SEMANTIC BICLUSTERING

FRANTIŠEK MALINKA

Doctoral Thesis

Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague

Ph.D. Programme:
Electrical Engineering and Information Technology
Branch of study:
Artificial Intelligence and Biocybernetics

SUPERVISOR:
Doc. Ing. Jiří Kléma, PhD.

SUPERVISOR-SPECIALIST:
prof. Ing. Filip Železný, PhD.

František Malinka: *Semantic Biclustering,* Doctoral Thesis
Ph.D. Programme:
Electrical Engineering and Information Technology
Branch of study: Artificial Intelligence and Biocybernetics

ABSTRACT

This thesis focuses on the problem of finding interpretable and predictive patterns, which are expressed in the form of biclusters, with an orientation to biological data. The presented methods are collectively called semantic biclustering, as a subfield of data mining. The term semantic biclustering is used here because it reflects both a process of finding coherent subsets of rows and columns in a 2-dimensional binary matrix and simultaneously takes into account a mutual semantic meaning of elements in such biclusters. In spite of focusing on applications of algorithms in biological data, the developed algorithms are generally applicable to any other research field, there are only limitations on the format of the input data.

The thesis introduces two novel, and in that context basic, approaches for finding semantic biclusters, as *Bicluster enrichment analysis* and *Rule and tree learning*. Since these methods do not exploit the native hierarchical order of terms of input ontologies, the run-time of algorithms is relatively long in general or an induced hypothesis might have terms that are redundant. For this reason, a new refinement operator has been invented. The refinement operator was incorporated into the well-known CN2 algorithm and uses two reduction procedures: *Redundant Generalization* and *Redundant Non-potential*, both of which help to dramatically prune the rule space and consequently, speed-up the entire process of rule induction in comparison with the traditional refinement operator as is presented in CN2. The reduction procedures were published as an R package that we called *sem1R*.

To show a possible practical usage of *semantic biclustering* in real biological problems, the thesis also describes and specifically adapts the algorithm for two real biological problems. Firstly, we studied a practical application of *sem1R* algorithm in an analysis of E-3 ubiquitin ligase in the gastrointestinal tract with respect to tissue regeneration potential. Secondly, besides discovering biclusters in gene expression data, we adapted the *sem1R* algorithm for a different task, concretely for finding potentially pathogenic genetic variants in a cohort of patients.

Keywords: biclustering, symbolic machine learning, ontology, taxonomy, gene expression, enrichment analysis, background knowledge, semantics

ABSTRAKT

Tato disertační práce se zaměřuje na problém hledání interpretovatelných a prediktivních vzorů, které jsou vyjádřeny formou dvojshluků, se specializací na biologická data. Prezentované metody jsou souhrnně označovány jako *sémantické dvojshlukování*, jedná se o podobor dolování dat. Termín sémantické dvojshlukování je použit z toho důvodu, že zohledňuje proces hledání koherentních podmnožin řádků a sloupců, tedy dvojshluků, v 2-dimensionální binární matici a zároveň bere také v potaz sémantický význam prvků v těchto dvojshlucích. Ačkoliv byla práce motivována biologicky orientovanými daty, vyvinuté algoritmy jsou obecně aplikovatelné v jakémkoli jiném výzkumném oboru. Je nutné pouze dodržet požadavek na formát vstupních dat.

Disertační práce představuje dva originální a v tomto ohledu i základní přístupy pro hledání sémantických dvojshluků, jako je *Bicluster enrichment analysis* a *Rule a tree learning*. Jelikož tyto metody nevyužívají vlastní hierarchické uspořádání termů v daných ontologiích, obecně je běh těchto algoritmů dlouhý či může docházet k indukci hypotéz s redundantními termy. Z toho důvodu byl vytvořen nový operátor zjemnění. Tento operátor byl včleněn do dobře známého algoritmu CN2, kde zavádí dvě redukční procedury: *Redundant Generalization* a *Redundant Non-potential*. Obě procedury pomáhají dramaticky prořezat prohledávaný prostor pravidel a tím umožňují urychlit proces indukce pravidel v porovnání s tradičním operátorem zjemnění tak, jak je původně prezentován v CN2. Celý algoritmus spolu s redukčními metodami je publikován ve formě R balíčku, který jsme nazvali *sem1R*.

Abychom ukázali i možnost praktického užití metody sémantického dvojshlukování na reálných biologických problémech, v disertační práci dále popisujeme a specificky upravujeme algoritmus *sem1R* pro dvě úlohy. Zaprvé, studujeme praktickou aplikaci algoritmu *sem1R* v analýze E-3 ubikvitin ligázy v trávicí soustavě s ohledem na potenciál regenerace tkáně. Zadruhé, kromě objevování dvojshluků v datech genové exprese, adaptujeme algoritmus *sem1R* pro hledání potenciálně patogenních genetických variant v kohortě pacientů.

Klíčová slova: dvojshlukování, symbolické strojové učení, ontologie, taxonomie, genová exprese, analýza obohacení, postranní znalost, sémantika

PUBLICATIONS

List of publications is presented for the purpose of dissertation defence.

Journal articles relevant to the topic of this thesis:

1. **Malinka, F.**; Železný, F.; Kléma, J. Finding Semantic Patterns in Omics Data Using Concept Rule Learning with an Ontology-based Refinement Operator. BioData Mining. 2020, 13(13), 1-22. ISSN 1756-0381. (Journal IF: 2.522)

2. Kléma, J.; **Malinka, F.**; Železný, F. Semantic biclustering for finding local, interpretable and predictive expression patterns. BMC Genomics. 2017, 18(4132), ISSN 1471-2164. (Journal IF: 3.969, WoS: 2, Google: 4)

3. Iatsiuk*, V.; **Malinka*, F.**; Pickova, M.; Tureckova, J.; Kléma, J.; Spoutil, F.; Novosadova, V.; Prochazka, J.; Sedlacek, R. Semantic clustering analysis of E3-ubiquitin ligases in gastrointestinal tract defines genes ontology clusters with tissue expression patterns. *Currently under journal review* (* Co-first author)

Other articles in journals:

1. Reis, M.L.; Sorokina, A.E.; Dudakova, L.; Moravikova, J.; Skalicka, P.; **Malinka, F.**; Seese, E.S; Thompson, S; Bardakjian, T.; Capasso, J.; Allen, W.; Glaser, T.; Levin, V.A.; Schneider, A.; Khan, A.; Liskova, P.; Semina, V.E. Comprehensive phenotypic and functional analysis of dominant and recessive FOXE3 alleles in ocular developmental disorders. Human Molecular Genetics, 2021. (Journal IF: 6.15)

2. Dudakova, L.; Skalicka, P.; Ulmanova, O.; Hlozanek, M.; Stranecky, V.; **Malinka, F.**; Vincent, A.L.; Liskova, P. Pseudodominant Nanophthalmos in a Roma Family Caused by a Novel PRSS56 Variant. Journal of Ophthalmology. 2020, 2020 ISSN 2090-004X. (Journal IF: 1.909, WoS: 1, Google: 1)

3. Skalicka, P.; Porter, L.F.; Brejchova, K.; **Malinka, F.**; Dudakova, L.; Liskova, P. Brittle cornea syndrome: Disease-causing mutations in ZNF469 and two novel variants identified in a patient followed for 26 years. Biomedical Papers. 2020, 164(2), 183-188. ISSN 1213-8118. (Journal IF: 1.245, WoS: 1, Google: 3)

4. Dudakova, L.; Evans, C.; Pontikos, N.; Hafforrd-Tear, N.J.; **Malinka, F.**; Skalicka, P.; Horinek, A.; Munier, F.L. et al. The Utility of Massively Parallel Sequencing for Posterior Polymorphous Corneal Dystrophy Type 3 Molecular Diagnosis. Experimental Eye Research. 2019, 182 160-166. ISSN 0014-4835. (Journal IF: 3.467, WoS: 2, Google: 4)

5. Moravikova, J.; Kozmik, Z.; Hlavata, L.; Putzova, M.; Skalicka, P.; Michaelides, M.; **Malinka, F.**; Dudakova, L.; Liskova, P. Phenotype Variability in Czech Patients Carrying PAX6 Disease-Causing Variants. (Journal IF: 0.906)

6. **Malinka, F.**; Zareie, A.; Procházka, J.; Sedláček, R.; Novosadova, V. Batch alignment via retention orders for preprocessing large-scale multi-batch LC-MS experiments. *Currently under journal review*

7. Liskova, P.; Hafford-Tear, N.; Skalicka, P.; **Malinka, F.**; Moravikova, J.; Dudakova, L.; Pontikos, N.; Davidson, A. Posterior corneal vesicles are not associated with the genetic variants that cause posterior polymorphous corneal dystrophy. *Currently under journal review*

Peer-reviewed conference papers:

1. Kléma, J.; **Malinka, F.**; Železný, F. Semantic Biclustering: A New Way to Analyze and Interpret Gene Expression Data. In: Bioinformatics Research and Applications. Heidelberg: Springer, 2016. pp. 332-333. ISSN 0302-9743. ISBN 978-3-319-38781-9.

2. **Malinka, F.**; Železný, F.; Kléma, J. Genomic single rule learning with an ontology-based refinement operator. In: ENBIK2018 Conference proceedings. Praha: Vysoká škola chemicko-technologická, 2018. p. 69. ISBN 978-80-7592-017-1.

3. **Malinka, F.**; Kléma, J.; Železný, F. Sémantická dvojshluková analýza dat genové exprese. In: Konferenční sborník ENBIK 2016. Praha: Centrální laboratoře, 2016. ISBN 978-80-7080-960-0.

4. Dudakova, L.; **Malinka, F.**; Liskova, P. Identification of two novel BCOR mutations with de novo occurrence. European Journal of Human Genetics. 2019, 27(2), 1874. ISSN 1018-4813.

5. **Malinka, F.** Prediction of Protein Stability Changes Upon One-point Mutations Using Machine Learning. In: Proceeding of the 2015 Research in Adaptive and Convergent Systems (RACS 2015). New York: ACM, 2015. pp. 102-107. ISBN 978-1-4503-3738-0.

Any parts of the original papers reused verbatim in this thesis have been included with the approval of the co-authors.

Naše moudrost, soukromá i veřejná,
náleží světu;
naše pošetilost náleží těm,
které milujeme.

— *Gilbert K. Chesterton*

ACKNOWLEDGEMENTS

I would like to thank my supervisor *doc. Ing. Jiří Kléma, Ph.D.* and my supervisor-specialist *prof. Ing. Filip Železný, Ph.D.* for their invaluable advice, continuous support, and patience during my PhD study.

My warm thanks go to my family for their love and support and also to my friends.

CONTENTS

NOTATION

Selected notation of Chapter 2

$\mathbb{M}$	2-dimensional matrix of real numbers with m rows and n columns
R	set of rows of $\mathbb{M}$
C	set of columns of $\mathbb{M}$
$\mathbb{M}_{IJ}$	submatrix of $\mathbb{M}$ (called bicluster) with a subset of rows and columns

Selected notation of Chapter 4

$\mathbb{A}$	original 2-dimensional matrix of binary values 0 and 1
$\mathbb{M}$	2-dimensional matrix representing training data (unrolled $\mathbb{A}$)
(G', S')	bicluster of genes and situations
B	system of biclusters
(T^γ, T^σ)	semantic bicluster
SB	system of semantic biclusters

Selected notation of Chapter 5

E^+, E^-	set of positive/negative learning examples
$O, \mathcal{O}$	ontology/set of ontologies
θ	cover operator

ACRONYMS

GSEA	Gene Set Enrichment Analysis
OBO	Open Biological and Biomedical Ontology
GO	Gene Ontology
DLO	Drosophila Location Ontology
DAO	Drosophila Anatomy Ontology
DOT	Dresden Ovary Table dataset
DISC	imaginal discs of Drosophila melanogaster
RIPPER	Repeated Incremental Pruning to Produce Error Reduction
NSGA-II	Nondominated Sorting Genetic Algorithm II
LC-MS	Liquid Chromatography – Mass Spectrometry

1 INTRODUCTION

Biclustering has become a very popular technique for discovering local and hidden patterns. Biclustering was first introduced in 1972 by Hartigan [70] and was called direct clustering. Recently, it has become widely applied to biological data [178] in general and gene expression data in particular. In this field, the main objective is to identify a subset of genes that exhibit coherent expression values across subsets of experimental conditions. These two-fold patterns are known to provide a local (and thus better) representation for genes with multiple functions regulated by several different transcription factors. The first biclustering application to the area of gene expression understanding was introduced by Cheng and Church [20] in 2000. Many algorithmic improvements and applications appeared later [102].

Terminology that refers to the same problem formulation is ambiguous, the biclustering is also called co-clustering, bi-dimensional clustering, block-clustering, two-way clustering or subspace clustering. The task of biclustering is an NP-hard problem as proved by Tanay et al. [160]. Simultaneously, it is known that one-way clustering is also an NP-hard problem [63]. Nevertheless, proposing a new effective heuristic function for biclustering is considerably more difficult than for one-way clustering.

As we outlined above, the bicluster is defined by a subset of genes and by a subset of experimental conditions. This biset-based description allows for arbitrary selection of rows and columns. In this thesis, we propose to address biclusters in a different way too. The approach that we call semantic biclustering defines a bicluster as a set of terms from the given prior knowledge where each term is associated with a gene or an experimental condition; thereby the bicluster is determined. In other words, the semantic biclustering utilizes a prior knowledge in the process of seeking homogeneous biclusters. This could be very helpful in a phase of finding biological interpretations, in revealing of unknown relationships across genes and experimental conditions. In addition, the necessary similarity in gene and sample description can help to reduce noise that is inherently present in the gene expression data and often leads to discovering biclusters that are too fragmented [65].

1.1 PROBLEM STATEMENT

When discussing biclustering, the first issue that has to be addressed is the quality of biclusters. Although it may not seem difficult at first glance, the noise inherently present in the data makes the task more challenging. Secondly, a way of seeking biclusters and their forms have to be considered as well.

Extending the conventional biclustering by prior knowledge brings new challenges to the task that we call semantic biclustering. Here, the essential decision that has to be made first is a definition of a hypothesis form in which biclusters are described. Simply put, a more complex form of hypothesis allows to describe more complex patterns in data. On the other hand, the more complex form of hypothesis might be problematic in interpretation and validation of results, especially in biological questions. As an example, we could mention first-order logic which enables to use predicates or function symbols, among other things, in the hypothesis form. In contrary, propositional logic does not provide these elements, thus making the computational runtime more feasible in general. In this thesis, we focus on an easily interpretable form of hypothesis. Concretely, only the conjunction is considered. Even under this simplification, some heuristics need to be applied for seeking semantic biclusters.

1.2 MAIN CONTRIBUTIONS

The main contributions of this thesis are novel algorithms for finding biclusters in a gene expression data that take into account both the gene expression and the semantic similarity of genes/conditions. Although the algorithms focus on the biological domain, they are applicable to any other domain that satisfies the required data format and hold all presented assumptions.

Fundamental theoretical aspects of semantic biclustering and two initially proposed approaches are introduced at the beginning of this thesis. Models induced especially by tree learning have usually very complex forms with high redundancy, which complicates the further process of data analysis like interpretation or hypothesis validation. To avoid potential term redundancy in such hypotheses, a new refinement operator has been formulated. This approach, which integrates the ontology-based refinement operator with CN2 algorithm, is published as R package and written in C++. The package is called *sem1R* and reports rapid runtime speed-up in comparison to the traditional refinement operator used in CN2 algorithm.

With certain adjustments, the package *sem1R* has been used in the field of ophthalmology for finding potential pathogenic genetic variants. Another real example of the useability of the proposed package is manifested in an analysis of E-3 ubiquitin ligase in the gastrointestinal tract.

Besides the semantic biclustering algorithms and their applications in various fields of biology, this thesis also studies a particular part of data preprocessing phase in a specific area of research. Concretely, two algorithms for approximating the correspondence problem of large-scale untargeted liquid chromatography–mass spectrometry (LC-MS) experiments are introduced. This method enables to preprocess considerable numbers of LC-MS experiments easily and then report the final peak (feature) table as a 2-dimensional matrix which can be potentially used as the input for the biclustering algorithms.

1.3 THESIS ORGANIZATION

This thesis is organized as follows. In Chapter 2, we introduce the necessary basic formal definitions for general n-dimensional clustering and then, we focus on the more specific case, 2-dimensional clustering, where we mention related terms in the context of gene expression analysis. Moreover, we review structures of biclusters and the most frequent types of biclusters suitable for general usage or specifically suitable for gene expression data. In Chapter 3, we introduce prior knowledge and its widespread representation in biology - ontology. In addition, we mention some limitations of Gene Set Enrichment Analysis, the method using the semantics in the form of gene annotations. That motivated us to this work. In Chapter 4, we spread out the definition of biclustering to the semantic biclustering and subsequently introduce two novel approaches that are tested on real gene expression datasets. Since the proposed tree and rule learning methods usually induce highly complex and redundant rules which complicate the hypothesis interpretation, in Chapter 5 we introduce a new refinement operator which eliminates these issues and consequently dramatically speeds-up runtimes of algorithm. The whole package is called *sem1R*. The next two chapters, Chapter 6 and 7, describe applications of *sem1R* in two different research areas. Concretely, Chapter 6 describes the application of *sem1R* on data of the gastrointestinal tract. In Chapter 7, we describe necessary adaptations of *sem1R* for finding common potentially pathogenic genetic variants in cohorts of patients. In Chapter 8, we propose a novel method for finding semantic biclusters as a combination of a multi-objective optimization technique and finding descriptions of biclusters using *sem1R*. In Chapter 9, we introduce an approach for preprocessing a considerable number of large-scale untargeted liquid chromatography–mass spectrometry experiments. The final outcome of the proposed approach is a 2-dimensional matrix which might be utilized as an input for the semantic biclustering task. Finally, Chapter 10 concludes the thesis. In addition, the last chapter stores documentation pages of developed and published R packages.

2 BICLUSTERING

In this section, we introduce a general definition of n-dimensional matrix with the elements being real numbers $\mathbb{R}$. Furthermore, we focus on a more specific version of the matrix, 2-dimensional matrix. This type of matrices is frequently used in attribute-value machine learning for its simplicity and often serves as the standard setting [14]. Then, we formulate a definition of *biclustering* and compare it with standard clustering, including its complexity. Besides, we review different types of biclusters and their structures with some corresponding well-established biclustering algorithms.

Many clustering methods have to deal with observed data, oftentimes in their rectangular form, 2-dimensional data matrix. Concurrently, for biological data it is typical *two-way two-mode* matrix depicted in Table 1. According to [169], *two-way* concept refers two-dimensional space and *two-mode* reports two-way data where the first and second dimensions refer to distinct sets of entities. For these two particular reasons, we exclusively restrict to only 2-dimensional matrices. A higher dimensional matrix is disregarded in the thesis.

Consider a new 2-dimensional matrix $\mathbb{M}$ that is defined by two sets $R = \{r_1, r_2, \ldots, r_m\}$ and $C = \{c_1, c_2, \ldots, c_n\}$ that denote a set of rows and columns, respectively. Moreover, assume that matrix $\mathbb{M}$ contains m rows, n columns and each element $m_{i,j} \in \mathbb{R}$ corresponds to a value representing the matrix element in the ith row and jth column.

From these assumptions, we can define three types of clusters on the most general level as *cluster of rows*, *cluster of columns*, and *cluster of rows and columns* known as a *bicluster*. These types were introduced in [102] and they are listed and briefly described below. For more details, we refer to the original paper [102].

A *row cluster* is a submatrix of M with a subset of rows $(I \subseteq R)$ defined over all columns C. In other words, a row cluster has a size of $k \times n$ where $k \leqslant m$.

Similarly, a *column cluster* is a submutrix of M with a subset of columns $(J \subseteq C)$ defined over all rows R. In other words, a column cluster has a size of $m \times k$ where $k \leqslant n$.

Finally, *a cluster of rows and columns* (known as a *bicluster*) $\mathbb{M}_{IJ}$ is a submatrix of M defined by a subset of rows $(I \subseteq R)$ and a subset of columns $(J \subseteq C)$. In contrast to the *cluster of rows* and *cluster of columns*, the size of bicluster $k_r \times k_c$ where $k_r \leqslant m$ and $k_c \leqslant n$ depends on two variable values, which are selected based on specific characteristics of homogeneity. Intuitively, the procedure of selecting the proper size of a bicluster is non-trivial. For this reason, the computational complexity of a task of the biclustering is much higher than in a task of the one-way clustering of rows or columns. The problem of complexity is discussed in more detail in Section 2.1.

For the sake of clarity, we stress the main differences between well-known standard one-way clustering and biclustering. The standard clustering derives a subset of rows (columns) according to the quality of cluster homogeneity throughout all columns (rows). In other words, the final quality of cluster is generally measured as a quantification of similar behavior across all rows (columns). However, there exists an extreme hypothetic situation, where the overall quality of cluster can be significantly influenced by only one element with extremely different value. For example, totally different gene expression values (an outlier) in one sample of gene expression data can dramatically influence the overall homogeneity of genes in the cluster. In [102] say that clustering derives a *global model*. On the other hand, biclustering produces a *local model* because this concept allows excluding the row, column, or specifically the gene from the previous example that significantly decreases the overall quality of homogeneity. The exact characteristics of homogeneity will be discussed in more detail in Section 2.2.

	Condition 1	...	Condition j	...	Condition n
Gene 1	$m_{1,1}$	...	$m_{1,j}$	...	$m_{1,n}$
Gene ...	...	...	...	...	...
Gene i	$m_{i,1}$	...	$m_{i,j}$	...	$m_{i,n}$
Gene ...	...	...	...	...	...
Gene m	$m_{m,1}$	...	$m_{m,j}$	...	$m_{m,n}$

Table 1: Gene Expression Data Matrix $\mathbb{M}$ with m rows and n columns.

2.1 PROBLEM OF BICLUSTERING COMPLEXITY

The biclustering complexity specifically depends on the merit function used to measure the quality of biclusters, where the vast majority of algorithms solving decision variants of this problem are NP-complete [102].

For exact mathematical proof, we need to utilize a procedure to transform a biclustering problem defined by 2-dimensional matrix onto a weighted bipartite graph. This procedure is very straightforward, we assume the following. A graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, is a *bipartite* graph if and only if the set of vertices V can be divided into two disjoint sets V_R and V_C: $V = V_R \cup V_C$ and $V_R \cap V_C = \emptyset$ and every edge $e \in E$ connects a vertex in V_R to one vertex in V_C. Suppose a 2-dimensional matrix $B = (R, C)$ that can be transformed onto weighted bipartite graph G if each row $r \in R$ corresponds to a node $n_{ir} \in V_R$ and each column $c \in C$ corresponds to a node $n_{ic} \in V_C$. The weight of edge $e_{ir,ic} \in E$ between the nodes n_{ir} and n_{ic} has a value corresponding to the value of element in the intersection between row ir and column ic in $\mathbb{M}$. Note that weight of edges represents a level of expression, in the case of gene expression data.

For the sake of simplicity, we further assume that matrix $\mathbb{M}$, defined above, is a binary matrix, where every element $m_{i,j}$ has a value 0 or 1. Then a bicluster corresponds to a *biclique* in the corresponding bipartite graph [102]. In [126], the authors prove that *maximum edge biclique problem*, the problem solving whether bipartite graph G contains a biclique with at least K edges, is NP-complete. Note that *maximum edge biclique problem* is equivalent to finding a maximum size bicluster [102]. In addition, the *maximum edge biclique problem* is NP-complete also if each edge of G has positive weight implies that the matrix may not be binary necessarily [160]. Given this, most of the well-established algorithms utilize a heuristic function to find an appropriate bicluster since an exhaustive search of the space of solutions may be infeasible.

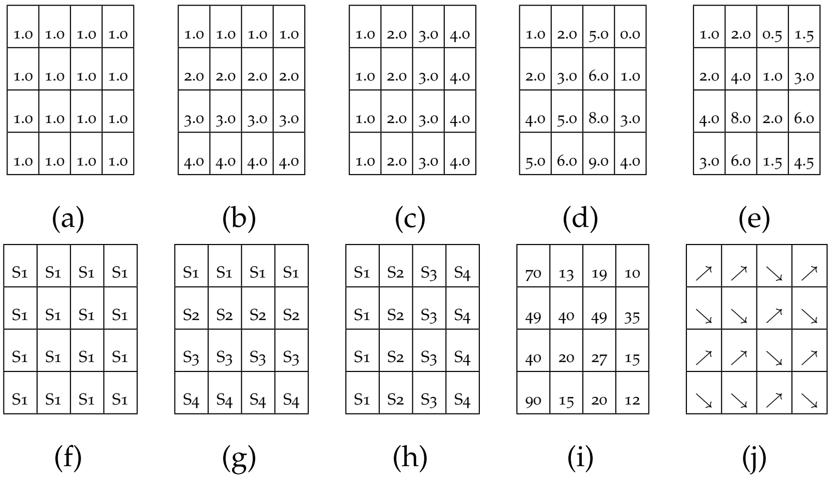
2.2 TYPES OF BICLUSTERS

At the beginning of this chapter, we introduced the term bicluster as a subset of rows that exhibit similar behavior across a subset of columns, and vice versa. However, the main question is still unanswered: how to determine the quality of bicluster? How to recognize that bicluster elements exhibit satisfactorily similar pattern? Or simply, how to find biclusters with regards to a specific application domain?

In order to get a correct answer, firstly, we define a function $h : \mathbb{M}_{IJ} \rightarrow \mathbb{R}$, where the input of h is a bicluster of rows $I \subseteq R$ and columns $J \subseteq C$ and the real value, as an output of the function h, represents the overall bicluster quality. Based on the form of function h, we identify the following classes of biclusters as are presented in [102, 131]:

1. Biclusters with constant values.
2. Biclusters with constant values on rows or columns.
3. Biclusters with coherent values.
4. Biclusters with coherent evolutions.

Several variations of the first four mentioned bicluster classes are depicted in Figure 1 that is taken from [102]. The first three classes evaluate the quality of bicluster based on the numerical values in data matrix. These behaviors can be observed on the rows, columns, or on both of them, see Figures 1(a), 1(b), 1(c), 1(d), and 1(e). On the other hand, biclusters with coherent evolutions view the elements in the data matrix as symbols. These symbols can represent: nominal values, as in Figures 1(f), 1(g), and 1(h); given order, as in Figures 1(i); or positive and negative changes relative to a normal value, as in Figure 1(j) [102].



1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0

(a)

1.0	1.0	1.0	1.0
2.0	2.0	2.0	2.0
3.0	3.0	3.0	3.0
4.0	4.0	4.0	4.0

(b)

1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0
1.0	2.0	3.0	4.0

(c)

1.0	2.0	5.0	0.0
2.0	3.0	6.0	1.0
4.0	5.0	8.0	3.0
5.0	6.0	9.0	4.0

(d)

1.0	2.0	0.5	1.5
2.0	4.0	1.0	3.0
4.0	8.0	2.0	6.0
3.0	6.0	1.5	4.5

(e)

S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1
S1	S1	S1	S1

(f)

S1	S1	S1	S1
S2	S2	S2	S2
S3	S3	S3	S3
S4	S4	S4	S4

(g)

S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4
S1	S2	S3	S4

(h)

70	13	19	10
49	40	49	35
40	20	27	15
90	15	20	12

(i)

↗	↗	↘	↗
↘	↘	↗	↘
↗	↗	↘	↗
↘	↘	↗	↘

(j)

Figure 1: **Examples of different types of biclusters.** (a) Constant bicluster, (b) constant rows, (c) constant columns, (d) coherent values (additive model), (e) coherent values (multiplicative model), (f) overall coherent evolution, (g) coherent evolution on the rows, (h) coherent evolution on the rows, (i) coherent evolution on the columns, and (j) coherent sign changes on rows and columns. Taken from [102].

Here, we briefly introduce only the first type of bicluster because it is the closest to *semantic biclustering*. For more details and descriptions, we refer to the original articles [102, 131].

2.2.1 *Biclusters with constant values*

One of the most straightforward approaches is to find a bicluster or several biclusters which are identified based on a constant value. In this case, we assume that the similar values in 2-dimensional matrix imply similar behavior across the corresponding rows and columns.

We define a *perfect constant bicluster* [1] as a submatrix $\mathbb{M}_{IJ}$ of matrix $\mathbb{M}$ where all elements in perfect bicluster are equal to a constant value π:

$$\forall i \in I, \forall j \in J : m_{i,j} = \pi.$$

This definition of perfect constant bicluster is useful especially for binary matrix, because for most cases the π value is equal to constant value 1 indicating the interesting behavior (e.g. gene expression). On the other hand, this definition is not appropriate for matrix containing real values, because constant biclusters are usually masked by noise. This means that the potential constant bicluster should be rather identified as $\pi + \eta_{ij}$, where η_{ij} represents the noise associated with the value π of m_{ij}. Given this, we can say that the evaluation function h finding constant biclusters can be represented by the variability. Hartigan [70] published one of the first biclustering algorithms (originally called *direct clustering*), although it was not applied to genetic data. The algorithm is based on the splitting approach, where 2-dimensional matrix is partitioned into a set of biclusters according to the quality of each bicluster $\mathbb{M}_{IJ}$. *Variance* is used as the evaluation function h [102, 131]:

$$VAR(\mathbb{M}_{IJ}) = \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} (m_{i,j} - \bar{\mathbb{M}}_{IJ})^2,$$

where $\bar{\mathbb{M}}_{IJ}$ is the mean of all elements in the bicluster $\mathbb{M}_{IJ}$. In order to avoid that the total number of biclusters would be equal to the total number of elements in matrix, because, of course, variance will be equal to zero, Hartigan proposed restriction to the optimal K number of biclusters. The algorithm stops when the matrix is splitted into K biclusters with the final quality expressed as the overall variance across all K biclusters [102]:

$$VAR(\mathbb{M}_{IJ})_K = \sum_{k=1}^{K} \sum_{i=1}^{|I|} \sum_{j=1}^{|J|} (m_{i,j} - \bar{\mathbb{M}}_{IJ})^2.$$

In this subchapter, we demonstrated the key problems that need to be dealt with. The function h must reflect the presence of noise that is inherently present in data and the proper size and the total number of biclusters as well. Concerning *semantic biclustering*, these issues are addressed in the following chapters.

