RESEARCH

Semantic biclustering for finding local, interpretable and predictive expression patterns

Jiří Kléma*, František Malinka and Filip Železný

*Correspondence:
klema@fel.cvut.cz
Department of Computer Science, Czech Technical University in Prague, Karlovo náměstí 13, 121 35 Prague 2, Czech Republic
Full list of author information is available at the end of the article

Abstract

Background: One of the major challenges in the analysis of gene expression data is to identify local patterns composed of genes showing coherent expression across subsets of experimental conditions. Such patterns may provide an understanding of underlying biological processes related to these conditions. This understanding can further be improved by providing concise characterizations of the genes and situations delimiting the pattern.

Results: We propose a method called semantic biclustering with the aim to detect interpretable rectangular patterns in binary data matrices. As usual in biclustering, we seek homogeneous submatrices, however, we also require that the included elements can be jointly described in terms of semantic annotations pertaining to both rows (genes) and columns (samples). To find such interpretable biclusters, we explore two strategies. The first endows an existing biclustering algorithm with the semantic ingredients. The other is based on rule and tree learning approaches known from machine learning.

Conclusions: The two alternatives are tested in experiments with two Drosophila melanogaster gene expression datasets. Both strategies are shown to detect sets of compact biclusters with semantic descriptions that also remain largely valid for unseen (testing) data. This desirable generalization aspect is more emphasized in the strategy stemming from conventional biclustering although this is traded off by the complexity of the descriptions (number of ontology terms employed), which, on the other hand, is lower for the alternative strategy.

Keywords: biclustering; enrichment analysis; symbolic machine learning; ontology; gene expression

Background

The general goal of *biclustering* (or *block-clustering, co-clustering*) [1] is to find interesting submatrices in a given data matrix. A submatrix is defined by a subset of rows and a subset of columns of the original matrix. In other words, it is a compact rectangular section of a matrix that can be obtained by permuting the rows and columns (respectively) of the input matrix. There are multiple ways to define the interestingness of biclusters; the simple view adopted here is that the biclusters cover as many as possible 1's within the containing binary matrix while leaving out as many as possible 0's. Biclustering has become remarkably popular in bioinformatics [2], especially in gene expression data analysis tasks [3, 4]. Here, biclustering detects an expression specific to a subset of genes in a subset of samples (situations).

Semantic clustering denotes conventional clustering augmented by the additional requirement that the discovered clusters are characterized through concepts defined as prior domain knowledge. The characterizations are obviously requested for the sake of easy interpretation of the analysis results. A popular activity in bioinformatics, where (ordinary) clusters of genes with similar expressions profiles are first detected and *enrichment analysis* [5] is subsequently applied on such clusters is in fact an example of ('manual') semantic clustering. The two steps in the latter workflow can also be merged into a single phase as demonstrated in [6]. Semantic clustering is also related to the subgroup discovery approach [7], although in an unsupervised setting. The term semantic clustering is also employed in the software-engineering context [8] and captures a roughly similar meaning as in the present context.

In this study we explore the combination of the two concepts, that is *semantic biclustering*. Specifically, we want to be able to detect biclusters as outlined above; however, we also want their elements to share a joint description as in semantic clustering. In the case of biclustering, the description pertains to both the rows (that is, genes) as well as the columns (that is, situations). We follow this goal because formal ontologies are frequently available and relevant to either dimension of the input data matrix. An example of such a data set is the *Dresden ovary table* [9, 10]. Simply put, our goal is to design an algorithm able to detect biclusters characterized e.g. as "glucose metabolism genes in late developmental stages" whenever such genes in such stages are uniformly expressed. To the best of our knowledge, the previous approach most related to semantic biclustering is [11], where formal knowledge associated with both rows and columns of a data matrix is used to specify filters for detected patterns.

In the rest of the paper we formalize the problem of semantic biclustering first. Then, we propose two strategies for semantic biclustering and test them comparatively on two experimental datasets. Our contributions also include a design of a suitable validation protocol, as evaluation criteria are not fully evident in unsupervised data analysis.

Methods

Problem formalization

We assume a set of genes Γ, a set of situations Σ, and a binary set of expression indicators $\{0, 1\}$. We further assume a joint probability distribution over these three sets $p : \{0, 1\} \times \Gamma \times \Sigma \to [0; 1]$. In a gene-expression assay, a set $G \subseteq \Gamma$ of genes and set $S \subseteq \Sigma$ of situations are selected and expression is sampled for all pairs of the selected genes and situations. In other words, a matrix $\mathbb{A} = (a_{g,s})$, $g \in G$, $s \in S$ is formed such that $a_{g,s} = 1$ with $p(1|g, s)$ (0 otherwise).

In standard multivariate analysis of gene expression, $A = (a_{g,s})$ represents a *sample set* in the sense that a *sample* corresponds to a column in $\mathbb{A}$. For benefits of statistical inference, it is typically assumed that samples are i.i.d; more precisely, that S is drawn i.i.d.[1] from the marginal $p(s)$. In the present biclustering context,

[1]The drawing is with replacement, so strictly speaking S (and G analogically) is a multi-set rather than a set. This distinction is however immaterial in the present context.

we put genes and situations (rows and columns) on equal footing. That is to say, a sample corresponds to a single measurement $a_{g,s}$. Under this view, the sample set $\{(a_{g,s}, g, s) : g \in G, s \in S\}$ is not an i.i.d. sample from $p(a, g, s)$ even if both G and S are i.i.d. samples from the respective marginals $p(g)$ and $p(s)$, which is due to the sample set's rectangularity. Indeed, if the latter contains a sample for a particular pair (g, s), it will necessarily also contain all pairs $(g', s), g' \in G$ and all pairs $(g, s'), s' \in S$, so the samples are mutually dependent.

