Bachelor Thesis

# Semantic Biclustering Optimization

**Petr Kubelka**

Supervisor: doc. Ing. Jiří Kléma Ph.D.

Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
May, 2018

# BACHELOR‘S THESIS ASSIGNMENT

## I. Personal and study details

Student's name: **Kubelka Petr** Personal ID number: **457022**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Branch of study: **Computer and Information Science**

## II. Bachelor’s thesis details

Bachelor’s thesis title in English:

**Semantic Biclustering Optimization**

Bachelor’s thesis title in Czech:

**Optimalizace sémantického dvojshlukování**

Guidelines:

1. Get familiar with the topic of clustering, biclustering and semantic biclustering.
2. Review the existing approaches to integration of prior knowledge into the process of biclustering, focus on the domain of molecular data.
3. Propose and implement a modification of the existing algorithms of semantic biclustering, concentrate on early concurrent application of all the characteristics of high-quality biclustetrs (size, accuracy, biological interpretability).
4. Work both with artificial and real data provided by your supervisor.
5. Evaluate your algorithm and its possible modifications on the data mentioned above. Compare your results with the existing benchmarks.

Bibliography / sources:

[1] Klema, J., Malinka, F., Zelezny, F.: Semantic biclustering for finding local, interpretable and predictive expression patterns. BMC Genomics, 18:752, 2017.

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

[3] Subramanian, A., Tamayo, P., Mootha, V. K., et al.: Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences, 102(43), 15545-15550, 2005.

[4] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 6(2), 182-197, 2002.

Name and workplace of bachelor’s thesis supervisor:

**doc. Ing. Jiří Kléma, Ph.D., Intelligent Data Analysis, FEE**

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **12.01.2018** Deadline for bachelor thesis submission: **25.05.2018**

Assignment valid until: **30.09.2019**

| | | |
|---|---|---|
| doc. Ing. Jiří Kléma, Ph.D.<br>Supervisor’s signature | doc. Ing. Tomáš Svoboda, Ph.D.<br>Head of department’s signature | prof. Ing. Pavel Ripka, CSc.<br>Dean’s signature |

# Author statement for undergraduate thesis:

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, date.............. ........................

signature

# Acknowledgements

I would like to thank doc. Ing. Jiří Kléma Ph.D. for his kind guidance, time, very helpful advice, and patience. I would also like to thank Ing. František Malinka who always quickly provided all the necessary data and answered all the questions. Special thanks go to my family for unconditional support and my girlfriend for her patience and graphics work.

Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

# Abstract

Biclustering is a popular approach to gene expression data analysis, namely for the discovery of gene sets that are functionally related by specific biological conditions. The purpose of the thesis is to test and compare the current semantic biclustering algorithm with a new approach. The new method uses a multi-criteria optimization that implements prior knowledge of genes and locations in contrast with current semantic biclustering algorithm when searching for biclusters. We propose three different approaches to aggregate a Pareto set solutions into a bicluster. These Pareto sets were obtained from modified multi-criteria optimization algorithm. We evaluate the good quality of acquired biclusters from Pareto sets by their following generalization ability to describe unseen entries of the gene expression dataset.

**Keywords:** Clustering, biclustering, semantic biclustering, bioinformatics

# Abstrakt

Dvojshlukování je populární způsob, jak analyzovat data genové exprese, zejména při objevování setů genů, které jsou si funkčně podobné v rámci specifických biologických podmínek. Cílem této bakalářské práce je otestovat a porovnat současný algoritmus sémantického dvojshlukování s novým přístupem. Nová metoda využívá vícekriteriální optimalizace, která implementuje předchozí znalosti o genech a lokacích při hledání dvojshluků na rozdíl od metody současné. Představujeme tři různé přístupy, jak agregovat Pareto množinu řešení do dvojshluku. Tyto Pareto množiny byly získány z modifikace algoritmu využívajícího vícekriteriální optimalizaci. Kvalitu obdržených dvojshluků z Pareto setů ověřujeme jejich následující shrnující schopností popsat neviděná data genové exprese.

**Klíčová slova:** Shlukování, dvojshlukování, sémantické dvojshlukování, bioinformatika

# Contents

**1 Introduction 11**

**2 Related algorithms 12**
2.1 Clustering . . . . . . . . 12
2.1.1 Centroid methods . . . . . . . . 12
2.1.2 Connectivity methods . . . . . . . . 12
2.1.3 Density methods . . . . . . . . 13
2.2 Biclustering . . . . . . . . 13
2.2.1 Biclustering algorithms based on evaluation measures . . . . 13
2.2.2 Non metric-based biclustering . . . . . . . . 15
2.3 Semantic biclustering . . . . . . . . 16

**3 Review of the existing approaches to integration of prior knowledge 17**
3.1 Semantic clustering . . . . . . . . 17
3.2 Semantic biclustering . . . . . . . . 18

**4 Data 20**
4.1 Expression matrix . . . . . . . . 20
4.2 Gene Ontology Terms . . . . . . . . 21
4.3 Kyoto Encyclopedia of Genes and Genomes . . . . . . . . 21
4.4 Location terms . . . . . . . . 21
4.5 Pareto set . . . . . . . . 21
4.6 Bicluster detailed description . . . . . . . . 22
4.7 Artificial dataset . . . . . . . . 23
4.8 Real datasets . . . . . . . . 23
4.8.1 Drosophila melanogaster . . . . . . . . 23
4.8.2 Gene expression matrix of Mice . . . . . . . . 24
4.9 OBO::Parser . . . . . . . . 24

**5 Implementation 26**
5.1 Gene expression matrix split . . . . . . . . 27
5.2 Approaches to Pareto set aggregation . . . . . . . . 27
5.2.1 Empirical distribution-based algorithm I . . . . . . . . 28
5.2.2 Empirical distribution-based algorithm II . . . . . . . . 29

5.2.3 Weighted random sample . . . . . . . . . . . . . . . . . . . . . 29
5.3 Using the semantic annotation . . . . . . . . . . . . . . . . . . . . . . . . 30
5.4 Annotation of genes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
5.4.1 Dividing ontology by molecular function . . . . . . . . . . . . 32
5.4.2 KEGG superiority . . . . . . . . . . . . . . . . . . . . . . . . . . 33
5.5 Annotation of locations . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
5.6 Applying the semantic . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33
5.6.1 Score distribution . . . . . . . . . . . . . . . . . . . . . . . . . . 33
5.7 The area under the Receiver Operating Characteristic curve . . . . . 34

**6 Evaluation** **35**
6.1 Dataset parameters . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
6.2 Tuning the thresholds for the gene and location scores . . . . . . . . 36
6.3 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
6.3.1 Artificial dataset . . . . . . . . . . . . . . . . . . . . . . . . . . 37
6.3.2 Imaginal disc dataset . . . . . . . . . . . . . . . . . . . . . . . . 38
6.3.3 Ovary dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
6.3.4 M2801 dataset . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
6.3.5 Runtimes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44
6.3.6 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

**7 Framework** **46**

**8 Conclusion** **48**

**9 DVD Content** **50**

**Bibliography** **51**

# List of Figures

4.1 Microarray containing genes and locations $^{1}$ . . . . . . . . . . . . . . . 20
4.2 Model situation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
4.3 Drosophila melanogaster, fruit fly$^{2}$ . . . . . . . . . . . . . . . . . . . . 24
4.4 Help for OBO::Parser . . . . . . . . . . . . . . . . . . . . . . . . . . 25
4.5 Short example of init1.txt . . . . . . . . . . . . . . . . . . . . . . . . 25

5.1 Algorithm workflow . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
5.2 Train / test description . . . . . . . . . . . . . . . . . . . . . . . . . 27
5.3 Weighted random sample example . . . . . . . . . . . . . . . . . . . . 29

6.1 AUROC of imaginal discs . . . . . . . . . . . . . . . . . . . . . . . . 41
6.2 AUROC of Ovary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43

7.1 File distribution in folders . . . . . . . . . . . . . . . . . . . . . . . . 47

9.1 DVD content . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50

# List of Tables

5.1 Confussion matrix . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

6.1 Number of unique terms . . . . . . . . . . . . . . . . . . . . . . . . . 35
6.2 Number of genes and locations described by terms . . . . . . . . . . 36
6.3 Thresholds for gene and location scores (G: 0.05, L: 0.1) . . . . . . . 36
6.4 Thresholds for gene and location scores (G: 0.05, L: 0.15) . . . . . . 36
6.5 Artificial Discs dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.1) . . . . . . . . . . . . . . . . . . . . . . . . 37
6.6 Artificial Discs dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.15) . . . . . . . . . . . . . . . . . . . . . . . 38
6.7 Imaginal Discs dataset results for current semantic biclustering algorithms (G: 0.05, L: 0.1) . . . . . . . . . . . . . . . . . . . . . . . . . 38
6.8 Imaginal Discs dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.1) . . . . . . . . . . . . . . . . . . . . . . . . 39
6.9 Imaginal Discs dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.15) . . . . . . . . . . . . . . . . . . . . . . . 39
6.10 Imaginal Discs dataset 2 results for our modified semantic biclustering algorithm (G: 0.05, L: 0.1) . . . . . . . . . . . . . . . . . . . . . . . . 40
6.11 Imaginal Discs dataset 2 results for our modified semantic biclustering algorithm (G: 0.05, L: 0.15) . . . . . . . . . . . . . . . . . . . . . . . 40
6.12 Ovary dataset results for current semantic biclustering algorithms (G: 0.05, L: 0.1) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
6.13 Ovary dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.1) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
6.14 Ovary dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.15) . . . . . . . . . . . . . . . . . . . . . . . . . . . 42
6.15 M2801 dataset thresholds for different methods . . . . . . . . . . . . 44
6.16 M2801 dataset results for our modified semantic biclustering algorithm 44
6.17 Runtimes of searching for optimal threshold . . . . . . . . . . . . . . 45
6.18 Runtimes of algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . 45

# Chapter 1

# Introduction

In bioinformatics, biclustering has become remarkable in gene enrichment analyses of gene expression data and its ability to describe gene sets featuring in these expression data. Application of these algorithms helps biologists properly understand underlying biological function processes to uncover pathways and structures of the diseases [Holden et al., 2008] or find new genes connected with spontaneous premature birth [Manuck et al., 2016]. This thesis continues in work on the topic of semantic biclustering of a [Klema et al., 2017] and Intelligent Data Analysis Research Group.

First, we familiarise the reader with the topic clustering, biclustering, and semantic biclustering. The second step is to review the existing approaches to integration of prior knowledge into the process of biclustering mainly focused on the domain of molecular data. Then we describe the given data and their properties. After the introduction into the problem we propose and implement a modification of the existing algorithm of semantic biclustering concentrated on the early concurrent application of all the characteristics of high-quality biclusters in terms of size, accuracy, and biological similarity. These steps are evaluated both on real and artificial datasets and compared with existing benchmarks.

We start with the split of data into the training and testing split. Then we present three techniques to acquire the aggregation of a Pareto set and later we use the semantic of genes and locations of the acquired bicluster to predict the unseen entries in gene expression matrix.

We will implement a framework in R language that is the language most used in bioinformatics and statistics. We provide the framework that takes any Pareto set of biclusters, if gene ontologies and locations ontology are provided, and run the enrichment analysis to predict unseen data entries of gene expression matrix. The measurement of quality of Pareto set and its aggregation represents the generalization ability of its semantic annotation to describe unseen data entries in gene expression matrix.

## Chapter 2

# Related algorithms

In this chapter, we introduce the concepts of clustering, biclustering, and semantic clustering. We give several examples of algorithms in each section to introduce with these algorithms and their properties.