2.3 STRUCTURES OF BICLUSTERS

The second aspect, which is necessary to take into consideration when the algorithm is selected, is a type of structure of discovered bicluster. Choosing the bicluster form should be discussed with respect to the connection with a domain of application. For example, overlapping biclusters is the more appropriate type for gene expression analysis because it reflects the specific property that many genes may belong to several biclusters depending on their influence in the different biological process [55]. There exists a lot of proposed algorithms utilizing the various restrictions on the form of bicluster during a process of its constructing in the task of discovering hidden patterns in gene expression analysis. Given [102], we can classify the fundamental types of biclusters in the following categories:

1. single bicluster,
2. exclusive row and column biclusters,
3. exclusive-rows biclusters,
4. exclusive-columns biclusters,
5. overlapping biclusters without restriction.

The types of biclusters are graphically depicted in Figure 2. In Figure 2(a) is shown only one bicluster that is highlighted by gray color. Of course, the original positions of rows and columns do not produce the compact bicluster as in Figure 2(a). For this visualization, it

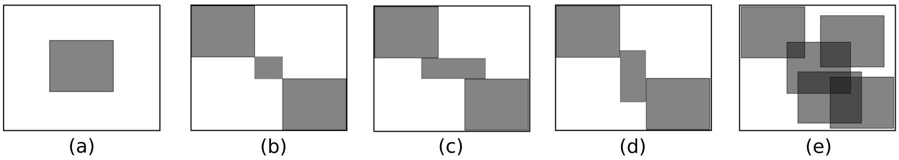


Figure 2: Various types of bicluster. (a) single bicluster, (b) exclusive row and column bicluster, (c) exclusive rows biclusters, (d) exclusive columns biclusters, (e) overlapping biclusters without restriction.

is usually necessary to properly rearrange the rows and columns of the original data matrix.

Firstly, assume that the rows can belong only to one bicluster, while the appearance of columns in biclusters is not restricted, so columns can belong to several biclusters. This structure, which is called exclusive-rows, is presented in Figure 2(c). This structure was used in work [144] and [161].

Not surprisingly, exclusive-columns bicluster depicted in Figure 2(d), where rows can belong to several biclusters, can be obtained by applying the same algorithms as for exclusive-rows, but with the opposite orientation of matrix.

Exclusive row and column bicluster in Figure 2(b) is a combination of exclusive row and column types meaning that all rows and all columns in the matrix belong exclusively to one of all expected biclusters.

The most general structure is shown in Figure 2(e), where restrictions, such as overlapping biclusters or exclusiveness of rows or columns, are not assumed generally. This general structure was used in work [20], [58], [161], [9], [119], [160], or [179].

In that context, *semantic biclustering* focuses on the most general type of biclusters, i.e., *overlapping biclusters without restriction*. This type enables to identify overlapping biclusters which are desirable for our primary datasets – gene expression datasets. We note that the task of semantic biclustering is defined over the binary matrix, therefore, all biclusters might be identified according to the *constant value* 1.

3 SEMANTICS IN GENE EXPRESSION DATA

In this chapter, we briefly outline a form of the background knowledge representing the semantics in biological data, the ontologies. Subsequently, we adduce a very popular method, Gene Set Enrichment Analysis (GSEA) [156], which utilizes the semantics for revealing interesting hypotheses from gene expression data. In this case, ontologies are not used, but only gene sets. In summary, the method is the standard in gene expression analysis. Finally, we discuss limitations in the form of hypothesis constructed by GSEA and outline an extension of the form of hypothesis which eventually leads to our main topic of the thesis, the *semantic biclustering*.

Nowadays, omics data analysis that integrates semantics in the form of external prior knowledge with raw measurements is becoming more and more popular in computational biology [125, 136, 155]. A typical example of integrative gene expression data analysis may deliver a direct link between a phenotype and existing annotation terms at different levels of generality. The integration helps scientists to interpret gene expression data easier because it can reveal gene sets that share common biological properties. Semantic data are stored in databases, oftentimes in an ontology format. In this area, an important role is played by The Open Biological and Biomedical Ontology (OBO) Foundry [145], which provides validation and assessment of ontologies to ensure their interoperability. Dozens of ontologies from various biological domains can be downloaded from http://www.obofoundry.org/.

3.1 ONTOLOGIES

The ontologies in OBO format consist of a set of terms (or classes or concepts) and relationships between them. The formal definition of ontology and associations between ontology terms and some external elements, oftentimes genes, are elaborated in Section 5.1. A graphical representation of a small subset of Gene Ontology [3, 29] is shown in Figure 3. The example shows three terms: *peptidyl-amino acid modification*, *negative regulation of kinase activity*, and *signal transduction* and the more general terms with the corresponding relationships.

3.2 GENE SET ENRICHMENT ANALYSIS

One of the most popular methods that works with semantics and employs gene annotations to interpret gene expression data is *enrichment analysis*. *GSEA* represents one of its most frequently used implementations. The enrichment analysis identifies a list of significantly enriched ontological terms that are associated with the given set of

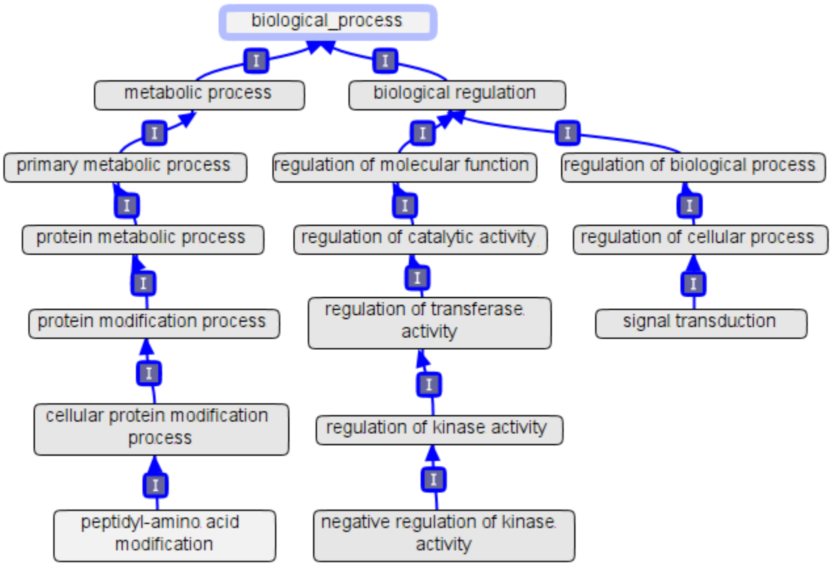

biological_process
metabolic process
biological regulation
primary metabolic process
regulation of molecular function
regulation of biological process
protein metabolic process
regulation of catalytic activity
regulation of cellular process
protein modification process
regulation of transferase activity
signal transduction
cellular protein modification process
regulation of kinase activity
peptidyl-amino acid modification
negative regulation of kinase activity
I

Figure 3: An example of a small subset of terms and their relationships that stems from Gene Ontology. The figure was plotted using OBO-Edit 2.0 [36].

differentially expressed genes. To discover a certain molecular function or biological process that is shared over the set of differentially expressed genes, Gene Ontology is an appropriate and often used annotation database. In addition, finding enriched biological pathways in gene expression data can be done similarly [33], in particular *KEGG* [77–79] and *Reactome* [32] databases are frequently employed in pathway analysis.

An example of GSEA outcome that is induced from data over the KEGG database can be the following:

$$\mathbb{H} = \{\mathit{KEGG\_WNT\_SIGNALING\_PATHWAY}, \\ \mathit{KEGG\_VEGF\_SIGNALING\_PATHWAY}, \\ \mathit{KEGG\_CELL\_CYCLE}\}.$$

In our view, this GSEA outcome corresponds to a hypothesis that can be seen as a collection of three simple rules where each rule has length one and says, independent of the other rules, that the corresponding term in the rule is significantly enriched in the reported set of genes against a background/control gene set. Unfortunately, GSEA in particular, and enrichment analysis in general, cannot produce more complex hypotheses. For example, the hypothesis above does not say that *KEGG_WNT_SIGNALING_PATHWAY* and *KEGG_CELL_CYCLE* are enriched simultaneously, in conjunction. The form of hypothesis only says that these terms are enriched individually. On the other hand, let R be the following rule:

$$\mathit{KEGG\_WNT\_SIGNALING\_PATHWAY} \wedge \mathit{KEGG\_CELL\_CYCLE}.$$

R says that simultaneous occurrence of the terms *KEGG_WNT_SIGNALING_PATHWAY* and *KEGG_CELL_CYCLE* in the annotation of a gene (frequently) leads to its upregulation. The upregulation score for the rule R is computed from a gene set where each gene has to be associated with both terms simultaneously. In our *semantic biclustering*, and unlike the traditional enrichment analysis, we will be able to cope with these conjunctive rules.

Moreover, the dimension of biological samples/conditions is disregarded in the enrichment analysis, only the dimension of genes is taken into consideration when searching for the enriched (ontology) terms. The enrichment analysis supposes a gene set of interest (e.g. genes that are differentially expressed) to be a part of the input. Consequently, these methods can only be applied in such biological experiments, where samples are split into two groups, treatments and controls. However, the treatment and control labels are often not available. In most cases, the split into groups is unclear, the sample groups may overlap or form complex taxonomies. Under these conditions, any set of differentially expressed genes cannot easily be determined. For this reason, we suppose that samples are described with a rich ontology of annotation terms (locations, conditions, situations, complex treatments, etc.) and bring an opportunity to further generalize the rules with extra terms from this ontology that can be added into the rules. This allows for inducing a rule that self-defines the semantically coherent joint groups of genes and samples; the genes tend to be upregulated in the sample group. The induction is fully automated and driven by the context provided in the measurements and annotation ontologies. In other words, GSEA uses a 1-dimensional space of deregulated genes to induce a list of significantly enriched annotation terms. In this work, we expand onto 2-dimensional expression space and consequently allow for generation of hypotheses that represent a set of genes upregulated in a specific set of samples/biological conditions. An example of the hypothesis could be the following rule:

$$\mathbb{H} = \{\ \mathit{KEGG\_WNT\_SIGNALING\_PATHWAY} \wedge \mathit{KEGG\_CELL\_CYCLE} \\ \wedge\ \mathit{WING\_VEIN\_SEGMENT}\ \}.$$

This hypothetical example shows the case where genes belonging to *KEGG_WNT_SIGNALING* and *KEGG_CELL_CYCLE* pathways are frequently upregulated in samples from *WING_VEIN_SEGMENT*, which makes a specific body part of Drosophila melanogaster.

In this part of the thesis, readers have been introduced to the basics of traditional biclustering and with the semantics oftentimes used in the field of biology. In the next chapter, we put these concepts together and formulate the core of this thesis - *semantic biclustering*.

4 SEMANTIC BICLUSTERING

In this chapter, we extend the definition of the ordinary bicluster used in the context of particular format of gene expression matrix to the semantic bicluster. Since *semantic biclustering* approach focuses only on binary data matrix and also there are various algorithms specifics, we bring a specific notation taking this into account. Furthermore, this chapter establishes the fundamental ideas of the thesis, the idea of semantic biclustering, and consequently outlines evaluation procedures to quantify the ability to seek reliable and predictive semantically compact biclusters. The next chapters extend the concepts that are listed here. We note that the content of this chapter has been taken from our publication [84].

The general goal of *biclustering* (or *block-clustering*, *co-clustering*) [169] 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 [102], especially in gene expression data analysis tasks [85, 160]. 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 GSEA [156] 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 [89, 170]. Semantic clustering is also related to the subgroup discovery approach [184], although in an unsupervised setting. The term semantic clustering is also employed in the software-engineering context [90] and captures a roughly similar meaning as in the present context.

In this chapter 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* [74]. 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 approaches most related to semantic biclustering are [153], where formal knowledge associated with both rows and columns of a data matrix is used to specify filters for detected patterns and [121, 122], which aim at biclustering of gene expression data with biclusters coherent in terms of gene functional annotation. The authors of [66] proposed a new iterative bi-clustering algorithm and applied it to a binary gene set expression dataset, i.e., the dataset where expression of whole gene sets was captured. They worked with the semantic annotation of the original gene expression data, but they employed the semantics solely in the preprocessing step.

In the rest of the chapter 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.

4.1 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, $\mathbb{A} = (a_{g,s})$ 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 independent and identically distributed (i.i.d.); more precisely, that S is drawn i.i.d.[1] from the marginal $p(s)$. In the present biclustering 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

[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.

$(g', s), g' \in G$ and all pairs $(g, s'), s' \in S$, so the samples are mutually dependent.

4.1.1 *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

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

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

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

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

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

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

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

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 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 $\widehat{\mathrm{Acc}}$ as a proxy for the true accuracy entails certain problems.

First, as we have commented already, the sample set where $\widehat{\mathrm{Acc}}$ is determined is not i.i.d. as normally required for a training set, although this could be tolerated if the intended use of $\widehat{\mathrm{Acc}}$ is as a heuristic guiding the search for B, rather than as an unbiased estimator. Second, $\widehat{\mathrm{Acc}}$ 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

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

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 [101, 112] 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 $\mathit{Test} = \{(g_k, s_k, a_k)\}$ drawn i.i.d. from $p(g, s, a)$, as

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

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*.

4.1.2 *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* $\mathit{SB} = \{(T^\gamma_k, T^\sigma_k)\}$ 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 $\mathit{ext}(\mathit{SB})$ of a system of semantic biclusters SB is not confined to the matrix of genes G and situations S

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

and thus also the indicator function $sb : \Gamma \times \Sigma \rightarrow \{0, 1\}$ 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.

4.1.3 *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

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

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.

4.2 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 one we employ an existing biclustering algorithm and subject its results to GSEA [156] algorithm, revealing annotation terms which are enriched on either dimension of the produced biclusters. The alternative strategy is materialized by an arrangement of classical symbolic machine-learning techniques known as decision rule and tree learning [138]. It is implemented in terms of two closely related methods that share the preprocessing step and differ in the consecutive learning step.

4.2.1 *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. In the field of biclustering, this is a well-known task that can be tackled with approximate pattern matching [101, 111, 177], non-negative matrix decomposition [185, 186], bipartite graph partitioning [38] or heuristic algorithms [19, 133, 137, 168]. The bicluster semantics are disregarded for the moment.

In our approach, we employed the popular PANDA+ tool [101] 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 [50] 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 [28, 56] and KEGG terms [77–79] to annotate the individual genes. The dedicated Drosophila location ontology (DLO) terms [39] and Drosophila anatomy ontology (DAO) terms [31] 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 hyperparameter 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 [2] to find the GO terms and the Fisher test to reveal the KEGG and location ontology terms enriched in the individual biclusters.

Algorithmus 1 : Bi-directional enrichment.

```
input  : $\mathbb{A}^{m \times n}$, $a_{i,j} \in \{0, 1, NA\}$; // NAs for testing fields
         $R_\gamma$; $R_\sigma$; // gene (GO, KEGG) and location annotation relations
output : $\Pi^S$; // 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($R_\gamma$, $G$); // filter all gene terms relevant to $\mathbb{A}$
$S \leftarrow$ getAllLocationNames($\mathbb{A}$); // all locations in $\mathbb{A}$
$\sigma \leftarrow$ getAllLocationTerms($R_\sigma$, $S$); // 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($B_{k,genes}$, $\gamma_i$, $G$, $R_\gamma$)
    end
    for $j \leftarrow 1$ to $s$ do
        $\Pi^S_{k,g+j} \leftarrow$ enrichmentGet($B_{k,locs}$, $\sigma_j$, $S$, $R_\sigma$)
    end
end
```

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 [102], the suboptimal heuristic algorithm is computationally scalable. The size of the input matrix influences mainly the initial bicluster search; time complexity of PANDA+ is $\mathcal{O}(|B|mn^2)$ [101] 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 $\mathcal{O}(|B|(|\gamma| * m + |\sigma| * n))$.

4.2.2 *Rule and tree learning*

The alternative approach is based on a reduction of the problem to a classification-learning problem. This entails a transformation of the original data matrix $\mathbb{A}$ into an auxiliary binary matrix $\mathbb{M}$ 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

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

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 we search for takes the form of a list of conjunctive decision rules [138], each of which acquires the form

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

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 for a pair (g, s), 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,

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

```
input  : $\mathbb{A}^{m \times n}$, $a_{i,j} \in \{0, 1, NA\}$; // NAs for testing fields
         $R_\gamma$; $R_\sigma$; // 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 */
$G \leftarrow$ getAllGeneNames($\mathbb{A}$); // all genes in $\mathbb{A}$
$\gamma \leftarrow$ getAllGeneTerms($R_\gamma, G$); // list all gene annotation terms
$S \leftarrow$ getAllLocationNames($\mathbb{A}$); // all locations in $\mathbb{A}$
$\sigma \leftarrow$ getAllLocationTerms($R_\sigma, S$); // list all location terms
$g \leftarrow |\gamma|$; $s \leftarrow |\sigma|$;
for $i \leftarrow 1$ to $m$ do
    $T \leftarrow 0^{|\gamma|+|\sigma|+1}$; // term indicator vector initialization
    for $j \leftarrow 1$ to $g$ do
        if $(\gamma_j, G_i) \in R_\gamma$ then $T_j \leftarrow 1$;
    end
    for $k \leftarrow 1$ to $n$ do
        for $j \leftarrow 1$ to $s$ do
            if $(\sigma_j, S_k) \in R_\sigma$ then $T_{g+j} \leftarrow 1$;
        end
        $T_{|\gamma|+|\sigma|+1} \leftarrow a_{i,k}$; // add expression indicator
        $\mathbb{M}_{(i-1) \cdot n+k,*} \leftarrow T$;
    end
end
$\mathbb{M} \leftarrow$ filterGeneTerms($\mathbb{M}, \Theta$); // wrt to a given threshold $\Theta$;
```

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 a variation to the workflow described, in which the rule set learner is replaced by a *decision tree* learner [138]. Each vertex in a learned tree corresponds to one ontology term, and the test represented by the vertex determines whether the term is among the annotation of the classified pair of gene and situation. Since all the attributes (including the class attribute) of the 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 [176] 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) [27]. J48 is an implementation of the well-known C4.5 algorithm [135].

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 [106], 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 enrichment. For this reason, we perform a feature selection step prior to the learning step. The published JRip's time complexity [27] implies the learning complexity for our problems $\mathcal{O}(mn\log^2(mn))$. In other

[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 [184].

words, a large number of samples in $\mathbb{M}$ indicates a time consuming run if compared to the other methods implemented in our work.

4.3 EVALUATION PROCEDURE

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 strategy based on conventional biclustering and subsequent enrichment analysis results in a set of soft semantic biclusters inducing the classification principle described by Eq. 6. The latter depends on the two hyper-parametric 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 $(\mathsf{FPr}, \mathsf{TPr})$ 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.

Algorithmus 3 : Predictive evaluation of bi-directional enrichment.

```
input      : $\Pi^S; \mathbb{A}^{m \times n}$, $a_{i,j} \in \{0, 1, NA\}$; // 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     : $\mathbb{P}^{m \times n}$, $p_{i,j} \in \{0, 1, NA\}$ // 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($R_\gamma, G$); // 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($R_\sigma, S$); // 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
```

4.4 EXPERIMENTAL DATASETS

We conducted our experiments on two real datasets. The first one is the Dresden ovary table [39]. 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 [74] 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. As such, the hierarchy deals with 100 terms. The gene ontology was used in its standard available form [2, 56] including 8,407 GO terms in total. 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 that 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 2.

The second experimental dataset comes from the same organism, i.e., Drosophila melanogaster, and captures the spatial gene expression in the larval imaginal discs (DISC). 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 4 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 3.

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. Furthermore, each segment of a particular imaginal disc was manually assigned a set of DAO terms. 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 4.

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.

	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 2: Drosophila ovary table 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 3: Imaginal disc dataset statistic.

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

Table 4: The number of annotation terms available for our experimental datasets.

4.5 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 reason for different values between the gene dimension and the situation dimension is that the number of situations is lower than the number of genes and the location ontology is less complex than the gene annotation. For this reason, even less significant location terms prove helpful when generalizing to unseen data. 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 5 shows the results including the AUROC achieved by the two proposed strategies (the rule and tree learning strategy is represented by the rule learning method and the tree learning method, they are evaluated independently) as well as further information re-

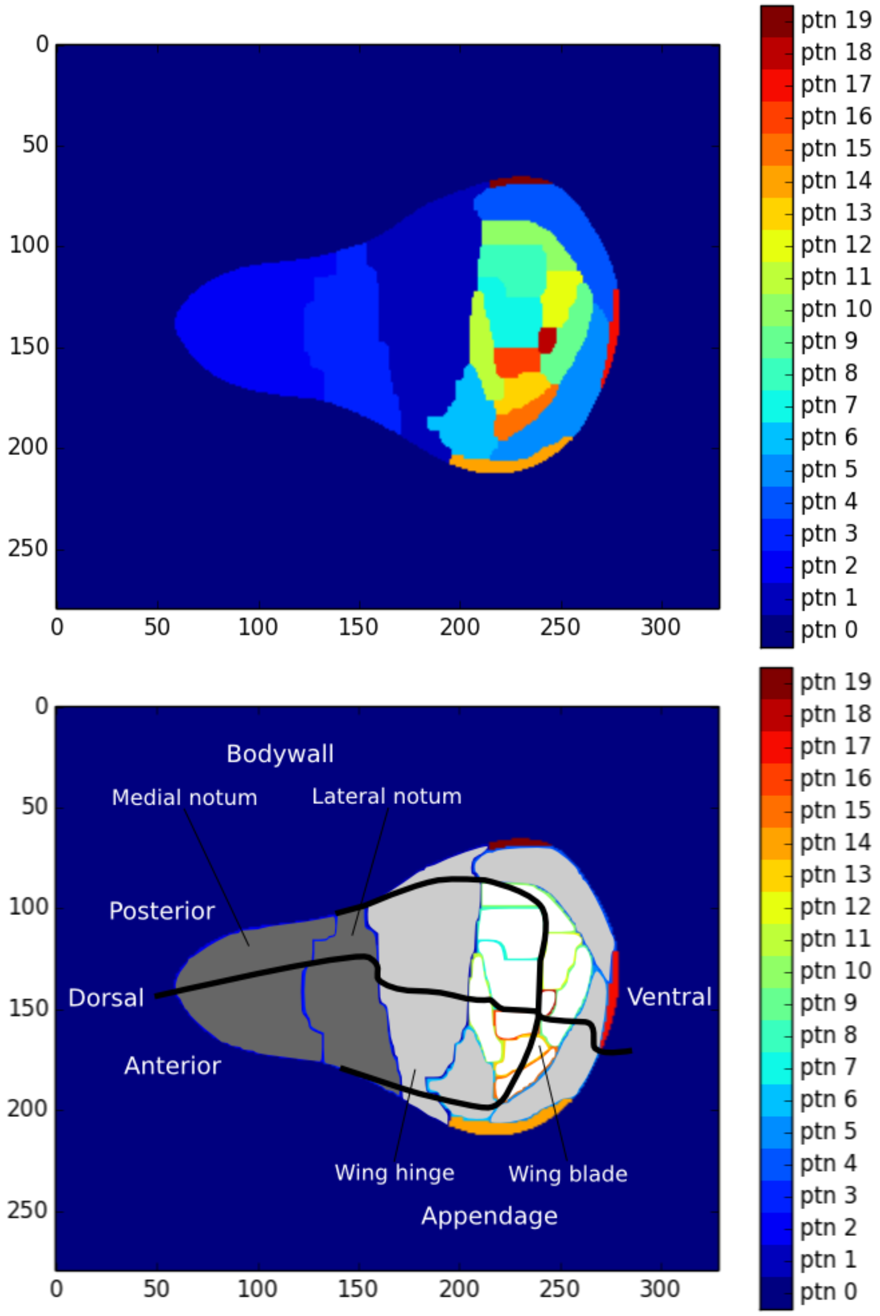


Figure 4: Segmentation of an imaginal disc. An example of segmentation of an imaginal disc (top), altogether with its annotation by the Drosophila ontology terms (bottom). The disc is split into 20 segments distinguished in colors, the split was found to best capture the gene expression patterns observed in the individual in situ hybridization images. The annotation stems from [62].

garding 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 strategies generalize far better than random. In other words, the semantic descriptions of the biclusters can be used to predict the expression for combinations of genes and situations not present in training data.

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

Table 5: Evaluation results of the proposed approaches to semantic biclustering. The Bicluster Enrichment method is denoted as BE.

4.6 discussion

The bicluster enrichment method 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 employs a 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 method 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. Table 6 evaluates biological homogeneity of the found biclusters in terms of their enrichment. The table shows the proportion of generated biclusters that have at least one enriched annotation term in each dimension at the level of significance 0.05. As the rule and tree learning methods directly define biclusters by the annotation terms, their proportions are naturally high. Biclusters without an annotation in one of the directions may originate namely if a bicluster is defined solely by one type of terms (either gene, or location terms). The proportions of enriched biclusters reached by bi-directional enrichment are lower but satisfactory too. We ascribe it to the PANDA's ability to cope with noise and search for large and semantically interpretable biclusters. The biological homogeneity is comparable with the result published in [122], where homogeneity in gene dimension only was measured.

Dataset	Method	% enriched
Ovary	Bicluster Enrichment	0.952±0.063
	Rules (JRip)	0.981±0.017
	Tree (J48)	0.974±0.021
DISC	Bicluster Enrichment	0.851±0.102
	Rules (JRip)	0.962±0.041
	Tree (J48)	0.931±0.043

Table 6: Biological homogeneity of the found biclusters in terms of their enrichment.

Figure 5 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 6. 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 7. The main conclusion is that it is much easier to

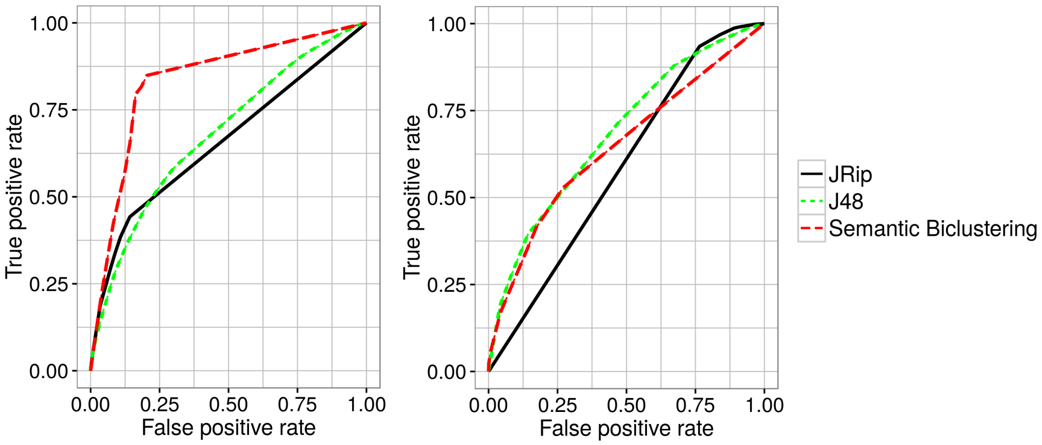


Figure 5: Semantic biclustering ROC curves for Drosophila ovary table (left) and Imaginal disc dataset (right).

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.

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
DISC	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 7: 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.