Ordinary biclusters

A *bicluster* in matrix $\mathbb{A} = (a_{g,s})$, $g \in G$, $s \in S$ is a submatrix defined by a subset of rows and columns, i.e., a tuple (G', S') where $G' \subseteq G$ and $S' \subseteq S$. A *system of biclusters* of $\mathbb{A}$ is $B = \{(G_k, S_k)\}$ where (G_k, S_k) are biclusters in $\mathbb{A}$. The *extension of* B is

$$ext(B) = \{(g, s) : g \in G', s \in S', (G', S') \in B\} \tag{1}$$

A usual requirement is that a system of biclusters covers regions of $\mathbb{A}$ that are *homogeneous* regarding the contained values. This may be interpreted in multiple ways and here we adhere to the simplest interpretation that the bicluster system B should ideally include all 1's present in $\mathbb{A}$ and exclude all 0's. Then a natural quality measure of B counts 1's inside its extension and 0's outside of it

$$\sum_{(g,s)\in ext(B)} a_{g,s} + \sum_{(g,s)\in G\times S\setminus ext(B)} 1 - a_{g,s} \tag{2}$$

For convenience, we introduce an *indicator function* $b : G \times S \to \{0, 1\}$

$$b(g, s) = 1 \text{ iff } (g, s) \in ext(B) \tag{3}$$

which allows us to rephrase the above quality measure as $|\{(g, s) \in G \times S : a_{g,s} = b(g, s)\}|$. Normalizing this to the interval $[0; 1]$, one obtains the formula

$$\widehat{Acc}(b) = \frac{|(g, s) \in G \times S : a_{g,s} = b(g, s)\}|}{|G||S|}$$

which is known as the *training (in-sample) accuracy* of b viewed as a classifier. This quantity provides an empirical approximation to the true b's accuracy on $G \times S$, which is $p(g, s, b(g, s)|(g, s) \in G \times S)$ according to our probabilistic model. The expression $(g, s) \in G \times S$ in the conditional part is important since b's domain is restrained to $G \times S$. On one hand, this classification viewpoint provides an additional motivation to maximize the ad-hoc formula (2). On the other hand, viewing $\widehat{Acc}$ as a proxy for the true accuracy entails certain problems.

First, as we have commented already, the sample set where $\widehat{Acc}$ is determined is not i.i.d. as normally required for a training set, although this could be tolerated if the intended use of $\widehat{Acc}$ is as a heuristic guiding the search for B, rather than an unbiased estimator. Second, $\widehat{Acc}$ can be trivially maximized by a system of single-element biclusters covering exactly all 1's in $\mathbb{A}$. Such an *overfitting* solution is

commonplace in classification and is usually avoided by an additional *regularization* term. Here, the latter could penalize small biclusters, or alternatively a high number of them. So one would search B maximizing

$$\widehat{Acc}(b) + \lambda/|B|$$

with λ determining the trade-off between accuracy and the size of the bicluster system. In fact, a regularizer is normally added to formula 2 in biclustering algorithms [12, 13] to prevent the trivial solution, irrespectively of any classification context.

The third problem lies in the restriction of b onto the $G \times S$ domain, which does not enable us to use b on genes and situations not in the training set. At first sight, this does not seem a problem if one is not interested in using the bicluster system B for classification. However, it makes the assessment of B's quality problematic in the following sense. Besides the training accuracy $\widehat{Acc}$ acting as a search heuristic, we are also interested in an unbiased estimate of the quality of the final system B produced by the biclustering algorithm. An ideal quality measure is the true accuracy $p(g, s, b(g, s))$ of b, which would normally be estimated using a *hold-out* or *testing* data set $Test = \{(g_k, s_k, a_k)\}$ drawn i.i.d. from $p(g, s, a)$, as

$$Acc(b) = \frac{|\{(g_k, s_k, a_k) \in Test : a_k = b(g_k, s_k)\}|}{|Test|} \tag{4}$$

However, this value cannot be established as b is not defined for arguments with values outside the training sample set and—to our best intelligence—there is no sensible way in which the bicluster system B could induce a classifier beyond the $G \times S$ domain. We will see in turn that this problem is overcome elegantly by *semantic biclusters*.

Semantic biclusters

Here we consider biclusters which are not defined by an enumeration of the selected rows and columns, but rather by enumerating conditions according to which the rows and columns are selected. In particular, the conditions are represented by semantic annotation terms pertaining to genes (rows) and situations (columns). Formally, we assume a set of gene annotation terms γ, and analogically situation annotation terms σ. Furthermore, relations $R_\gamma \subseteq G \times \gamma$, $R_\sigma \subseteq S \times \sigma$ are defined, associating genes and situations with selected annotation terms.

For an arbitrary gene set G, a term set $T^\gamma \subseteq \gamma$ induces the set $\{g \in G : \forall t \in T^\gamma, (g, t) \in R_\gamma\}$ of exactly those genes in G that comply with all the terms in T^γ. We denote this induced set as $G(T^\gamma)$. Similarly for a situation set S and a situation term set T^σ, $S(T^\sigma) = \{s \in S : \forall t \in T^s, (s, t) \in R_\sigma\}$.