## 2.1 Clustering

*Clustering* [Hartigan, 1975] is a machine learning technique that groups similar objects into clusters. We offer the summarization three types of clustering methods:

### 2.1.1 Centroid methods

In this section, the well-known algorithms for clustering objects are The k-means algorithm, the k-means algorithm, the k-medoids algorithm. The most basic one, K-means needs an initial number of k cluster centroids. This algorithm minimizes within-cluster sum of squares. K-means assigns for each sample t $\in$ T one cluster centroid c $\in$ C, where the distance between each centroid c and sample t is minimized. Formally:

$$\underset{C}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in C_i} \left\| t - \mu_i \right\|^2$$

where $\mu$ is mean of points in T.

### 2.1.2 Connectivity methods

Hierarchical clustering algorithm [Pedregosa et al., 2011] build a tree where the root a is cluster containing all the other clusters. The end of the tree are leaves belonging only to one cluster. This kind of graph is called a dendrogram. Here we need to choose the metric between every two points for the algorithm, such as Euclidian distance, Cosine, Minkowski, Chebychev and linkage method. The second method that has to be chosen is the decision when two clusters are merged, such as the average method, the nearest neighbor, the farthest neighbor.

### 2.1.3 Density methods

DBSCAN and OPTICS define clusters as connected dense regions in the data space. Modifications of DBSCAN are used to find patterns in biological data.

DBSCAN [Edla and Jana, 2012] stands for Density-Based Spatial Clustering of Applications with Noise. In contrast with the K-means algorithm, we do not specify the number of clusters in DBSCAN. Instead, we specify the maximum distance between two points for them to be in same the cluster, the minimal sample in the cluster and the metric to compute the distance between every two points.

## 2.2 Biclustering

More suitable technique for gene expression is biclustering [Pontes et al., 2015a] that searches for local patterns of gene expression simultaneously. This algorithm was first introduced by [Cheng and Church, 2000] who proposed a biclustering algorithm of biological gene expression data and it becomes a popular technique ever since. Most of the biclustering techniques use measure and cost functions to evaluate the quality of bicluster. The biclustering was proved to be NP-hard problem [Orlin, 1977] however heuristic solutions are used to solve the problem.

The algorithm generates submatrices which show similar behavior for genes if in the same bicluster. Nowadays biclustering is a favorite technique in gene expression problems.

We divide biclustering algorithms into two types [Pontes et al., 2015b], biclustering algorithms based on evaluation measures and non-metrics based biclustering algorithms.

### 2.2.1 Biclustering algorithms based on evaluation measures

In this section, we review numerous algorithms used in biclustering based on evaluation measures. Biclustering is NP-hard problem, thus heuristics solutions are needed to be able to search the space of solutions. The NSGA-II [Deb et al., 2002] is reviewed in this section since it belongs to this class of biclustering algorithms. We review this algorithm at the end since it is the most complex one.

#### Cheng and Church

The first ever generalized bicluster algorithm for biological expression data was introduced by [Cheng and Church, 2000]. Their algorithm uses Mean Squared Residue (MSR) measure that evaluates the coherence of the genes and conditions of a bicluster.

Let $a_{ij}$ be the element of the expression matrix $\mathbf{A}$ representing the logarithm of the relative abundance of the mRNA of the $i$ gene under the condition $j$. The pair $(\mathbf{I}, \mathbf{J})$ species a submatrix $\mathbf{A_{IJ}}$ with the following mean squared residue score.

Let **I** be a set of genes and **J** the set of locations.
where:

$$a_{iJ} = \frac{1}{|J|}\sum_{j} a_{ij}, a_{Ij} = \frac{1}{|I|}\sum_{j} a_{ij}, a_{IJ} = \frac{1}{|I||J|}\sum_{i,j} a_{ij} \tag{2.1}$$

where: $a_{iJ}$ is the row means for the bicluster, $a_{Ij}$ is the column means for the bicluster and $a_{IJ}$ is the overall means for the bicluster.
The residue of element is $a_{ij} - a_{iJ} - a_{Ij} + a_{IJ}$ and the biclusters mean squared residue score is:

$H(I,J) = \frac{1}{|I||J|}\sum_{i,j}(a_{ij} - a_{iJ} - a_{Ij} + a_{IJ})^2$

A submatrix $A_{IJ}$ is called a $\delta$-bicluster

if H(I,J) $\leq \delta$ for some $\delta \geq 0$. This algorithm uses greedy iterative search to minimize MSR and finds bicluster one by one and then hides the found bicluster by replacing each element with a random number from chosen range and following which it searches for different bicluster again.

### SMSR-based biclustering (SMSR-CC)

SMSR-based biclustering (SMSR-CC) implemented by [Mukhopadhyay et al., 2009] use similar strategy to [Cheng and Church, 2000] (CC) but the Mean Squared Residue (MSR) is scaled and called (Scaling MSR). Here CC is used twice, first MSR is used as the evaluation measure and then SMSR is used.

### Maximum Similarity bicluster algorithm (MSB)

[Liu and Wang, 2006] This algorithm first introduces similarity score for biclusters. It has the advantage of finding the optimal solution in polynomial time in contrast to the rest of the algorithms.

### Weighted Fuzzy-Based Maximum Similarity Bicluster algorithm (WF-MSB)

Weighted Fuzzy-Based Maximum Similarity Bicluster algorithm is an improvement based on MSB algorithm. This algorithm [Chen et al., 2011] provides us biclusters that are the most similar and biclusters that are the most dissimilar as well.

### Flexible Overlapped biClustering (FLOC)

FLOC [Yang et al., 2005] is an improvement of Cheng and Church algorithm with different measuring technique. FLOC use iterative process from random biclusters to improve the total quality of the biclustering by finding k overlapping biclusters by one run.

#### NSGA-II

A multiobjective optimization problem is solved by evolving a population of solutions and conclude Pareto-optimal set from the population in a single run. One of the algorithms is NSGA-II [Deb et al., 2002], NSGA-II stands for nondominated sorting genetic algorithm II. NSGA-II uses crossover, mutation and genetic operators to provide a population of solutions in form of Pareto set. This algorithm has low complexity $O(MN^2)$ and has the ability to find a Pareto-optimal set of solutions in one simulation run. Later in Section 4.5 we describe the use of the modification of NSGA-II to our problem.

#### Mitra's and Banka's first use of NSGA-II

Mitra and Banka [Mitra and Banka, 2006] first implemented a Multi-Objective evolutionary algorithm (MOEA) based on Pareto dominance and demonstrating that it performs better than [Yang et al., 2005] or [Cheng and Church, 2000].

### 2.2.2 Non metric-based biclustering

In this section, we review biclustering algorithms that search the space of solutions without any evaluations measures.

#### Statistical-Algorithmic Method for Bicluster Analysis (SAMBA)

SAMBA [Tanay et al., 2002] first builds a bipartite graph with locations and genes, where edges correspond to significant expression changes. Every pair has assigned weights so heavy bipartite sub-graphs correspond to significant bicluster.

#### QUBIC

QUBIC [Li et al., 2009] has been presented as a QUalitative BIClustering algorithm. This algorithm recognizes biclusters with the shifting pattern and in addition, it recognizes scaling patterns as a new feature. This algorithm is one of a few that was implemented in C language unlike in R language.

#### CoBi (Pattern-based Co-Regulated Biclustering of Gene Expression Data)

CoBi [Roy et al., 2013] makes use of a tree to group, expand and merge genes according to their expression patterns. In order to group genes in the tree, a pattern similarity between two genes is defined.

#### FABIA

FABIA [Hochreiter et al., 2010] stands for factor analysis for bicluster acquisition. This algorithm assumes realistic non-gaussian signal distributions with heavy tails. Authors of FABIA concluded that FABIA is superior to CC, SAMBA, FLOC, and another 8 algorithms mentioned in [Hochreiter et al., 2010].

## 2.3 Semantic biclustering

This technique was first introduced by [Kléma et al., 2016] and further developed in [Klema et al., 2017] . *Semantic biclustering* searches for homogenous submatrices of data in the same way that we did in standard biclustering, but in addition, we demand that it to be possible to annotate the produced bicluster in terms of semantic annotations. Two strategies are presented, biclustering algorithm using the semantic and tree learning algorithm.

In case of **Bicluster enrichment analysis** first the biclusters are found in dataset using PANDA+ [Lucchese et al., 2014] algorithm. Then each found bicluster is annotated by terms found by enrichment analysis [Subramanian et al., 2005] so its semantics is revealed. Finally, biclusters are applied to classify the unseen data entries.

The second strategy **Rule and tree learning** transfer the problem into a classification-learning problem. Dataset is unrolled into a vector and rule-set learner or decision tree learner is used. [Klema et al., 2017] used implementation of JRip proposed by [Cohen, 1995] and J48 [Quinlan, 2014] algorithms from the WEKA machine-learning software [Hall et al., 2009].

A disadvantage of semantic biclustering presented by [Klema et al., 2017] is that when the homogenous biclusters are being found in a dataset, no prior knowledge is used since PANDA+ only searches for homogenous binary biclusters. This is however the advantage of our algorithm, where multicriterial optimization is used to find biclusters considering size, accuracy, and biological similarity.

# Chapter 3

# Review of the existing approaches to integration of prior knowledge

In this chapter, we review the existing approaches to integration of prior knowledge into biclustering algorithms. We introduce semantic clustering that benefits from the integration of prior knowledge and then we focus on biclustering algorithms using integration of the prior knowledge in the domain of molecular data. However, we also present approaches that integrate the prior knowledge in semantic web and coordinated relationships in text analysis.

## 3.1 Semantic clustering

First, we introduce semantic clustering that expands common clustering with the condition to find clusters based on their given prior domain knowledge. This allows us to conveniently represent the results. In bioinformatics, usually clusters of genes are found and then gene enrichment analysis is used. This is what we call semantic clustering. This approach is used in [Krejnik and Klema, 2012], [Verbanck et al., 2013] and [Kuhn et al., 2007]. Semantic clustering also used in software-engineering [Kuhn et al., 2007].

[Krejnik and Klema, 2012] introduce **functional clustering** (FC) and show its ability to be useful in gene expression data. The paper is focused on a method that reduces the dimensionality of gene expression data. The gene expression data describing genes are replaced by features that correspond to the centroids of the gene clusters obtained by K-centroids algorithm and are then used for classifier learning. Five different classification algorithms were used. Support vector machines, random forests, C4.5, naïve Bayes and nearest neighbor. [Krejnik and Klema, 2012] did not conclude significant difference between any two classification algorithms except random forests versus C4.5 and random forests versus support vector machines. [Krejnik and Klema, 2012] conclude that functional clustering can overcome the performance of random clustering where no biological knowledge is used.

[Verbanck et al., 2013] employs the external biological knowledge into clustering to discover possible relationships between genes. A new distance between genes is used. This distance between genes is computed from prior biological knowledge and then used in K-means algorithm to acquire gene clusters. Their usage of K-means is compared with heatmap and propose a number of better cluster candidates for interpretation than heatmap. The [Verbanck et al., 2013] believe their obtained clusters are able to help biologists to develop a new hypothesis on relationships of genes.

[Mitra and Ghosh, 2012] employs the algorithm called clustering large applications based on RAN-domized search (CLARANS). [Mitra and Ghosh, 2012] also focus on reduction of dimensionality and use feature selection using prior biological knowledge. CLARANS is medoid-based clustering algorithm. When CLARANS searches for k medoids, the problem is converted into a search through a graph. Each iteration CLARANS select a set of new neighbor nodes as a new medoids, where neighbors are considered every two nodes that differ by one object. Medoids obtained by CLARANS are then used by classifications algorithms which show better accuracy for all classifications algorithms when using biological knowledge.

[Kuhn et al., 2007] suggests to use the information obtained by gaining linguistic information (semantic) found in source code, such as names of variables and code documentation. [Kuhn et al., 2007] claims that in order to look after the code, 60 % of the time is spent on understanding what code does. The authors use latent semantic indexing to build similarity index for further clustering. The clustering algorithm of their choice is dendrogram. The clusters are called linguistic topics since they contain the info about used language.