Runtimes of all the three implemented methods are summarized in Tables 8 and 9. All tests were performed with the same configuration: 8-core Intel Xeon E5-2630v3 2.40GHz. We measured runtimes in 10 experimental runs with different random train-test splits. The tables distinguish the individual subtasks that underlie the implemented methods. Table 9 for bi-directional enrichment distinguishes the preparatory subtask (data and ontology upload, train-test split preparation), the model building (biclustering in PANDA+) and the model testing (annotation of the individual biclusters and their application to test data). Table 8 splits the runtime between the ARFF building (process of unrolling the gene expression matrix into the ARFF file), the model building (learning of decision trees or rule sets) and the model testing (the application of the trees or rules to test data). The runtimes

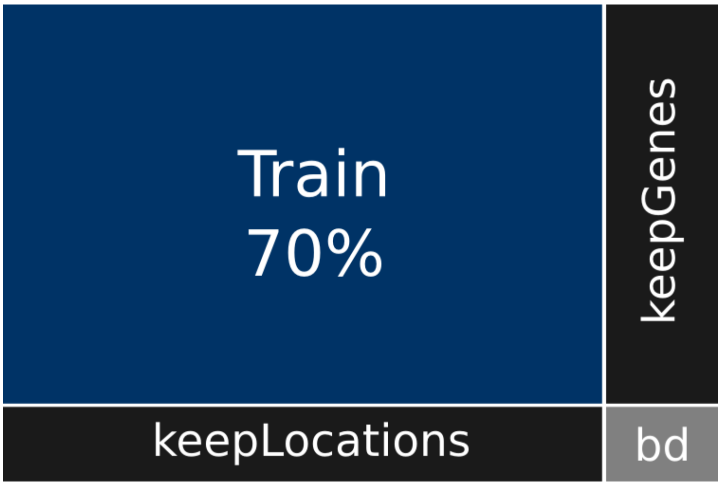


Figure 6: Train and test matrices.

show that biclustering enrichment method is in the order of magnitude faster than rule and tree learning. Firstly, it is the result of large semantic description as discussed during the theoretical complexity analysis. Secondly, it stems from efficient implementation of PANDA+ in C while the rest of the code runs in R, Perl and Java. Consequently, only the building of ARFF file in rule and tree learning takes more time than bi-directional enrichment. These two reasons also contribute to the fact that bicluster annotation and application to test data is more time consuming than bicluster construction in bi-directional enrichment. It is also clear that JRip algorithm is much less computationally efficient than J48.

Split	DOT					DISC				
	build ARFF	build model		test model		build ARFF	build model		test model	
		J48	JRip	J48	JRip		J48	JRip	J48	JRip
1	1,033	1,237	26,810	17.00	23.44	274	59.59	510.84	3.08	3.11
2	1,091	1,503	21,384	19.45	18.67	272	38.03	557.92	2.93	3.19
3	1,042	1,076	19,519	19.09	18.19	287	71.62	363.00	3.16	3.16
4	1,096	1,300	20,054	17.59	19.07	270	64.65	438.87	3.16	3.25
5	1,127	2,010	20,605	18.61	21.22	278	39.47	941.30	3.20	3.64
6	1,121	1,999	24,568	19.38	18.69	260	39.77	550.50	3.11	3.05
7	1,097	1,656	25,279	18.90	18.60	281	47.61	288.14	2.98	3.00
8	1,058	1,087	22,459	26.47	18.48	269	44.00	641.16	3.14	3.26
9	1,023	1,236	14,062	17.81	18.24	288	54.83	201.10	3.25	2.91
10	1,268	1,583	27,299	18.81	21.07	276	42.83	506.14	2.96	3.06
$\bar{x}$	1,096	1,469	22,204	19.31	19.57	629.4	50.24	499.9	3.10	3.16
$\mathrm{sd}(x)$	±70.6	±343	±3,995	±2.64	±1.75	±32.3	±11.78	±204.8	±0.11	±0.2

Table 8: Runtimes (in seconds) of rule and tree learning methods on DOT and DISC datasets. The process of transforming original matrix onto ARFF file (build ARFF) and the process of building classification models were measured separately.

Split	DOT			DISC		
	prepare data	build model	test model	prepare data	build model	test model
1	21.80	74.75	278.79	14.75	133.14	70.42
2	20.44	122.27	233.85	13.96	112.36	53.41
3	14.76	100.80	259.17	10.11	101.49	49.12
4	16.05	87.42	223.64	9.36	107.10	47.32
5	14.54	120.49	266.52	9.28	72.78	60.17
6	16.98	110.70	228.80	13.87	124.81	45.06
7	14.79	100.55	231.63	9.51	153.33	82.83
8	14.43	80.02	229.41	14.08	144.09	50.18
9	14.58	94.29	204.34	9.73	176.95	61.83
10	14.02	103.77	230.10	15.60	90.13	45.86
$\bar{x}$	16.24	99.51	238.63	12.03	121.62	56.62
$sd(x)$	±2.73	±15.88	±22.46	±2.61	±31.26	±12.30

Table 9: Runtimes (in seconds) of bi-directional enrichment on DOT and DISC datasets.

4.7 CONCLUSION

We have motivated and defined the task of semantic biclustering and proposed two strategies 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 bicluster enrichment method achieves the best performance in terms of the area under the ROC curve, at the price of employing a large number of ontology terms to describe the discovered biclusters.

Furthermore, an attempt to develop a new method for semantic biclustering that combines the complementary advantages of the proposed approaches has been made. The method is described in Chapter 8. In principle, 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. Besides, a new refinement operator improving the traditional rule learning operator has been proposed and it is described in Chapter 5. Finally, a biological interpretation of the results reached in a particular domain, the domain of gastroenterology, is provided in Chapter 6.

We made the project publicly available through GitHub [143]. The repository contains source code of both the implemented strategies as well as both the experimental datasets.

5 CONCEPT RULE LEARNING WITH AN ONTOLOGY-BASED REFINEMENT OPERATOR

This chapter seamlessly follows the work from the previous chapter and accentuates potential disadvantages and complications that rise up in the process of seeking biclusters. The evident difficulties come especially from rule learning techniques such as JRip and J48. From the previous experiments, we observe that J48 tends to generate highly complex models (hypotheses) and, simultaneously with JRip, hypotheses are usually redundant. The redundancy means that a hypothesis contains such a pair of ontological terms where a relationship between them exists. Given this, redundant hypothesis does not improve predictive accuracy in comparison to the corresponding non-redundant hypothesis; it only makes the hypothesis more complex and therefore difficult to interpret or validate. Hence this chapter describes adjustments of the traditional refinement operator of CN2 algorithm, the well-known algorithm for rule learning, which lead to inducing non-redundant hypotheses. Furthermore, we complete the precise formal definition of background knowledge (ontology) and its assumptions, and a relationship between background knowledge and the input 2-dimensional matrix. Concurrently, this chapter represents an effort to reduce a potentially high number of negative ontological terms in the resulting hypothesis since they make it difficult to interpret the hypothesis in general because it is necessary to know overall domain of background knowledge. It is therefore an improvement and extension of the originally proposed method for semantic biclustering. We note that the content of this chapter has been taken from our publication [103].

We use rule learning to construct the conjunctive hypotheses, to exemplify, let us consider the same hypothesis we discussed in Chapter 3:

$$\mathbb{H} = \{\ \mathit{KEGG\_WNT\_SIGNALING\_PATHWAY} \wedge \mathit{KEGG\_CELL\_CYCLE} \\ \wedge \mathit{WING\_VEIN\_SEGMENT}\ \}.$$

Rule learning refers to a class of supervised machine learning methods that induce a set of classification rules from a given set of training examples [88]. For a binary task, training examples are assigned to two disjoint sets of positive and negative examples. The rule is an if-then statement where the antecedent is in the form of a conjunction of positive or negative logical terms, and the consequent is a class label. The final decision regarding an unseen example is provided by a set of rules or their ordered list. The rules are widely used in the fields of medicine and biology for their easy and clear interpretation [8, 17, 72] contrary to neural networks, for instance.

As previously mentioned, one of the things that can help scientists interpret their data in a more natural way is background knowledge.

We have already mentioned Gene Ontology [3, 29] and KEGG [77–79], which can also be interpreted as an ontology or a taxonomy. Moreover, medicine employs Disease Ontology [82, 141] or SNOMED-CT, natural language processing makes use of WordNet [113] or YAGO [157], dedicated ontologies are often encountered in industry too.

In our work, these two concepts, rule learning and ontologies or taxonomies, are combined. We observed that the ontologies reasonably increase accuracy and robustness of induced rules. However, they also reasonably raise the number of logical terms available for rule construction, which consequently leads to prohibitive growth of hypothesis space and inefficiency of rule learning. This inefficiency can reasonably be reduced with consistent utilization of the known hierarchical relationships between the ontology terms that cannot be handled with the traditional rule learning methods [26, 27]. In accordance with the previous chapter, we will focus on the binary task (positive and negative examples, two classes only) and multiple rule models (the output of the learning algorithm is multiple rules).

The main motivation for this chapter was our work presented in Chapter 4, in which we introduced a technique called *semantic biclustering*. This type of biclustering infers a human easily readable form of hypothesis describing only a single target class (also known as the target concept). This technique is applied to a gene expression data where highly expressed genes in corresponding samples are considered as the target class. One of the proposed methods solves the problem of semantic biclustering by linearizing a two-dimensional binary data matrix and a set of ontologies to an attribute-value representation that can be figured out using one of the well-known rule learning algorithms such as CN2 [25, 26], RIPPER [27], or PRIM [51]. However, current ontologies, such as Gene Ontology, contain tens of thousands of hierarchically ordered terms. As a result, building a classification model without a preprocessing step is time and memory consuming. For this reason, we introduce a new refinement operator for a rule learning algorithm that examines properties between given data, ontologies, and its mutual relations to speed-up and improve the process of learning.

One of the related subfields of machine learning that can handle a large amount of prior knowledge (i.e. ontology or taxonomy) is Inductive Logic Programming (ILP) where a key challenge is to prune a search space. This is caused by the fact that hypotheses are formulated in first-order logic and ILP algorithms have to search over a large hypothesis space. For its ability to work with this form of prior knowledge, we were inspired by this subfield. In [183], the authors proposed a refinement operator to construct conjunctive relational features that use taxonomies to speed-up the process of propositionalization. This algorithm uses taxonomies to exclude conjunction from the exploration process if the conjunction contains a feature together with any of its subsumees. In [159], the authors find and prune such hypotheses that are equivalent to a previously considered hypothesis. To test such equivalency in given domain theory, they proposed a saturation method for a first-order logic clause with the property that two clauses are equivalent whenever their saturations are isomorphic. However, ILP can build a highly complex hypothesis whose interpretation can prove difficult, especially in a biological domain where validation of such a hypothesis can hardly be feasible. For this reason, we decided to focus on the easily interpretable hypotheses produced by rule learning algorithms.

In this work, we propose a new rule learning algorithm that builds classifiers, as is usual for rule learning, from given positive and negative examples. To build such a classifier, we were inspired by the well-known CN2 algorithm that is based on the Beam search heuristic [26]. A refinement operator of CN2, that is used for a rule space exploration, extends the current rule by trial adding all features from a set of features to the antecedent of the rule. For example, with a set of four features $\{t1, t2, t3, t4\}$, the current rule $t1 \wedge t2 \rightarrow yes$ (all the examples with positive values of features t1 and t2 belong to the positive class) could be extended to two rules $t1 \wedge t2 \wedge t3 \rightarrow yes$ and $t1 \wedge t2 \wedge t4 \rightarrow yes$[1]. In particular, we introduce a new refinement operator that, due to ontologies, significantly reduces a search space of rules and consequently reduces run time of rule learner in comparison to the traditional refinement operator without a loss of accuracy. In our example, if knowing that t4 is more general than t1, the second rule extension can be rejected for redundancy without testing. The proposed ontology-based refinement operator uses two reduction procedures: a *Redundant Generalization* that omits candidate rules based on a relation generalization-specialization and a *Redundant Non-potential* that omits the candidate rules which cannot improve classification accuracy. We demonstrate effectiveness and efficiency of our algorithm on three real gene expression datasets: *Dresden ovary table* (DOT) [39, 74], *Drosophila imaginal discs* (DISC) [12], and dataset of *Strand-specific RNA-seq of nine mouse tissues* (m2801) [109].

5.1 PROBLEM DEFINITION

To start with, we formally define two basic concepts: ontology and example as inputs of the proposed algorithm.

Firstly, assume ontologies as a partial-ordered set $< T, \succeq >$, where T represents a set of all terms that are presented in the given ontologies and $\succeq$ is a binary relation defined in T such that $(g, s) \in \succeq \subseteq T \times T$. In other words, term g is $\succeq$-related to term s. For example, in the context of Gene ontology [3, 29], we can say that the term *developmental process* is a subtype of term *biological process*, written as (*biological process*, *developmental process*) $\in \succeq$. According to partial-order set definition, the $\succeq$ relation is reflexive, transitive, and antisymmetric. Consequently, ontologies cannot contain any cycles. For a better understanding, we define the following concepts that will be used in this thesis later.

Definition 1. *Let $x, y \in T$, x is called a* generalization *of y (or x is more general than y) iff $x \succeq y$.*

[1] As we deal with the binary class we will skip the right hand side of the rules in the rest of text and implicitly assume that all the rules target the positive class.

Definition 2. *Let* $x, y \in T$*,* y *is called a* specialization *of* x *(or* y *is more specific than* x*) iff* $x \succeq y$*.*

Given these two definitions, we are able to express basic elements defined over ontologies. As an example, we can define all top elements in ontologies. In the context of ontology, top elements correspond to roots, i.e. the terms that do not have any ancestors/more general terms.

Example 1. *Let ontology* $O =< T, \succeq^* >$ *be a partial-ordered set that is shown in Figure 7 where* $T = \{t0, t1, t2, t3, t4, t5, t6\}$*,* $\succeq = \{(t0, t1), (t0, t2), (t0, t3), (t1, t4), (t2, t4), (t2, t5), (t3, t5), (t3, t6)\}$*, and* $\succeq^*$ *is a reflexive transitive closure of the relation* $\succeq$*. Then* t0 *is more general than* t1, t2, t3, t4, t5 *and* t6 *and simultaneously* t0 *is a root since there is no other more general term of* t0*.*

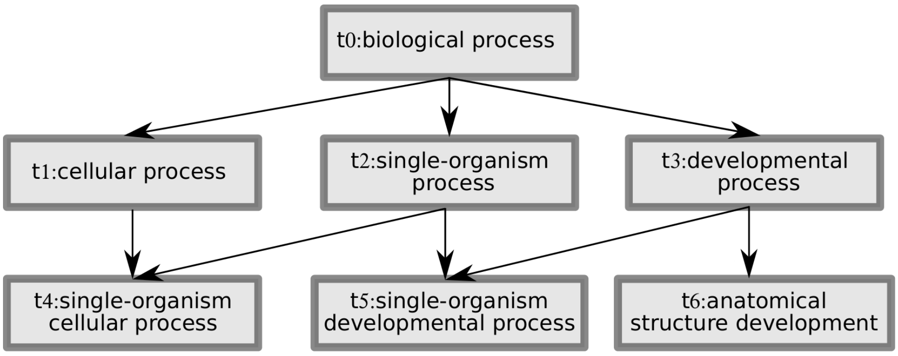

t0:biological process
t1:cellular process
t2:single-organism process
t3:developmental process
t4:single-organism cellular process
t5:single-organism developmental process
t6:anatomical structure development

Figure 7: An example of partial-order binary relation $\succeq$ over a set of terms T. The partial-ordered set is depicted in the form of Hasse diagram. The terms and relations come from Gene Ontology.

Secondly, every ontology refers to a certain set of ground level objects. In this work we will call them examples. In the case of gene ontology, the examples could be the set of measured genes, in the case of location ontology it can be the set of body parts from which the measurements come. We define a set of examples $E = E^+ \cup E^-$ where E^+ represents a set of positive examples (e.g. a set of up-regulated genes or a gene set of interest) and E^- represents a set of negative examples (e.g. a set of down-regulated genes or a control gene set). Finally, we define an association between examples and ontology terms and vice versa. These associations are essential for the novel refinement operator and for the two reduction procedures.

Since each example is annotated by a subset of terms from an ontology, we assume a mapping

$$M : E \to \mathcal{P}(T) \tag{9}$$

that maps examples from the set of examples E to elements of the power set of terms T from a given ontology. To illustrate, an example is represented by the Drosophila's gene *Phosphoenolpyruvate carboxykinase* that is annotated by the following Gene ontology terms: *GTP binding*, *phosphoenolpyruvate carboxykinase activity*, *gluconeogenesis*, and *mitochondrion*. In most cases, this mapping function M is defined manually by a user based on their expert knowledge, or automatically by well-known tools that help map a text associated with an example to an ontology as [182].

On the other hand, there are associations from terms to examples. For this, we define a reverse mapping

$$M' : T \rightarrow \mathcal{P}(E) \tag{10}$$

that maps elements from the set of terms T to elements of the power set of examples E. The formal definition of this function follows:

$$M'(t \in T) = \{\forall e \in E : t \subseteq M(e)\}. \tag{11}$$

Example 2. *Suppose the ontology as a partial-order set that is defined in Example 1. Let* $E = \{e_1, e_2, e_3\}$ *be a set of examples, and the mapping* M*, that assigns terms from the ontology* O *to the specific example, is defined manually in the following way:* $M(e_1) = \{t4\}, M(e_2) = \{t5, t6\}, M(e_3) = \{t2\}$*. The mapping* M' *that reversely assigns set of examples to the specific term from the ontology* O *is following:* $M'(t0) = \{\emptyset\}, M'(t1) = \{\emptyset\}, M'(t2) = \{e_3\}, M'(t3) = \{\emptyset\}, M'(t4) = \{e_1\}, M'(t5) = \{e_2\}, M'(t6) = \{e_2\}$*. The graphical representation of the partial-order set, examples, and their associations with terms is shown in Figure 8.*

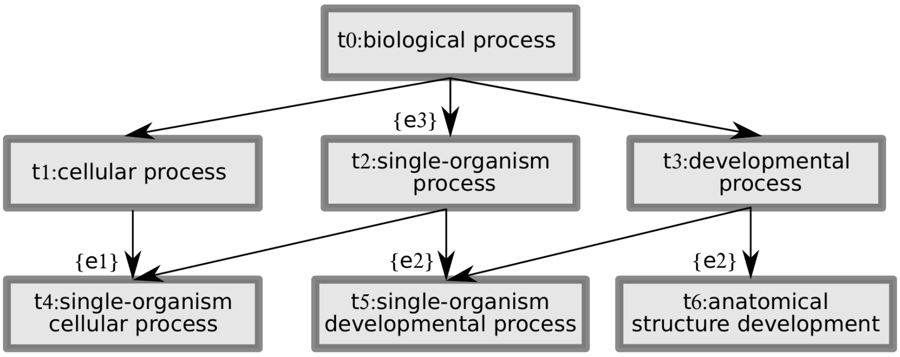

t0:biological process
{e3}
t1:cellular process
t2:single-organism process
t3:developmental process
{e1}
{e2}
{e2}
t4:single-organism cellular process
t5:single-organism developmental process
t6:anatomical structure development

Figure 8: The extended version of the original Figure 7. Elements in curly brackets represent examples that are associated with the terms according to the mapping M'.

Since the associations between examples and terms have been defined, we may propagate these associations to the more general terms. It holds that if an example associates with a term then by default it associates with all generalizations of this term as well. The propagation is shown in Figure 9 and it is concurrently defined in Eq. 12. The process of spreading information in ontologies is represented by a mapping S where for each term we firstly find a corresponding set of all its specializations, and then we unite all examples associated with them.

$$S : T \rightarrow \mathcal{P}(E)$$

$$S(t_r \in T) = \bigcup_{t \in \{\forall x \in T : x \text{ is specialization of } t_r\}} M'(t) \tag{12}$$

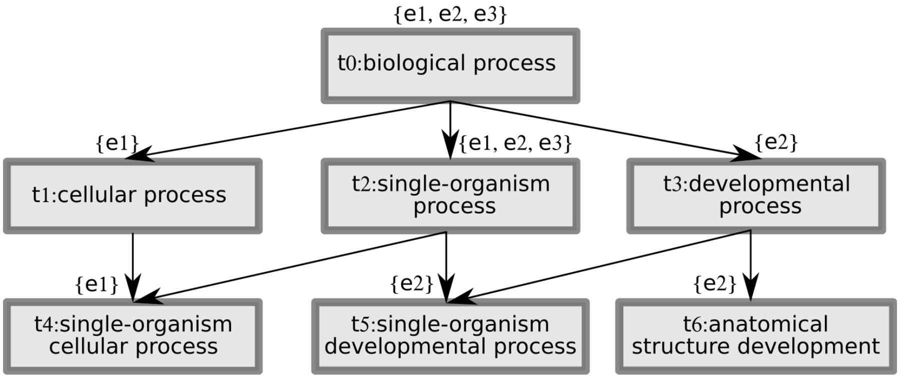

{e1, e2, e3}
t0:biological process
{e1}
{e1, e2, e3}
{e2}
t1:cellular process
t2:single-organism process
t3:developmental process
{e1}
{e2}
{e2}
t4:single-organism cellular process
t5:single-organism developmental process
t6:anatomical structure development

Figure 9: An example of spreading information about associations between examples and terms over the whole ontology. The mapping S was applied on each term comes from Figure 8 which was defined previously in Examples 1 and 2.

Example 3. *Consider the ontology* O *from Example 1,* M *and* M' *from Example 2. We apply* S *to each term in* O*, i.e.*

$$S(t4) = M'(t4) = \{e1\}$$

$$S(t5) = M'(t5) = \{e2\}$$

$$S(t6) = M'(t6) = \{e2\}$$

$$S(t1) = M'(t1) \cup M'(t4) = \{e1\}$$

$$S(t2) = M'(t2) \cup M'(t4) \cup M'(t5) = \{e1, e2, e3\}$$

$$S(t3) = M'(t3) \cup M'(t5) \cup M'(t6) = \{e2\}$$

$$S(t0) = M'(t0) \cup M'(t1) \cup \cdots \cup M'(t6) = \{e1, e2, e3\}$$

The result of this operation is shown in Figure 9. In this example, the most general term $t0$ named as *biological process* is associated with all examples in E, i.e. $e1$, $e2$, and $e3$. Intuitively, this is an expected result since other terms (*cellular process*, *single-organism process*, *developmental process*, *single-organism cellular process*, *single-organism developmental process*, and *anatomical structure development*) are also a *biological process* thanks to the $\succeq^*$-relation. In other words, the most general term is associated with all examples that are associated with all specializations of that term.

Before we introduce a rule space and a corresponding form of rules, we formulate a cover operator Θ. The cover operator determines which examples satisfy a rule condition, and we say that the rule covers such examples. Formally, Θ is a mapping from a power set of terms T to a power set of examples E, i.e.

$$\Theta : \mathcal{P}(T) \rightarrow \mathcal{P}(E) \tag{13}$$

and is defined in the following form:

$$\Theta(T_r \subseteq T) = \bigcap_{t \in T_r} S(t). \tag{14}$$

The cover operator Θ takes a set of terms and returns all the examples concurrently annotated by all the terms from this set. The cover is applied after the annotation propagation step mentioned above.

Rule space $\mathcal{R} =< R, \succeq_r >$ is defined as a quasi-ordered set where R represents a set of rules and a binary relation $\succeq_r$ such that $\succeq_r \subseteq R \times R$ is reflexive and transitive. Further, the rule syntax is restricted on a conjunction of positive terms. For the sake of simplicity, we omit a propositional logic notation of rules and represent the conjunction of positive terms as a set of terms, i.e. propositional logic notation of rule $t1 \wedge t3 \wedge t4$ is represented as $\{t1, t3, t4\}$. Notice that conjunctive interpretation corresponds to the definition of cover operator as well.

Example 4. *Suppose the ontology* O *from Example 1, mapping* M *and* M' *from Example 2, and mapping* S *from Example 3. Let rules be* $R1 = \{t1, t2\}$, $R2 = \{t2\}$, *and* $R3 = \{t0\}$. *Then* $R1$ *covers* $\Theta(R1) = S(t1) \cap S(t2) = \{e1\} \cap \{e1, e2, e3\} = \{e1\}$. *We say that example* $e1$ *is covered by rule* $R1$. *Equivalently,* $R2$ *covers examples* $e1, e2, e3$ *and* $R3$ *covers the same set of examples as* $R2$.

The binary relation $\succeq_r$ is defined in the set of rules R as follows:

$$R_f1 \succeq_r R_f2 \iff \Theta(R_f2) \subseteq \Theta(R_f1) \tag{15}$$

where $R_f1, R_f2 \in R$.

Example 5. *Suppose the ontology* O *from Example 1, mapping* M *and* M' *from Example 2, mapping* S *from Example 3, and rules from Example 4. Then the relations between rules are as follows:* $R2 \succeq_r R1$, $R3 \succeq_r R1$, $R3 \succeq_r R2$ *and* $R2 \succeq_r R3$.

5.2 PROPOSED ALGORITHM

The algorithm proposed in this work induces a hypothesis from data in the form of a set of rules (conjunctions). To induce a hypothesis consisting of more rules we apply a covering algorithm that has its origin in the AQ family of algorithms [110] and it is also used in CN2. The covering algorithm consists of two steps: (1) induce a single rule from the current set of examples, (2) exclude the examples that are covered by this single rule from the current set of examples; these two steps are iteratively applied starting with the the set of all examples until all positive examples are covered or a certain number of induced rules is reached. This process is described in Algorithm 4 and that algorithm we refer to as *sem1R*. As an input, the following data are required: a set of positive E^+ and negative E^- examples, a set of ontologies $\mathcal{O}$, and a maximal size of the set of induced rules k. An output is a set of induced rules. An *induceSingleRule* function returns the best rule based on selected evaluation function. The function *induceSingleRule* is described in Algorithm 6, all evaluation functions can be found in the Evaluation Criteria section.

Contrary to CN2, the *sem1R* algorithm has the relations over terms that are explicitly specified in provided ontologies. Intuitively, if this

Algorithmus 4 : sem1R

```
input  : $E^+$, $E^-$, $\mathcal{O}$, k
output : $\mathbb{H}$ // hypothesis
$\mathbb{H} \leftarrow \emptyset$
foreach $i \in \{1, 2, \cdots, k\}$ do
    $newRule \leftarrow$ induceSingleRule($E^+$, $E^-$, $\mathcal{O}$, k)
    $E^+, E^- \leftarrow$ removeCoveredExamples($newRule$, $E^+$, $E^-$)
    $\mathbb{H} \leftarrow \mathbb{H} \cup newRule$
end
return $\mathbb{H}$
```

kind of knowledge were exploited, then we would expect some benefits during the process of inducing rules because the structure of terms is known. In this case, the main benefit is speeding up the process of inducing rules and removing obvious redundancy between the terms in rules. This was the main motivation for the following reduction procedures.

5.2.1 *Reduction Procedures*

In this section, we formulate two procedures that significantly reduce a rule space in comparison with the traditional rule learning methods such as CN2.

5.2.2 *Redundant Generalization*

This reduction method eliminates such terms occurring in a rule which are more general than any other term of the rule. Such terms in the rule do not affect a set of examples covered by the rule and consequently do not change its impact. Evidently, the set of covered examples is only affected by the most stringent sets of examples according to the mapping S.

Theorem 1. *Let* $R1$ *be a rule and suppose that term* $t1 \in R1$ *and a term* $t2 \in R1$ *where* $t1$ *is more general than* $t2$*. Then, the rule* $R1$ *covers an equal set of examples as a rule* $\overline{R1} = R1 \setminus \{t1\}$ *that does not contain* $t1$*:*

$$\Theta(\overline{R1}) = \Theta(R1)$$

and the rule $R1$ *is called a redundant generalization of* $\overline{R1}$*.*

Proof. For simplicity, we take into consideration only rules with cardinality 1. Given this, mapping S can be seen as a cover operator Θ because it only makes an intersection over all sets of examples according to S. Also, a rule of cardinality 1 will be denoted as a term because we do not want to distinguish the relations over the set of terms and the set of rules. In this case, the $\succeq$ relation over terms is equivalent to $\succeq_r$ relation over rules. This simplification does not lose generality.

In Eq. 12, we use the mapping S that finds all specializations for a term and afterwards, using mapping M', it unites all the examples assigned to these specializations into one set. This is done for each term in an ontology. Apparently, terms that are more specific cannot be associated with a higher number of examples than their more general counterparts. Concurrently, examples associated with a more specific term make a subset of examples associated with a more general term, written as $t1 \succeq t2 \Rightarrow S(t2) \subseteq S(t1)$ where $t1, t2 \in T$. Now, let rule $R1 = \{t1, t2\}$ consist of two terms such that $t1 \succeq t2$ and rule $\overline{R1} = \{t2\}$ consists of only term $t2$. Then $R1$ covers an equal set of examples as $\overline{R1}$. This equality is proven below.

$$\begin{gathered}\Theta(R1) = \Theta(\overline{R1})\\ S(t1) \cap S(t2) = S(t2)\\ \{e \in E : S(t2) \subseteq S(t1)\} = S(t2)\\ S(t2) = S(t2).\end{gathered}$$

□

Example 6. *Consider the ontology O from Example 1, mapping M and M' from Example 2, and mapping S from Example 3. Let rule $R1 = \{t0, t2\}$, term $t0$ is more general than $t2$ ($t0 \succeq t2$) and this rule covers examples $e1, e2, e3$ because $\Theta(R1) = \Theta(\{t0, t2\}) = S(t0) \cap S(t2) = \{e1, e2, e3\}$. Now, consider a rule $\overline{R1} = \{t2\}$ that also covers examples $e1, e2, e3$ since $\Theta(\overline{R1}) = S(t2) = \{e1, e2, e3\}$ and as we can see, term $t0$ occurring in the rule $R1$ does not influence a set of covered examples. Given this, rule $R1$ covers the same set of examples as rule $\overline{R1}$. For this reason, rule $R1$ is Redundant Generalization and rule $\overline{R1}$ is not Redundant Generalization.*

To achieve a non-Redundant Generalization rule, i.e. the rule where the relation $\succeq$ does not exist between any terms in the rule, we have to apply Redundant Generalization procedure until the relation $\succeq$ between terms in the rule has not been found. As we can see in Example 6, this reduction procedure decreases the cardinality (length) of the rules.