Thus within a matrix of genes G and situations S, a *semantic bicluster* (T^γ, T^σ) induces a unique ordinary bicluster $(G(T^\gamma), S(T^\sigma))$ and a *system of semantic biclusters* $SB = \{(T^\gamma_k, T^\sigma_k)\}$ defines a unique ordinary system of biclusters B. Due to this correspondence between SB and B, SB can be searched using the heuristic $\widehat{Acc}(B)$ we elaborated above.

Unlike the extension of an ordinary system of biclusters (Eq. 1), the extension $ext(SB)$ of a system of semantic biclusters SB is not confined to the matrix of genes G and situations S

$$ext(SB) = \{(g, s) : g \in \Gamma(T^\gamma), s \in \Sigma(T^\sigma), (T^\gamma, T^\sigma) \in SB\} \tag{5}$$

and thus also the indicator function $sb : \Gamma \times \Sigma \to \{0, 1\}$ defined as in (3) now has all genes and situations in its domain. (Note that the restriction of $ext(SB)$ to the matrix $G \times S$ coincides with the extension $ext(B)$ of the ordinary system B of biclusters defined by SB; this is easy to see by replacing Γ and Σ respectively by G and S in Eq. 5.)

This means that for a system SB of semantic biclusters, we can obtain an extra-sample (testing) quality estimate $Acc(sb)$ per Eq. 4 which was not possible with ordinary biclusters. Note that the testing sample set $Test = \{(g_k, s_k, a_k)\}$ needed for the estimate is drawn i.i.d. from $p(g, s, a)$ and is not expected to form a matrix. This has a positive practical implication for the evaluation procedure, which will be commented further in the experimental section.

Soft semantic biclusters

The last extension we introduce is that of *soft* semantic biclusters, motivated by the fact that in the terms sets T^γ, T^σ defining a semantic bicluster (T^γ, T^σ), some of the terms may be more important than others. The reason for this will follow from the algorithm implementations elaborated below. Here we simply assume that the sets T^γ, T^σ consist of pairs (t, w) where $t \in \gamma$ $(t \in \sigma)$ and the weight $w \in (0; 1]$. In this situation, we adapt the classification function to

$$\begin{aligned} sb(g, s) = 1 \text{ iff } & (T^\gamma, T^\sigma) \in SB \\ \text{and } & \sum_{(t,w)\in T^\gamma, (g,t)\in R_\gamma} w \geq \theta_G \\ \text{and } & \sum_{(t,w)\in T^\sigma, (g,t)\in R_\sigma} w \geq \theta_S \end{aligned} \tag{6}$$

where $\theta_G, \theta_S \in R$ are some real thresholds (hyper-parameters). Informally, the classifiers outputs 1 iff at least one of the biclusters in SB *supports* the classified tuple (g, s). The tuple is supported by a bicluster (T^γ, T^σ) if the weights of terms which are simultaneously (i) assumed by T^γ (T^σ, respectively), (ii) and among the annotations of g (s), sum up to at least θ_G (θ_S). The earlier definitions of $\widehat{Acc}$ and Acc apply to this redefined classifier sb as well.

Algorithms

At least two different strategies lend themselves to find a good system of semantic biclusters SB. The first option is to find a system B of ordinary biclusters first, and then identify the characteristic annotation terms T^γ and T^σ for each of the biclusters in B. The second option is to search directly in the space of (sets of) semantic biclusters, i.e. explore systematically various combinations of the annotation terms. We explore both strategies henceforth. In the first approach we employ an

Algorithm 1: Bi-directional enrichment.

input : $\mathbb{A}^{m\times n}$, $a_{i,j} \in \{0, 1, NA\}$; // NAs for testing fields
R_γ; R_σ; // gene (GO, KEGG) and location annotation relations
output: Π^S; // the matrix of gene and location p-values

/* Get list of biclusters, i.e., bi-sets of gene/location indices */
$A \leftarrow$ convertToSparseFIMIFormat($\mathbb{A}$);
$B \leftarrow$ PANDA+(A); // obtain ordinary biclusters
/* Get actual genes and locations, e.g., from $\mathbb{A}$ row/column names */
$G \leftarrow$ getAllGeneNames($\mathbb{A}$); // all genes in $\mathbb{A}$
$\gamma \leftarrow$ getAllGeneTerms(R_γ, G); // filter all gene terms relevant to $\mathbb{A}$
$S \leftarrow$ getAllLocationNames($\mathbb{A}$); // all locations in $\mathbb{A}$
$\sigma \leftarrow$ getAllLocationTerms(R_σ, S); // filter all location terms relevant to $\mathbb{A}$
$g \leftarrow |\gamma|$; $s \leftarrow |\sigma|$; $\Pi^S \leftarrow 0^{k\times(|\gamma|+|\sigma|)}$;
/* Annotate the individual biclusters */
for $k \leftarrow 1$ **to** $|B|$ **do**
  for $i \leftarrow 1$ **to** g **do**
    $\Pi^S_{k,i} \leftarrow$ enrichmentGet($B_{k,genes}$,γ_i,G, R_γ)
  end
  for $j \leftarrow 1$ **to** s **do**
    $\Pi^S_{k,g+j} \leftarrow$ enrichmentGet($B_{k,locs}$,σ_j,S, R_σ)
  end
end

existing biclustering algorithm and subject its results to an *enrichment analysis* [5] algorithm, revealing annotation terms which are enriched on either dimension of the produced biclusters. The alternative approach is materialized by an arrangement of classical symbolic machine-learning techniques known as decision rule and tree learning.