## 3.2 Semantic biclustering

[Liu et al., 2004] first implements an algorithm that generates gene clusters reflecting gene function categories from Gene Ontology. The algorithm is called Smart Hierarchical Tendency Preserving clustering (SHTP-clustering) and directly includes Gene Ontology Information into clustering process. By using bicluster model called the Tendency Preserving cluster, SHTP algorithm generates TP-cluster tree where any subtree can be reverted back to Gene Ontology hierarchy.

[Nepomuceno et al., 2015] implements an algorithm that uses similarity biological measures FracGO using biological enrichment and SimNTO using overlapping among GO annotations of pairs of genes. The most important part of [Nepomuceno et al., 2015] is the fitness function that integrates the biological knowledge that is used to determine the quality of the found bicluster from the biological perspective.

[Soulet et al., 2007] integrates prior knowledge with the genetics domain into constraints that are connected both with rows and columns of a dataset. This knowledge is then used to filter irrelevant biclusters.

[Pio et al., 2012] integrate microRNA data by biclustering techniques for signaling networks analysis.[Pio et al., 2012] algorithm HOCCLUS2 can rank biologically significant interaction networks.

[Gohari and Tarokh, 2016] demonstrate a hybrid system using semantic information and biclustering technique. [Gohari and Tarokh, 2016] is used in semantic web recommendations for a user. The ontology is made up of user demographic information, user preferences of product and item ontology. Missing data are predicted from the users' ontology and the item ontology. The second step uses biclustering algorithm to obtain clusters of users and items from which a list of top recommendations for users is generated.

[Sun et al., 2016] introduced the algorithm called BiSet. BiSet's main goal is to clearly show coordinated relationships between objects that might be hidden or hard to see. The algorithm uses semantic edge bundling that acquires data from biclustering algorithms. This biclustering algorithm finds biclusters from computing coordinated relationships. First entities have been extracted from a document with entity recognizer algorithm. Then LCM [Uno et al., 2004] and CHARM [Zaki and Hsiao, 2005] are used as biclustering algorithms to obtain biclusters from entities. Biclusters are bundled and connected with edges to reveal their semantic.

# Chapter 4

# Data

In this thesis, we deal with two principal data types. The first is provided measurements, the second is generated annotations from the provided measurements. The provided measurements are described by two sets, one describing the row (genes) of bicluster and the second describing the column (locations) of bicluster. These measurements are annotated by their terms. We focus on Gene ontology (GO) database, however, our framework is capable of handling any kind ontology database if terms for genes and columns are provided in the proper format as it is shown in Section 4.9. We acquired 3 real datasets. Two datasets describe *Drosophila melanogasters* (fruit fly), namely its ovaries and imaginal discs. One new dataset that has not been tested by [Klema et al., 2017] is *Mus musculus* (mice tissue).

## 4.1 Expression matrix

The dataset is obtained from measurements that are stored in so-called ”microarrays”. Below we can see two samples of microarrays.

Figure 4.1: Microarray containing genes and locations [1]

## 4.2 Gene Ontology Terms

Gene Ontology Terms (GO terms) [Consortium, 2016] [Ashburner et al., 2000] are structured controlled ontologies or vocabularies that describe the products of genes and yhe relationships between them in their association in:

1. cellular components (CC),
2. biological processes (BP),
3. molecular functions (MF).

These terms are provided by Gene Ontology Consortium. Most of the annotations are created by automated electronic annotation. [Balakrishnan et al., 2013] In 2013, 1.1 million annotations were made by biological curators. The rest, over 126 million annotations were created by automated electronic annotations.

## 4.3 Kyoto Encyclopedia of Genes and Genomes

KEGG [Kanehisa and Goto, 2000] (Kyoto Encyclopedia of Genes and Genomes) is alternative to GO terms. However, KEGG does have a significantly lower number of annotations than GO have due to the fact that KEGG is manually created by biological curators that acquire the knowledge from literature and scientific papers.

## 4.4 Location terms

For each dataset, there is a description of each location by its corresponding terms. For example, in case of *drosophila melanogaster*, the locations are described by Drosophila location ontology (DLO) terms [Jambor et al., 2015] and Drosophila anatomy ontology (DAO) terms [Costa et al., 2013]. These terms describe the development in each stage and its anatomical locations.

## 4.5 Pareto set

Our input data are sets of Pareto optimal solutions. Each Pareto set was generated using NSGA-II algorithm [Deb et al., 2002] with certain modifications made by František Malinka. This algorithm searches for $k$ biclusters where each bicluster is described by Pareto set with the size of 200 biclusters. This modification optimizes 3 criteria:

1. the size of the bicluster - the bicluster must not be a 1x1 matrix, generally, the bicluster covers a large part of a dataset,

[1] By Schutz `https://commons.wikimedia.org/wiki/User:Schutz`, licensed under [CC BY-SA 2.5 (https://creativecommons.org/licenses/by-sa/2.5)] `https://commons.wikimedia.org/wiki/File:Affymetrix-microarray.jpg`

2. the accuracy of the bicluster - we want our bicluster to contain as many 1's as possible while leaving 0's outside the bicluster,
3. their semantic similarity - we require a semantic similarity between rows and columns (genes with similar function, the same location of a gene or same development phase).

Each bicluster in generated Pareto set is described by a pair of row $r$ and column $c$ vector and thus simplified as a bicluster in terms of size. This offers us a simplified computational task and lower requirements on disc space. Later in Algorithm 1 and Algorithm 2 we refer to Pareto set as a $paretoSet^{k \times m \times n}$ where $k$ is the number of Pareto sets, $m$ is the length of row vector and $n$ is the length of column vector.

## 4.6 Bicluster detailed description

Each bicluster in Pareto set is binary matrix $\mathbb{A}^{m \times n}$ where each binary element $a_{i,j} \in \{0, 1\}$ can be constructed as the minimum of corresponding elements of the vectors $r = (r_1, \ldots, r_m)$ and $c = (c_1, \ldots, c_n)$, $m, n \in \mathbb{N}$, where $\mathbb{A}_{i,j} = min(r_i, c_j)$. If $a_{i,j} \in \mathbb{A}$ contains 1, it indicates the gene $i$ in location $j$ is present, otherwise $a_{i,j}$ contains 0. Every gene and location is described by its terms creating the semantic of rows and columns. Every bicluster has submatrix containing as many 1's while leaving as many 0's out as possible.

The Pareto set consists of the biclusters displayed by colors of grey shades. One of the possible aggregations of the Pareto set is displayed as the red rectangle. Let us note that the final bicluster is not a full rectangle of 1's (annotated genes) and areas that of 0's (locations) can be found.

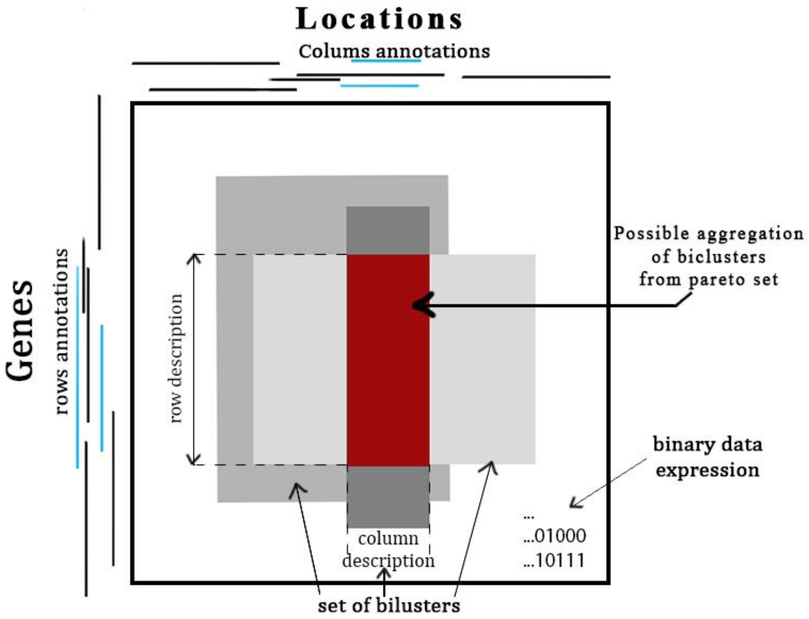

Figure 4.2: Model situation

## 4.7 Artificial dataset

For the artificial data, we use annotations of Drosophila melanogaster specifically measurements of the imaginal disc. Our input data are Pareto set generated from randomly generated expression matrix, where the gene expression matrix is covered on $(m \times n)/2$ of the area, where $m$ is a number of the rows and $n$ is a number of the columns. This equal distribution of 0's and 1's helps us simulate the real gene expression dataset. Rows of the bicluster contain genes' ID's and columns of the bicluster represent locations of genes. These biclusters contain more than 1200 genes and 70 locations.

## 4.8 Real datasets

### 4.8.1 Drosophila melanogaster

*Drosophila melanogaster* is popular dataset since it's one of the most examined and described organism. *Drosophila* has completely described set of genetic instructions. Another benefit of using *Drosophila* is its short lifespan, 8 - 14 days, so it is not consuming to observe the whole life cycle of *Drosophila* and gain all the information about *Drosophila* [fac, 2015].

Figure 4.3: Drosophila melanogaster, fruit fly[2]

### Gene expression matrix of imaginal disc

The first dataset is imaginal discs of a fruit fly. Each bicluster in a Pareto set has 1207 possible genes described by 5181 GO terms and 423 individual genes described by 114 KEGG terms. The biclusters have 72 locations that are described by 157 location terms.

### Gene expression matrix of ovaries

The second dataset is ovaries of a fruit fly. There are 6510 possible genes described by 8540 GO terms and 1605 genes described by 135 KEGG terms. Locations are described by 111 unique location terms.

### 4.8.2 Gene expression matrix of Mice

The last dataset was not tested by [Klema et al., 2017] and is exclusively tested by our implemented algorithm. This dataset comes from [Merkin et al., 2012] and contains 12225 genes and 26 possible locations. We refer to this dataset as to the m2801 since it is the code name of this dataset.

## 4.9 OBO::Parser

To generate the ontology descriptions for each bicluster the OBO::Parser [OBO, 2018] package in Perl language was used. The ontology of entire dataset needs to be generated as well.

We use OBO::Parser to create an ontology for both genes and locations. OBO::Parser needs 2 files to generate the ontologies. The first file is ontology stored in OBO format describing directed acyclic graph. The second file is init file. Each init file

[2] By André Karwath aka Aka [CC BY-SA 2.5 (https://creativecommons.org/licenses/by-sa/2.5)], from Wikimedia Commons, `https://commons.wikimedia.org/wiki/File:Drosophila_melanogaster_-_side_(aka).jpg`

consists of names of genes or locations and its corresponding terms, that are leaves directed of an acyclic graph.

The OBO::Parser collects all terms through the path from the leave to the root and saves them with the name of the gene to our chosen output file.