5.2.3 *Redundant Non-potential*

In the previous case, the Redundant Generalization method reduces a rule space as a result of its ability to decrease the cardinality of rules. Specifically, this reduction capability is applied to the refinement operator that gradually extends rules by adding new terms into them. Redundant Non-potential method can generate fewer candidate rules because terms that are in a relation with another term are not appended to the refined rule.

Contrary to the previous method, the Redundant Non-potential method does not utilize relations among terms to reduce a rule space but compares rules with each other and removes such rules that cannot reach a higher quality value than the current best rule has. The ability to recognize non-potential rules can be used for a direct reduction of rules in a rule space and also for eliminating a number of candidate rules in a process of rules refining. Firstly, we define two types of evaluation function: Q evaluating a quality of rule based on the number of covered/uncovered examples, and Q_p that evaluates a potentially maximum quality of rule that could possibly be achieved over its future refinements. Examples of Q functions are depicted in Eq. 23, 25, and 27. Corresponding Q_p functions are depicted in Eq. 24, 26, and 29. For the moment, we can say that Q_p function expresses an upper boundary of a rule quality. This upper bound can be reached when we know that rule refinements can only reduce the set of examples the rule covers. Then, the best potential refinement does not lose any positive examples from the current cover while ceasing to cover all the current negative examples. A Redundant Non-potential rule and all its more specific rules can be safely disregarded in the single rule induction process because there is a guarantee that these rules cannot exceed an upper boundary of the rule quality represented by Q_p.

To illustrate, consider an arbitrary rule R1 and its more specific rule R2 ($R1 \succeq_r R2$) which was created by refinement operator application. Given Eq. 15, R2 covers a subset of examples covered by R1 ($\Theta(R2) \subseteq \Theta(R1)$). Unfortunately, ACC or F1-score are not monotone functions, meaning that it is not guaranteed that R2 must always have a higher ACC or F1-score than R1. For this reason, R2 cannot be safely pruned from a rule space because it is not obvious whether other refinements of R2, which are more specific than R2, can potentially achieve a higher score than R1 even though R2 could have a worse score than R1. To prune the rule space safely, we maintain the upper bound of rule quality Q_p. Given this, if rule R2 (refinement of R1) has a lower Q_p value than R1's value of Q then R2 is a *Redundant Non-potential* and this rule, along with all its more specific extensions/refinements, can be safely pruned from a rule space.

Theorem 2. *Let $\mathcal{R} =< R, \succeq_r >$ be a quasi-ordered set representing a rule space, where $R = \{R1, R2, R_{best}\}$. Binary relation $\succeq_r$ is defined on R1 and R2 as $\succeq_r = \{(R1, R2)\}$ meaning that R2 is more specific than R1; relation of R_{best} is disregarded - may be arbitrary. If potential quality (Q_p) of the rule R1 is smaller than the quality Q of rule R_{best} then the rule R1 and all its potential more specific rules, i.e. R2, can be pruned from the set of rules R thus from the rule space $\mathcal{R}$. Then the rules R1 and R2 are called Redundant Non-potentials.*

Proof. First of all, suppose that a target class is represented by positive examples. Secondly, suppose an evaluation function whose highest value is returned when all positive examples and none of the negative examples are covered. An example of this function can be ACC or F1-score. Note, that ACC is given by equation $TP + TN/(TP + TN + FP + FN)$ (see the Evaluation Criteria section) and the reason, why we affect only TP and not TN, is simple. An example that is classified as TP has to be covered by a rule. On the other hand, an example classified as TN does not have to be covered by a rule. Since we focus on the target class, an arbitrary rule reaches a higher score if a new rule covers the same set of positive examples as a rule and does not cover any other negative example. □

Example 7. *Suppose the ontology* O *from Example 1, mapping* M *and* M' *from Example 2, mapping* S *from Example 3, and two rules* $R1 = \{t2\}$ *and* $R2 = \{t3\}$*. Furthermore, we define a set of positive examples* $E^+ = \{e1, e3\}$ *and a set of negative examples* $E^- = \{e2\}$*. Firstly, we evaluate the quality of the rules according to ACC measure (see Eq. 23)*

$$Q_{ACC}(R1) = \frac{TP + TN}{TP + TN + FP + FN} = \frac{2+0}{2+0+1+0} = \frac{2}{3} \tag{16}$$

$$Q_{ACC}(R2) = \frac{TP + TN}{TP + TN + FP + FN} = \frac{0+0}{0+0+1+2} = 0 \tag{17}$$

Now, we compute a potential quality score of $R2$ *(see Eq. 24):*

$$Q_{p\_ACC}(R2) = \frac{TP + TN + FP}{TP + TN + FP + FN} = \frac{0+0+1}{0+0+1+2} = \frac{1}{3} \tag{18}$$

Evidently, the potential quality of $R2$ *is smaller than the quality of* $R1$ *so we can exclude the rule* $R2$ *and all its more specific rules (e.g.* $\{t5, t6\}$*) from the rule space. Note that an example of how to compute evaluation measures can be found in the next section.*

To achieve the most effective pruning of rule space, we store a value of the highest quality rule that has been discovered during the learning process in $\mathbb{R}_{BEST\_SCORE}$ variable, see Algorithm 5. If the potential quality ($Q_p(R)$) of currently examined rule R is less than the $\mathbb{R}_{BEST\_SCORE}$, then the rule R and all its more specifics rules are *Redundant Non-potential* and can be excluded from a rule space.

5.3 EVALUATION CRITERIA

It is necessary to know the quality of each rule because the rule with the highest value is needed for the final hypothesis. In this case, we define three evaluation functions: accuracy (ACC), F1-score (F1), area under the ROC curve (AUC), and their adjusted versions for evaluating the potentially best results that the current rule can achieve after refinements in future evaluations. Accuracy works well for balanced problems (the number of positive examples is similar to the number of negative ones) and both classes are equally important. F1 and AUC help when dealing with imbalanced classes, F1 puts more emphasis on the positive class.

First of all, we define four elements of confusion matrix: number of true positives (TP), number of false positives (FP), number of false negatives (FN), and number of true negatives (TN) examples that are covered by an arbitrary rule R, see Figure 10.

TP is given as a cardinality of the intersection of two sets, a set of examples that are covered by the rule R and a set of positive examples E^+. FP is given as a cardinality of the intersection of two sets, a set of examples that are covered by the rule R and a set of negative examples E^-. TN is given as a cardinality of the subtraction of two set, a set of

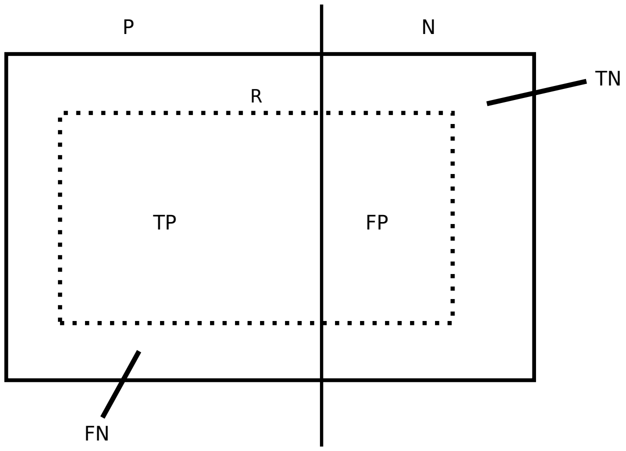


Figure 10: A graph representing a set of positive examples P and negative examples N and the way they are covered by a rule R assuming that R is focused on the classification of positive examples. Subspaces corresponding to TP, FP, FN and TN examples are also depicted.

negative examples E^- and a set of examples that are covered by the rule R. Finally, FN is given as a cardinality of subtraction of two sets, a set of positive examples E^+ and a set of examples that are covered by the rule R. All equations are shown below.

$$TP = |\Theta(R) \cap E^+| \tag{19}$$

$$FP = |\Theta(R) \cap E^-| \tag{20}$$

$$TN = |E^- \backslash \Theta(R)| \tag{21}$$

$$FN = |E^+ \backslash \Theta(R)| \tag{22}$$

Corresponding accuracy (ACC) of an arbitrary rule R can be computed by the widely known equation below:

$$Q_{ACC}(R) = \frac{TP + TN}{TP + TN + FP + FN}. \tag{23}$$

However, the potentially highest accuracy of rule refined from R is computed differently. In Eq. 23, we see that the eventual accuracy is given by the numerator (TP and TN) whereas the denominator has the normalization function. The refinement may improve the rule quality in such a way that the examples that are classified as FP will be re-classified to TN, i.e. the numerator of Q_{p_ACC} may at best be given by the sum of TN,TP, and FP. The equation for the potentially highest quality reached through refinement follows:

$$Q_{p\_ACC}(R) = \frac{TP + TN + FP}{TP + TN + FP + FN} \tag{24}$$

The computation of Q_{p_ACC} in Eq. 24 assumes that the rule R aims to cover positive examples rather than negative ones. In other words, examples that are covered by the rule R are classified as positive. Secondly, we propose another evaluation measure that is based on F1-

score that implicitly does not take into account the number of TNs. Its common form is depicted in Eq. 25.

$$Q_{F1}(R) = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{25}$$

The corresponding version of potentially best accurate rule created by applying refinement operator to rule R that is based on the $F1$ measure takes the following form:

$$Q_{p\_F1}(R) = \frac{2 \times TP}{2 \times TP + FN}, \tag{26}$$

where all negative examples covered by rule R (FP) are excluded from the denominator in comparison with Eq. 25. Since there is still the possibility of finding such a rule which covers all examples determined as TP and none of the FPs.

Example 8. *Consider the ontology O and mappings M, M', S from Example 1, and a set of positive (E^+) and negative (E^-) examples from Example 7. Furthermore, we define a rule $R = \{t2\}$. First of all, we find examples that are covered by the rule using Θ operator, i.e. $\Theta(\{t2\}) = S(t2) = \{e1, e2, e3\}$. Secondly, we compute TP, FP, FN and TN:*

$$TP = |\Theta(r) \cap E^+| = |\{e1, e2, e3\} \cap \{e1, e3\}| = 2$$

$$FP = |\Theta(r) \cap E^-| = |\{e1, e2, e3\} \cap \{e2\}| = 1$$

$$TN = |E^- \backslash \Theta(r)| = |\{e2\} \cap \{e1, e2, e3\}| = 0$$

$$FN = |E^+ \backslash \Theta(r)| = |\{e1, e3\} \cap \{e1, e2, e3\}| = 0$$

Finally, we substitute these numbers in Eq. 23 and 24:

$$Q_{ACC}(R) = \frac{TP + TN}{TP + TN + FP + FN} = \frac{2 + 0}{2 + 0 + 1 + 0} = \frac{2}{3}$$

$$Q_{p\_ACC}(R) = \frac{TP + TN + FP}{TP + TN + FP + FN} = \frac{0 + 0 + 1}{0 + 0 + 1 + 2} = \frac{1}{3}$$

The final ACC of rule R over the set of positive and negative examples is $\frac{2}{3}$ and the potential best ACC for the set rule and the set of examples is $\frac{1}{3}$.

Finally, let us give the rule quality in terms of AUC. The area under the curve can be computed easily. Since only the single rule is taken into consideration, its quality is determined by a single point in the ROC plot and it can be computed as a sum of areas of two triangles and one rectangle using an Eq. 27.

$$Q_{AUC}(R) = FPR \times TPR + (1 - FPR) \times TPR + \frac{(1 - FPR) \times (1 - TPR)}{2} \tag{27}$$

TPR (true positive rate) and FPR (false positive rate) are calculated as follows:

$$TPR = \frac{TP}{TP + FN}, FPR = \frac{FP}{FP + TN} \tag{28}$$

$$Q_{p\_AUC}(R) = TPR + \frac{(1 - TPR)}{2} \tag{29}$$

The adjusted version of AUC computing a potentially best AUC that a rule can achieve is shown in Eq. 29. In contrast to Eq. 27, Q_{p_AUC} supposes that FPR goes to zero.

5.3.1 *Feature Construction*

In the Problem definition section, we defined the rule space $\mathcal{R}$ as a quasi-ordered set that is expressed as a pair of a set of rules and the relation $\succeq_r$ between rules. In addition, the form of rules is determined by propositional logic; more precisely, the rule is restricted to a conjunction of positive terms, i.e.

$$R = t1 \wedge t2 = \{t1, t2\}, t1, t2 \in O.$$

The first step in the rule learning process is feature construction because rule learning employs features as their basic building blocks. In this work, features are constructed trivially from a set of terms T which comes from the ontology O where each ontology term corresponds to one feature.

5.3.2 *Feature Selection*

Oftentimes, a constructed feature set is extremely large and also redundant since it contains many features that are not associated with any example. For this reason, a feature selection method is highly recommended. Given this, we propose three various feature selection methods.

5.3.2.1 *FS_atLeastOne*

The first feature selection method excludes such terms from a constructed feature set which are not associated with at least one example from a set $E^+ \cup E^-$. In other words, this feature selection method removes such terms that are highly specific or do not cover any example. This method guarantees that removed terms cannot positively affect the final evaluation score of a rule because these terms cover an empty set of examples. For this reason, if such terms appeared in a rule then the rule would cover an empty set of examples.

5.3.2.2 *FS_onlySig*

The second feature selection method preserves only features whose terms are significant. P-values are calculated using a Likelihood Ratio Statistic (LRS) as is presented in [26]. The LRS for the two-class problem measures differences between two distributions: the positive and negative class probability distribution within the set of covered examples and the distribution over the whole example set. It is computed as follows:

$$LRS(r) = 2 \times \left(TP \times \log_2 \frac{\frac{TP}{TP+TN}}{\frac{TP+FN}{|E|}} + TN \times \log_2 \frac{\frac{TN}{TP+TN}}{\frac{FP+TN}{|E|}} \right) \tag{30}$$

This measure is distributed approximately as χ^2 distribution with 1 degree of freedom for two classes. If the LRS is above the specific significance threshold, then the term is considered significant.

5.3.2.3 *FS_sigAtLeastOne*

The third feature selection method combines the two previous feature selection methods. A term that belongs to the feature set has to satisfy two conditions: 1) that term covers at least one example, and 2) the term is significant which is calculated by the LRS or the term is a generalization of some significant term. This method combines requirements from the previous two selection methods, its selectivity will be experimentally verified later.

5.3.3 *Rule Construction*

Rule construction is the second step which aims to find a rule that optimizes a given quality criterion in the search space of rules.

The description of the algorithm for single rule learning is depicted in Algorithm 5 where input is a set of positive examples E^+, a set of negative examples E^-, a set of ontologies $\mathcal{O}$, a function *buildMapping* that creates a link between the ontology and the set of examples E ($E = E^+ \cup E^-$), and a parameter k that represents the maximal length of induced rules. Note that this function is defined manually by a user. The first step in Algorithm 5 is to find all features. This operation is represented by the function *featureConstruction* at line 4 that assigns all terms from the set of ontologies $\mathcal{O}$ to a set of features $\mathbb{F}$. To remove irrelevant features from the set of features $\mathbb{F}$, we propose a function *featureSelection* at line 5. Here, three various feature selection methods are provided as we mentioned in the Feature Selection section, i.e. *FS_atLeastOne*, *FS_onlySig*, and *FS_sigAtLeastOne*.

The main part of this algorithm is presented in lines 8-24. In this while loop, candidate rules are gradually refined until the maximal length of the rule is reached (l variable represents the current length of rule) or there is nothing to refine, i.e. the algorithm did not create any new rule in the previous iteration. In the for loop (lines 11-21), new candidate rules are generated using the application of the refinement operator on the corresponding parental rules. The algorithm iterates over each rule that is presented in the set of rules $\mathbb{R}$. To this rule, we apply a new ontology-based refinement operator which is represented at line 12 by the function *refineRule* that uses the Redundant Generalization and Redundant Non-potential reduction procedures. Similar to the traditional CN2 refinement operator, the ontology-based refinement operator appends a feature to the refined rule. For example, in the case of a conjunction of terms $R = \{t1, t2, t3\}$, a new rule is created as the union of term $t4$ and terms in rule R, i.e. $R\_new = \{t1, t2, t3\} \cup \{t4\}$. A new refinement operator requires the following inputs: rule r to refine, a set of features $\mathbb{F}$, an ontology $\mathcal{O}$ for information about relationships, a score of the best rule $\mathbb{R}_{\text{BEST\_SCORE}}$ that has been discovered, a set of positive and negative examples E, and a mapping M' that represents a connection between ontologies and examples. The operator returns a set of all refined rules that are not Redundant Generalizations nor Redundant Non-potentials and assigns them to *newCandidates* set.

Algorithmus 5 : induceSingleRule

```
input  : $E^+$, $E^-$, $\mathcal{O}$, $k$
output : $\mathbb{R}_{BEST}$ // conjunction of selectors
$\mathbb{R}_{BEST} \leftarrow \emptyset$
$\mathbb{R}_{BEST\_SCORE} \leftarrow 0$, $l \leftarrow 0$
$\mathbb{M}' \leftarrow$ buildMapping($\mathcal{O}$, $E^+$, $E^-$)
$\mathbb{F} \leftarrow$ featureConstruction($\mathcal{O}$)
$\mathbb{F} \leftarrow$ featureSelection($\mathbb{F}$, $E^+$,$E^-$, $\mathcal{O}$, $M'$)
$\mathbb{R} \leftarrow \mathbb{F}$
// discover rules until stopConditions
while $\mathbb{R} \neq \emptyset$ and $l < k$ do
    $\mathbb{R}_{new} \leftarrow \emptyset$
    // Refine all rules in R
    foreach $r \in \mathbb{R}$ do
        $newCandidates \leftarrow$ refineRule($r$, $\mathbb{F}$,$\mathcal{O}$,
         $\mathbb{R}_{BEST\_SCORE}$, $E^+ \cup E^-$, $M'$)
        $\mathbb{R}_{new} \leftarrow \mathbb{R}_{new} \cup newCandidates$
        // Find the best rule
        foreach $nc \in newCandidates$ do
            $score \leftarrow$evaluateCandidate($nc$, $E^+$, $E^-$, $\mathcal{O}$, $M'$)
            if $score \geqslant \mathbb{R}_{BEST\_SCORE}$ AND isSignificant($nc$,
             $E^+$, $E^-$, $\mathcal{O}$, $M'$) then
                $\mathbb{R}_{BEST} \leftarrow nc$
                $\mathbb{R}_{BEST\_SCORE} \leftarrow score$
        end
    end
    $\mathbb{R} \leftarrow$filterRules($\mathbb{R}_{new}$)
    $l \leftarrow l + 1$ // increment the rule length by one
end
return $\mathbb{R}_{BEST}$
```

The *refineRule* function that is described in Algorithm 6 starts with an empty set $\mathbb{S}$ where a content of this set will be returned at the end of the function at line 10. The cycle from lines 3 to 6 appends every feature to the rule that should be refined. Up to this part, the algorithm is similar to the traditional refinement operator. However, all rules that are not *Redundant Generalization* are excluded from the set $\mathbb{S}$ using the ontology $\mathcal{O}$ that provides relationships among terms. This is done by calling a function *removeRedundGeneralizations* at line 8. The function *removeRedundNonPotentials* removes such rules that satisfy the definition of *Redundant Non-potential* rules. In this case, the function continuously checks the following: 1) $R \succeq_r \forall s \in \mathbb{S} \cup R$. This is true since each element s represents a rule that is created as a refinement of rule R. 2) For each s, if its potential quality $Q_p(s)$ is less than the quality $Q(\mathbb{R}_{BEST})$ then remove s and all its more specific rules from the set $\mathbb{S}$. In other words, all rules in $\mathbb{S}$ whose potential quality can be greater than the rule with the greatest quality $\mathbb{R}_{BEST}$ are preserved.

Algorithmus 6 : refineRule

```
input  : $r$, $F$, $\mathcal{O}$, $\mathbb{R}_{BEST\_SCORE}$, $E$, $M'$
output : $\mathbb{S}$ // set of refined rules
$\mathbb{S} \leftarrow \emptyset$
// Append all features to the rule
foreach $f \in F$ do
    $newRule \leftarrow r \cup f$
    $\mathbb{S} \leftarrow \mathbb{S} \cup newRule$
end
// Filter rules
$\mathbb{S} \leftarrow$ removeRedundGeneralizations($\mathbb{S}$, $\mathcal{O}$)
$\mathbb{S} \leftarrow$ removeRedundNonPotentials($\mathbb{S}$, $r$, $\mathcal{O}$, $\mathbb{R}_{BEST\_SCORE}$, $E$,
  $M'$)
return $\mathbb{S}$
```

All candidate rules that were generated in *refineRule* function are assigned to the set of new rules $\mathbb{R}_{new}$. In addition, all *newCandidates* are evaluated by the function *evaluateCandidate* and its corresponding quality score is compared to the rule with the highest quality stored in a $\mathbb{R}_{BEST\_SCORE}$. If such a compared rule has a better quality then this rule is assigned to the $\mathbb{R}_{BEST}$ variable and the score is stored in the $\mathbb{R}_{BEST\_SCORE}$ variable. Simultaneously, the rule has to be significant. To compute this significance, we use LRS as we did in feature selection.

At the end of the algorithm, the best rule of the all rules that have been discovered is returned. If the function *filterRules* at line 22 is omitted then the Algorithm 5 is called a *brute-force exhaustive search* that explores the whole search space and leads to a combinatorial explosion. For this reason, an appropriate heuristic should be provided for reducing the search space. In this work, we use Beam search that expands only the most promising rules based on the evaluation function. Other rules are disregarded.

5.4 RESULTS AND DISCUSSION

In this section, we propose an evaluation procedure that experimentally confirms the efficiency of the new ontology-based refinement operator using two reduction procedures: the *Redundant Generalization* and the *Redundant Non-potential*. The algorithm with the ontology-based operator is called *sem1R* and it is compared against the traditional refinement operator used in CN2, which does not exploit any external knowledge during the rule refining process. Here, it is called *exhaustive refinement*. The ability to reduce a search space is tested on three different datasets with three feature selection methods (*FS_atLeastOne*, *FS_onlySig*, and *FS_sigAtLeastOne*) and with three different evaluation functions (ACC, AUC, and F1-score). Observed parameters as a total number of explored rules, which must be refined to find the best rule, and also run times, were measured for the *sem1R* and *exhaustive refinement*. All presented algorithms are implemented in C++ and work with the Open Biological and Biomedical Ontology (OBO) format. Note that the algorithms require at least one ontology.

Because the proposed algorithm requires three inputs, we define their format as it is used in our R package. The datasets are represented as a two-dimensional binary matrix D with i rows, j columns, a set of row ontologies R, and a set of column ontologies C. The mapping M' is constructed such that each row and column is associated with a subset of ontology terms. This construction step has to be done manually by a user based on expert knowledge. In practice, it is necessary to have specific identifiers for rows and columns and these identifiers are associated with corresponding ontology terms. In gene expression analysis, such an identifier can be gene ID (e.g. FBgn for Drosophila melanogaster, ENSB for human or mouse musculus) for rows and sample ID (e.g. FBbt for anatomy compartments of Drosophila melanogaster or Experimental Factor Ontology for experiment metadata) for columns.

To transform a dataset from a two-dimensional binary matrix to the set of positive and negative examples, we design the following procedure. First of all, we suppose that each element of the matrix D represents one example. Then all matrix elements containing 1s are assigned to the set of positive examples E^+ and elements with 0s are assigned to the set of negative examples E^-. For a non-binary matrix D, binarization is necessary.

The first tested dataset comes from [12] and describes the gene expression of imaginal discs of Drosophila melanogaster (DISC) where rows of the dataset correspond to genes and columns correspond to samples. Note that this format is used for all tested datasets. Rows (genes) of DISC dataset are described by Gene ontology [3, 29] and KEGG BRITE database. Columns (samples) are described by Drosophila anatomy ontology (DAO) [31]. The second dataset called Dresden Ovary Table (DOT) [39, 74] describes gene expression and RNA localization in fly ovaries using Gene ontology, KEGG BRITE database, and an ontology provided by the authors is freely available at [39], respectively. Note that DOT and DISC are originally formed as a binary matrix. Last but not least, the third dataset was downloaded via Expression Atlas [129] where it is called *Strand-specific RNA-seq of nine mouse tissues*[109] (m2801) and using Gene ontology and Experimental Factor Ontology (EFO) [105]. For binarization, we set up cutoff to 0.5 TPM (Transcripts Per Kilobase Million) because it is presented as a default value in Expression Atlas and it maintains comparable numbers of positive and negative examples. If a value in the matrix is higher than 0.5 TPM then the value is set to 1 and the element is assigned as a positive example otherwise the value is 0 and the element goes to the set of negative examples E^-.

Also, it may be desirable to find descriptive rules only for predefined rows (genes) or columns (samples) that are relevant to specific research. Specifically, it can be significantly expressed genes in a treatment group against the control group. In this case, the matrix

D has only i_s rows corresponding to significantly expressed genes and j_t columns corresponding to samples belonging to the treatment group and j_c columns belong the control group. Here, each of the elements belonging to the treatment group is set up to 1 and is considered to be positive, others are 0 which means negatives. The total number of examples is $i_s \times j_t$ and $i_s \times j_c$ for positive and negative examples, respectively.

Basic statistics of tested datasets, as a number of rows and columns, a number of positive and negative examples, and a number of ontology terms for given ontologies, are depicted in Table 10. Because there are some terms that do not associate with any example and such terms are not good candidates to be a feature since they do not cover any example, the final feature sets can be given by one of the three feature selection methods mentioned in the Feature Selection section. The numbers of features that were used for each rule induction step are shown in Figure 11.

These experiments clearly confirm our presumptions, defined in the Feature Selection section, where we assumed that the most reducing feature selection method is *FS_onlySig*. On the other hand, the most benevolent or conservative method is *FS_atLeastOne*, which guarantees that any of the relevant features possibly positively affecting the quality score of the hypothesis will not be discarded from the feature set. The *FS_sigAtLeastOne* demonstrates a similar behavior to *FS_atLeastOne*. Concretely, the *FS_sigAtLeastOne* method produces a smaller feature set than *FS_atLeastOne*. However, the differences are not huge.

To avoid a combinatorial explosion problem in exploring the rule space, we use a Beam search which is represented by *filterRules* function in Algorithm 5. The width of the beam was set no higher than the 100 best rules, the rules are sorted according to their quality score calculated with one of the given evaluation functions. We decided to use this threshold, because greater beam widths result in huge run times in *exhaustive refinement*. Higher beam widths also increase memory requirements. At the same time, the ability of *sem1R* to reduce the search space and consequently reduce run time is obvious even below the beam width of 100. Theoretically, it is anticipated that the ability to reduce a search space grows with the beam width since there are potentially more rules to prune especially for *Redundant Non-potential* procedure.

Total run time and total number of explored rules were observed for rules with the maximum length of 10 because longer rules can be more difficult to interpret in real problems, especially in a biology domain. The total number of induced rules for each dataset was set to 10, for the same reason as previously mentioned. The final results of experiments as total run time in seconds and total number of explored rules are depicted in Table 11 for *sem1R* and in Table 12 for *exhaustive refinement*.

A graphical representation is shown in Figure 12 and Figure 13. The first one shows run times in logarithmic scale depending on the number of induced rules for *sem1R* (dashed line) and *exhaustive re-*

Dataset	Size	# of pos/neg examples	# of ontology terms
DOT	6,510 × 100	309,593/341,407	42,964 (GO)/32,488 (BRITE)/140 (DOT)
DISC	1,207 × 72	65,537/21,367	42,964 (GO)/32,488 (BRITE)/9,255 (DAO)
m2801	12,225 × 26	124,032/193,818	42,964 (GO)/18,786 (EFO)

Table 10: Statistics for DOT, DISC, and m2801 dataset.