Bicluster enrichment analysis

The enrichment approach to semantic biclustering first searches for a set of ordinary biclusters. The goal is to find a small set of biclusters that cover as many 1's as possible and as few 0's as possible. In other words, we search for the most concise biset-based description that minimizes the occurrence of false positives and false negatives. The bicluster semantics are disregarded for the moment. In the field of biclustering, this is a well-known task that can be tackled with approximate pattern matching [13, 14, 15], non-negative matrix decomposition [16, 17], bipartite graph partitioning [18] or heuristic algorithms [19, 20, 21].

In our approach, we employed the popular PANDA+ tool [13] to accomplish the first step. PANDA+ adopts a greedy search that iteratively builds a sequence of biclusters. The constructed bicluster set gradually increases its coverage of the input matrix. This bicluster set is initially required to be noise-less, i.e. without false positives. In a subsequent step, PANDA+ extends the biclusters by allowing false positives. The main guiding 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 [22] before calling PANDA+.

In the second step, the biclusters are annotated in terms of prior domain knowledge, i.e., their semantics are revealed. In our case, we use the gene ontology (GO) terms [23, 24] and KEGG terms [25] to annotate the individual genes. The dedicated Drosophila location ontology terms [9] and Drosophila anatomy ontology

terms [26] were used to annotate the situations; in particular, these terms define the developmental stages and anatomical locations of the sample. Each non-trivial bicluster (comprising more than 1 gene and 1 stage) is annotated by all the terms (GO+KEGG and situation/anatomy ontology, respectively) whose enrichment exceeds the predefined statistical significance threshold. In order to avoid this hyper-parameter in our workflow, we propose setting 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 are scored proportionally to their statistical significance, yielding the weights w assumed by the classification principle in Eq. 6. We employed the topGO Bioconductor package [27] to find the GO terms and the Fisher test to reveal the KEGG and location ontology terms enriched in the individual biclusters.

This approach to semantic biclustering could as well be referred to as *bi-directional enrichment*. The procedure pseudocode is shown in Algorithm 1. Despite the NP-complexity of the general problem of finding the optimal set of biclusters [2], the suboptimal heuristic algorithm is computationally scalable. The size of the input matrix influences mainly the initial bicluster search; time complexity of PANDA+ is $\mathcal{O}(|B|mn^2)$ [13] where $|B|$ is the number of biclusters and $m = |G|, n = |S|$ are the dimensions of the expression matrix. The sizes $|\gamma|$, $|\sigma|$ of the annotation vocabularies influence solely the annotation step whose time complexity is $\mathcal{O}(|B|(|\gamma|*m+|\sigma|*n))$.

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 $\mathbb{M}$ of dimensions $(|G| \cdot |S|) \times (|\gamma| + |\sigma| + 1)$. Matrix $\mathbb{A}$ is unrolled into $\mathbb{M}$ so that each row of $\mathbb{M}$ corresponds to one element $a_{i,j}$ of $\mathbb{A}$ and has the form

$$t_1, t_2, \ldots t_{|\gamma|}, t_{|\gamma|+1}, t_{|\gamma|+2}, \ldots t_{|\gamma|+|\sigma|}, \mathit{expression} \tag{7}$$

where the first $|\gamma|$ numbers are binary indicators of annotation terms (acquiring a value of 1 iff the corresponding term is associated with gene in i'th row of $\mathbb{A}$), the subsequent $|\sigma|$ numbers are analogical indicators of situation ontology-terms for situation in j'th column of $\mathbb{A}$, and the last number is the expression indicator for the said gene and situation, and thus equals $a_{i,j}$. The transformation details are shown in Algorithm 2.

The next step is learning a classification model to predict *expression* from $t_1, \ldots t_{|\gamma|+|\sigma|}$. To this end, $\mathbb{M}$ represents the training data with individual rows such as (7) corresponding to learning examples with the last element being the class indicator. The model searched for takes the form of a list of conjunctive decision rules [28], each of which acquires the form

$$\wedge_{i \in I} t_i \wedge_{j \in J} t_{j+|\gamma|} \rightarrow \mathit{expression} \tag{8}$$

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

input : $\mathbb{A}^{m\times n}$, $a_{i,j} \in \{0,1,NA\}$; // NAs for testing fields
R_γ; R_σ; // gene (GO, KEGG) and location annotation relations
output: $\mathbb{M}^{(m\cdot n)\times(|\gamma|+|\sigma|+1)}$, $b_{i,j} \in \{0,1\}$

```
/* Genes are represented by a set of FBgn identifiers                    */
G ← getAllGeneNames(𝔸); // all genes in 𝔸
γ ← getAllGeneTerms(R_γ, G); // list all gene annotation terms
S ← getAllLocationNames(𝔸); // all locations in 𝔸
σ ← getAllLocationTerms(R_σ, S); // list all location terms
g ← |γ|; s ← |σ|;
for i ← 1 to m do
    T ← 0^(|γ|+|σ|+1); // term indicator vector initialization
    for j ← 1 to g do
        if (γ_j, G_i) ∈ R_γ then T_j ← 1;
    end
    for k ← 1 to n do
        for j ← 1 to s do
            if (σ_j, S_k) ∈ R_σ then T_{g+j} ← 1;
        end
        T_{|γ|+|σ|+1} ← a_{i,k}; // add expression indicator
        𝕄_{(i−1)·n+k,*} ← T;
    end
end
𝕄 ← filterGeneTerms(𝕄, Θ); // wrt to a given threshold Θ;
```