Our init files are stored in X-ontoDesc.RData files, where X is the number of the split. We provide a R script ontodesctotxt.R in */GenerateData/* that extracts the leave terms from RData format and saves them to the init files in text format to further use in OBO::Parser.

```
./generateOntologyFile.pl -f go-basic.obo -i init1.txt -o disccol1.txt

description: Get all ancesters defined in init file.
usage      : getOntologyFile.pl [options]
options    :
-f    OBO input file
-i   init file
-o   output file
example:
```

Figure 4.4: Help for OBO::Parser

```
FBgn0033019 GO:0006338 GO:0031011 GO:0006355
FBgn0263251 GO:0005575 GO:0008080 GO:0048477 GO:0022008 GO:0016573
FBgn0037224 GO:0008010 GO:0005578 GO:0040003
FBgn0038013 GO:0008150 GO:0003674 GO:0005575
FBgn0037358 GO:0005509 GO:0005886
FBgn0035252 GO:0005829 GO:0016021
```

Figure 4.5: Short example of init1.txt

# Chapter 5

# Implementation

Here we start with the implementation of our algorithm. There is a workflow of our algorithm depicted below in the Figure 5.1. The workflow is split into two parts. The first part is implemented by Ing. František Malinka, who randomly splits the gene expression matrix into training data and test data. Following which he runs NSGA-II to obtain Pareto set of solutions. From now on, the Pareto sets and test data are given to us to test the generalizing ability of Pareto set to unseen gene expression matrix entries. This is symbolized by the red line in Figure 5.1.
The right part of Figure 5.1 is our implementation. First, we use one of our methods of aggregating the Pareto set into a bicluster. Then we obtain the semantic annotation of the bicluster using gene enrichment analysis (G. E.). This semantic is later used to predict the unseen data entries in the gene expression dataset and evaluated from the point of view of semantics generalization ability using AUROC explained in Section 5.7.

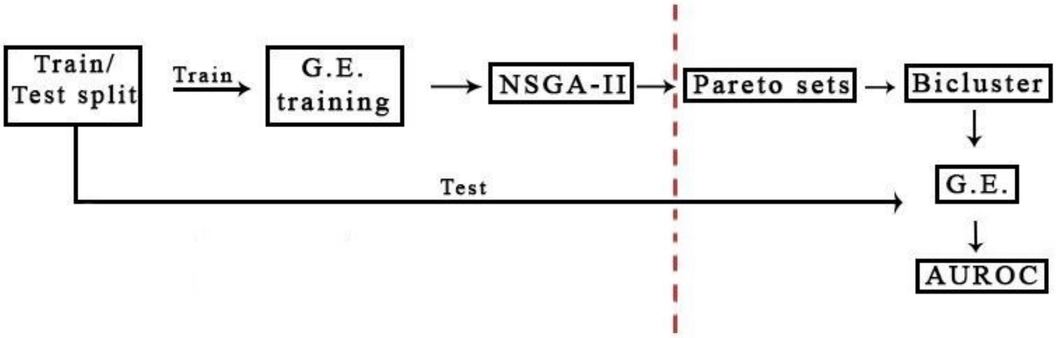

Figure 5.1: Algorithm workflow

## 5.1 Gene expression matrix split

At the very beginning, we split gene expression matrix into 10 training data and testing data splits. Our goal is to use our semantic description of aggregated biclusters from Pareto set to predict the unseen gene expression matrix data. Therefore we use the unseen data as test data. The training data are used to create Pareto sets in NSGA-II. The gene expression matrix is split into 70 % of training dataset and 30 % of test dataset. Later in Section 6.3 we compare our algorithm in 4 categories.

1. all - we generalize on the whole dataset,
2. both dimensions - we generalize in terms of locations and genes,
3. keep genes - we generalize only in terms of locations, genes are kept,
4. keep locations - we generalize only in terms of genes, locations are kept.

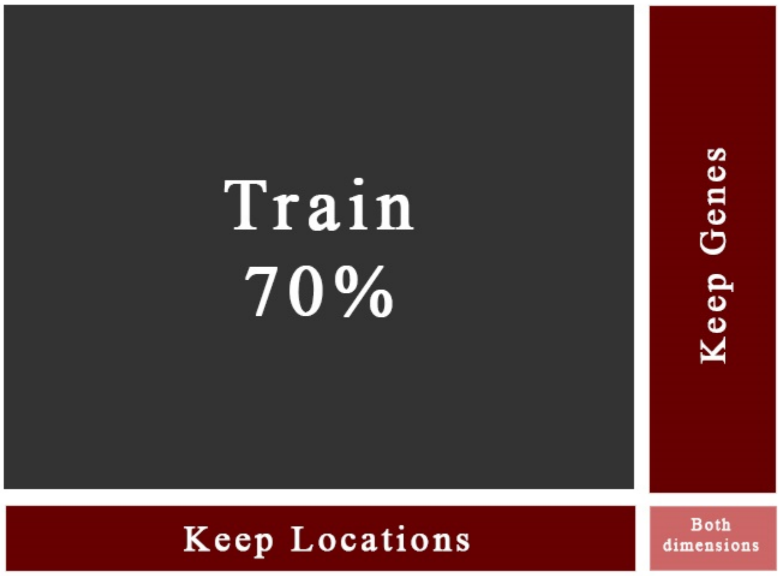

Figure 5.2: Train / test description

## 5.2 Approaches to Pareto set aggregation

We introduce three algorithms to obtain a Pareto set aggregation for further enrichment analysis.

The first approach finds the aggregation based solely on the relative frequency of each gene $i$ and each location $j$ over all biclusters in a Pareto set. This method is aptly called the *Empirical distribution-based algorithm*.

The second approach computes the mean of each gene $i$ and each location $j$ and keeps only values that exceed the mean of these values.

The third approach uses weighted random sampling that computes relative frequency of each element that is used as the weight of element and decides if the element is kept or not based on the given random value.

### 5.2.1 Empirical distribution-based algorithm I

The first method to aggregate a Pareto set is to discard all genes $i$ and locations $j$ which occur in less than $k\%$ of the Pareto set.

First, we obtain a vector of row sums of the set and another vector of column sums. These vectors are then normalized to unit sums so they can be interpreted as probability distributions. Finally, only genes $i$ and locations $j$ whose corresponding values are at least $k\%$ are kept. Later we refer to this method as to the "aggregation (k/k)", where $k \in (0, 1)$. For example, the aggregation that keeps locations and genes that exceeds at least 50 % is named as "(0.5/0.5)".

**Algorithm 1:** Empirical distribution

```
Input  : $paretoSet^{k \times m \times n}$ //set of biclusters in Pareto set
         , thresholdRows, thresholdCols
Output: paretoRows //aggregated bicluster row
         paretoColumns //aggregated bicluster column

/* initialize empty matrix counting hits                              */
paretoRows $\leftarrow$ 0
paretoColumns $\leftarrow$ 0
/* build biclusters and sum them into one matrix                      */
for $t \leftarrow 1$ to $k$ do
    /* Constructs bicluster using outer product                       */
    $paretoRows \leftarrow paretoRows + paretoSet[[t]][[1]]$
    $paretoColumns \leftarrow paretoColumns + paretoSet[[t]][[2]]$
end
/* normalize hits in vectors                                          */
$paretoRows \leftarrow paretoRows/k$
$paretoColumns \leftarrow paretoColumns/k$
/* delete elements that does not exceed threshold                     */
$paretoRows[paretoRows < thresholdRows] \leftarrow 0$
$paretoRows[paretoRows > 0] \leftarrow 1$
$paretoColumns[paretoColumns < thresholdCols] \leftarrow 0$
$paretoColumns[paretoColumns > 0] \leftarrow 1$
```

### 5.2.2 Empirical distribution-based algorithm II

The second approach is a modification of the first algorithm. However, we compute the mean from rows and columns and keep only values exceeding these numbers in each vector. We refer to this method as to the ”aggregation Mean”.

### 5.2.3 Weighted random sample

The third method employs weighted random sample. First, we obtain the relative frequency of rows and columns from Pareto set. Then we use the obtained values as weights to each of the element in a vector. We use a random number generator to determine whether we classify the element as 0 or 1 using the weighted vector. We refer to this method as to the ”aggregation WR”.

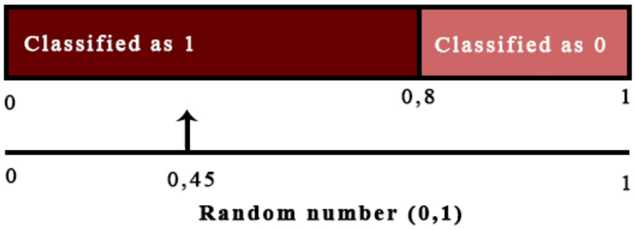

Figure 5.3: Weighted random sample example

**Algorithm 2:** Weighted random sample

```
Input  : $paretoSet^{k \times m \times n}$ //set of biclusters in Pareto set
Output: paretoRows //aggregated bicluster row
        paretoColumns //aggregated bicluster column
/* initialize empty matrix counting hits */
$paretoRows \leftarrow 0$
$paretoColumns \leftarrow 0$
for $t \leftarrow 1$ to $k$ do
    $paretoRows \leftarrow paretoRows + paretoSet[[t]][[1]]$
    $paretoColumns \leftarrow paretoColumns + paretoSet[[t]][[2]]$
end
$probCols \leftarrow paretoCols/k$
$probRows \leftarrow paretoRows/k$
for $x \leftarrow 1$ to $length(probRows)$ do
    $rand \leftarrow runif(n = 1, min = 0, max = 1)$
    if $rand < probRows[x]$ then
        $paretoRows[x] \leftarrow 1$
    else
        $paretoRows[x] \leftarrow 0$
    end
end
for $y \leftarrow 1$ to $length(probCols)$ do
    $rand \leftarrow runif(n = 1, min = 0, max = 1)$
    if $rand < probCols[y]$ then
        $paretoCols[y] \leftarrow 1$
    else
        $paretoCols[y] \leftarrow 0$
    end
end
```

## 5.3 Using the semantic annotation

As mentioned earlier in Section 4.6, nearly every gene and location is described by sets of gene and location terms. Genes are mostly described by GO terms and can be described by KEGG terms as well. Locations are described by location terms. Biclusters deal with sets of genes and locations. Each of these sets can be tested for the increased occurrence of a certain annotation term. We refer to this increase as the enrichment.

Gene set enrichment analysis [Subramanian et al., 2005] is used to reveal insight into genes with common biological function and is a core of our algorithm. To achieve the enrichment one-sided Fisher exact test is used since we need only term counts in a sample and no other information in order to run the enrichment.

To compute the enrichment we use R library function fisher.test() that uses phyper() method.

We compute the confusion matrix for each term and call the Fisher exact test method. Terms are considered to be significant if their *p-value* is smaller than 0.05 if not specified otherwise.

The confusion matrix for our problem looks as follows:

Table 5.1: Confussion matrix

| | |
|---|---|
| Term do not belong in gene / location outside the bicluster | Term belongs in gene / location outside the bicluster |
| Term do not belong in gene / location in the bicluster | Term belongs in gene / location in the bicluster |

myTerm is the selected term for which we want to know if it is enriched. SampleNames are all genes or locations from selected bicluster. Ontology is all genes or location described by its terms. Variable *Objects* describes either genes or locations.

**Algorithm 3:** GO term Enrichment

```
input  : myTerm, sampleNames, ontology
output: name term with its p-value
parameter: A parameter for the algorithm

Prepare confusion matrix for each GO Term
/* get all objects in the bicluster                                  */
$objectInBicluster \leftarrow ontology[names(ontology)\ \%in\%\ sampleNames]$
/* get all objects outside the bicluster                             */
$namesOutsideBicluster \leftarrow$
  $ontology[!names(ontology)\%in\%sampleNames]$
/* find which objects are described by term in the bicluster         */
$inBic \leftarrow sapply(objectInBicluster, function(x)myTerm\%in\%x)$
/* find which objects are described by term outside the
   bicluster                                                         */
$outBic \leftarrow sapply(namesOutsideBicluster, function(x)myTerm\%in\%x)$
/* count occurrences                                                 */
$taggedInBicluster \leftarrow length(which(inBic\%in\%TRUE))$
$taggedOutsideBicluster \leftarrow length(which(outBic\%in\%TRUE))$
$notTaggedInBicluster \leftarrow length(which(inBic\%in\%FALSE))$
$notTaggedOutsideBicluster \leftarrow length(which(outBic\%in\%FALSE))$
/* build confusion matrix                                            */
$confusionMatrix \leftarrow$
  $matrix(c(notTaggedOutsideBicluster, notTaggedInBicluster,$
  $TaggedOutsideBicluster, taggedInBicluster), 2, 2)$
/* compute p-value                                                   */
$pValueFisher \leftarrow fisher.test(confusionMatrix, alternative =' $
  $greater')\$p.value$
```

## 5.4 Annotation of genes

First, we obtain enriched GO terms for aggregated bicluster. We withdraw all genes that are classified as 1's and extract GO terms corresponding to each gene. We run enrichment analysis and obtain *p-values* for each GO term using Fisher exact test. We keep GO terms with *p-values* smaller than 0.05 if not set otherwise. The same process is applied to obtain enriched KEGG terms. These GO and KEGG terms and its *p-values* are saved.