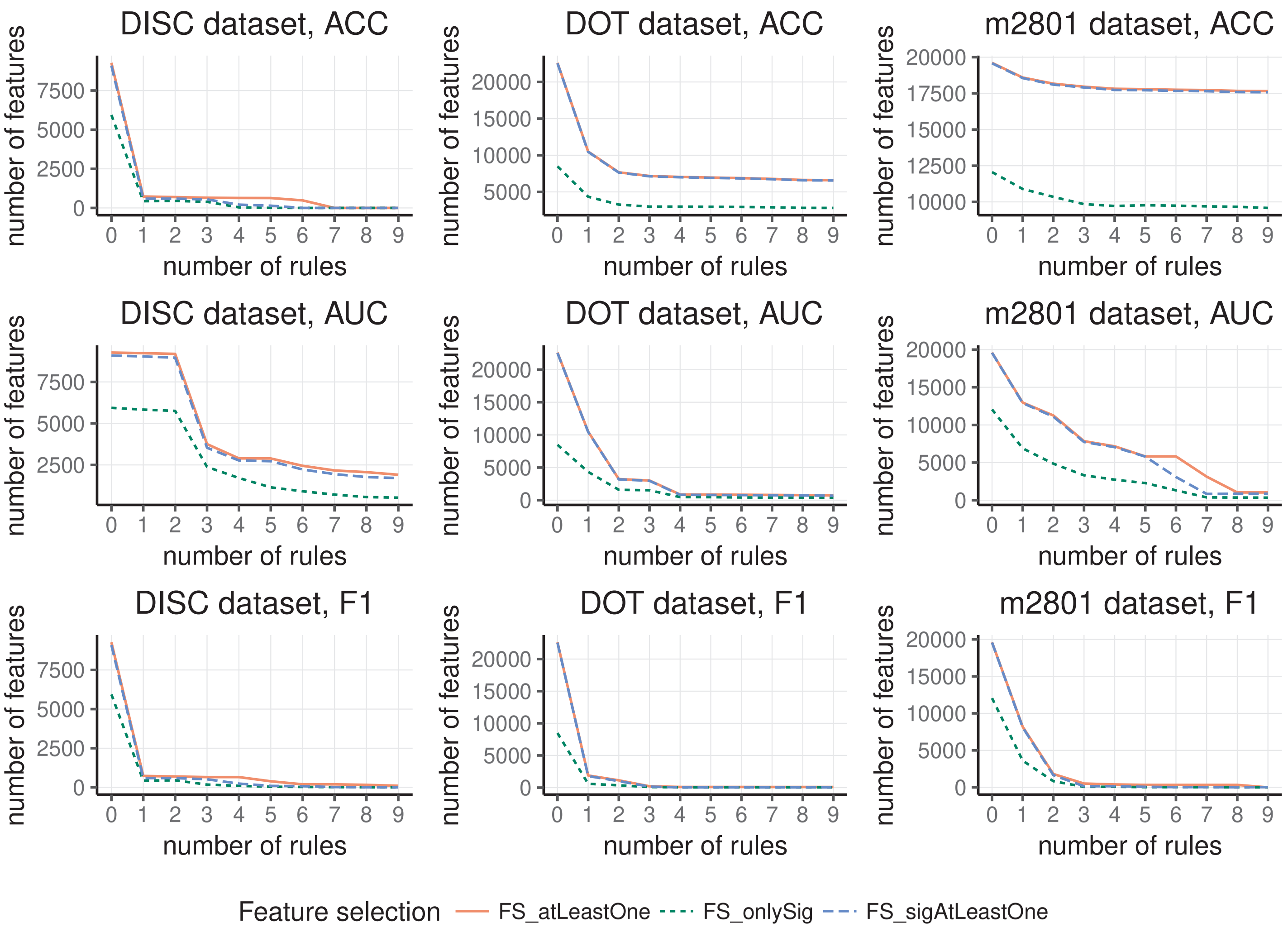


Figure 11: An average number of features across DISC, DOT, and m2801 datasets for three various feature selection methods *FS_atLeastOne*, *FS_onlySig*, and *FS_sigAtLeastOne*. These results were computed using three evaluation functions ACC, AUC, and F1-score.

Dataset	Feature selection	ACC score		F1 score		AUC score	
		total time	# of rules	total time	# of rules	total time	# of rules
DOT	*FS_atLeastOne*	303.636	107,964	22.381	26,460	142.302	52,638
	FS_onlySig	235.947	54,167	11.427	9,780	102.760	25,817
	FS_sigAtLeastOne	250.633	107,535	19.813	25,756	115.994	52,136
DISC	*FS_atLeastOne*	10.737	102,219	8.059	178,346	60.780	609,937
	FS_onlySig	1.777	87,671	1.109	7,223	33.304	67,558
	FS_sigAtLeastOne	1.955	13,041	1.330	11,270	25.861	91,003
m2801	*FS_atLeastOne*	699.273	461,745	28.087	80,079	168.210	225,081
	FS_onlySig	914.283	340,039	21.594	18,787	148.992	82,456
	FS_sigAtLeastOne	802.176	433,393	18.939	32,054	123.561	124,002

Table 11: Total runtime [s] and a total number of explored rules of *sem1R* algorithm for DOT, DISC, and m2801 dataset.

Dataset	Feature selection	ACC score		F1 score		AUC score	
		total time	# of rules	total time	# of rules	total time	# of rules
DOT	*FS_atLeastOne*	33,800.529	62,192,307	12,807.090	21,977,679	22,814.993	37,604,456
	FS_onlySig	15,849.761	30,991,466	5,049.444	8,075,976	9,042.203	15,672,577
	FS_sigAtLeastOne	33,638.549	61,912,986	12,743.265	21,674,132	22,726.459	37,176,099
DISC	*FS_atLeastOne*	996.587	10,681,537	881.007	11,017,701	2,214.819	38,874,717
	FS_onlySig	623.078	6,125,970	524.412	6,041,883	1,323.920	21,618,242
	FS_sigAtLeastOne	963.291	9,406,704	817.055	9,484,372	2,145.880	37,172,305
m2801	*FS_atLeastOne*	53,163.030	153,778,914	6,573.700	26,766,658	12,766.542	64,329,543
	FS_onlySig	29,641.080	86,150,004	3,873.421	14,168,195	6,741.368	29,298,233
	FS_sigAtLeastOne	53,019.570	153,255,327	6,431.049	25,255,830	12,391.710	59,322,805

Table 12: Total runtime [s] and the total number of explored rules of exhaustive refinement for DOT, DISC, and m2801 dataset.

finement (full line). Run time was measured for three datasets with three different evaluation functions and with three different feature selection methods. Evidently, in all cases, the run time of *sem1R* is significantly lower. Figure 13 shows the total number of rules that have been evaluated in a logarithmic scale that depends on the number of rules. As in the previous figure, the number of rules was measured for three datasets with three different evaluation functions and with three different feature selection methods. But even in this case, *sem1R* with its *Redundant Generalization* and *Redundant Non-potential* procedures prunes the rule space more rapidly in comparison with the traditional *exhaustive refinement*. Note that using *FS_onlySig* method, the smallest number of rules is evaluated. This corresponds to the results in Figure 11.

In all various experimental settings, both *exhaustive refinement* and *sem1R* induce rules with the same quality score across corresponding experiments. The level of significance was set to 99% for feature selection method *FS_onlySig* and *FS_sigAtLeastOne* and also the same significance level for finding the best rule in *induceSingleRule* function. From Figures 12 and 13 it is obvious that F1-score prunes the search space most and the run of the algorithm is fastest. One of the reasons is that only TP, FP, and FN must be calculated here. On the other hand, AUC is less strict in the pruning of the search space and it is also the slowest, because Eqs. 27, 28 and 29 have to be calculated for every candidate solution and the algorithm has to evaluate the highest number of candidate rules. There is a clear trade-off between the efficiency and complexity of evaluation that stands behind AUC. All results of the experiments are appended to Additional file 1 of the original article [103].

For illustration and better understanding, we present an example of 2-terms long rule induced from the DISC dataset, where each term comes from a different ontology. The rule is following: GO:0002181 AND FBbt:00000015. This reported rule is enriched (it covers far more positive examples than expected by random). The FBbt identifier refers to a term from Drosophila anatomy ontology and the GO identifier refers to a term from Gene ontology. In this particular case, the rule says that all genes that are associated with a cytoplasmic translation process (the chemical reactions and pathways resulting in the formation of a protein in the cytoplasm) tend to be over-represented in thorax segment of Drosophila melanogaster.

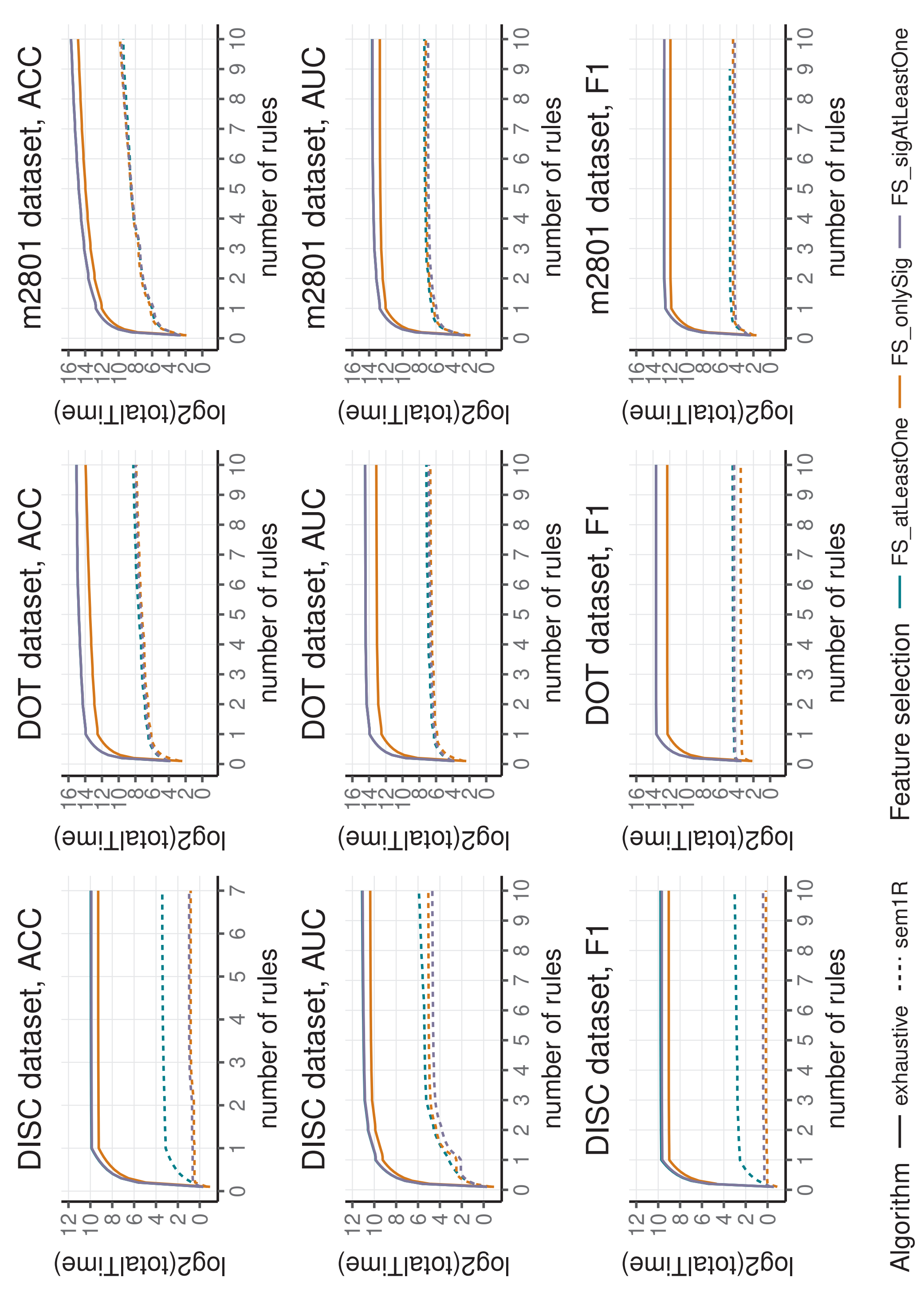


Figure 12: Total run time in logarithmic scale depending on the number of induced rules for three datasets (DISC, DOT, and m2801). ACC, AUC, and F1-score were used for evaluating the quality of rules and three feature selection methods (*FS_atLeastOne*, *FS_onlySig*, and *FS_sigAtLeastOne*) were applied before rule induction. Dashed line represents *sem1R*, full line represents *exhaustive refinement*.

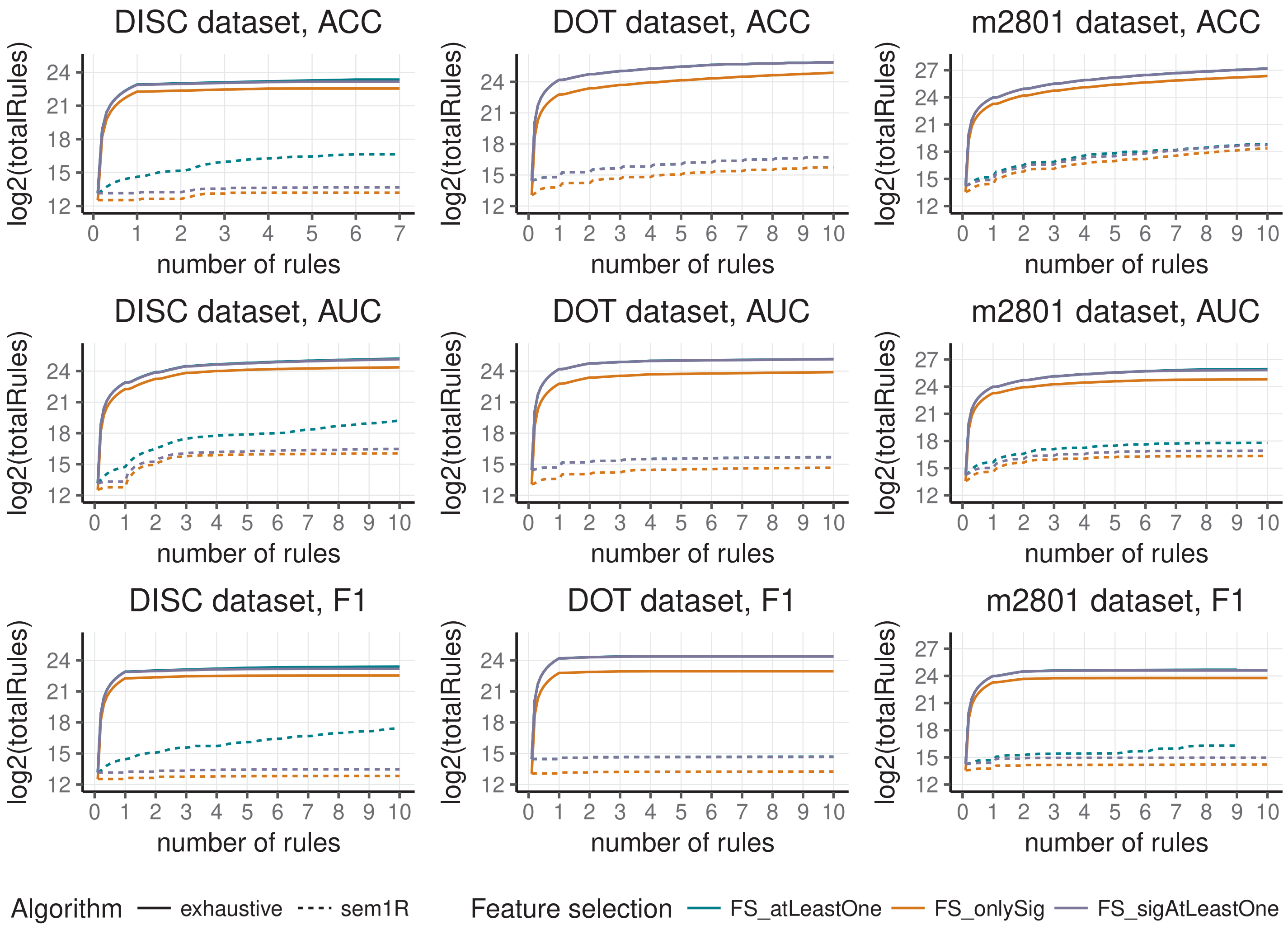


Figure 13: Total number of candidate rules in logarithmic scale depending on the number of induced rules for three datasets (DISC, DOT, and m2801). ACC, AUC, and F1-score were used for evaluating the quality of rules and three feature selection methods (*FS_atLeastOne*, *FS_onlySig*, and *FS_sigAtLeastOne*) were applied before rule induction. Dashed line represents *sem1R*, full line represents *exhaustive refinement*.

5.5 CONCLUSION

We proposed and implemented a new rule learning algorithm that induces a set of rules related to ontologies or taxonomies. Using two novel reduction procedures *Redundant Generalization* and *Redundant Non-potential*, which are part of the proposed ontology-based refinement operator, we dramatically reduce the search space. Consequently, runtime of the algorithm is decreased rapidly as well. These procedures guarantee that any removed rule cannot positively affect the quality of the final hypothesis. Also, three various feature selection methods that help to increase the efficiency of the algorithm were proposed. The algorithm is implemented in C++ and it is available at http://github.com/fmalinka/sem1r as R package. We demonstrated our algorithm on three real gene expression datasets, however, it is generally applicable to any learning task that combines measurements and ontologies, including metabolomics, etc.

6 SEMANTIC CLUSTERING ANALYSIS OF E3-UBIQUITIN LIGASES IN GASTROINTESTINAL TRACT

In this chapter, we accommodate the general framework of the semantic biclustering algorithm for finding tissue-specific gene expression patterns. The established protocol that incorporates the semantic biclustering algorithm exploiting the semantic of genes shows the potential to reveal interesting patterns in data. Simply put, this chapter describes a recipe to design and modify the specific data to run the *sem1R* algorithm and interpret the results.

Here, in contrast to the previous chapter, we denote the proposed approach as a *semantic clustering*, not the *semantic biclustering* as usual. The reason behind this decision comes from the fact that we control the sample dimension in this real experiment. Practically, we assign the samples to groups manually according to the requirements of biologists who interpret the data. Then, for each group, we find clusters of genes with respect to gene expression and their semantic similarity. This helps the biologists to recognize promising patterns in the data more easily.

As we show below, the *sem1R* is a practically applicable tool that formulates relevant biologically related hypotheses. In this chapter, the *sem1R* is being meaningful for studying redundancy of enzymes belonging to other families, like proteases or phosphatases. In comparison to the conventional GSEA method [156] which is oftentimes used as well, the *sem1R* easily defines rules/hypotheses using terms of various ontologies that extensive a hypothesis language and consequently oftentimes improve the predictive accuracy.

This chapter has been created with the cooperation of scientists from *Czech Centre for Phenogenomics*. I give my thanks to them. Note that this work is being considered for publication as [73].

6.1 BACKGROUND

Ubiquitination [48] is the most common post-translational protein modification, during which small protein ubiquitin (Ub) is covalently attached to the substrate. Ubiquitination can either direct proteins for degradation to the proteasome system or modulate their intracellular localization, vesicular trafficking, activation of signaling pathways and alteration of DNA transcription [54, 139]. The enzymes responsible for transferring ubiquitin to protein are called E3 Ub-ligases. To ensure high specificity during selection of target proteins there have been predicted more than 600 genes encoding E3-Ub ligases in human genome [98, 108]. They are divided into three basic classes, the

RING, HECT, and RBR according to the conserved domains and the mechanism of transfer of the Ub from E2 ubiquitin-conjugation enzyme to the substrate [10, 98, 117]. The E3 Ub-ligases are involved in all regulatory pathways in cellular signaling, physiology regulation and metabolism. Individual Ub-ligases recognize their targets in strictly regulated manner and without any respect to their sequence similarities. Depicting of regulatory roles of Ub-ligases within complex regulatory network can be hampered by strong parallel compensation mechanisms among Ub-ligases either recognizing the same substrate or affecting different nodes of same regulatory pathway [87, 139]. This makes it very hard to predict alternative compensating enzyme in reverse genetics approach. Thus, more functional classification of Ub-ligases is needed.

The way of classification of E3 ubiquitin ligases according to their function uses the Gene ontology [3], which describes three aspects of the biological domain as molecular function, cellular component, and biological process [108]. In addition, there are hundreds of other ontologies that do not specialize only onto genes and their properties. Such ontologies can describe developmental stages or influence of treatment and environment. The most popular method that employs this type of classification was GSEA. However, this type of analysis is limited by restriction to specific type of evaluation, and provides only a sorted list of genes together with their ontological annotation [103]. For this reason, semantic analysis methods were introduced in Chapters 4 and 5 that allow to determine and describe semantically comprehensive gene biclusters. So, these methods provide a more complete picture of functional gene classification for specific cell type in the tissue. We note that some differences between GSEA and the *semantic biclustering* methods have been already discussed in Chapter 3.

The gastrointestinal tract (GIT) is a system with high rate of regeneration. It consists of variety of diverse epithelial cell populations with different morphology and function, such as nutrients absorption, hormone production, barrier function, responding to microorganisms, coordination of immune response and self-renewing [140, 151]. Those features are determined by unique gene signature and regulatory pathway cooperation that is individual for specific cell type, and can be found in their RNA profile [67]. Therefore, GIT represents a valuable model system to study parallel regulatory networks in the context of tissue homeostasis, regeneration and response during pathogenic processes. In addition to different population of epithelial cells, stem cells and mucosa-associated lymphoid tissue can be found along the gastrointestinal tract [52, 142]. Tissue specific stem cells are of epithelial origin and they continuously divide, proliferate and differentiate to ensure the turnover of cells and the overall tissue homeostasis [7]. The multiple signaling pathways, such as Wnt, Notch or EphrB3, have been known to be critical for regulation of stem cell niche and differentiation of progeny cells [11, 151]. However, little is known about how ubiquitin ligases are involved in such physiological regulatory processes despite increasing evidence that an aberrant function or dysregulation of the expression of the E3 Ub-ligases can cause pathological changes resulting in dysplasia, metaplasia or even cancer.

Thereby, in this chapter we aim to identify GIT specific Ub-ligases and their role in tissue homeostasis. Here we provide a semantic clustering method combined with the expression profiling of E3 Ub-ligases in stomach, small intestine and colon parts of gastrointestinal tract in order to specify dominant biological roles and their possible prediction for alternative compensation in different part of GIT and during tissue homeostasis and regeneration. Also, by using already published single-cell RNA sequencing data [30], we make an attempt to identify cell-specific Ub-ligases in colon. We demonstrate that the individual Ub-ligase may be typical for several cell types, but its expression is determined by the tissue homeostasis status and could differ during injury response or regeneration.

6.2 METHODS

6.2.1 *Animals*

For this chapter were used C57BL/6NCrl mice (Charles River Laboratories). For the expressional profiling were used three 12-week-old C57BL/6NCrl males. Stomach, small intestine and colon were dissected and immediately proceeded for RNA isolation.

We refer readers to the original article [73] for a detailed review of data preparation methods.

6.2.2 *Statistical analysis*

qPCR data were normalized on Hsp gene expression. Missing data were replaced by maximum value +2 for a given gene, recalculated to relative quantities and log transformed. The ANOVA test with Tukey post test was used for analyzing different gene expression in different GIT parts. As significance level we used p-value = 0.01. Comparison of DSS treated and untreated distal colon was not performed due to small sample size. As primary criterion for selection potential interesting genes, the absolute difference higher than 1.25 delta Cq was used and all values from one had to be higher/smaller compare to any value from the second group. Fisher test was used for comparison of category data (distribution of ontology terms in different tissue and structural groups).

6.2.3 *Ontology and semantic analysis*

Ontologies that were used in all experiments are the following: Gene ontology [3], Pathway ontology [128], and KEGG Brite database [77–79]. These ontologies contained 45044, 2601, and 63263 ontological terms, respectively. Entire gene set of 370 genes was split into three groups according to the samples as follows: Small intestine vs colon -

Group A, stomach vs colon Group B, and stomach vs small intestine Group C. Then, the enrichment score (statistical significance) of each ontological term of all presented ontologies was calculated for each group of genes (Group A, Group B, Group C) separately. For this analysis, the *sem1R* package [103] was used with *computeTermsEnrichment* function. The results are presented in the original article [73].

6.2.4 *Semantic cluster analysis*

To perform a semantic cluster analysis, *sem1R* algorithm was used in this chapter. The algorithm induces a set of predictive rules that describe coherent biclusters using ontology terms from input data. In this case, the input data mean a gene set of significant and non-significant differentially expressed genes for each comparison (Small intestine vs colon - Group A, stomach vs colon Group B and stomach vs small intestine Group C), and a set of ontologies. Each rule was formulated as a conjunction of ontology terms where a group of genes covered by the rule had to be associated with all ontology terms appearing in that rule.

The concept of semantic cluster analysis is illustrated in Figure 14. The figure shows a process of inducing hypotheses for each set of significantly and non-significantly expressed genes of the original qPCR dataset that is divided into three groups of samples, i.e. Group A, Group B, and Group C.

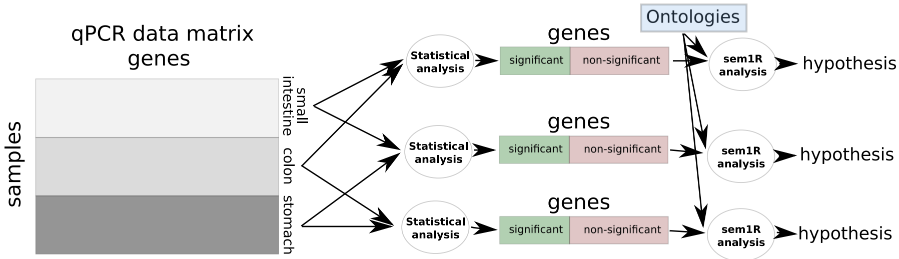


Figure 14: A scheme of semantic cluster analysis using *sem1R* algorithm.

6.2.5 *Selection rules definition*

The selected groups of genes were sorted according to the t-score and number of differences between significant and non-significant differentially expressed genes (minimum difference was set up arbitrarily equal to 3) for each ontology level. For each group (Group A, Group B, and Group C) we run *sem1R* algorithm that is restricted to find maximum 10 best rules (groups of genes) according to an evaluation function. To get more different rules and consequently more different covered groups of genes, all supported evaluation functions (ACC, AUC, and F1-score) were used in the process of rule learning. To control a level of specificity of rules, 'minLevel' parameter was set up to 0, 2, 3, 4, 5, and 6 for all runs of *sem1R* algorithm. Defining a minimal level of specificity prevents to induce too general or too specific rules that cover too many or too few genes, respectively. From all of these runs of various settings, interesting rules and corresponding groups of genes were selected.

6.3 RESULTS

6.3.1 *Organ-specific gene combinations*

The expression profiling of E3-Ub ligases was performed in stomach, small intestine and colon of WT mice, respectively. We found that each organ has their specific set of up- and downregulated genes (Figure 15), suggesting their organ specific role. For further analysis, the genes were divided into three groups. Colon upregulated 118 genes (sum of genes which were upregulated in colon over intestine or stomach), intestine upregulated 22 (sum of genes which were upregulated in intestine compared to colon or stomach) and stomach upregulated 78 (sum of genes which were upregulated in stomach compared to colon or intestine). No significant difference was found in the representation of individual structural classes with p-value 0.736 for upregulated genes in each organ. Genes from this cluster expressed at the same level in stomach, small intestine and colon and might have the same functional activity for each organ.

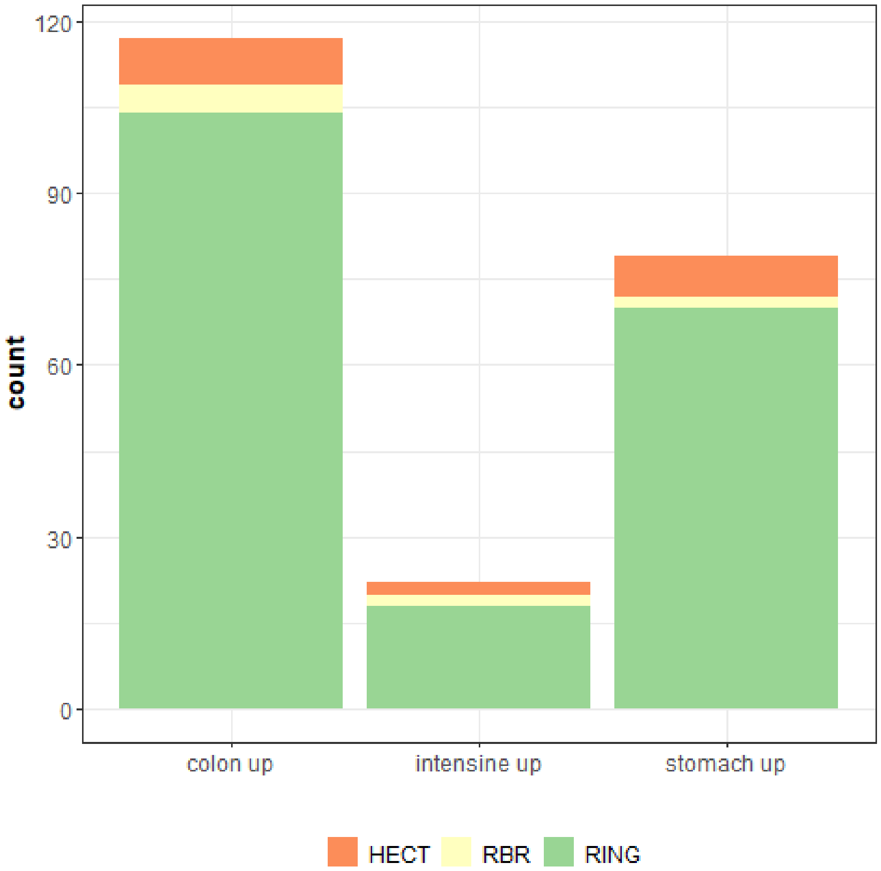


Figure 15: Representative distribution of upregulated genes in stomach, small intestine and colon divided into main Ub-ligase classes.

In the next step, we found ontology terms for each gene and compared distribution of ontology terms with theoretical distribution. We found 26 significantly enriched terms for genes, which were differentially expressed in the specific part of GIT [73]. These ontology terms displayed specific functions of given genes.

Ontology clusters of stomach represent genes that are involved in stress response by regulating various intracellular signal transduction with association of SCF ubiquitin ligase [166]. Ontology groups displays that small intestine is mostly represented by genes playing roles in immune and inflammatory response. This group is represented by the suppressor of cytokine signaling (SOCS) family of protein encoded genes – Socs1 and Socs3. Those genes responsible for negative regulation of cytokine signaling through the JAK/STAT3 pathway and was mentioned as a probable substrate recognition component of a SCF-like ECS E3 ubiquitin-protein ligase complex [49, 76]. Next, there was a group of upregulated genes in small intestine (14% out of all upregulated) which are responsible for negative regulation of insulin receptor signaling pathway [73]. The most representative gene for this group was Cish, which is also a member of SOCS family [21].