where the rule conditions $I \subseteq [1;|\gamma|], J \subseteq [1;|\sigma|]$ are learned selections of gene and situation ontology terms. The rule stipulates that a gene annotated with all the gene-ontology terms indexed by I is likely to be expressed in situations annotated with all the situation-ontology terms indexed by J. 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 (g, s) satisfying the conditions of rule (8). It is easy to see that P forms a submatrix of $\mathbb{A}$, i.e., there exists a permutation of $\mathbb{A}$'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 (8) is satisfied perfectly by the genes in the intersection of G and S, which is thus a rectangle.[2] Therefore, each rule such as (8) identifies a bicluster in $\mathbb{A}$.

Moreover, a rule set optimized for classification accuracy on training data such as (7) will produce those biclusters of $\mathbb{A}$ which contain a high number of 1's. 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's and no 0's 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 (8). Thus, the procedure described does indeed convey the semantic biclustering task.

In addition, we propose an variation to the workflow described, in which the rule-set learner is replaced by a *decision tree* learner [28]. Each vertex in a learned tree corresponds to one ontology term, and the test represented by the vertex determines

[2]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 [7].

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 training data (7) are binary, the learned tree is also binary. Each path from the root to one positive leaf can be rewritten as a rule in the form (8), except that some of the literals may be negated. For example, literal $\neg t_1$ expresses the condition that t_1 is *not* among the annotation terms. So the learned decision tree defines a set of semantic biclusters as the rule-set does, except these biclusters are defined in a more expressive language (allowing negation) than we considered in the original formalized model.

The main reason for exploring this decision tree alternative is that it is often claimed that decision trees 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 [29] to learn the rule-sets and decision trees, respectively. The JRip algorithm is an implementation of a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER) [30]. J48 is an implementation of the well-known C4.5 algorithm [31].

The time complexity of this approach is determined by the complexity of converting the $\mathbb{A}$ into $\mathbb{M}$, which is $\mathcal{O}(mn(|\gamma|+|\sigma|))$, and the complexity of the subsequent learning algorithm. In the case of binary decision trees, the runtime of the heuristic J48 algorithm grows linearly with the number of training instances and quadratically with the number of features [32], in our problem it is $\mathcal{O}(mn(|\gamma|+|\sigma|)^2)$. As the total number of annotation terms can be large, the actual runtime of this approach would be much larger than for the bi-directional clustering. For this reason, we perform a feature selection step prior to the learning step. The published JRip's time complexity [30] implies the learning complexity for our problems $\mathcal{O}(mnlog^2(mn))$. In other words, a large number of samples in $\mathbb{M}$ indicates a time consuming run if compared to its counterparts.

Evaluation method

Both biclustering and enrichment analyses are unsupervised data mining methods and the exact way of validating 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 semantic annotations discovered may either represent genuine characteristics of the biclusters, or the included terms may be enriched merely by chance. Distinguishing these two effects through a statistical test involves distributional assumptions which we cannot guarantee.

We solve the latter dilemma by measuring the quality of semantic biclusters from the point of view of *predictive classification*, and particularly using an extra-sample (testing) accuracy estimate as proposed in Eq. 4. This assumes that the available data is split randomly into a training partition where the semantic biclusters are found, and a testing partition where they are evaluated. The training split is a (strict) submatrix of the input matrix and thus its complement (the testing split) is not a matrix. Fortunately, a matrix form is not required of the testing split as explained in the Problem formalization section.

As stated already, the approach based on conventional biclustering and subsequent enrichment analysis results in a set of soft semantic biclusters inducing the

Algorithm 3: Predictive evauation of bi-directional enrichment.

input : Π^S; $\mathbb{A}^{m\times n}$, $a_{i,j} \in \{0, 1, \mathsf{NA}\}$; // NAs for training fields
$R_\gamma; R_\sigma$; // gene (GO, KEGG) and location annotation relations
parameters: $\theta_G; \theta_S$; // gene and location term score thresholds
p_{perm}; // p-val permutation threshold
output : $\mathbb{P}^{\mathrm{m}\times\mathrm{n}}$, $p_{i,j} \in \{0, 1, \mathsf{NA}\}$ // 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 and KEGG term indication vectors for all genes */
$G \leftarrow$ getAllGeneNames($\mathbb{A}$); // all genes in $\mathbb{A}$
$\mathbb{T}_G \leftarrow$ getTermsForGenes(R_γ, G); // a binary m×g incidence matrix
/* Get location term indication vectors for all stages */
$S \leftarrow$ getAllLocationNames($\mathbb{A}$); // all locations in $\mathbb{A}$
$\mathbb{T}_S \leftarrow$ getTermsForStages(R_σ, S); // 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** $|\gamma| + |\sigma|$ **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 $\mathbb{P}$ */
  $\mathbb{P}[\mathbb{T}_G\Pi^S_{k,1\ldots g} > \theta_G, \mathbb{T}_S\Pi^S_{k,g+1\ldots|\gamma|+|\sigma|} > \theta_S] \leftarrow 1$
end

classification principle described by Eq. 6. The latter depends on the two hyperparametric thresholds θ_G and θ_S, and their different choices result in different values of the accuracy measure (4). In such a situation, it is convenient to visualize the global performance profile through *ROC analysis*. Here, the accuracy measure (4) is decomposed into the *false positive rate* component FPr and the *true positive rate* TPr, both of which are functions of θ_G and θ_S. By varying these hyperparameters, a set of (FPr, TPr) points is obtained, forming the *ROC curve*. The area under this curve (termed AUC) represents the quality of the classifier for the entire range of the hyperparameters. The semantic biclustering validation procedure is summarized in Algorithm 3.