### 5.4.1 Dividing ontology by molecular function

Since one of our goals was to substitute TopGO [Alexa and Rahnenfuhrer, 2010] dependancy we imitated the function of TopGO that splits environment into three ontologies: BP, CC, and MF as mentioned in Section 4.2. Therefore we ran enrich-

ment analysis for those three ontologies.
At first, we separated provided GO terms to ontology categories and ran three enrichment analysis using Fisher exact test. Unfortunately, we still needed TopGO [Alexa and Rahnenfuhrer, 2010] package to distinguish where each GO term belonged. However, we conducted a test that enriches not three split ontologies but uses one united gene to go terms description. We concluded that there is no difference between running enrichment analysis for three separated ontologies or one united when searching for the enriched terms. Nevertheless running one enrichment analysis over three enrichment analysis saves computing time significantly. Thus we eliminate the need for TopGO package completely. The experiment is available in $/experiments/DividingOntology$ folder.

### 5.4.2 KEGG superiority

Since KEGG ontology database covers usually fewer genes than GO ontology, we rely on GO ontology. However, KEGG pathways are derived from biological books and experiments by biological curators in contrast with GO terms mostly acquired by computers. We tried to give a higher score for genes annotated by KEGG terms to see if KEGG ontology can provide us more accurate results, but it did not help us significantly $/experiments/keggSuperiority$.

## 5.5 Annotation of locations

A similar approach is used for locations enrichment. Every location is described by location terms. As we stated earlier, for each bicluster, we single out all locations that were classified as 1's and afterward we run enrichment analysis for each location term and obtain *p-values* that we threshold by 0.1 if not set otherwise. These location terms and its *p-values* are saved.

## 5.6 Applying the semantic

In the previous steps, we obtained a semantic description of each bicluster. Now, we will use this semantic description for classification of unseen data entries in the binary expression dataset. As mentioned in Section 5.1 we test the generalization on whole test dataset, on genes only, on locations only and on both dimensions.

### 5.6.1 Score distribution

We use the enriched terms to give score to all genes and locations containing these terms. This is the step, where we classify even entries we have not seen since they can contain the enriched term. The score is computed as $-log_{10}(pvalue)$ same as [Klema et al., 2017], where a *p-value* is corresponding value of the term. After the score is distributed, we scale the score to the interval (0,1). We save these evaluated genes and locations for further cut-off by thresholds. Now we have identified genes and locations that are affected by enriched terms in gene expression matrix and obtained their score.

## 5.7 The area under the Receiver Operating Characteristic curve

AUROC [Hanley and McNeil, 1982] stands for the area under the Receiver Operating Characteristic curve. We use the AUROC as our quality of classifier. After each fold, we predict unseen entries of gene expression dataset and compute AUROC for each test bicluster. Since we have two hyperparameters to optimize the result by cutting off the scores of genes and locations, the true positive rate and the false positive rate is employed to demonstrate the area under the Receiver Operating Characteristic curve. After certain cut-off, we classify every (gene; location) pair in binary matrix as 1. When predicting only genes or only locations, we use the values from aggregated biclusters to substitute the missing part of data and thus we predict in only one dimension.
AUROC is computed as:

$AUROC = \frac{FPr \cdot TPr}{2} + TPr \cdot (1 - FPr) + \frac{1-TPr}{2}$

where variables are:

True positive (TP) - gene $i$ in situation $j$ was originally 1 and correctly classified as 1

False negative (FN) - gene $i$ in situation $j$ was originally 1 but was misclassified as 0

True negative (TN) - gene $i$ in situation $j$ was originally 0 and correctly classified as 0

False positive (FP) - gene $i$ in situation $j$ was originally 0 but was misclassified as 1

True positive rate, $TPr = \frac{TP}{TP+FN}$

False positive rate, $FPr = \frac{FP}{FP+TN}$

# Chapter 6

# Evaluation

The main goal of this chapter is to verify whether the early consideration of both the key aspects of biclusters (homogeneity and semantic coherence) brings their increased predictive strength. We employ the classification accuracy as the objective of bicluster quality. We compare our method with two real experimental datasets used in [Klema et al., 2017] and add one brand new dataset. Also, a test on artificial data is made.

We will refer to further *genes p-value* as to the "G" and to the *locations p-value* as to the "L". These are the *p-values* we use as the cut-off during enrichment analysis in Section 5.3.

The names of our aggregation methods are shortened in order to fit in the table. The abbreviations for our three aggregation methods are:

- Weighted random sample - WR,
- Empirical distribution-based algorithm I - (X, Y), where X, Y are threshold values
- Empirical distribution-based algorithm II - Mean.

## 6.1 Dataset parameters

Here we present number of unique terms in Table 6.1 and number of genes and locations in Table 6.2 for each dataset.

Table 6.1: Number of unique terms

| | GO Terms | KEGG Terms | Location Terms |
|---|---|---|---|
| Imaginal disc | 5181 | 114 | 157 |
| Ovary | 8540 | 135 | 111 |
| M2801 | 19852 | - | 39 |

Table 6.2: Number of genes and locations described by terms

| | Genes | Locations |
|---|---|---|
| Imaginal disc | 1207 | 72 |
| Ovary | 6510 | 100 |
| M2801 | 12225 | 26 |

## 6.2 Tuning the thresholds for the gene and location scores

When searching for optimal threshold settings we implemented an algorithm that tried every combination of thresholds on the interval (0, 0.3). The best thresholds are chosen for the highest value of the empirical mean of AUROC for generalization on full dataset.

For most of our aggregations, thresholding the scores that did not overcome 30 % of score maximum performs the best while not decreasing the size of predicted values. We believe this value is suitable since we employ a large number of GO terms to predict the gene expression matrix and not all the genes score is high enough to classify the gene.

Here we present thresholds for each aggregation methods displayed as a pair (genes threshold; locations threshold).

Table 6.3: Thresholds for gene and location scores (G: 0.05, L: 0.1)

| | WR | Mean | 0.5/0.5 | 0.7/0.7 | 0.75/0.75 | 0.9/0.9 |
|---|---|---|---|---|---|---|
| Imag. disc | (0.26; 0) | (0.2; 0.21) | (0.13; 0.29) | (0.14; 0.28) | (0.3; 0.25) | (0.21; 0) |
| Imag. disc2 | (0.19; 0.23) | (0.27; 0) | (0; 0.24) | (0.28; 0) | (0.18; 0) | (0.16; 0) |
| Ovary | (0.29; 0) | (0.3; 0) | (0.3; 0) | (0.3; 0) | (0.3; 0) | (0.3; 0) |
| Artif. disc | (0; 0.07) | (0; 0.17) | (0; 0.06) | (0; 0.15) | (0; 0.14) | (0; 0.16) |

Table 6.4: Thresholds for gene and location scores (G: 0.05, L: 0.15)

| | WR | Mean | 0.5/0.5 | 0.7/0.7 | 0.75/0.75 | 0.9/0.9 |
|---|---|---|---|---|---|---|
| Imag. disc | (0.19; 0.1) | (0.15; 0.27) | (0.19; 0.1) | (0.14; 0.26) | (0.21; 0.29) | (0.21; 0) |
| Imag. disc2 | (0.15; 0) | (0.27; 0) | (0; 0.19) | (0.27; 0) | (0.18; 0.06) | (0.27; 0) |
| Ovary | (0.26; 0) | (0.3; 0) | (0.3; 0) | (0.3; 0) | (0.29; 0) | (0.3; 0) |
| Artif. disc | (0; 0.18) | (0; 0.16) | (0; 0.15) | (0; 0.16) | (0; 0.08) | (0; 0.12) |

## 6.3 Results

Unfortunately we were not able to reproduce the same results as stated for Bicluster enrichment in [Klema et al., 2017]. These are the highest values obtained by us for imaginal discs (genes threshold = 1; locations threshold = 50) in Table 6.7 and ovaries (genes threshold = 1; locations threshold = 1) in Table 6.12 when using the recommended hyperparameters in [Klema et al., 2017]. We present graph for each dataset to compare our best achieved AUROC and [Klema et al., 2017]'s, except m2801 that was not tested by [Klema et al., 2017].

Our training/test splits were dependent on a limited number of given Pareto sets. In case of our implementation, each split has 2 Pareto sets thus 2 biclusters aggregated from training data. Each of this Pareto sets consists of 200 overlapping biclusters. We compare the algorithms in 4 categories as stated in Section 5.1. We generalize on entire gene expression dataset, in terms of locations, in terms of genes and on both dimensions simultaneously.

### 6.3.1 Artificial dataset

Here we provide test on artificial data. The problem with the test on artificial data is that we cannot control the semantics of the data since the gene expression matrix was randomly covered with 1's as mentioned in Section 4.7. We can see that keeping only genes and locations that occur in at least 50 % generalize the best in both *location p-values.*

Table 6.5: Artificial Discs dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.1)

| | AUROC | keep genes | keep locations | both dimensions |
|---|---|---|---|---|
| WR | 0.772 ± 0.068 | 0.693 ± 0.048 | 0.789 ± 0.035 | 0.795 ± 0.082 |
| Mean | 0.790 ± 0.032 | 0.666 ± 0.035 | 0.819 ± 0.038 | 0.807 ± 0.072 |
| 0.5/0.5 | **0.791 ± 0.032** | 0.658 ± 0.035 | 0.821 ± 0.040 | 0.807 ± 0.072 |
| 0.7/0.7 | 0.769 ± 0.087 | 0.598 ± 0.028 | 0.844 ± 0.030 | 0.785 ± 0.105 |
| 0.75/0.75 | 0.769 ± 0.087 | 0.596 ± 0.031 | 0.843 ± 0.031 | 0.785 ± 0.105 |
| 0.9/0.9 | 0.763 ± 0.111 | 0.557 ± 0.032 | 0.768 ± 0.099 | 0.777 ± 0.158 |

Table 6.6: Artificial Discs dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.15)

| | AUROC | keep genes | keep locations | both dimensions |
|---|---|---|---|---|
| WR | 0.755 ± 0.106 | 0.653 ± 0.088 | 0.800 ± 0.035 | 0.771 ± 0.143 |
| Mean | **0.791 ± 0.032** | 0.668 ± 0.037 | 0.819 ± 0.038 | 0.807 ± 0.072 |
| 0.5/0.5 | **0.791 ± 0.032** | 0.656 ± 0.036 | 0.821 ± 0.040 | 0.807 ± 0.072 |
| 0.7/0.7 | 0.773 ± 0.089 | 0.602 ± 0.028 | 0.844 ± 0.030 | 0.794 ± 0.109 |
| 0.75/0.75 | 0.769 ± 0.087 | 0.595 ± 0.030 | 0.843 ± 0.031 | 0.785 ± 0.105 |
| 0.9/0.9 | 0.758 ± 0.111 | 0.561 ± 0.027 | 0.793 ± 0.085 | 0.769 ± 0.159 |

### 6.3.2 Imaginal disc dataset

Here in Table 6.7 we recomputed the results for Bicluster Enrichment, since it is the algorithm we try to overcome. We obtained similar results for Rules (JRip) and Tree (J48) thus we used the reference values from [Klema et al., 2017]. The resulting graph shows Bicluster Enrichment method in compare with our best approach. In case of this dataset we have two training / test splits unlike the others datasets. The first training / test split is evaluated in Table 6.8 and Table 6.9. The second training / test split is evaluated in Table 6.10 and Table 6.11.