As for colon, ontology clustering of colon upregulated genes showing us enrichment for DNA repair, apoptosis and catabolic processes specific genes. For instance, upregulation of E3 ubiquitin-protein ligase Trim62 works as positive regulation in I-kappaB kinase/NF-kappaB signaling and DNA-binding of transcription factors (for more details, see [73]) [167]. Mul1 and Trim13 (also known as Ret finger protein 2, RFP2), among others, take a role in a positive regulation of cell death modulating innate immune response against viruses [75].

By applying semantic ontology analysis, we were able to find the groups of genes which belonged to the same ontology cluster but which had unique tissue expression pattern. This kind of analysis allowed us to identify possible genes which can share similar function in parallel regulatory networks.

In few following paragraphs, several cases of the ontology term combinations will be shown: GO:0018193: peptidyl-amino acid modification and GO:0042326: negative regulation of phosphorylation that includes five genes: Socs4, Socs5, Cbl, Socs1, Socs3 (Figure 16 A, B). Inside this ontology, group genes Socs1, Socs3 are upregulated in small intestine and downregulated in colon, whereas genes Socs5 and Cbl exhibited the opposite expression. On the contrary, Socs4 does not show significant difference in expression for colon and small intestine.

Ontology term combination GO:0045309: protein phosphorylated amino acid binding and GO:0044267: cellular protein metabolic process unite Fbxw7, Nedd4, Btrc, Cblb and Socs3 genes (Figure 16 C, D). In this group, Socs3 is upregulated in small intestine and downregulated in colon, while genes Nedd4, Btrc, Cblb are characterized by the opposite expression pattern. Expression of Fbxw7 did not significantly differ between small intestine and colon. Interestingly, the same gene set (with additional Cbl gene, which is a paralog of Cblb) belongs to another ontology term combination GO:0045309:

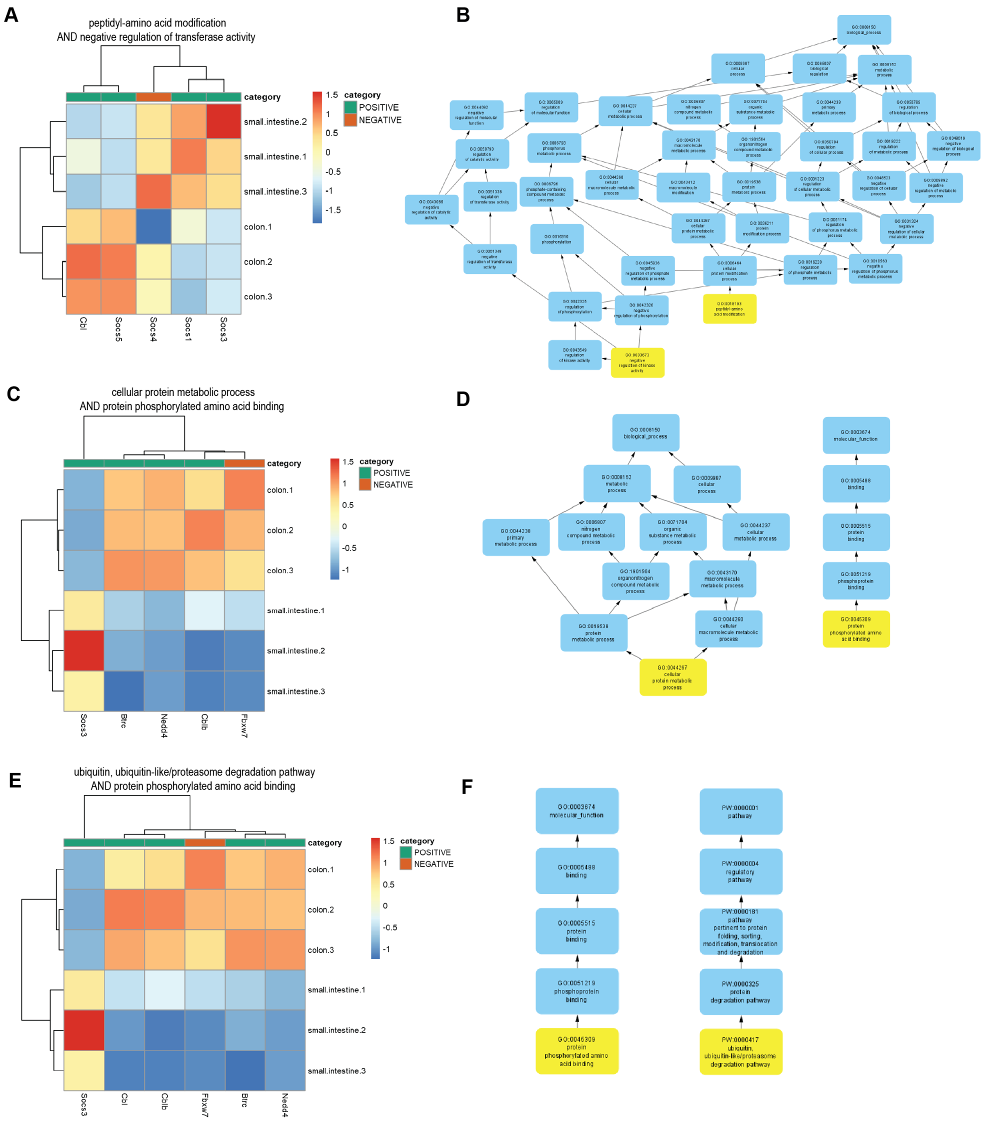


Figure 16: Selected examples of ontologically related genes in GIT. Heatmaps for genes that are annotated simultaneously by ontology terms GO:0018193: peptidyl-amino acid modification and GO:0042326: negative regulation of phosphorylation (A), GO:0045309: protein phosphorylated amino acid binding and GO:0044267: cellular protein metabolic process (C), and GO:0045309: protein phosphorylated amino acid binding and PW:0000417: ubiquitin, ubiquitin like/proteasome degradation pathway (E). Schematic visualization of gene ontology and their more general terms for GO:0018193 and GO:0042326 (B), GO:0045309 and GO:0044267 (D), GO:0045309 and PW:0000417 (F). Selected ontologies illustrate the ability of semantical clustering to group genes that are carry the same biological function in different parts of the organ. Scheme showing the relationships among ontology terms related to biological processes.

protein phosphorylated amino acid binding and PW:0000417: ubiquitin, ubiquitin-like/proteasome degradation pathway. For this ontology group they perform similar tissue expression pattern (Figure 16 E, F) representing that the same genes might share similar functions in multiple regulatory pathways.

Besides, Socs genes represented the most illustrative expression pattern in the GIT, particularly Socs1, Socs3, Socs4 and Socs5. Those genes appeared in 9 out of 10 ontology combinations. Thus, Socs5 was always downregulated in small intestine and upregulated in colon and stomach. Socs1 and Socs3 showed upregulation in small intestine and downregulation in colon. Also, Socs1 was downregulated in the stomach tissue, while Socs4 did not show any difference in expression between SI and colon, what indicates its equal contribution for homeostasis of these tissues.

6.3.2 *Epithelial damage in colon*

In order to reveal possible parallel networks, we used model of epithelial regeneration. We hypothesized that the genes involved in tissue regeneration might be masked by steady state homeostasis and their function thus might become apparent after tissue challenged conditions, such as epithelia inflammatory damage. For this purpose, we induced epithelial damage by treating mice with dextran sulfate sodium (DSS), a chemical compound that is widely used for mouse colitis models [42]. The expression site of 35 Ub-ligase genes was monitored in the DSS treated and untreated distal colon tissue. It was observed that most of the tested genes changed their expression pattern top-body-base along the colon crypt, which might in accordance with disruption of cell balance in the crypt after treatment (Figure 17 A). For untreated colon 22 out of 35 Ub-ligases were detected on the crypt top, whereas after DSS treatment only 12 genes remained at the same expression position and others – translocated either to the crypt base or spread over the crypt body (Figure 17 B-D). This could be associated with the damaged and missing cells at the apical site due to the treatment. Similar situation was observed for the genes that originally expressed in the crypt base.

As for the crypt body expression site, we observed more genes that had expression in the affected area, some of which showed strong signal (Cbl, Fbxl5, Rnf19b, Apc2) (Figure 17 B-E). This observation might be a result of inflammation and/or robust regeneration that occurs after treatment. Besides, there were genes with significantly decreased expression after treatment. Some of them kept their original intramucosal location of expression (for example, Trim25, Smurf2, Brcc3, Trim11), another – doesn't showed visible expression area (such as Bmi1, Asb11, March7, Btrc).

We further focused on the response of genes that were grouped into the same ontology term combination GO:0045309 and GO:0044267, GO:0045309 and PW:0000417 (for more details, see [73]). On the ontology combination table, see [73], Socs1 and Socs3 genes were downregulated for colon and Btrc, Bmi1, Cbl – upregulated. With the help of in situ hybridization we identified specific expression region in colon for each of these genes. Also we saw that expression region can differ under pathological conditions (Figure 17 E). In homeostasis Btrc was highly expressed by the cells of the crypt apex, but its

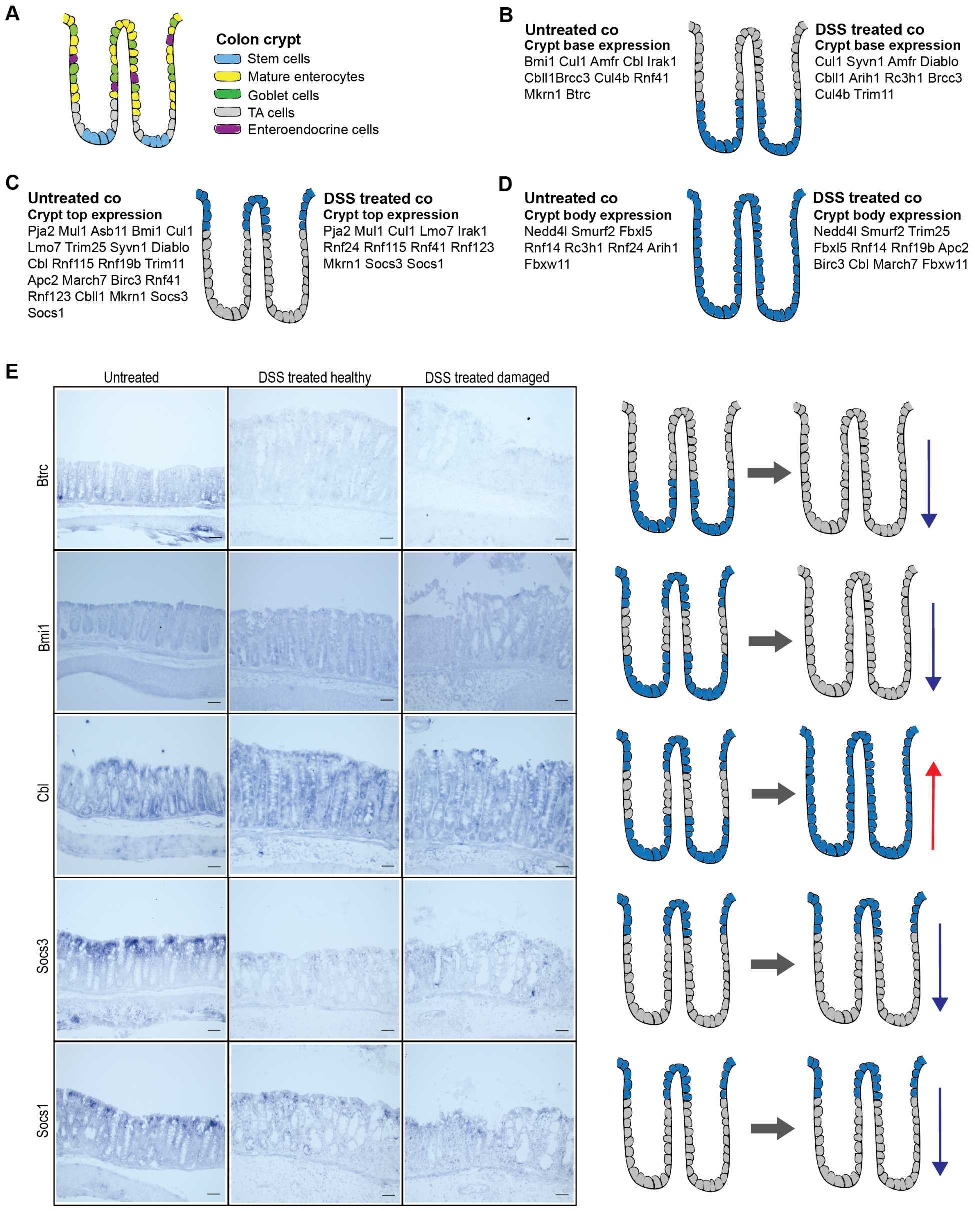


Figure 17: Differential expression of Ub-ligases of the same ontology group after induced epithelial damage. **A** Schematic illustration of the cell type distribution in colon in homeostasis (based on [30, 34, 57, 180]). **B-D** Epithelial damage causes a switch of ubiquitin ligases expression pattern in distal colon crypt in the top-body-base manner. e In situ hybridization images of the DSS treated and untreated colon demonstrate changes in expression of genes obtained from the ontology combination GO:0045309 and GO:0044267, GO:0045309 and PW:0000417. Scale = 50 um.

expression went remarkably down after injury. Similar situation was observed for Bmi1 originally present in the crypt top and crypt base, respectively. Socs1 and Socs3 are localized to the crypt apex. In the damaged or regenerated tissue, they keep the place of expression, but their expression level appears much lower because of either missing or re-structured epithelia (Figure 17 E).

On the contrary, Cbl showed high expression both at the crypt top and the crypt base but the DSS-induced damage significantly upregulated its expression through the entire crypt body. This could be explained by the potential communication of Cbl with signaling pathways maintaining stem/progenitor/mature cell balance during tissue regeneration (for example, protein tyrosine kinases mediated signaling) [115].

6.3.3 *Contemporary distribution in several cell types*

To determine cell type specific distribution of Ub-ligases in the colon, we used published single-cell RNA sequencing data of murine colon as a reference [30]. Only Ub-ligase related genes were chosen from the global scRNA-seq dataset (n=367) and were processed by Seurat package. For a cell subtype visualization, we performed principal component analysis (PCA), then the 10 most significant principal components were projected to two dimensions with UMAP, and the cells were colored by their classification label [16]. We used established cell markers to determine cell types in proximal and distal colon, including enterocytes (Krt20+, Slc26a3+) [34], goblet cells (Atoh1+, Spdef+) [64, 180], tuft cells (Dclk1+) [57], chromaffin (also known as enteroendocrine) cells (Chga+, Chgb+) [45], proliferating (Lgr5-, Mki67+) and non-proliferating (Lgr5+, Mki67-) stem cells (SCs) [7]. With the help of UMAP visualization we showed that Ub-ligases can be grouped into several clusters that demonstrate cell specificity (Figure 18 A). However, there is not very strict tissue specificity between distal and proximal parts of colon, and clusters there demonstrate some overlapping.

To see the more detailed Ub-ligase distribution throughout the colon, we focused on genes that were clustered into the same ontology combination groups GO:0018193 and GO:0042326, GO:0045309 and GO:0044267, GO:0045309 and PW:0000417 (for more details, see [73]). Those genes showed a differential expression after DSS-induced inflammation (Figure 17 E). Thus, Socs1 and Socs3 were mostly expressed by stem cells that clustered as Lgr5+ undifferentiated, Lgr5+ amplifying undifferentiated SCs and goblet cells both in proximal and distal colon, together with enterocyte cells of proximal colon (Figure 18 D, E). Besides this, Socs3 were typical for Lgr5- undifferentiated SCs cluster (Figure 18 E).

Gene Btrc was abundant in the clusters of goblet cells, Lgr5+ amplifying undifferentiated and Lgr5+ undifferentiated SCs (Figure 18 B). Also, it was spotted in chromaffin, enterocyte, and tuft cell clusters. As for Cbl, it displayed similar distribution in all cell clusters with the higher concentration in the enterocyte and Lgr5+ undifferentiated SCs clusters, respectively (Figure 18 C). Finally, Bmi1 showed the lowest cell cluster specificity with equal distribution through all clusters (Figure 18 F). These results illustrate the fact that Ub-ligases are not present solely in one given cell type, but they seem to be expressed by various cell types along the tissue. Yet, the discussed Ub-ligases could carry out different functions or could be expressed under specific conditions (as was observed after the colon injury).

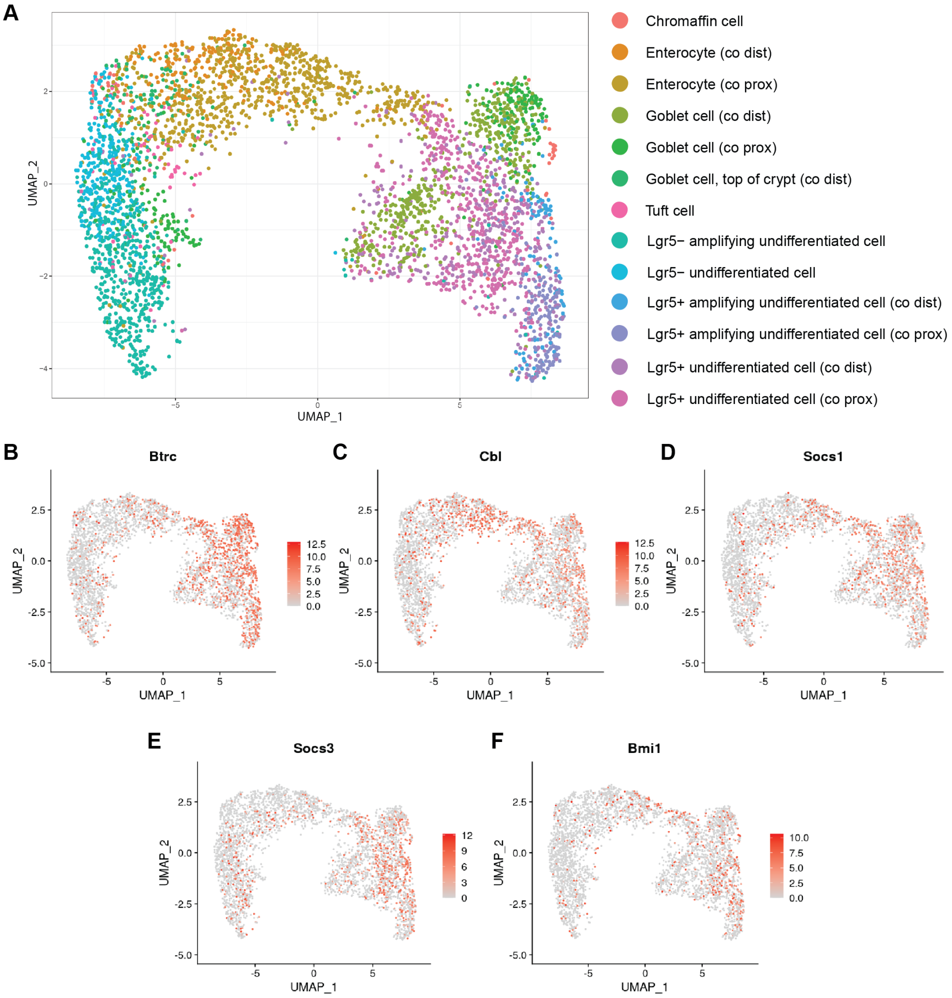


Figure 18: Distribution of Ub-ligases in colon. **A** UMAP analysis demonstrates that the colon Ub-ligases may be grouped into 13 cell specific clusters (labeled by colors). **B-F** Ub-ligases from the same ontology combination group displaying distributional expression between several cell types.

6.4 DISCUSSION

Up to date, there have been published many reports on E3 Ub-ligases based on in vitro investigation. It gives valuable data regarding cellular physiology and homeostasis such as proliferation, cell growth, apoptosis, nucleic acids maintenance, metabolism, cell cycle etc., with either overexpressed or absent E3 Ub-ligases [10, 98, 108]. However, contextual information about their effect on a complex tissue, organ and organism, including reciprocal regulations within subpopulation of cells is missing in such models. Therefore, studying E3 Ub-ligases in vivo gives more information about the biological role of these enzymes and their implementation in the physiology of the entire organism. Yet, in vivo models are subjected to strong regulatory mechanisms relying on compensatory effects of alternative pathways.

The ability of biological system to maintain homeostasis in the presence of mutations is described by the term genetic robustness. This feature is evolutionarily essential for the organism surviving in case of gene misfunction and can be achieved via regulatory pathways intercommunication [43, 44]. However, this could cause difficulties for researches to analyze the animal models, when gene targeting does not lead to the expected abundant or severe phenotype. After being first reported in Drosophila as transcriptional dosage compensation of X chromosome [118], genetic robustness was then described in many model organisms from yeast [60] to mammals [175]. To explain genetic robustness phenomenon, researchers proposed several mechanisms, such as functional redundancy of homologous genes [162], adaptive mutations [163], rewriting of genetic network [6], genetic compensation, and transcriptional adaptation [44].

To gain deeper understanding of genetic compensation, we proposed the usage of Semantic biclustering analysis (presented in Chapters 4 and 5) to statistically predict and describe semantically coherent gene biclusters in the context of functional gene classification for specific cell type in the tissue. In this manner, E3 ubiquitin ligases have been chosen as the testing model of our hypothesis. We compared expression of E3 ubiquitin ligases in three main segments of the gastrointestinal tract, i.e. in stomach, small intestine and colon. As first outcome, small intestine appears having all the ligases expressed at the lowest level. Knowing this, we used expression in small intestine as a reference level for stomach and colon for the ontology analyses dividing expressed genes according to their function in cells and tissues. These analyses revealed that small intestine is characterized by genes involved in maintenance of the immune system, and that genes playing roles in the catabolic processes are typical for colon. It has been discussed if compensatory activity of redundant genes may or may not correlate with their similarities in sequence or structure and in common origin [18]. These facts complicate compensatory pathways identification. Taking into account the theory above, by applying semantic clustering analysis we were able to reveal ten groups of Ub-ligases that share the same ontologies, but that carry the unique tissue expression pattern. Notably, the genes from the same ontology combination group were not described before as redundant what gives an interesting hint for detailed studying those genes together.

In order to test identified possible parallel networks in biological system, we used mouse model of epithelial regeneration. We hypothesized that genes involved in tissue regeneration might be masked by steady state homeostasis, but expose their function after tissue challenged conditions. Therefore, we induced epithelial damage by treating mice with DSS. We observed that epithelial damage in colon activated intracellular signaling transduction with activation of genes different from that functioning in homeostasis. This suggestion was also supported by our approach classifying Ub-ligases based on their cell specificity. We have not observed a strict cell specificity and tested Ub-ligases were found present in various cell types playing different roles. This observation refers to the ability of Ub-ligases to participate in regulation of several signaling pathways in specific clusters. Yet, such regulation can be significantly different depending on tissue type, developmental stage and homeostatic condition.

Taken together, the important outcome of our study was that semantic clustering analysis of GIT specific Ub-ligases allows us to statistically define compensatory genes clusters consisting of the same genes involved in the distinct regulatory pathways vs few different genes playing roles in the functionally similar signaling pathways. Such an approach could find potential application in the cancer therapy development as genetic redundancy has also been described during cancerogenesis. In this case redundant genes cover potential harmful effect of mutation and cancer progression depends on the effective functional setup between defective genes and their compensatory partners [18]. The most illustrative expression pattern in GIT semantic ontologies combinations showed members of Socs family. Besides their role in the immune response regulation as suppressors of cytokine signaling, some members of the Socs family were described to participate in tumor progression [59]. For instance, SOCS1 downregulation was described in hepatocellular carcinoma [181], cervical [83], ovarian and breast cancer [158]. Aberrant expression of SOCS1 and SOCS3 has been described in human colorectal cancer, when SOCS3 overexpression inhibits proliferation, migration and invasiveness of tumor cells [24], while SOCS1 overexpression has pro-oncogenic activity [165]. In this manner, it would be meaningful to further study Socs genes together with other genes from the same ontology group in terms of compensatory behavior during cancerogenesis and other GIT diseases progression.

Having obtained our overview of Ub-ligases clustering, it seems to be meaningful to apply semantic clustering approach for studying redundancy of enzymes belonging to other families, like proteases, phosphatases, kinases etc. Their important biological roles indirectly suggest their high compensatory potential. Operating with the knowledge of ontology relationship among genes will help to choose the relevant animal model for study of a particular disease and future therapy development.

6.5 CONCLUSION

The aim of this chapter was to explore gastrointestinal tract specific Ub-ligases, define their dominant biological roles at homeostasis and possible contribution to alternative compensatory networks. By applying improved ontology-based clustering method *sem1R*, we performed Ub-ligases profiling and revealed ten ontology combination of Ub-ligases groups that potentially exhibit redundant features in GIT. The compensatory biological networks identified through testing showed that genes from the same ontology cluster alter their expression pattern after induced epithelial damage exposing their compensatory activity during tissue regeneration.

Besides the biological interpretations, we provide guidance of using the *sem1R* algorithm for this specific GIT experimental design. Hence, our previous research effort introduced in the previous chapters lead to the practical application in biology.

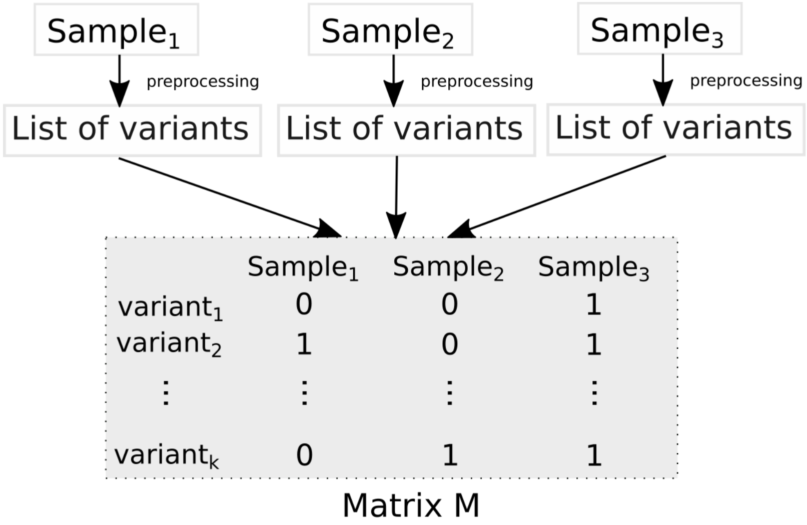
7 SEMANTIC BICLUSTERING FOR REVEALING PATHOGENIC LOW-FREQUENCY GENETIC VARIANTS IN A COHORT OF PATIENTS

An application of the semantic biclustering technique may not be necessarily restricted only to gene expression datasets. Another biological application may aim at genetic variant data. In this chapter, we will demonstrate suitability of semantic biclustering for finding potential disease-causing genetic variants inferred from an observed cohort of affected ophthalmological patients. The implementation of the concept rule learning algorithm with an ontology-based refinement operator, called *sem1R*, is potentially applicable for solving that task only with partial algorithmic adjustments. The ability of the algorithm to find relevant results will be discussed in this chapter. Moreover, we will describe the conceptual adaptations of the original *sem1R* algorithm that are necessary to make for solving such a specific task.

The idea of adapting the *sem1R* algorithm was initiated by our work presented in [40] where we focus on revealing genetic mutations that cause posterior polymorphous corneal dystrophy. Generally speaking, finding pathogenic genetic variants across individuals who share the same symptoms of the disease might be problematic from at least two aspects. Firstly, from a biological aspect, there is no only one biological model explaining disease risk in a population, various biological models are currently taken into consideration [61]. Secondly, from a computer scientist's point of view, searching in the space of hypotheses is time-consuming, especially for large cohorts. For this reason, we established a software tool that helps to find common pathogenic variants in a cohort of rare disease patients.

The proposed tool was developed for an analysis of all variants that were discovered in the process of variant calling and it might be generally useful for finding interesting mutations in any kind of disease. For the variant analysis, we were inspired by our previous work presented in Chapter 5. This algorithm was originally developed for finding hidden and nontrivial patterns in gene expression data; however, it might be utilized generally for any kind of binary classification problem. Only a binary matrix and at least one ontology is required for input. For adapting the original algorithm *sem1R* to the problem of finding common/shared rare genetic variants in an arbitrary genome, it is necessary to rebuild the required algorithm's input as a binary matrix, ontology, or any other hierarchically ordered structure in the proper data format.

BioBin algorithm [116] automates the process of binning low frequency variants for association testing. In contrast to BioBin, we introduce the algorithm which enables to form genomic boundaries using genomic locations and their mutual intersections as well. Furthermore, the proposed algorithm may perform gene enrichment analysis



$$\begin{array}{cccc} & \text{Sample}_1 & \text{Sample}_2 & \text{Sample}_3 \\ \text{variant}_1 & 0 & 0 & 1 \\ \text{variant}_2 & 1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \text{variant}_k & 0 & 1 & 1 \end{array}$$

Figure 19: A scheme showing the process of constructing the matrix M.

to reveal the enriched genomic regions of gene-based associations using Gene ontology or any other ontology like the biological Pathway ontology.