The approach based on rule and tree learning produces crisp semantic biclusters, and as such it induces classifiers in the standard form given by (3). For the sake of unified comparison, we also evaluate these classifiers through ROC analysis although they do not contain explicit threshold parameters. This is made possible by the employed JRip and J48 algorithms which provide confidence values along with the expression predictions. We make a positive expression call only if the corresponding confidence value exceeds a threshold Θ, and we obtain the ROC curve by varying Θ.

Results

Experimental datasets

We conducted our experiments on two real datasets. The first one is the Dresden ovary table [9]. 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 authors of the table believe [10] 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 in 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 [23, 27], there were 8,407 GO terms available altogether. The set of KEGG terms was considerably smaller, we dealt with 133 terms that annotated a limited set of 1,605 genes. For this reason, the importance of KEGG is smaller than the importance of GO. After minor data cleansing, the expression matrix has 6,510 rows (genes) and 100 columns (situations) with 47.5% positive data instances. The detailed data statistics can be found in Table 3.

The second experimental dataset comes from the same organism, i.e., Drosophila melanogaster, and captures the spatial gene expression in the larval imaginal discs. An imaginal disc is a part of insect larva from which the adult body parts develop. The dataset is a binary representation of an automatically processed large collection of fluorescent in situ 2D hybridization images. The images were collected for more than 1,000 genes in 4 different imaginal discs (wing, antenna-eye, leg and haltere). About 20 distinct locations (image segments) were distinguished for each disc, see Figure 1 for further details. A set of semantically annotated biclusters may help to reveal and understand the local expression patterns in larval development. Altogether, the binary imaginal disc dataset contains the expression of 1,207 genes in 72 different locations with 75.4% positive data entries. The detailed data statistics can be found in Table 4. Similarly to the Dresden ovary table, we assigned a set of GO and KEGG terms to each gene. 114 KEGG terms appeared in the annotation records of 423 distinct genes. Further, each segment of a particular imaginal disc was manually assigned a set of Drosophila anatomy ontology (DAO) terms [26]. The DAO consists of over 8,000 terms with broad coverage of Drosophila anatomy including the descriptions of imaginal discs and their compartments, we made use of 148 distinct terms. The summary ontology term counts are available in Table 5.

For the evaluation purposes, each data set was randomly split into a submatrix containing 70% of the original matrix elements, and the complement which was used as the testing set.

Experimental protocol

The bicluster enrichment method was run with the PANDA+ noise parameters that minimized the total cost of biclusters in the training set (i.e., the summarizing criterion that controls both bicluster size and the number of false positives and negatives). This setting can be reached in a fully unsupervised way and avoids both too noisy and too detailed sets of biclusters. For the ovary dataset, the statistical significance thresholds were set to 0.05 for genes and 0.1 for situations. For the imaginal disc dataset, the statistical significance thresholds were set to 0.01 for genes and 0.1 for situations. The method was run repeatedly with the following sets of match thresholds: $\theta_G \in \{1, 5, 10, 50\}$ and $\theta_S \in \{1, 5, 10, 50\}$. The results in ovary dataset suggested that precision decreases slowly with decreasing match

thresholds while recall grows quite rapidly. The best precision/recall trade-off is thus achieved 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. To the contrary, in imaginal discs we were able to find biclusters with strongly related location terms. For this reason, $\theta_S = 50$ seems to be the best threshold as it already provides a sufficient recall and its decrease only leads to decreasing precision.

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 the $\mathbb{M}$ matrix) that occurred in fewer than approximately 1‰expression entries (the train matrix rows) were removed. The cut-off threshold was found with the feature frequency histograms. Eventually, we worked with a train matrix size of 457,548×326 and 60,600×403, respectively. Besides speeding up the learning process, we avoided the annotation 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 the expression of unmeasured genes in unseen developmental stages.

Discussion

The bicluster enrichment approach seems to be the most reliable predictive method in datasets that can be described by a coherent biclusters whose size allows their reliable subsequent annotation. In the ovary dataset, the mean bicluster size exceeded 30,000 entries and the biclusters proved to generalize well. If given an unseen pair of positive (present) and negative (absent) expression entries, it correctly guesses the positive entry with more than a 82% 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, KEGG and location terms per bicluster was 59, 2 and 4 respectively (as the KEGG and location ontology deal with a smaller number of terms). This number of terms may make the interpretation hard for a human expert. At the same time, in more fragmented and difficult domains such as the imaginal disc dataset, the mean size of biclusters drops (we observed the mean bicluster size 3,998 in the imaginal disc dataset) and the biclusters seem to generalize worse. J48 proved to be the method that copes well with this increased fragmentation. The decision tree grows without an immediate decrease in its generalization power. 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 a tendency to be short and crisp with potentially lower recall.