Table 6.7: Imaginal Discs dataset results for current semantic biclustering algorithms (G: 0.05, L: 0.1)

| | AUROC | keep genes | keep locations | both dimensions |
|---|---|---|---|---|
| Bic. Enrich. | 0.548 ± 0.042 | 0.541 ± 0.038 | 0.559 ± 0.057 | 0.563 ± 0.020 |
| Rules (JRip) | 0.565 ± 0.010 | 0.588 ± 0.010 | 0.546 ± 0.010 | 0.537 ± 0.020 |
| Tree (J48) | 0.627 ± 0.050 | 0.630 ± 0.060 | 0.627 ± 0.050 | 0.602 ± 0.040 |

#### First split

In Table 6.8 we provide the result for settings, where *p-value* for genes is 0.05 and *p-value* for locations is 0.1. Since the sizes of our biclusters from Pareto sets employ larger size, the enrichment analysis performs better if *p-value* for locations is 0.15 instead of 0.1. An aggregation method where we keep genes and locations that occur in more than 50 % performs the best with additional setting of thresholds demonstrated in Table 6.3.

Table 6.8: Imaginal Discs dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.1)

| | AUROC | keep genes | keep locations | both dimensions |
|---|---|---|---|---|
| WR | 0.530 ± 0.031 | 0.513 ± 0.04 | 0.557 ± 0.026 | 0.533 ± 0.039 |
| Mean | 0.546 ± 0.030 | 0.534 ± 0.03 | 0.559 ± 0.035 | 0.547 ± 0.033 |
| 0.5/0.5 | **0.549 ± 0.036** | 0.523 ± 0.022 | 0.561 ± 0.035 | 0.545 ± 0.046 |
| 0.7/0.7 | 0.538 ± 0.026 | 0.520 ± 0.026 | 0.550 ± 0.031 | 0.540 ± 0.030 |
| 0.75/0.75 | 0.531 ± 0.038 | 0.514 ± 0.024 | 0.547 ± 0.028 | 0.537 ± 0.043 |
| 0.9/0.9 | 0.538 ± 0.028 | 0.511 ± 0.012 | 0.546 ± 0.033 | 0.534 ± 0.032 |

In Table 6.9 we provide the result for settings, where *p-value* for genes is 0.05 and *p-value* for locations is 0.15. Here an aggregation method where the weighted random sample is used performs the best, however the result of this method is not consistent thus we select aggregation method 0.7/0.7 that is slightly worse, but replicable.

Table 6.9: Imaginal Discs dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.15)

| | AUROC | keep genes | keep locations | both dimensions |
|---|---|---|---|---|
| WR | **0.557 ± 0.037** | 0.530 ± 0.027 | 0.579 ± 0.015 | 0.559 ± 0.040 |
| Mean | 0.550 ± 0.033 | 0.535 ± 0.025 | 0.569 ± 0.029 | 0.549 ± 0.032 |
| 0.5/0.5 | 0.552 ± 0.033 | 0.522 ± 0.020 | 0.561 ± 0.034 | 0.553 ± 0.032 |
| 0.7/0.7 | **0.556 ± 0.041** | 0.528 ± 0.027 | 0.572 ± 0.022 | 0.557 ± 0.043 |
| 0.75/0.75 | 0.539 ± 0.026 | 0.520 ± 0.025 | 0.565 ± 0.023 | 0.541 ± 0.035 |
| 0.9/0.9 | 0.549 ± 0.023 | 0.514 ± 0.012 | 0.558 ± 0.026 | 0.543 ± 0.031 |

### Second split

Since we were provided a second training / test split, we can observe that it performs better and overcome both **Biclustering enrichment** and **Rules (JRip)**) algorithms. Again, the higher *p-value* for locations helps us to get better result. In both tables, An aggregation method where we keep genes and locations that occur in more than 50 % performs the best.

Table 6.10: Imaginal Discs dataset 2 results for our modified semantic biclustering algorithm (G: 0.05, L: 0.1)

| | AUROC | keep genes | keep locations | both dimensions |
|---|---|---|---|---|
| WR | 0.546 ± 0.042 | 0.533 ± 0.038 | 0.569 ± 0.018 | 0.551 ± 0.058 |
| Mean | 0.557 ± 0.014 | 0.531 ± 0.018 | 0.563 ± 0.033 | 0.560 ± 0.021 |
| 0.5/0.5 | **0.599 ± 0.054** | 0.549 ± 0.033 | 0.552 ± 0.036 | 0.603 ± 0.061 |
| 0.7/0.7 | 0.547 ± 0.021 | 0.522 ± 0.016 | 0.563 ± 0.033 | 0.551 ± 0.027 |
| 0.75/0.75 | 0.559 ± 0.034 | 0.524 ± 0.019 | 0.563 ± 0.031 | 0.563 ± 0.044 |
| 0.9/0.9 | 0.558 ± 0.022 | 0.519 ± 0.011 | 0.562 ± 0.032 | 0.559 ± 0.031 |

Since the obtained biclusters employ a large part the of bicluster, the increase of *p-value* and therefore more obtained terms for locations help us gain better results in Table 6.11.

Table 6.11: Imaginal Discs dataset 2 results for our modified semantic biclustering algorithm (G: 0.05, L: 0.15)

| | AUROC | keep genes | keep locations | both dimensions |
|---|---|---|---|---|
| WR | 0.558 ± 0.053 | 0.538 ± 0.044 | 0.575 ± 0.023 | 0.567 ± 0.059 |
| Mean | 0.564 ± 0.018 | 0.537 ± 0.020 | 0.568 ± 0.030 | 0.564 ± 0.030 |
| 0.5/0.5 | **0.601 ± 0.053** | 0.550 ± 0.027 | 0.554 ± 0.033 | 0.601 ± 0.061 |
| 0.7/0.7 | 0.556 ± 0.016 | 0.528 ± 0.015 | 0.571 ± 0.032 | 0.559 ± 0.028 |
| 0.75/0.75 | 0.565 ± 0.030 | 0.528 ± 0.019 | 0.572 ± 0.033 | 0.567 ± 0.039 |
| 0.9/0.9 | 0.564 ± 0.018 | 0.537 ± 0.020 | 0.568 ± 0.030 | 0.564 ± 0.030 |

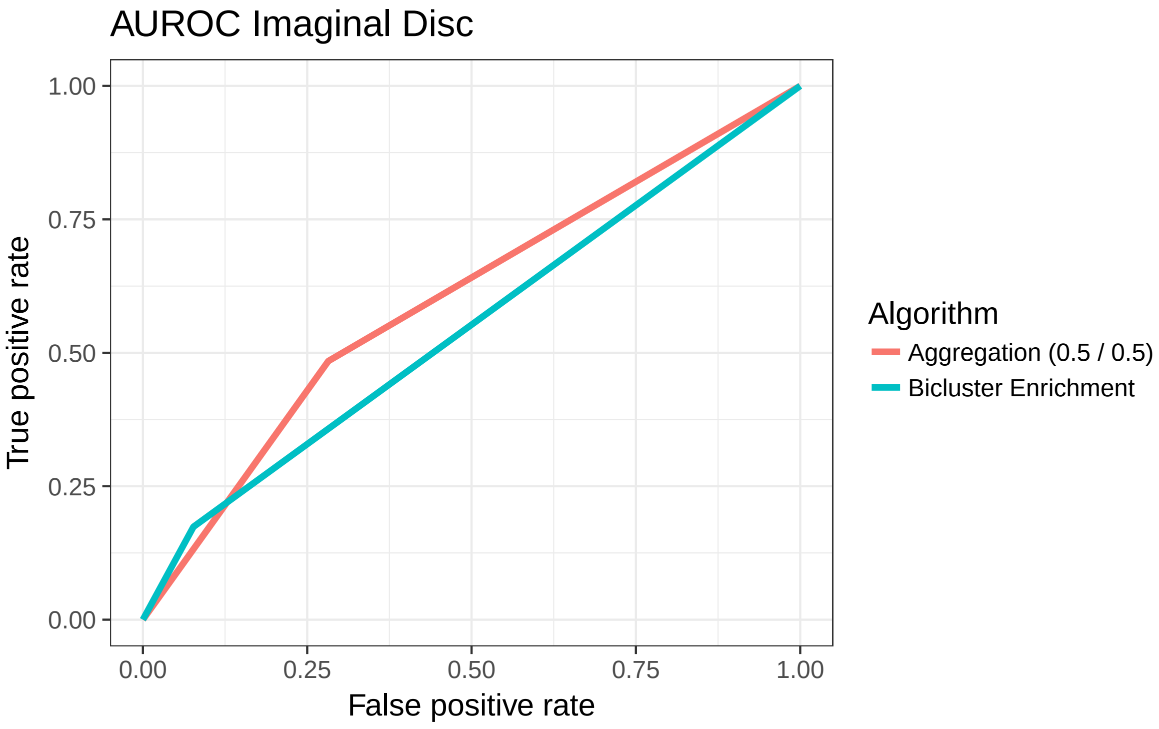

Figure 6.1: AUROC of imaginal discs

### 6.3.3 Ovary dataset

In case of this dataset it is even more convenient to use the *p-value* 0.15 for locations since there are only 111 unique location terms to describe 100 locations so the locations are strongly correlated. Since our biclusters employ large size of the matrix, it is harder to obtain enriched values. Here we present two tables comparing results for *p-value* 0.1 for locations in Table 6.13 and *p-value* 0.15 for locations in Table 6.14. The resulting graph shows Bicluster Enrichment method in comparison with our best approach.

Table 6.12: Ovary dataset results for current semantic biclustering algorithms (G: 0.05, L: 0.1)

| | AUROC | keep genes | keep locations | both dimensions |
|---|---|---|---|---|
| Bic. Enrich. | 0.536 ± 0.011 | 0.606 ± 0.037 | 0.527 ± 0.014 | 0.537 ± 0.020 |
| Rules (JRip) | 0.636 ± 0.010 | 0.588 ± 0.010 | 0.546 ± 0.010 | 0.537 ± 0.020 |
| Tree (J48) | 0.659 ± 0.010 | 0.630 ± 0.060 | 0.627± 0.050 | 0.602 ± 0.040 |

Since the strong correlation of locations and its terms and the fact that the aggregated biclusters covers large part of gene expression set, There are not many enriched locations under the location threshold 0.1 in Table 6.8. We can see that we generalize worse if we keep genes and generalize in terms of locations than if we keep locations and generalize in terms of genes. Thus we try to lift the *p-value* for locations to obtain more terms.

Table 6.13: Ovary dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.1)

| | AUROC | keep genes | keep locations | both dimensions |
|---|---|---|---|---|
| WR | 0.524 ± 0.027 | 0.512 ± 0.016 | 0.528 ± 0.026 | 0.523 ± 0.027 |
| Mean | 0.518 ± 0.030 | 0.508 ± 0.020 | 0.527 ± 0.018 | 0.517 ± 0.029 |
| 0.5/0.5 | 0.526 ± 0.028 | 0.508 ± 0.021 | 0.538 ± 0.023 | 0.523 ± 0.028 |
| 0.7/0.7 | 0.525 ± 0.027 | 0.505 ± 0.014 | 0.521 ± 0.017 | 0.524 ± 0.029 |
| 0.75/0.75 | **0.531 ± 0.031** | 0.503 ± 0.014 | 0.520 ± 0.021 | 0.527 ± 0.036 |
| 0.9/0.9 | **0.531 ± 0.031** | 0.505 ± 0.011 | 0.523 ± 0.022 | 0.533 ± 0.034 |

Here an aggregation 0.75 / 0.75 performs the best keeping genes and locations that occur in at least 75 % of biclusters.