7.1 DATA MATRIX

Since *sem1R* utilizes two sets of examples for hypothesis induction, the proper format has to be defined first. All learning examples are decoded into a 2-dimensional binary matrix M, where rows represent preselected variants from an explored genome. Each variant is given by its genome coordinates, i.e., chromosome, start, and end position of variant. Columns of the matrix M represent samples/individuals who share the same disease or the same symptoms. The presence of a variant in the matrix M is expressed by one in the corresponding row and column, respectively. Otherwise, the absence of a variant in a sample is expressed by zero. Note that the matrix M contains all preselected variants that were discovered by any of *variant calling* methods together in the given samples/individuals.

Figure 19 shows a scheme of constructing the matrix M from three samples where each sample is firstly preprocessed by any of the available tools, see Section 7.6. Then, all variants are combined into one binary matrix.

Practically, M is the sparse matrix because the number of equivalent variants that match with each other across different samples is small. Generally, this we consider to be a problem of variant granularity; variants are generally too specific for their common presence across various samples/individuals. In practical applications, it is less probable that a specific rare variant would be spread over all samples in a large cohort of individuals. Besides biological reasons, one technical explanation of false positives or negatives might be the fact that the results of variant calling highly vary, among the other things, on input data quality, value of hyper-parameters, or type of pipeline. For this reason, we propose to extend the hypothesis language by arbitrary genomic elements on a different level of details. This extension brings an opportunity to cover more than one variant easily. For illustration, suppose only such elements that are defined according to gene coordinates. Then, only genes would be described and covered. Therefore, a common gene pattern might be found easily because a gene can be associated with many variants - many samples from the cohort of individuals can be covered. On the other hand, the pattern description might be too general - many other samples from the control group are covered as well.

Another example, the extreme case from the other side, is considering chromosomes, the top elements in the genome hierarchy, as the only genome regions of the hypothesis language. Then, the matrix M will be formed by the same number of rows as how many chromosomes the given samples contain. In that case, one will be assigned to the element of the matrix M if at least one variant appears in the corresponding chromosome. However, this information does not bring new nontrivial knowledge. The probability that at least one variant occurs in a chromosome is thus extremely high. Therefore, this considering is useless in results interpretation.

Various genomic regions and their generalizations/specializations relationships can be represented by an ontology as the set of partially-ordered elements. An example of a small subset of genomic elements and their relationships among them is depicted in Figure 20. The ontology was constructed from the real GFF file of the human genome, restricted to chromosome 10.

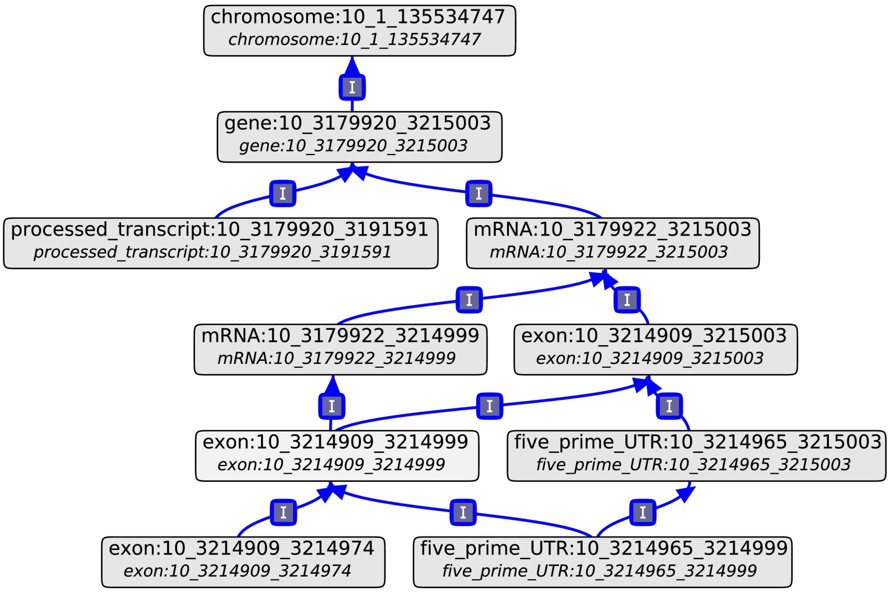


Figure 20: An example of ontology constructed from the real GFF file showing relationships among various genomic elements.

7.2 FORM OF HYPOTHESIS

For finding the most significant hypothesis covering a large number of examples from a data matrix, it is necessary to extend the hypoth-

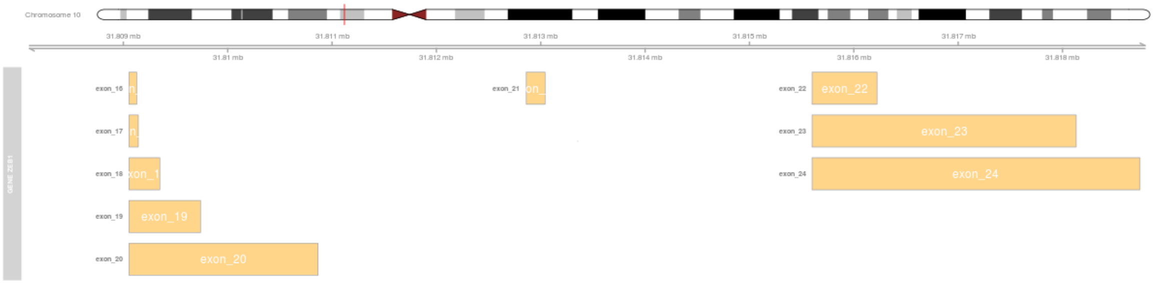


Figure 21: An example of selected genomic features of human gene ZEB1. All of shown features are various exons on chromosome 10.

esis language by some elements which can positively improve the generalization in the context of machine learning and making easier interpretation and further validation of the final hypothesis. As a good candidate for extending the hypothesis language, we considered a GFF3 (General Feature Format) file format that annotates all different features in the selected genome. These genomic features are described by their type (e.g., gene, transcript, exon, miRNA, etc.) and coordinate. The coordinates are basically expressed by their chromosome number, start, and end positions. In addition, all of these features can be ordered based on a subinterval relation, denoted as $\sqsubseteq_{SINT}$. This binary relation is reflexive, antisymmetric, and transitive. Formally $\sqsubseteq_{SINT}$ is defined as follows: let a genomic feature a be $a = (ch_1, s_1, e_1) \in CH \times \mathbb{N}^+ \times \mathbb{N}^+$ and a genomic feature b be $b = (ch_2, s_2, e_2) \in CH \times \mathbb{N}^+ \times \mathbb{N}^+$ where CH is a set of chromosome identifiers (i.e. chr1, chr2, etc. for human genome GRCh37). s_1, s_2 represent the start position of the feature, and finally e_1, e_2 represent the end position of the feature. Then, $a \sqsubseteq_{SINT} b$ if and only if $ch_1 = ch_2 \wedge s_1 \geqslant s_2 \wedge e_1 \leqslant e_2$. All presented genomic features F and the relation $\sqsubseteq_{SINT}$ correspond to the required definition for background knowledge given from *sem1R*. Since F and $\sqsubseteq_{SINT}$ are considered as the partial-ordered set $< F, \sqsubseteq_{SINT} >$. For this reason, the algorithm *sem1R* and both of the proposed reduction procedures can be applied to this kind of problem. An example of selected genomic features of human gene *ZEB1* (from human reference genome GRCh37) is shown in Figure 21.

Since hypothesis induction is the process of learning from examples, a set of examples has to be established from the given data matrix firstly. Here, the set of all examples counts $n \times m$ elements, where n/m is the total number of rows/columns of the matrix M, respectively. Elements containing ones are considered as positive examples. Otherwise, elements are considered as negative examples. Note that positive examples are referred also as a target class, because only such target class is described by the induced rule. In more detail, an example is a variant which is determined by its coordinates (chromosome, start and end position) in genome.

Both, the example and the genomic feature, use the same coordinate system (the same notation of position in an arbitrary genome) and therefore they can be compared to each other using the relation

$\sqsubseteq_{SINT}$. This property allows to formulate a cover operator easily. For sake of clarity, the cover operator says whether an example is or is not covered by a rule. This knowledge is essential for a rule quality evaluation. It is essential to determine false positives, false negatives, true positives, or true negatives in binary classification.

Example 9. *Suppose a hypothesis* H *containing a one-term rule* $r = \{(chr10, 31608151, 31816222)\}$ *and a set of examples* $E = \{e1, e2, e3\}$ *where* $e1 = (chr10, 31803516, 31803516)$, $e2 = (chr10, 31810012, 31810013)$, *and* $e3 = (chr16, 69973127, 69973127)$. *Then, the rule* r *covers the examples* $e1, e2$ *and does not cover* $e3$.

However, an one-term rule cannot explain more complicated patterns in data because the hypothesis language is limited by the predefined genomic features and theirs coordinates. Therefore, a more sophisticated form of rules is needed to bring new genomic regions into account. As a good trade-off between the complexity of a rule's form and its interpretability, it seems to be a rule in the form of conjunctions. Here, the conjunction can be interpreted as an intersection of two genomic intervals, so the new interval is introduced into the hypothesis space, and thereafter the new interval can be exploited for a pattern explanation.

Example 10. *For example, let rule* r_2 *that contains two genomic features in the conjunction is defined as* $r_2 = \{(chr1, s_1, e_1), (chr1, s_2, e_2)\}$ *and if* $chr1 = chr1$ *and* $\{x \in \mathbb{N}^+ | s_1 < x < e_1 \wedge s_2 < x < e_2\} \neq \emptyset$, *then the final interval is the following:* $(chr1, max(s1, s2), min(e1, e2))$.

7.3 PROGRAM SETTINGS SCENARIO

The original *sem1R* works in a simple scenario: induce a set of rules that covers as many positive examples as possible and concurrently covers a minimum of negative examples, according to the chosen evaluation function. However, this scenario does not allow to take into account the natural distribution of genetic variations in a population and therefore the discovered genetic variants do not have to be rare disease-causing. These variants are too widespread across population in general. To include control/background samples into the process of induction, we define and develop the second scenario. The control samples help to eliminate frequent variants. Both scenarios are described in more detail below.

7.3.1 *Scenario 1 - no background samples available*

The first scenario handles data in the same way as the original *sem1R* algorithm. No additional improvements were needed. The input matrix contains only the samples of interest, the samples showing similar symptoms of disease. Given this, positive examples are such variants that appear in the provided samples. Intuitively, negative examples are variants that do not appear in the samples. Notice once again that the induced rule set covers the maximum positive examples and minimum negative examples. Practically, this scenario does not meet the fundamental criteria that are required for the task, i.e., discovering patterns that try to explain a common disease throughout the rare variants of individuals. The problematic part is the determination of negative examples because, in this case, the algorithm penalizes even variants which should have neutral effects in the context of quantification of rule quality (i.e. does not positively or negatively affect the quality of the tested rule). This is shown in Example 11.

Example 11. *Suppose the binary data matrix represented by Table 13 that contains two variants* v_1 *and* v_2*, and two samples* s_1 *and* s_2*, where* v_1 *is present in sample* s_1 *and variant* v_2 *is present in sample* s_2*. Furthermore,*

	sample s_1	sample s_2
variant v_1	1	0
variant v_2	0	1

Table 13: An example of data matrix.

suppose a genomic region g *such that* $v_1 \sqsubseteq_{\mathrm{SINT}} g$ *and* $v_2 \sqsubseteq_{\mathrm{SINT}} g$*. In other words,* g *covers both variants and we can consider it to be the ideal rule because it covers all affected samples* s_1 *and* s_2*. More precisely, the rule* g *covers two positive examples (variant* v_1 *of sample* s_1 *and variant* v_2 *of sample* s_2*) and two negative examples (variant* v_1 *of sample* s_2 *and variant* v_2 *of sample* s_1*). Consequently, for PN evaluation function, that is given as the number of positive examples minus the number of negative examples covered by a rule, the score of rule* g *is 0. Note that a rule which does not cover any example has PN score equal to 0 as well.*

The *Scenario 1* serves as a motivation to incorporate additional changes into *sem1R* algorithm that would address the mentioned scientific question in a better way.

7.3.2 *Scenario 2 - background samples available*

The second scenario eliminates shortcomings that arise from *Scenario 1*. In order to make *sem1R* algorithm more useful in discovering potentially interesting variants, we provide an opportunity to add background samples that allow to filter out false positive variants. The background samples represent individuals unaffected by a disease that we are interested in. In comparison with the *Scenario 1*, the second scenario differs in determining positive and negative examples. Practically, the input matrix stays the same, but a vector of binary numbers is added as the required information which determines assignments to a cohort of individuals having the same symptoms of disease of interest. Certainly, the input matrix includes background samples. The set of positive examples contains only variants occurring in the samples of interest, while negative examples are represented by variants occurring in the background samples. An extended version of Table 13, is presented below in Example 12.

Example 12. *Suppose an extended version of binary data matrix presented in Example 11. The binary matrix and a vector of binary values, represented here in a row format, depicting whether the corresponding sample belongs to the background samples or not, are both depicted in Table 14. Then, the set of positive examples consists of variant v_1 (sample s_1) and v_2 (sample s_2). On the other hand, the set of negatives consists of three examples: variant v_1 of sample s_4 and variant v_2 of samples s_3 and s_4. Moreover, suppose a rule demonstrating an arbitrary genomic region g covering both variants v_1 and v_2, i.e., $v_1 \sqsubseteq_{\mathrm{SINT}} g$ and $v_2 \sqsubseteq_{\mathrm{SINT}} g$. Then, g covers two positive examples and three negative examples. PN evaluation function thus returns a score equal to minus one.*

	sample s_1	sample s_2	sample s_3	sample s_4
assignments	1	1	0	0
variant v_1	1	0	0	1
variant v_2	0	1	1	1

Table 14: An extended version of data matrix where samples s_1 and s_2 that belong to the cohort of samples that are in interest, and a sample s_3 and s_4 that both belong to the background samples.

This example shows that background samples can decrease the score of PN evaluation function and thus only such rule that significantly favorizes variants in non-background samples can be revealed. Note that all zeros appearing in the input binary matrix are omitted; it is completely in contrast to the first mentioned scenario.

7.4 BICLUSTER FORM

A further aspect of the *sem1R* algorithm, that needs to be discussed, is the form of biclusters. Here, a bicluster is formulated as a subset of variants occurring in a subset of samples that are covered by an induced rule. According to the requirements specified in *Scenario 2*, a perfect bicluster should contain all samples of interest because all of these samples/patients report symptoms of a disease. Thus they should be included into one coherent bicluster. However, this would assume that false negative and false positive variants are not present in M. Nowadays, this is an unfulfillable presumption because there are many steps in the whole *variant calling* pipeline and at each step noise can be added into the data. For example, loss of true positive variants in variant calling is inevitable, especially using hard filtering (GATK hard filtering [4]). Thus, some important variants can be filtered out because of inappropriate values of parameters. For this reason, a crisp and strict partitioning of biclusters containing all samples of interest is not a good idea. Therefore, we need to establish a more flexible form of bicluster taking into account the presence of false positives or negatives in the sample dimension.

In the *sem1R* algorithm, the form of biclusters, rows and columns, is generated natively according to one of the proposed evaluation functions (ACC or F1-score). More precisely, first of all, the rule with the highest score of the evaluation function is selected. Then, the examples that are covered by the rule form the bicluster.

To control the sample dimension of biclusters, we have introduced modified versions of the evaluation functions of *sem1R*. Without the modifications, the original algorithm had a tendency to induce biclusters that cover only a few samples. Simply, there is not a huge presence of common variants across samples regardless of the scenario. The equation for computing ACC of rule R is newly formulated as the following:

$$Q_{ACC}(R) = \frac{(TP + TN) \times w}{TP + TN + FP + FN} \tag{31}$$

The equation to compute the potential quality of rule R is adjusted on the following:

$$Q_{p\_ACC}(R) = \frac{(TP + TN + TP) \times w}{TP + TN + FP + FN} \tag{32}$$

Both equations (Eq. 31 and 32) contain newly introduced variable w that is a real number from the range $[0, 1]$. Here, w plays a role of penalization element where the quality of the rule is linearly penalized according to the number of noncovered samples of interest. More precisely, w is defined as a ration between the number of covered samples of interest and the total number of samples of interest. It means that for a rule that covers all samples that belong to the examined set of individuals, w is equal to 1. On the other hand, for a rule that does not cover any such sample, w is equal to 0 and thus the total quality score is equal to 0.

In the same principle, the corresponding equations for F1-score are adapted on the following form:

$$Q_{F1}(R) = \frac{2 \times TP \times w}{2 \times TP + FP + FN} \tag{33}$$

$$Q_{p\_F1}(R) = \frac{2 \times TP \times w}{2 \times TP + FN} \tag{34}$$

In summary, the introduced element w effectively controls the size of biclusters regarding the sample dimension, that is for both presented scenarios. It allows us to consider even such rules that do not cover all samples of interest because of false negative error, i.e., a variant is not present in a list of variants for a sample that belongs to the sample group of interest.

7.5 VARIANT FILTERING

Since this chapter targets rare human diseases, it is highly reasonable to filter-out variants that are highly frequent in a population because evidently, by definition, the occurrence of rare diseases in the human population should be infrequent. For this reason, Genome Aggregation Database (*gnomAD*) [80] that collects structural variants from 14,891 genomes across diverse global populations seems to be an appropriate tool to filter-out too frequent variants.

To accommodate the information of variant frequency from gnomAD into our framework, we propose the same approach for both scenarios. As positive examples, we considered only such variants that are present in no more than 5% of the human population, and simultaneously these variants belong to at least one sample from the sample of interest. On the contrary, negative examples are all variants in samples from a set of background samples. There are no restrictions on the frequency of variants in human populations. Generally, this approach tends to create a relatively small set of positive highly specific examples, i.e., less common variants from the samples of interest, contrary to a large set of negative examples that represent the background samples. The large negative set of examples can help to eliminate false positives, especially.

7.6 WHOLE WORKFLOW

To provide a complete overview of the process of variant calling and the following incorporation of *sem1R* algorithm into the whole pipeline, we established the following steps that are going from sequenced data/reads to the induction of the most significant variants for the given cohort and a control group of individuals (background samples). The overview arises from GATK Best Practices [4]. In Figure 22, we depicted the whole pipeline that is divided into the following three phases that have to be performed sequentially:

1. **raw data pre-processing** phase handles NGS data from physical DNA samples to the file in BAM format. Noise in data, that has not been removed here, negatively affects the results of the following phases [164]. The input samples might be whole-genome sequencing (WGS) or whole-exome sequencing (WES) data. After reads were sequenced, raw data quality in the *fastq* format is verified and only appropriate data continue in the pipeline. To reconstruct the full DNA sequence from fragments, reads are aligned to the reference human genome using some aligner, e.g. Burrows-Wheeler Aligner (bwa) [96, 97]. Because some DNA molecules can be sequenced more times due to the polymerase chain reaction (PCR) [95] and their multiple counting might affect the process of variant calling negatively, read duplicates are located, marked, and eventually filtered out by *Samtools* software. In order to efficient accesses and manipulation, the data are sorted by their coordinates using Picard command-line tool MarkDuplicates. The final step *Base recalibration* empirically recalibrates the quality score of each base hence the original raw quality score does not reflect the true base-calling error rates [124]. The importance of base recalibration is emphasized by the fact that variant calling algorithms,

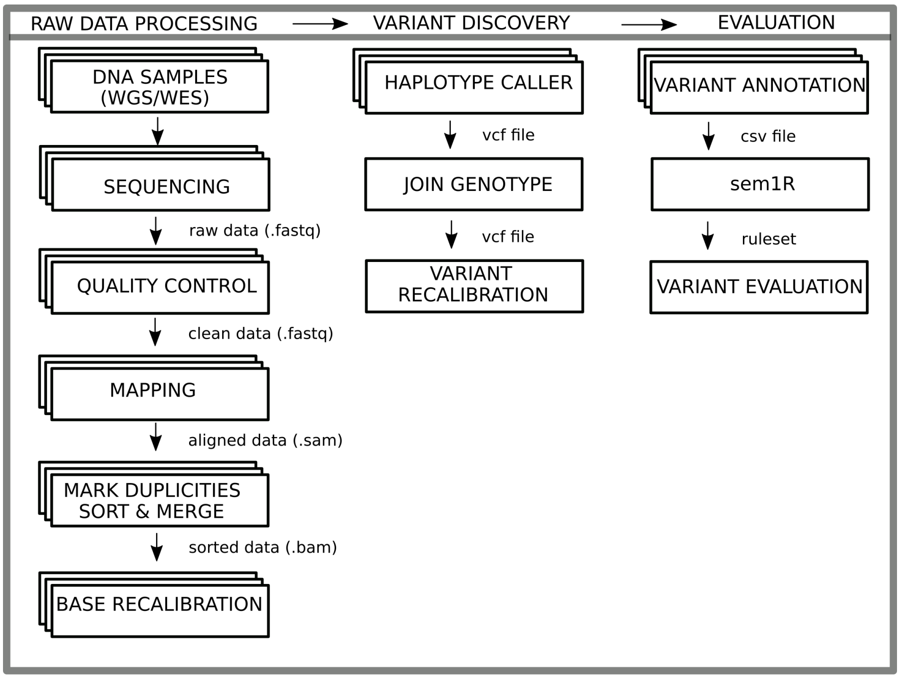


Figure 22: **Variant discovery pipeline.** The overview of variant discovering pipeline that consists of three phases: raw data pre-processing, variant discovery, and evaluation.

such as GATK *BaseRecalibrator*, utilize quality score as an important base feature to making their decision. Simply put, base recalibration improves variant calling accuracy [124].

2. **variant discovery** phase contains three sequentially ordered procedures that from the given individual bam files generate eventually one multiple VCF (Variant Call Format) file. All these steps are implemented and provided by GATK. Firstly, to identify variants in the complex genome, *Haplotype caller* is a popular choice since is capable of calling both variants, SNPs and indels, using denovo local assembly [4]. Haplotype caller works on the per-sample approach so it is necessary to run the algorithm for each bam file separately. Subsequently, *Join Genotype* unions all vcf files into one huge multi vcf file and prepare the file for hard filtering represented here by *Variant recalibration* step.

3. **evaluation** phase requires three tools. Firstly, we transform the multiple VCF file to corresponding csv files, each csv describes variants for one sample. For transformation to csv files we suggest *ANNOVAR* [173] tool. A further used tool is *gnomAD* which serves to append information about the frequencies of variants to csv files. As we mentioned in *Variant filtering* section, frequencies are used to filter out variants that are too common in the population. The last step requires to determine a design of the experiment, i.e. a set of samples that belongs to the cohort of interest (individuals with the same disease). Or alternatively, for *Scenario 2*, a set of samples representing control samples need to be provided too. Then, *sem1R* algorithm is ready to use.

7.7 EXPERIMENTS

In this section, we evaluate the ability of the *sem1R* algorithm to discover potentially interesting genomic regions in a cohort of individuals that share the same symptoms of the disease. Accordingly, we present an evaluation procedure that estimates this ability in various real experiments in which variants, specifically for some diseases, have been reported and published by our collaborators from the ophthalmology field. For the experiments, we report basic statistics as a number of samples and variants, a number of terms appearing in ontologies, etc. In addition to measuring the accuracy of predicted variants, we also measure the speed of induction. All these measurements are performed simultaneously for the two different scenarios, for two various evaluation functions (ACC and F1-score), and three various feature selection methods (FS_atLeastOne, FS_onlySig, and FS_sigAtLeastOne). The evaluation functions and the feature selection methods were introduced in Section 5.3.

7.7.1 *Evaluation procedure*

For the evaluation of the proposed algorithm, we collect and consider only a set of experiments where pathogenic variants are known for each sample. Firstly, all of these samples were preprocessed by the given pipeline depicted in Figure 22. After files are reached in *variant annotation* step, then the *sem1R* algorithm is run with various settings meaning evaluation functions, feature selection methods, and in one of the two scenarios. In addition to the sample files, *sem1R* requires GFF file that defines genomic regions of the human genome. Note that GFF can be manually edited and thus user-defined regions can be appended easily.

The evaluation procedure is defined in the following steps. Firstly, the final hypothesis, in the form of ordered rule set, is induced according to *Scenario 1* or *Scenario 2*. Then, each rule in the rule set is compared with the manually labeled experimental pathologic variant. If the genomic interval of the rule is a superset of the labeled variant, the order of the rule in the rule set is reported as the final distance. Otherwise, the next rule in the rule set is compared.

We note that this evaluation procedure reflects the ability of an algorithm to reveal and report the genomic region where the manually labeled variant appears. Better position of the variant in the rule set enables its earlier verification since less computation time is required.

7.7.2 *ZEB1 experiment*

ZEB1 experiment consists of 6 WES samples sharing the same symptoms of the disease. Firstly, all of these samples were preprocessed by the pipeline depicted in Figure 22. Then, we performed two previously mentioned scenarios. Both scenarios are more specified below.

1. ***Scenario 1*** considers only 6 WES samples on the input, namely, *K01*, *K02*, *S1910*, *S1930*, *S1937*, and *S2406*. No background samples are available. To prevent too common variants in the human population, we filtered out non-significant variants that are given by the gnomAD database. A list of presented samples (denoted as Cohort) is summarized in Figure 23 where numbers of significant and non-significant variants are highlighted. As we can see, significant variants constitute a small portion of all presented variants. For the first scenario, most variants are disregarded in considering.

   For estimating the importance of using various genomic regions as predictive rules where the regions have different levels of specificity, we examined the common variants across all samples in the cohort. The most largest sets of intersections of variants are shown in Figure 24 using *Upset* plot [94] which is a suitable form of plots for finding and visualizing intersections for more than three sets. It is a good alternative to the Venn diagram. As we can see in that figure, the same variants are spread across all samples very rarely. More precisely, samples K01 and K02 share 6,051 variants in the whole human genome, however, only 1,706 variants are shared across the three samples S1910, S1930, and S1937. Subsequently, only 673 variants are shared across S1910, S1930, S1937, and S2406. The sample S1910 contains 12,490 variants that are only specific for this sample, i.e., such variants that are not present in any other sample. In addition, there is no variant that is shared simultaneously over all six samples. Therefore, a rule consisting of only variants or their conjunctions cannot cover all samples in that example. Variants are too specific in this case. Given this, it is inevitable to extend the hypothesis language by various more general genomic elements.

   To see the distribution of rare variants across samples, we established a binary matrix that has been previously defined in *Introduction* section. The matrix M was constructed from the input WES files restricted to chromosome 10. The other human chromosomes were disregarded, since *Zeb1* gene occurs only in chromosome 10. Totally, M has 2,734 rows and 6 columns. The rows represent particular variants occurring on chromosome 10. The columns represent the input files. Ones in the matrix express the appearance of particular variants in the corresponding samples. Otherwise, the elements in M are equal to zero.

   The cardinality of the set of positive examples is equal to 3,468, the negative set of samples contains 12,936 examples. The total number of terms, i.e. genomic regions, introduced into the process of induction is equal to 42,086, that is, only for chromosome 10. However, many terms are disregarded from consideration because of the feature selection methods that reduce a feature space dramatically. Because the *sem1R* algorithm uses the *covering* algorithm to induce multiple rule sets, feature selection

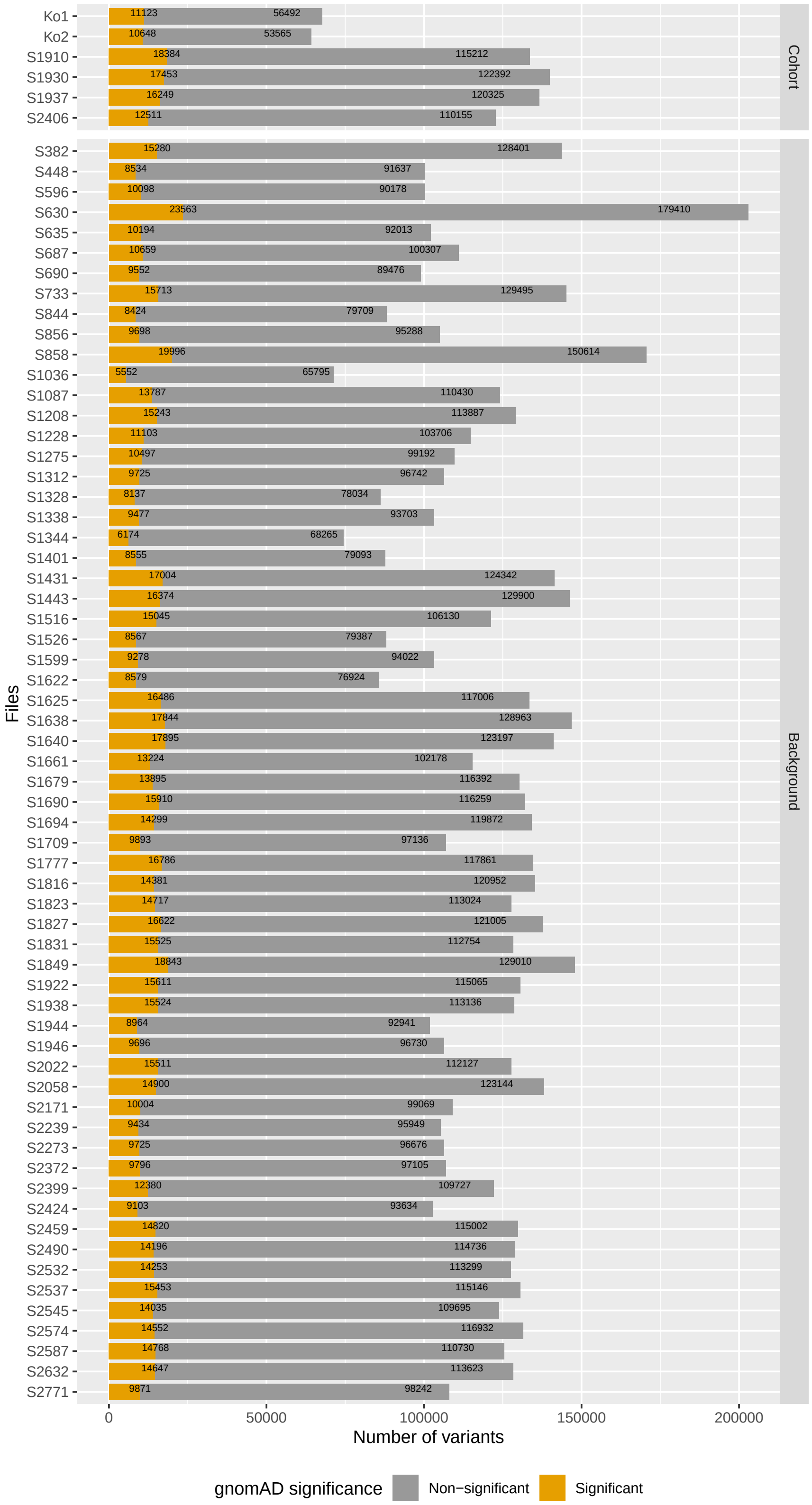


Figure 23: **Number of variants in samples.** Bar plot shows numbers of variants, significant and non-significant, in input files. Files are separated into two groups. The cohort represents the group of samples of interest, the background represents control samples. Significance is determined by the gnomAD database.