Figure 2 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. Here, we work with the probabilistic outcome, the curve is constructed with different probability thresholds for assigning an example to the positive class.

Eventually, we compared the generalization ability independently in terms of gene and location annotation terms. Under this evaluation protocol, the test matrices were split into three parts, see Figure 3. The first submatrix denoted as kG (keepGenes), contains only the rows whose gene identifiers were already observed in the complementary train set while its columns correspond to the locations that were not observed there. Consequently, each biclustering method has to generalize towards the locations. The second submatrix denoted as kL (keepLocations), covers the locations already observed in the train set and the previously unobserved genes. Each biclustering method has to employ gene annotation terms to be able to predict here. Finally, the third submatrix *bd* contains the rest of testing entries. Bi-directional generalization has to be applied here. The results are summarized in Table 2. The main conclusion is that it is much easier to generalize in terms of locations than in terms of genes. The locations common for a bicluster tend to share location annotation terms observed for other genes with a similar local expression pattern. On the contrary, the description in terms of genes is often extensive with more difficult application to external genes. The bicluster enrichment method provides the best generalization for the *bd* region, where both the genes and locations were previously unseen.

Conclusion

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 and imaginal disc 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 biclusters.

In future work, the statistical implications of the non-standard way of splitting the data matrix into the (rectangular) training set and the testing set could be investigated. Furthermore, a method for semantic biclustering that would combine the complementary advantages of the proposed approaches could be devised. In principal, the biclustering enrichment ignores prior knowledge when searching for

biclusters. None of the biclusters have to be interpretable as a result. The rule and tree-based methods directly stem from prior knowledge and search for the most general conjunctive concepts that fit the training data at the risk of their overfitting. Finally, a biological interpretation of the results reached in particular domains could be provided.

Declarations

Ethics approval and consent to participate
Not applicable

Consent for publication
Not applicable

Availability of data and material
The Dresden ovary table is a publicly available dataset [9]. The Imaginal disc dataset is a dataset being built and analyzed in terms of our Czech Science Foundation project 14-21421S. The dataset will be made publicly available.

Competing interests
The authors declare that they have no competing interests.

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

Authors' contributions
JK proposed, implemented and tested the bidirectional enrichment method. FZ proposed the tree and rule learning method. FM implemented and tested it and was a major contributor in raw data preparations including the dedicated location ontologies. FZ and JK wrote the manuscript. All authors read and approved the final manuscript.

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

Authors' information (optional)
todo

References

1. 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)
2. 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)
3. 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)
4. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics **18**(suppl 1), 136–144 (2002)
5. Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., Mesirov, J.P.: Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A **102**(43), 15545–15550 (2005)
6. 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)
7. Zelezny, F., Lavrac, N.: Propositionalization-based relational subgroup discovery with RSD. Machine Learning **62**(1-2), 33–63 (2006)
8. Kuhna, A., Ducasseb, S., Girbaa, T.: Semantic clustering: Identifying topics in source code. Information and Software Technology **49**(3), 230–43 (2007)
9. Dresden Ovary Table. `http://tomancak-srv1.mpi-cbg.de/DOT/main`. [Online; accessed 15-February-2016]
10. 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)
11. Soulet, A., Kléma, J., Crémilleux, B.: In: Džeroski, S., Struyf, J. (eds.) Efficient Mining Under Rich Constraints Derived from Various Datasets, pp. 223–239. Springer, Berlin, Heidelberg (2007)
12. Miettinen, P., Vreeken, J.: Model order selection for boolean matrix factorization. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 51–59 (2011). ACM
13. 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)
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. 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)
16. 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)
17. Žitnik, M., Zupan, B.: Nimfa: A python library for nonnegative matrix factorization. The Journal of Machine Learning Research **13**(1), 849–853 (2012)
18. 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 (2001). ACM
19. 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), 71680 (2013)
20. 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)
21. van Uitert, M., Meuleman, W., Wessels, L.: Biclustering sparse binary genomic data. Journal of Computational Biology **15**(10), 1329–1345 (2008)
22. Frequent Itemset Mining Implementations Repository. `http://fimi.ua.ac.be/`. [Online; accessed 15-February-2016]
23. Gene Ontology Consortium. `http://geneontology.org/`. [Online; accessed 15-February-2016]
24. Consortium, G.O., *et al.*: Gene ontology consortium: going forward. Nucleic acids research **43**(D1), 1049–1056 (2015)
25. Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., Tanabe, M.: Kegg as a reference resource for gene and protein annotation. Nucleic acids research, 1070 (2015)
26. Costa, M., Reeve, S., Grumbling, G., Osumi-Sutherland, D.: The drosophila anatomy ontology. Journal of Biomedical Semantics **4**(1), 1–11 (2013). doi:10.1186/2041-1480-4-32
27. Alexa, A., Rahnenfuhrer, J.: topGO: topGO: Enrichment Analysis for Gene Ontology. (2010). R package version 2.4.0
28. Russell, S.J., Norvig, P., Davis, E.: Artificial Intelligence, 3rd ed. edn. Prentice Hall, Upper Saddle River (c2010)
29. Witten, I.H., Frank, E., Hall, M.A.: Data Mining, 3rd ed. edn. Morgan Kaufmann, Burlington (c2011)
30. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on Machine Learning, pp. 115–123 (1995)
31. Quinlan, J.R.: C4.5. Morgan Kaufmann Publishers, San Mateo, Calif. (c1993)
32. Martin, J.K., Hirschberg, D.: On the complexity of learning decision trees. In: International Symposium on Artificial Intelligence and Mathematics, pp. 112–115 (1996). Citeseer
33. Gomez-Skarmeta, J.L., Campuzano, S., Modolell, J.: Half a century of neural prepatterning: the story of a few bristles and many genes. Nature Reviews Neuroscience **4**(7), 587 (2003)

