrichment method was run with the default PANDA+ parameters. The statistical significance thresholds were set to 0.05 for genes and 0.1 for situations. The method was run repeatedly with the following sets of match thresholds: $\theta_G \in \{1, 2, 5, 10\}$ and $\theta_S \in \{1, 2, 5, 10\}$. The results suggested that precision decreases slowly with decreasing match thresholds while recall grows quite rapidly. The best precision/recall trade-off is thus reached for the minimum match threshold values $\theta_G = \theta_S = 1$. The size of bicluster description does not directly change with the match threshold values, their decrease raises the number of genes and developmental stages matched by bicluster annotation terms.

The rule and tree learning was performed with the default WEKA parameters for JRip and J48. In order to work with a reasonable number of features, feature selection was employed first. All the features (annotation terms) of the train matrix (originating from $\mathbb{B}$ matrix) that occurred in fewer than 500 expression entries (the train matrix rows) were removed. The cut-off threshold was found with the feature frequency histogram. Eventually, we worked with the

train matrix of the size 457,548×194. Besides speeding up the learning process, we avoided the GO terms that cannot generalize over a reasonable number of locations.

Table 1 shows the results including the AUROC achieved by the two methods as well as further information regarding the found biclusters. The table summarizes 10 experimental runs, each for a different random train-test split. Note that the traditional cross-validation scenario cannot be applied in the two-dimensional setting.

AUROC evaluates the proposed methods from the point of view of their generalization ability. Importantly, both the proposed methods generalize far better than random. In other words, the semantic descriptions of the biclusters can be used to assume on the expression of unmeasured genes in unseen developmental stages. The bicluster enrichment approach seems to be the most reliable predictive method. If given an unseen pair of positive (present) and negative (absent) expression entries, it correctly guesses the positive entry with approximately 75% chance. On the other hand, the method asks for a relatively large number of bicluster annotation terms to reach a reasonable recall. In our experiments, the average number of GO and location terms per bicluster was 44 and 4 respectively (as the location ontology deals with a smaller number of terms). This number of terms may make the interpretation hard for a human expert. JRip outputs the most concise bicluster description, its disadvantages lie in the low AUROC and by far the slowest runtime.

The experimental results conform to expectations. The bicluster enrichment approach ignores the semantic description when building the biclusters. Consequently, they tend to faithfully fit the expression matrix and compactly represent the expression patterns that the matrix contains. On the other hand, their postponed semantic annotation may turn out complex and fuzzy. The rule and tree learning does just the opposite. It directly searches for concise semantic descriptions that separate positive and negative expression values in training data. As a result, the descriptions have tendency to be short and crisp with potentially lower recall.

Method	AUROC	# of biclusters	Avg. # of terms per bicluster
Bicluster Enrichment	0.769±0.013	11.8±1.5	47.9±2.13
Rules (JRip)	0.636±0.01	93.7±17.4	7.0±0.40
Tree (J48)	0.713±0.01	1±0	27.5±0.89

Table 1. Evaluation results of the proposed approaches to semantic biclustering.

Figure 1 presents the individual ROC curves. For the bicluster enrichment method, the curve is constructed as a convex hull for 16 binary classifiers reached for different θ_G and θ_S settings. However, the curve suggests that one of the classifiers (namely the one for $\theta_G = \theta_S = 1$) makes the major contribution to the aggregate AUROC while the other classifiers approach the trivial convex hull or

fall under it. J48 and JRip can provide both binary and probabilistic outcomes. In here, we work with the probabilistic outcome, the curve is constructed with different probability thresholds for assigning an example to the positive class.

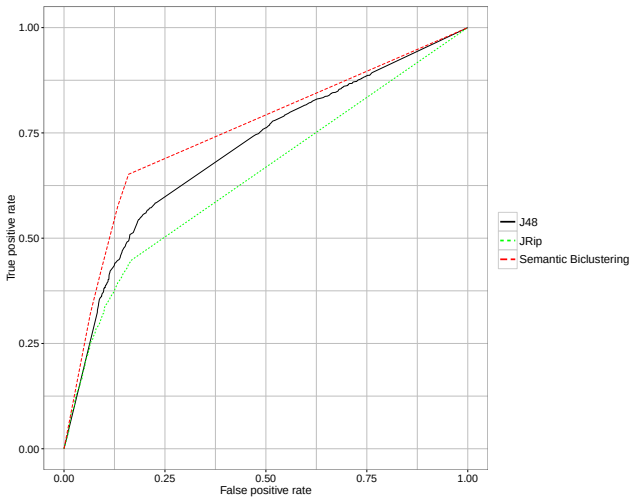

Fig. 1. Semantic biclustering ROC curves for Drosophila ovary gene expression data.