Table 6.14: Ovary dataset results for our modified semantic biclustering algorithm (G: 0.05, L: 0.15)

| | AUROC | keep genes | keep locations | both dimensions |
|---|---|---|---|---|
| WR | 0.544 ± 0.027 | 0.524 ± 0.019 | 0.546 ± 0.023 | 0.543 ± 0.032 |
| Mean | 0.524 ± 0.027 | 0.508 ± 0.020 | 0.532 ± 0.020 | 0.524 ± 0.026 |
| 0.5/0.5 | 0.536 ± 0.028 | 0.515 ± 0.020 | 0.547 ± 0.022 | 0.533 ± 0.031 |
| 0.7/0.7 | 0.543 ± 0.028 | 0.510 ± 0.013 | 0.527 ± 0.023 | 0.544 ± 0.029 |
| 0.75/0.75 | **0.550 ± 0.020** | 0.508 ± 0.010 | 0.525 ± 0.021 | 0.550 ± 0.019 |
| 0.9/0.9 | 0.546 ± 0.029 | 0.508 ± 0.012 | 0.531 ± 0.022 | 0.547 ± 0.034 |

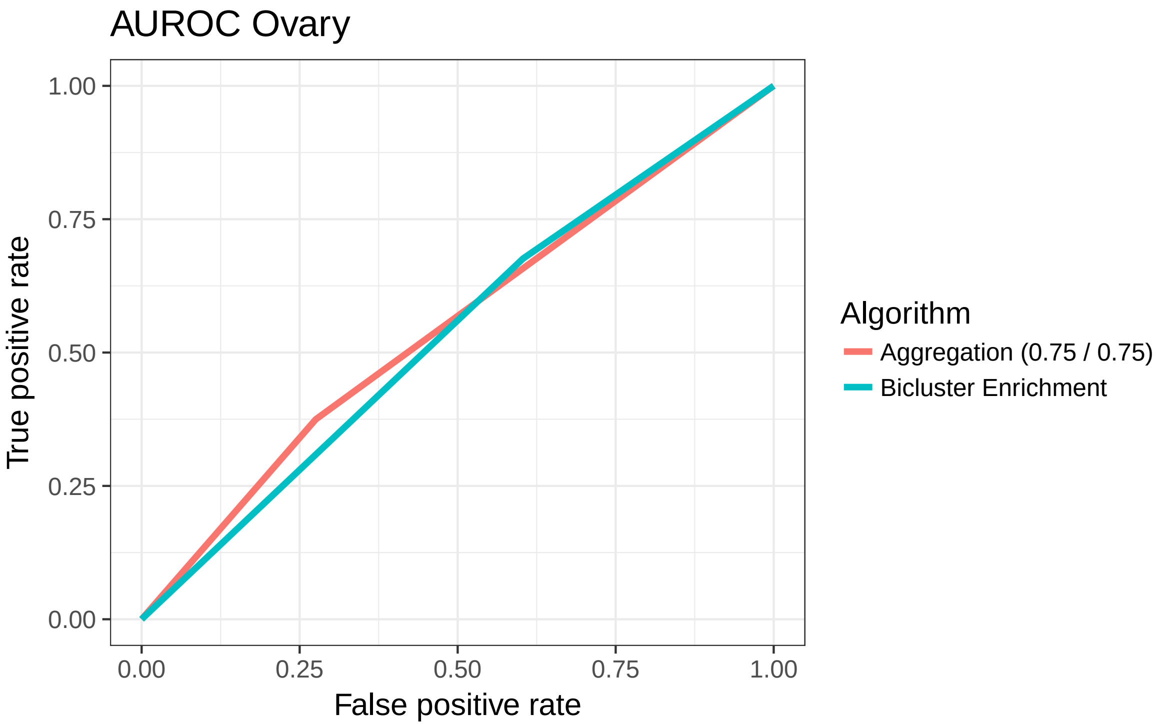

Figure 6.2: AUROC of Ovary

### 6.3.4 M2801 dataset

M2801 dataset contains only 26 locations described by 39 terms. Thats why we truly need to lift the cut-off *p-value* to get location enriched terms. we tried several set-ups for different aggregation methods. Here we keep *p-value* for genes to 0.05 and we change *p-value* for locations.

Here in Table 6.15 and Table 6.16 we will store the location *p-value*, the type of method and possible threshold in following vector:

[locations p-value; number of method; threshold if method 3 is selected]

where number of method refer to:

- 1 - Aggregation method weighted random sample,
- 2 - Aggregation method empirical distribution-based algorithm II,
- 3 - Aggregation method empirical distribution-based algorithm I, with chosen threshold (X; Y).

Table 6.15: M2801 dataset thresholds for different methods

| | Thresholds |
|---|---|
| [0.1; 3; 0.9] | [0.15; 0] |
| [0.15; 3; 0.9] | [0.15; 0] |
| [0.2; 3; 0.9] | [0.15; 0.12] |
| [0.3; 3; 0.9] | [0.15; 0.14] |
| [0.3; 3; 0.7] | [0.15; 0.14] |
| [0.3; 2] | [0.15; 0.14] |
| [0.4; 1] | [0.15; 0.15] |
| [0.4; 2] | [0.15; 0] |
| [0.4; 3; 0.5] | [0.15; 0.15] |
| [0.4; 3; 0.7] | [0.15; 0] |
| [0.4; 3; 0.9] | [0.15; 0] |

Here in Table 6.16 we present results for different cut-off *p-values* for location terms and aggregation methods. We can observe that we cannot obtain reasonable score, if we keep *p-value* for location terms low and therefore gain a little number or no enriched terms. The best result is obtained if we keep location terms smaller than *p-value* 0.4.

Table 6.16: M2801 dataset results for our modified semantic biclustering algorithm

| | AUROC | keep genes | keep locations | both dimensions |
|---|---|---|---|---|
| [0.1; 3; 0.9] | 0.497 ± 0.015 | 0.498 ± 0.006 | 0.500 ± 0.001 | 0.490 ± 0.030 |
| [0.15; 3; 0.9] | 0.510 ± 0.030 | 0.504 ± 0.015 | 0.487 ± 0.026 | 0.514 ± 0.064 |
| [0.2; 3; 0.9] | 0.513 ± 0.025 | 0.503 ± 0.013 | 0.515 ± 0.015 | 0.517 ± 0.049 |
| [0.3; 3; 0.9] | 0.520 ± 0.019 | 0.501 ± 0.009 | 0.494 ± 0.018 | 0.484 ± 0.039 |
| [0.3; 3; 0.7] | 0.529 ± 0.013 | 0.503 ± 0.012 | 0.495 ± 0.017 | 0.494 ± 0.035 |
| [0.3; 2] | 0.530 ± 0.013 | 0.505 ± 0.019 | 0.496 ± 0.017 | 0.493 ± 0.033 |
| [0.4; 1] | 0.524 ± 0.018 | 0.495 ± 0.030 | 0.487 ± 0.012 | 0.487 ± 0.042 |
| [0.4; 2] | 0.538 ± 0.027 | 0.512 ± 0.028 | 0.492 ± 0.006 | 0.513 ± 0.054 |
| [0.4; 3; 0.5] | 0.515 ± 0.029 | 0.485 ± 0.040 | 0.490 ± 0.005 | 0.470 ± 0.056 |
| [0.4; 3; 0.7] | **0.538 ± 0.027** | 0.512 ± 0.028 | 0.492 ± 0.006 | 0.513 ± 0.054 |
| [0.4; 3; 0.9] | 0.536 ± 0.022 | 0.506 ± 0.009 | 0.491 ± 0.018 | 0.506 ± 0.041 |

### 6.3.5 Runtimes

Here we present runtimes. The algorithms were run on set-up with Intel i5-3470 and 8 GB RAM.

Table 6.17: Runtimes of searching for optimal threshold

| Dataset | Time |
|---|---|
| IDisc | 2574 s |
| Ovary | 4197.2 s |
| M2801 | 18748.5 s |

Table 6.18: Runtimes of algorithm

| Dataset | Time |
|---|---|
| IDisc | 517 s |
| Ovary | 3612 s |
| M2801 | 5025 s |

### 6.3.6 Discussion

In Table 6.7 and Table 6.12 we can see, that JRip and J48 perform better than Semantic biclustering as stated in [Klema et al., 2017], however, the authors stated that JRip and J48 tend to overfit, whereas semantic biclustering does not. We demonstrate that our three proposed aggregation methods can slightly overcome the results of Biclustering enrichment algorithm in ovary dataset approximately by 2 % and imaginal discs by 5 % . Nevertheless there is still space for improvement. We satisfy our expectations that the key factors of biclusters, homogeneity and semantic coherence help us achieve moderately better results. The disadvantage of our measurement is the fact, that we were limited by having only one training/test split for each dataset (except imaginal discs, where second training/test split helped us to overcome both **Biclustering enrichment** and **Rules (JRip)**) that we could use since we were not provided more dataset splits. Thus there is possibility of better results. Also a larger number of biclusters in Pareto sets might help us to get better accuracy. [Klema et al., 2017] benefits from the fact that their implementation is able to run the algorithm multiple times with different training/test split every time.

We can see that our methods did well if the threshold for genes was employed. This is caused by the fact that we use a large number of GO terms, but not every GO term has a high score. If we use the right *p-value* for location terms, we do not need necessarily to threshold them as it is with genes score. We can see, that the key aspects of biclusters from NSGA-II (homogeneity and semantic coherence) brings increased predictive strength.

# Chapter 7

# Framework

Our framework was implemented in R language since it is the most used programming language in bioinformatics and in statistical computing. R is open-source license language available for free.

To run our algorithm, it is required to install:

- R-3.5.0, we also recommend RStudio IDE
- R's tictoc package
- Perl
- OBO::Parser

All the scripts are the same for each dataset. However the only script that differ is main.R script. Main.R loads corresponding data for each dataset. We provide two modifications of main.R script. The first one solely runs the algorithm based on specified genes and locations thresholds. The second modification of the script has these thresholds set to 0 and searches for optimal thresholds by itself on given interval.

After we manually generate the ontology files for each training/test split and ontology for entire dataset we can run the main.R that is responsible for loading all the other scripts, data and running the workflow.R.

Here we need to specify:

- cut-off *p-value* for locations,
- cut-off *p-value* for genes,
- Genes threshold,
- Locations threshold,
- aggregation method and threshold, if method 2 is selected,

- Path to root file.

The rest of the data is loaded automatically, if root folder is specified in main.R and data are placed in right folders. The file distribution in folders is displayed below in Figure 7.1.

```
root folder
 |__main.R
 |__workflow.R
 |__results
 |__logs
 |__scripts
 |__data
     |__Gene expression matrix
     |__Dataset GO ontology
     |__Dataset Location ontology
     |__KEGG ontology (optional)
     |__datasets
     |   |__TRAIN / TEST DATA
     |__pareto
     |   |__Pareto sets
     |__generatedontologies
         |__TRAIN / TEST data ontology
```

Figure 7.1: File distribution in folders

Object with biclusters details is saved into */results/* folder, the file is named by time it was saved in milliseconds in RDS format. The logs with the most important values for each split are saved in */logs/* folder.