Figures

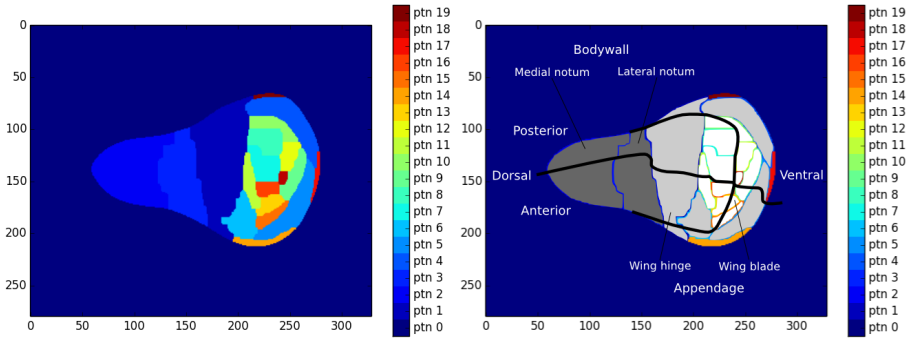

Figure 1 Segmentation of an imaginal disc. An example of segmentation of an imaginal disc (left), altogether with its annotation by the Drosophila ontology terms (right). The disc is split into 20 segments distinguished in colors, the split was found to best capture the gene expression patterns observed in the individual in situ hybridization images. The annotation stems from [33].

Tables

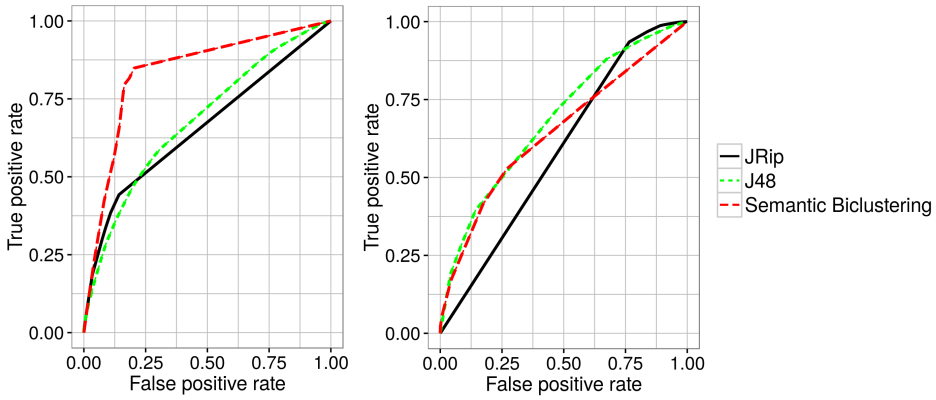

Figure 2 Semantic biclustering ROC curves for Drosophila ovary table (left) and Imaginal disc dataset (right).

Figure 3 Train and test matrices.

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

Dataset	Method	AUROC	# of biclusters	# of terms per bicluster
Ovary	Bicluster Enrichment	0.823±0.006	11.8±1.5	64.8±3.4
	Rules (JRip)	0.636±0.01	102.6±21.5	7.1±0.61
	Tree (J48)	0.659±0.01	109.9±5.2	25.4±2.0
IDiscs	Bicluster Enrichment	0.608±0.03	16.4±4.7	47.9±2.13
	Rules (JRip)	0.565±0.01	25.9±6.2	7.89±0.53
	Tree (J48)	0.627±0.05	20.6±11.09	11.01±4.71

Table 2 Generalization in terms of genes and locations. The table compares the AUROC for three different settings. kG tests the generalization across locations, kL the generalization across genes and bd the generalization in both the dimensions.

Dataset	Method	kG	kL	bd
Ovary	Bicluster Enrichment	0.929±0.013	0.677±0.03	0.818±0.014
	Rules (JRip)	0.692±0.02	0.583±0.01	0.583±0.02
	Tree (J48)	0.725±0.002	0.604±0.01	0.604±0.02
IDiscs	Bicluster Enrichment	0.705±0.06	0.560±0.02	0.593±0.03
	Rules (JRip)	0.588±0.01	0.546±0.01	0.537±0.02
	Tree (J48)	0.630±0.06	0.627±0.05	0.602±0.04

Table 3 Drosophila ovary table statistic.

	complete dataset	Train	Test		
		all	keepLocations	keepGenes	bd
#of rows/genes	6,510	5,447	1,063	5,447	1,063
#of columns/locations	100	84	84	16	16

Table 4 Imaginal disc dataset statistic.

	complete dataset	Train	Test		
		all	keepLocations	keepGenes	bd
#of rows/genes	1,207	1,010	197	1,010	197
#of columns/locations	72	60	60	12	12

Table 5 The number of annotation terms available for our experimental datasets.

	GO	KEGG	DAO	DLO
Ovary	8,407	1,605	-	100
IDisc	5,083	423	147	-