5 Conclusions and Future Work

We have motivated and defined the task of semantic biclustering and proposed two approaches to solve the task, based on adaptations of current biclustering, enrichment, and rule and tree learning methods. We compared them in experiments with Drosophila ovary gene expression data. Our findings indicate that the semantic biclustering method achieves the best performance in terms of the area under the ROC curve, at the price of employing a large number of ontology terms to describe the discovered bicluster.

In future work, we mainly want to investigate the statistical implications of the non-standard way to split the data matrix into the (rectangular) training set and the testing set. Furthermore, we plan to devise a specialized method for semantic biclustering that would combine the advantages of the proposed approaches. In principal, the biclustering enrichment ignores the prior knowledge when searching for biclusters. None of the biclusters has to be interpretable as a result. The rule and tree-based methods directly stem from the prior knowledge and search for the most general conjunctive concepts that fit the training data under the risk of their overfitting.

Acknowledgments. This work was supported by Czech Science Foundation project 14-21421S.

References

1. Dresden Ovary Table. `http://tomancak-srv1.mpi-cbg.de/DOT/main`, [Online; accessed 15-February-2016]
2. Frequent Itemset Mining Implementations Repository. `http://fimi.ua.ac.be/`, [Online; accessed 15-February-2016]
3. Gene Ontology Consortium. `http://geneontology.org/`, [Online; accessed 15-February-2016]
4. Alexa, A., Rahnenfuhrer, J.: topGO: topGO: Enrichment analysis for Gene Ontology (2010), r package version 2.4.0
5. Chen, H.C., Zou, W., Tien, Y.J., Chen, J.J.: Identification of bicluster regions in a binary matrix and its applications. PloS one 8(8), e71680 (2013)
6. Consortium, G.O., et al.: Gene ontology consortium: going forward. Nucleic acids research 43(D1), D1049–D1056 (2015)
7. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 269–274. ACM (2001)
8. Jambor, H., Surendranath, V., Kalinka, A.T., Mejstrik, P., Saalfeld, S., Tomancak, P.: Systematic imaging reveals features and changing localization of mRNAs in Drosophila development. eLife 4(e05003) (2015)
9. Kluger, Y., Basri, R., Chang, J.T., Gerstein, M.: Spectral biclustering of microarray data: Coclustering genes and conditions. Genome Research 13(4), 703–716 (2003)
10. Krejnik, M., Klema, J.: Empirical evidence of the applicability of functional clustering through gene expression classification. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 9(3), 788–798 (2012)
11. Kuhna, A., Ducasseb, S., Girbaa, T.: Semantic clustering: Identifying topics in source code. Information and Software Technology 49(3), 230–43 (2007)
12. Lucchese, C., Orlando, S., Perego, R.: A unifying framework for mining approximate top-binary patterns. Knowledge and Data Engineering, IEEE Transactions on 26(12), 2900–2913 (2014)
13. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE Transactions on Computational Biology and Bioinformatics 1(1), 24–45 (2004)
14. Miettinen, P., Mielikainen, T., Gionis, A., Das, G., Mannila, H.: The discrete basis problem. Knowledge and Data Engineering, IEEE Transactions on 20(10), 1348–1362 (2008)
15. Prelić, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9), 1122–1129 (2006)
16. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach (3rd Edition). Prentice Hall (2009)
17. Soulet, A., Kléma, J., Crémilleux, B.: Efficient mining under rich constraints derived from various datasets. In: Knowledge Discovery in Inductive Databases, pp. 223–239. Springer (2006)

18. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics 18(suppl 1), S136–S144 (2002)
19. van Mechelen, I., Bock, H.H., De Boeck, P.: Two-mode clustering methods: a structured overview. Statistical Methods in Medical Research 13(5), 363–94 (2004)
20. van Uitert, M., Meuleman, W., Wessels, L.: Biclustering sparse binary genomic data. Journal of Computational Biology 15(10), 1329–1345 (2008)
21. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2011)
22. Xiang, Y., Jin, R., Fuhry, D., Dragan, F.F.: Summarizing transactional databases with overlapped hyperrectangles. Data Mining and Knowledge Discovery 23(2), 215–251 (2011)
23. Yoon, S., Nardini, C., Benini, L., De Micheli, G.: Discovering coherent biclusters from gene expression data using zero-suppressed binary decision diagrams. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 2(4), 339–354 (2005)
24. Zelezny, F., Lavrac, N.: Propositionalization-based relational subgroup discovery with RSD. Machine Learning 62(1-2), 33–63 (2006)
25. Zhang, Z.Y., Li, T., Ding, C., Ren, X.W., Zhang, X.S.: Binary matrix factorization for analyzing gene expression data. Data Mining and Knowledge Discovery 20(1), 28–52 (2010)
26. Žitnik, M., Zupan, B.: Nimfa: A python library for nonnegative matrix factorization. The Journal of Machine Learning Research 13(1), 849–853 (2012)