# Chapter 8

# Conclusion

In this thesis, we presented a modified current semantic biclustering algorithm that uses Pareto sets obtained by multi-criteria optimization algorithm NSGA-II that focuses on all aspects of good quality bicluster (size, accuracy, biological interpretability) to predict unseen gene expression data. With these Pareto sets we employ three aggregation methods, weighted random sample and two empirical distribution-based algorithms to define one bicluster for further enrichment analysis. We compared our proposed modification of semantic biclustering with the current version of semantic biclustering algorithm [Klema et al., 2017] and their generalization ability to describe the unseen gene expression data.

First, we define one bicluster from given Pareto set using one of the aggregation methods. Then we employ the semantics of data and use enrichment analysis to obtain key terms of bicluster. With this semantic description, we compare its generalization ability to other semantic biclustering algorithms.

Our result for imaginal disc dataset overcome the results of Biclustering enrichment by 5% and Rules (JRip) by 3.5 %. The result for the ovary dataset is almost identical, but we believe we can obtain solid result for ovary dataset as well, if more training / test splits for each dataset are provided. Also, a larger number of biclusters in Pareto set could help us achieve better results.

We can see that our methods of aggregating biclusters from Pareto sets are suitable for the description of unseen data entries in gene expression set. We satisfy our hypothesis that biclusters found using prior knowledge has better generalization ability to describe unseen data both in case of imaginal disc dataset and ovary dataset than biclusters found solely as binary biclusters.

However there is still a space for improvement. The framework is prepared for further testing of another aggregation methods of Pareto sets. Another contribution of our thesis is testing the semantic biclustering on the new dataset of mice tissue. We also present that our implementation offers commented, explained and simplified code.

In future work, a substitution for enrichment analysis could be implemented and compared with [Klema et al., 2017]. Here a beam search or using the ontology description logic could be implemented. Also, a new aggregation function for a Pareto set could be considered. We believe that it would be beneficial to connect generating the Pareto sets by NSGA-II and evaluating the aggregated Pareto sets into one framework. An automation of generating the ontologies by using Perl script could be also implemented in R language to substitute the manual way of generating the ontology data.

# Chapter 9

# DVD Content

The content of DVD contains folders with 4 datasets. Each dataset has 2 folders, one for pure evaluation of Pareto sets and the other for searching the optimal threshold. The folder GenerateData contains all generated data by OBO::Parser and OBO::Parser itself. Experiments contains two experiment with ontologies. We also provide our thesis in pdf format and following .tex file.:

```
|  Experiments - Kegg superiority and dividing ontology
|__Results - Logs from thresholding algorithms and 2 plots
|__Artificial
|__ArtificialThresholds
|__Disc
|__DiscThresholds
|__Disc2
|__DiscThresholds2
|__Ovary
|__OvaryThresholds
|__M2801
|__M2801Thresholds
|__GenerateData - contains generated data, ontodesctotxt.R and
|  OBO::Parser
|__Thesis - contains .tex file and pdf version of thesis
```

Figure 9.1: DVD content

# Bibliography

[fac, 2015] (2015). Why use the fly in research? https://www.yourgenome.org/facts/why-use-the-fly-in-research.

[OBO, 2018] (2018). Obo::parser http://search.cpan.org/dist/onto-perl/lib/obo/parser/oboparser.pm.

[Alexa and Rahnenfuhrer, 2010] Alexa, A. and Rahnenfuhrer, J. (2010). topgo: enrichment analysis for gene ontology. *R package version*, 2(0).

[Ashburner et al., 2000] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25.

[Balakrishnan et al., 2013] Balakrishnan, R., Harris, M. A., Huntley, R., Van Auken, K., and Cherry, J. M. (2013). A guide to best practices for gene ontology (go) manual annotation. *Database*, 2013:bat054.

[Chen et al., 2011] Chen, L.-C., Yu, P. S., and Tseng, V. S. (2011). Wf-msb: A weighted fuzzy-based biclustering method for gene expression data. *International journal of data mining and bioinformatics*, 5(1):89–109.

[Cheng and Church, 2000] Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Ismb*, volume 8, pages 93–103.

[Cohen, 1995] Cohen, W. W. (1995). Fast effective rule induction. In *Machine Learning Proceedings 1995*, pages 115–123. Elsevier.

[Consortium, 2016] Consortium, G. O. (2016). Expansion of the gene ontology knowledgebase and resources. *Nucleic acids research*, 45(D1):D331–D338.

[Costa et al., 2013] Costa, M., Reeve, S., Grumbling, G., and Osumi-Sutherland, D. (2013). The drosophila anatomy ontology. *Journal of biomedical semantics*, 4(1):32.

[Deb et al., 2002] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2):182–197.

[Edla and Jana, 2012] Edla, D. R. and Jana, P. K. (2012). A prototype-based modified dbscan for gene clustering. *Procedia Technology*, 6:485 – 492. 2nd International Conference on Communication, Computing And Security [ICCCS-2012].

[Gohari and Tarokh, 2016] Gohari, F. S. and Tarokh, M. J. (2016). New recommender framework: combining semantic similarity fusion and bicluster collaborative filtering. *Computational Intelligence*, 32(4):561–586.

[Hall et al., 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

[Hanley and McNeil, 1982] Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

[Hartigan, 1975] Hartigan, J. A. (1975). Clustering algorithms.

[Hochreiter et al., 2010] Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., Kasim, A., Khamiakova, T., Van Sanden, S., Lin, D., Talloen, W., Bijnens, L., Göhlmann, H. W. H., Shkedy, Z., and Clevert, D.-A. (2010). Fabia: factor analysis for bicluster acquisition. *Bioinformatics*, 26(12):1520–1527.

[Holden et al., 2008] Holden, M., Deng, S., Wojnowski, L., and Kulle, B. (2008). Gsea-snp: applying gene set enrichment analysis to snp data from genome-wide association studies. *Bioinformatics*, 24(23):2784–2785.

[Jambor et al., 2015] Jambor, H., Surendranath, V., Kalinka, A. T., Mejstrik, P., Saalfeld, S., and Tomancak, P. (2015). Systematic imaging reveals features and changing localization of mrnas in drosophila development. *Elife*, 4.

[Kanehisa and Goto, 2000] Kanehisa, M. and Goto, S. (2000). Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30.

[Kléma et al., 2016] Kléma, J., Malinka, F., and Zelezny, F. (2016). Semantic biclustering: a new way to analyze and interpret gene expression data. *Bioinformatics Research and Applications, Minsk, Belarus, Springer*, pages 332–3.

[Klema et al., 2017] Klema, J., Malinka, F., and Železný, F. (2017). Semantic biclustering for finding local, interpretable and predictive expression patterns. 18:41–53.

[Krejnik and Klema, 2012] Krejnik, M. and Klema, J. (2012). Empirical evidence of the applicability of functional clustering through gene expression classification. *IEEE/ACM transactions on computational biology and bioinformatics*, 9(3):788–798.

[Kuhn et al., 2007] Kuhn, A., Ducasse, S., and Gîrba, T. (2007). Semantic clustering: Identifying topics in source code. *Information and Software Technology*, 49(3):230 – 243. 12th Working Conference on Reverse Engineering.

[Li et al., 2009] Li, G., Ma, Q., Tang, H., Paterson, A. H., and Xu, Y. (2009). Qubic: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic acids research*, 37(15):e101–e101.

[Liu et al., 2004] Liu, J., Wang, W., and Yang, J. (2004). Gene ontology friendly biclustering of expression profiles. In *Computational Systems Bioinformatics Conference, 2004. CSB 2004. Proceedings. 2004 IEEE*, pages 436–447. IEEE.

[Liu and Wang, 2006] Liu, X. and Wang, L. (2006). Computing the maximum similarity bi-clusters of gene expression data. *Bioinformatics*, 23(1):50–56.

[Lucchese et al., 2014] Lucchese, C., Orlando, S., and Perego, R. (2014). A unifying framework for mining approximate top-$k$ binary patterns. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2900–2913.

[Manuck et al., 2016] Manuck, T. A., Watkins, S., Esplin, M. S., Parry, S., Zhang, H., Huang, H., Biggio, J. R., Bukowski, R., Saade, G., Andrews, W., et al. (2016). 242: Gene set enrichment investigation of maternal exome variation in spontaneous preterm birth (sptb). *American Journal of Obstetrics & Gynecology*, 214(1):S142–S143.

[Merkin et al., 2012] Merkin, J., Russell, C., Chen, P., and Burge, C. B. (2012). Evolutionary dynamics of gene and isoform regulation in mammalian tissues. *Science*, 338(6114):1593–1599.

[Mitra and Banka, 2006] Mitra, S. and Banka, H. (2006). Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, 39(12):2464–2477.

[Mitra and Ghosh, 2012] Mitra, S. and Ghosh, S. (2012). Feature selection and clustering of gene expression profiles using biological knowledge. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1590–1599.

[Mukhopadhyay et al., 2009] Mukhopadhyay, A., Maulik, U., and Bandyopadhyay, S. (2009). A novel coherence measure for discovering scaling biclusters from gene expression data. *Journal of bioinformatics and computational biology*, 7(05):853–868.

[Nepomuceno et al., 2015] Nepomuceno, J. A., Troncoso, A., Nepomuceno-Chamorro, I. A., and Aguilar-Ruiz, J. S. (2015). Integrating biological knowledge based on functional annotations for biclustering of gene expression data. *Computer Methods and Programs in Biomedicine*, 119(3):163 – 180.

[Orlin, 1977] Orlin, J. (1977). Contentment in graph theory: Covering graphs with cliques. *Indagationes Mathematicae (Proceedings)*, 80(5):406 – 424.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[Pio et al., 2012] Pio, G., Ceci, M., Loglisci, C., Malerba, D., and D'Elia, D. (2012). The integration of microrna target data by biclustering techniques opens new roads for signaling networks analysis. *EMBnet.journal*, 18(B).

[Pontes et al., 2015a] Pontes, B., Giráldez, R., and Aguilar-Ruiz, J. S. (2015a). Biclustering on expression data: A review. *Journal of biomedical informatics*, 57:163–180.

[Pontes et al., 2015b] Pontes, B., Giráldez, R., and Aguilar-Ruiz, J. S. (2015b). Biclustering on expression data: A review. *Journal of Biomedical Informatics*, 57:163 – 180.

[Quinlan, 2014] Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

[Roy et al., 2013] Roy, S., Bhattacharyya, D. K., and Kalita, J. K. (2013). Cobi: pattern based co-regulated biclustering of gene expression data. *Pattern Recognition Letters*, 34(14):1669–1678.

[Soulet et al., 2007] Soulet, A., Kléma, J., and Crémilleux, B. (2007). Efficient mining under rich constraints derived from various datasets. In Džeroski, S. and Struyf, J., editors, *Knowledge Discovery in Inductive Databases*, pages 223–239, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Subramanian et al., 2005] 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., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102(43):15545–15550.

[Sun et al., 2016] Sun, M., Mi, P., North, C., and Ramakrishnan, N. (2016). Biset: Semantic edge bundling with biclusters for sensemaking. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):310–319.

[Tanay et al., 2002] Tanay, A., Sharan, R., and Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(suppl_1):S136–S144.

[Uno et al., 2004] Uno, T., Asai, T., Uchida, Y., and Arimura, H. (2004). An efficient algorithm for enumerating closed patterns in transaction databases. In Suzuki, E. and Arikawa, S., editors, *Discovery Science*, pages 16–31, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Verbanck et al., 2013] Verbanck, M., Lê, S., and Pagès, J. (2013). A new unsupervised gene clustering algorithm based on the integration of biological knowledge into expression data. *BMC bioinformatics*, 14(1):42.

[Yang et al., 2005] Yang, J., Wang, H., Wang, W., and Yu, P. (2005). An improved biclustering method for analyzing gene expression profiles. *International Journal on Artificial Intelligence Tools*, 14(5):771–789.

[Zaki and Hsiao, 2005] Zaki, M. J. and Hsiao, C. J. (2005). Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):462–478.