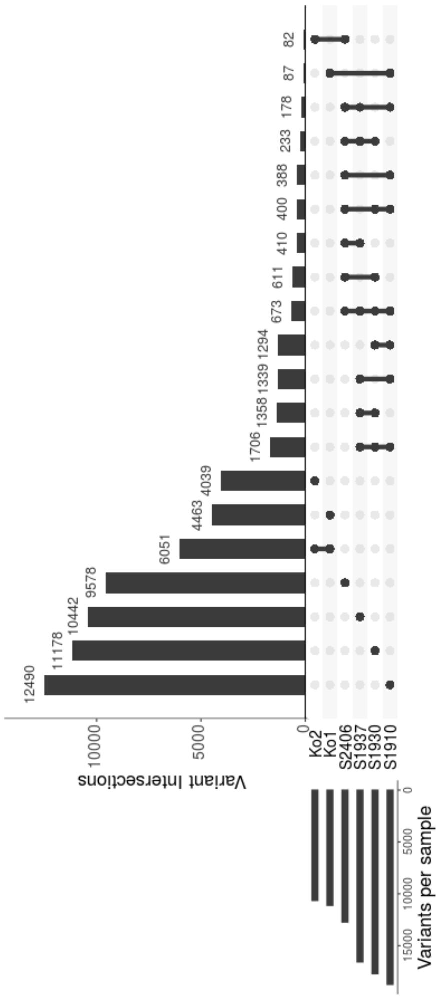


Figure 24: Upset plot shows the top 20 most large sets of intersections of variants across the samples of interest for *Scenario 1*. Note that non-significant variants were filtered out and thus are not shown. Dark circles in the matrix below the bars indicate sets that are part of the intersection.

methods are called before every single rule induction step. At first sight, this is reasonable for rules covering a relatively large number of examples since due to the covering algorithm, the set of examples might be reduced more dramatically. Then, a feature selection method might subsequently reduce a number of terms going to the induction process. Otherwise, for rules covering a small number of examples, the feature selection methods might slow down the speed of the algorithm since the selection methods are time consuming.

The cardinalities of feature sets according to the feature selection methods are depicted for each induced rule in Figure 25. Evidently, *FS_atLeastOne* method is not so radical in prunning of feature space than *FS_onlySig* and *FS_sigAtLeastOne*. Therefore, *FS_atLeastOne* is able to induce more rules. Concretely, the *sem1R* algorithm induced only 3 (resp. 7) rules for *FS_onlySig* (resp. *FS_sigAtLeastOne*) feature selection method. Then, the algorithm was terminated since the feature set has been empty.

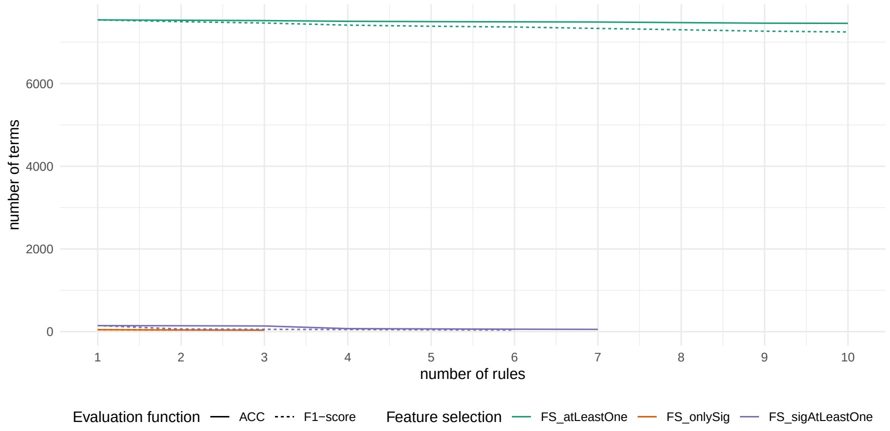


Figure 25: Cardinality of the feature sets that were established during the process of induction of 10 rules for the human chromosome 10 in the *Scenario 1* setting.

The run times of the *sem1R* algorithm for different feature selection methods and evaluation functions are depicted in Figure 26. In contrary to the previous results presented in Figure 12, *FS_atLeastOne* method is the fastest. The reason for this behavior is the different time requirements of the feature selection methods. Intuitively, without any exact proof, *FS_atLeastOne* should be the fastest because makes only a binary decision whether a term is covered by at least one example or not. *FS_onlySig* should be slower because a more complicated formula that computes a Likelihood Ratio Statistic (LRS) defined in Eq. 30 needs to be computed. Finally, the slowest method should be *FS_sigAtLeastOne* since firstly the algorithm computes LRS for each term and then is examined whether they are covered by at least one example. The thesis is supported by the collected data shown in Figures 26 and 28.

The estimated times needed for the feature selection methods are represented by the area in the plot where the curves grow relatively enormously in comparison to the rest. In summary, the implemented feature selection methods take a relatively large amount of time compared to the induction process itself. We explain this by the fact that the genomic region-related ontology is wider than deeper and generally the hypothesis language is more limited in a sence of relations than we suppose originally in the *sem1R* algorithm. Then, the induced rule covers fewer examples, and therefore the size of the set of examples that goes to the next iteration will not change dramatically. To support this, make a comparison with Figures 25 and 11.

The result showing the ability of the algorithm to recognize and describe genomic regions that are specific for the cohort of individuals is summarized in Table 15. For the given parameter settings, a rule set is induced by the *sem1R* algorithm and sub-

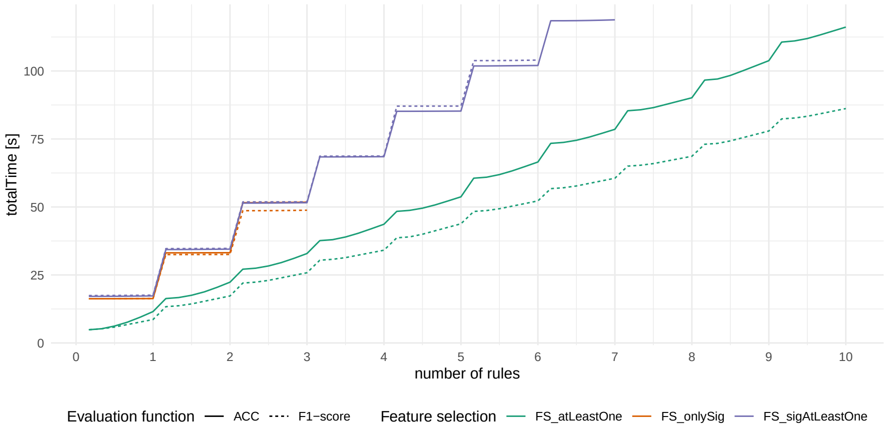


Figure 26: Total run time for the induction of 10 rules for the human chromosome 10 in the *Scenario 1* setting. The algorithm performed for different evaluation functions and different feature selection methods.

Evaluation function	Feature selection	Position in the rule set
ACC	*FS_atLeastOne*	17
	FS_onlySig	NA
	FS_sigAtLeastOne	NA
F1-score	*FS_atLeastOne*	32
	FS_onlySig	NA
	FS_sigAtLeastOne	NA

Table 15: Results for the *Scenario 1* where positions of rules that cover the manually labeled genomic region are reported. Better position of the rule in the rule set enables its earlier verification. The size of the rule set was restricted to 50.

sequently, the order of the rule that covers the region manually specified for each sample, the reference variant, is recorded in the table. NA value means that no rule that covers such a region has been found in the rule set. The lower the number is, the rule is more significant and consequently, it is faster and easier to find it.

Evidently, the suitable evaluation function for this kind of scenario is ACC since the discovered corresponding rule is on the 17th position in the rule set in contrast to the 32nd position for F1-score. Furthermore, the experiments where *FS_onlySig* and *FS_sigAtLeastOne* were used did not discover the region of interest. Additionally, both of them reduce the feature space too radically and consequently did not induce the required rule set of 10 rules.

2. ***Scenario 2*** considers the same 6 WES samples on the input as in the case of *Scenario 1*, i.e., *K01*, *K02*, *S1910*, *S1930*, *S1937*, and *S2406*. However, there are also 62 samples that play the role of control samples. Similarly to the previous scenario, we fil-

tered out non-significant variants with a p-value > 0.05 given the gnomAD database. The numbers of variants for the samples of interest and the control samples are summarized in Figure 23. Contrary to the *Scenario 1*, variants in the control samples are used both significant and non-significant. This leads to a disproportion between the size of positive and negative examples.

The matrix M was constructed from the input WES files including the samples of interest and the control samples. In both cases, they are restricted to chromosome 10. M has 27,273 rows and 68 columns. The interpretation of rows, columns, and elements is the same as the previous scenario.

The number of positive examples is equal to 3,468 examples, the negative examples are 287,577. The total number of terms introduced into the process of induction is equal to 66,625 for chromosome 10. The cardinalities of feature sets given by feature selection methods are depicted for each induced rule in Figure 27. Moreover, the same trend of pruning that we described for *Scenario 1*, we observed even for *Scenario 2*. *FS_atLeastOne* method prunes the feature space not so radically as *FS_onlySig* or *FS_sigAtLeastOne*. Although using *Scenario 1* induced only 3 (resp. 7) rules for *FS_onlySig* (resp. *FS_sigAtLeastOne*) feature selection method, for *Scenario 2* the algorithm induced all 10 rules that were required for all proposed feature selection methods.

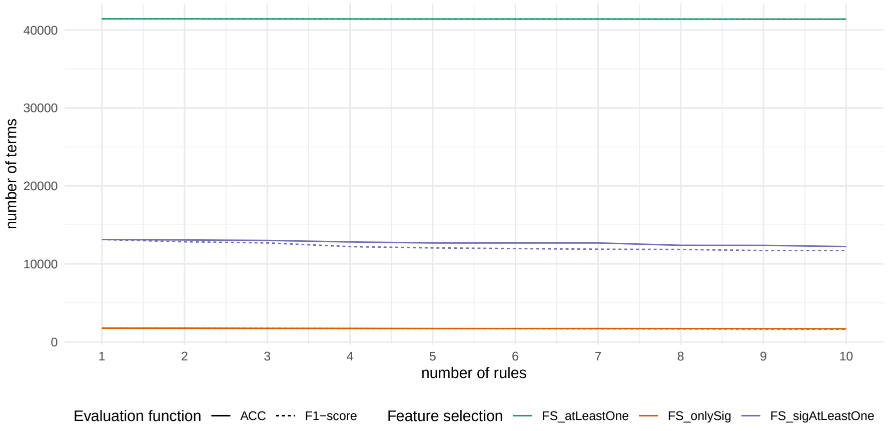


Figure 27: Cardinality of the feature sets that were established during the process of induction of 10 rules for the human chromosome 10 in the *Scenario 2* setting.

The runtimes of the algorithm for different feature selection methods and evaluation functions are depicted in Figure 28. The trends in the measured runtimes correspond to the trend that we observed in *Scenario 1*. This means that *FS_atLeastOne* method is the fastest, *FS_onlySig* reaches the runtime in the middle, and *FS_sigAtLeastOne* is the slowest method. Even in *Scenario 2* that contains much more negative examples, the runtime of the feature selection method plays a dominant part of the total runtime in that particular task.

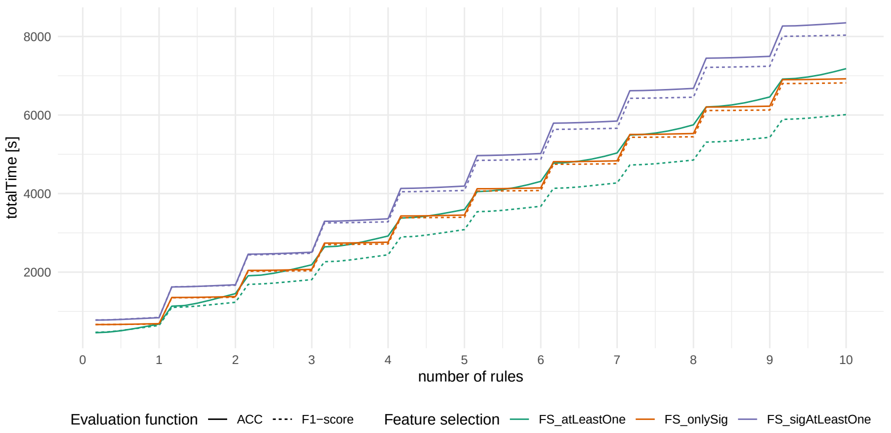


Figure 28: Total run time for the induction of 10 rules for the human chromosome 10 in the *Scenario 2* setting. The algorithm performed for different evaluation functions and different feature selection methods.

The results of discovering the region of interest for *Scenario 2* restricted to chromosome 10 are summarized in Table 16. Indeed, the added control samples help to control too common variants in a population and improve the ability to find the region of interest for all evaluation functions and all feature selection methods. Using ACC, the induced rule that covers the region of interest was reached on the second position in the rule set. The same position in the rule set we got for each feature selection method. In other words, for ACC, the feature selection methods do not influence the position of the rule that covers the region of interest. In the case of experiments where F1-score was used as the evaluation function, the position of the rule in the rule set reached a higher/better position for *FS_onlySig* and *FS_sigAtLeastOne* in comparison to the experiments of *Scenario 1*. However, using *FS_atLeastOne* the rule was found on the 35th position. Clearly, F1-score does not get better results than ACC in any feature selection method. This we explain by the fact that F1-score focuses more on the positive examples than ACC. More precisely, F1-score does not take into account examples classified as true negative in the evaluation formula.

To see the distribution of variants across genomic coordinates of a particular rule, we plot the distribution via a density plot in Figure 29. The figure was created for *Scenario 2* using ACC and *FS_atLeastOne* and the whole dataset was restricted only to chromosome 10. The figure shows the distribution of significant variants, denoted as positives (yellow color), and both significant and non-significant variants of control samples. For example, rule 2 covers variants in a genomic region that appears on the following coordinate: chr10:31,650,000:31,800,000. A yellow peak on the right side of the figure highlights a small region where a significant number of variants that belong to the set of positive examples are accumulated. Simultaneously, a subset of

Evaluation function	Feature selection	Position in the rule set
ACC	*FS_atLeastOne*	2
	FS_onlySig	2
	FS_sigAtLeastOne	2
F1-score	*FS_atLeastOne*	35
	FS_onlySig	17
	FS_sigAtLeastOne	17

Table 16: Results for the *Scenario 2* where positions of rules that cover the manually labeled genomic region are reported. Better position of the rule in the rule set enables its earlier verification. The size of the rule set was restricted to 50.

negative variants is accumulated on the left side of the region of positive examples. In spite of the dominance of negative examples, the rule is considered significant since the cardinality of positive and negative sets is unbalanced.

7.7.3 *Algorithm acceleration*

The *sem1R* algorithm as described in Chapter 5 defines repeatedly a feature set in each iteration step when inducing a single rule. Calling *featureConstruction* and *featureSelection* function has its own rationality.

In contrast to the common variant discovery task, the originally formulated algorithm and the three gene expression datasets presented in Chapter 5 exploit Gene ontology which is able to construct rules that cover a relatively large number of examples. For these experiments, the average number of features across DIST, DOT, and m2801 datasets is depicted in Figure 11. In this case, the cardinality of feature sets drops down relatively fast with the number of induced rules. This, we explain by the fact that the induced rules cover a huge number of examples that are associated with many ontology terms. These terms are further excluded from the induction process in the next iteration. However, in the case of variant discovery, the number of covered examples is much lower for each iteration than the number of covered examples of the originally motivated task. This also reflects the cardinality of the feature set that is depicted in Figures 25 and 27 for *Scenario 1* and *Scenario 2*, respectively.

Since the feature set itself does not change rapidly during the process of rule induction, we move *buildMapping*, *featureConstruction*, and *featureSelection* functions outside of *induceSingleRule* function. The adjusted version of the *sem1R* algorithm, named *variant-sem1R*, is presented in Algorithm 7. Three functions, *buildMapping*, *featureConstruction*, and *featureSelection*, are now located in lines 2-4. These time-consuming functions are called only once when the algorithm is running.

In contrast to the original *sem1R* algorithm, when a new rule is induced, only positive examples that are covered by the rule are re-

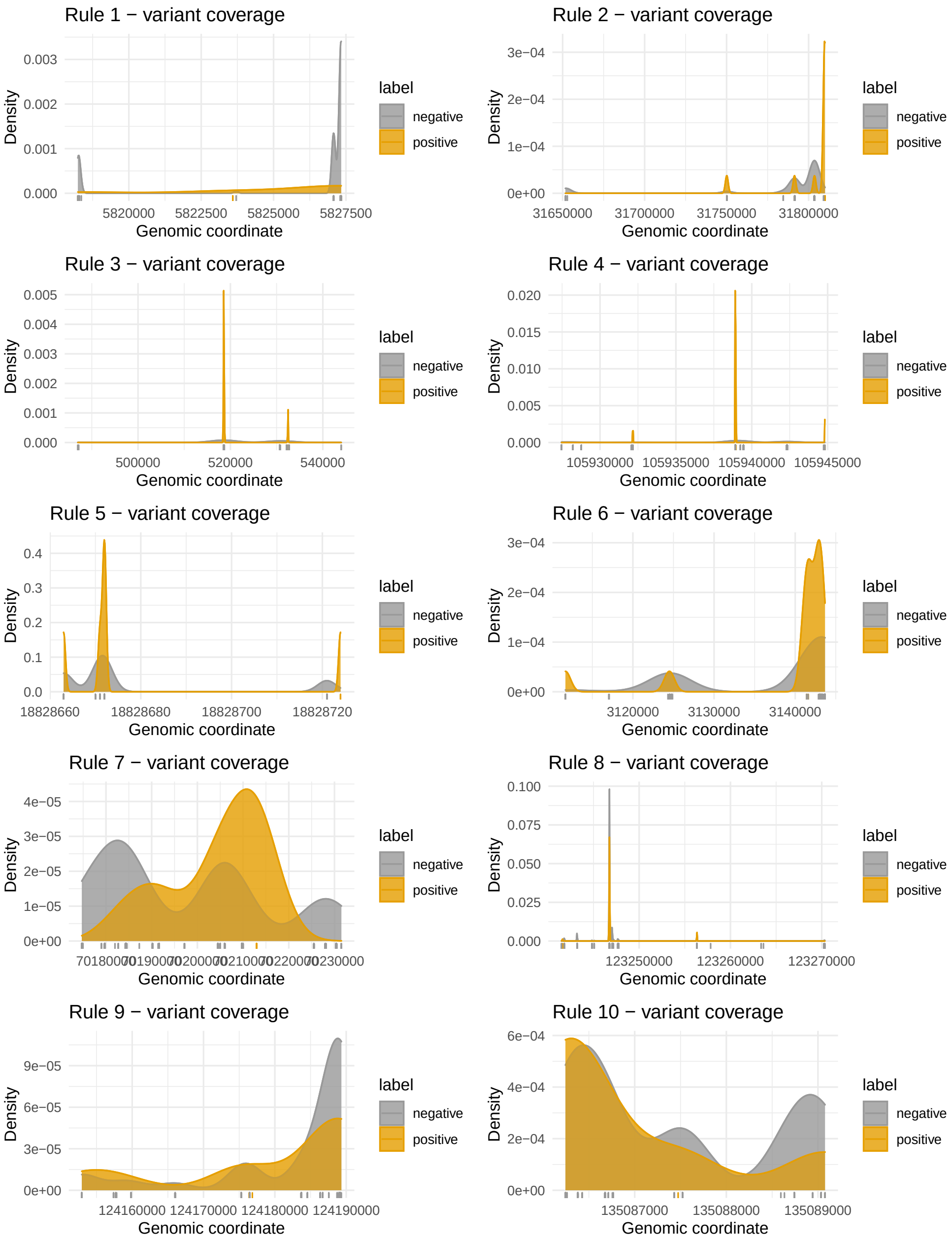


Figure 29: Density plot for the top 10 rules induced by the algorithm for *Scenario 2* using ACC and *FS_atLeastOne* feature selection method.

moved from the set of positive examples E^+. The covered negative examples are not removed from the set of negative examples E^- because they are still being considered as control samples that serve to filter out too common variants. There is no rational reason for restriction of the set of control samples.

Moreover, the new version of the function responsible for single rule induction, the *variant-induceSingleRule*, is shown in Algorithm 8. This function remains almost the same, only two algorithm inputs, $\mathbb{R}$ and $\mathbb{F}$, are added to make the algorithm coherent. For this reason, further algorithm explanation is redundant. For more details of the original work, see Algorithm 5.

Algorithmus 7 : variant-sem1R

```
input  : $E^+$, $E^-$,$\mathcal{O}$, k
output : $\mathbb{H}$ // hypothesis

$\mathbb{H} \leftarrow \emptyset$
$\mathbb{M}' \leftarrow$ buildMapping($\mathcal{O}$, $E^+$, $E^-$)
$\mathbb{F} \leftarrow$ featureConstruction($\mathcal{O}$)
$\mathbb{F} \leftarrow$ featureSelection($\mathbb{F}$, $E^+$,$E^-$, $\mathcal{O}$, $M'$)
$\mathbb{R} \leftarrow \mathbb{F}$
foreach $i \in \{1, 2, \cdots, k\}$ do
    $newR \leftarrow$ variant-induceSingleRule($E^+$, $E^-$, $\mathcal{O}$, k, $\mathbb{R}$, $\mathbb{F}$)
    $E^+ \leftarrow$ removeCoveredExamples($newR$, $E^+$)
    $\mathbb{H} \leftarrow \mathbb{H} \cup newR$
end
return $\mathbb{H}$
```

Algorithmus 8 : variant-induceSingleRule

```
input  : $E^+$, $E^-$, $\mathcal{O}$, k, $\mathbb{R}$, $\mathbb{F}$
output : $\mathbb{R}_{BEST}$ // conjunction of selectors

$\mathbb{R}_{BEST} \leftarrow \emptyset$
$\mathbb{R}_{BEST\_SCORE} \leftarrow 0$, $l \leftarrow 0$
// discover rules until stopConditions
while $\mathbb{R} \neq \emptyset$ and $l < k$ do
    $\mathbb{R}_{new} \leftarrow \emptyset$
    // Refine all rules in R
    foreach $r \in \mathbb{R}$ do
        $newCandidates \leftarrow$ refineRule($r$, $\mathbb{F}$,$\mathcal{O}$,
         $\mathbb{R}_{BEST\_SCORE}$, $E^+ \cup E^-$, $M'$)
        $\mathbb{R}_{new} \leftarrow \mathbb{R}_{new} \cup newCandidates$
        // Find the best rule
        foreach $nc \in newCandidates$ do
            score $\leftarrow$evaluateCandidate($nc$, $E^+$, $E^-$, $\mathcal{O}$, $M'$)
            if score $\geqslant \mathbb{R}_{BEST\_SCORE}$ AND isSignificant($nc$,
             $E^+$, $E^-$, $\mathcal{O}$, $M'$) then
                $\mathbb{R}_{BEST} \leftarrow nc$
                $\mathbb{R}_{BEST\_SCORE} \leftarrow$ score
        end
    end
    $\mathbb{R} \leftarrow$filterRules($\mathbb{R}_{new}$)
    $l \leftarrow l + 1$ // increment the rule length by one
end
return $\mathbb{R}_{BEST}$
```

As we can see in both Figures 30 and 31, these improvements accelerate the algorithm dramatically. Equally important is the fact that changes in the algorithm do not change the accuracy nor the order of the induced rules.

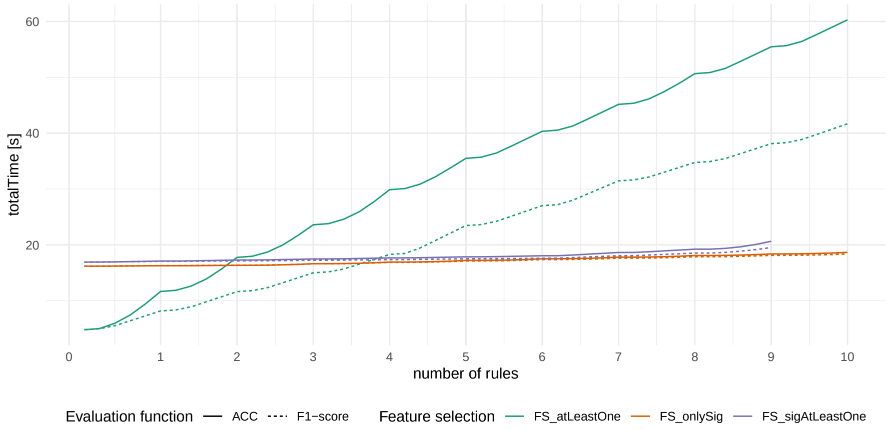


Figure 30: Total run time for the induction of 10 rules for the human chromosome 10 in the *Scenario 1* setting with the accelerated version of *sem1R* algorithm. The algorithm performed for different evaluation functions and different feature selection methods.

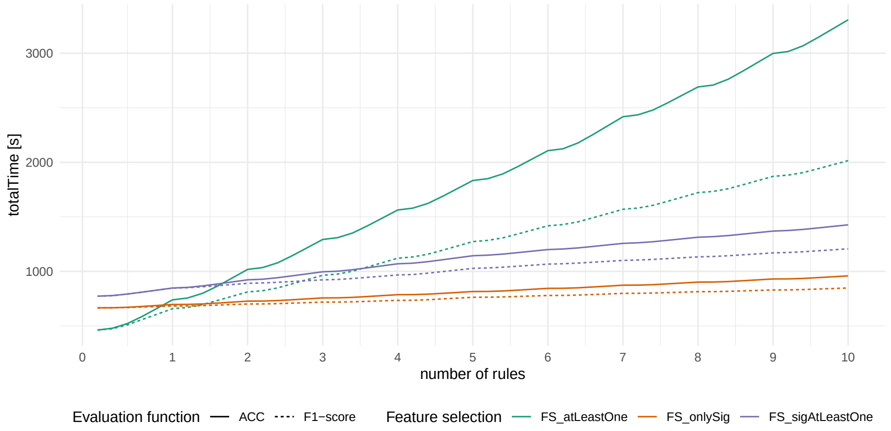


Figure 31: Total run time for the induction of 10 rules for the human chromosome 10 in the *Scenario 2* setting with the accelerated version of *sem1R* algorithm. The algorithm performed for different evaluation functions and different feature selection methods.

7.8 RESULTS

In order to make the results and the algorithm itself more understandable for a wider audience, we bring an example of induced rules. The rule that we present in this section covers the region of interest in *Zeb1* gene and it is on the second position in the ordered rule set for ACC evaluation function using *FS_atLeastOne* feature selection method, see Table 16. The rule has the following form:

mRNA:10:31608151-31816222 and
processed_transcript:10:31608141-31812935

where the first gene region represents *mRNA* in chromosome 10 which starts at 31608151 and ends at 31816222 position. The second region represents a transcript that does not contain an open reading frame and starts at 31608141 and ends at 31812935 position on chromosome 10 as well. Here, the conjunction ("and") can be interpreted as the grand intersection of all regions appearing in the rule. Specifically, the rule covers variants in a region that starts at 31608151 and ends at 31812935 position on chromosome 10. In this case, the covered variants have appeared in the region of interest in all samples of the cohort.

7.9 CONCLUSION

We adapted the original version of *sem1R* algorithm to discover common genomic regions that are shared over a set of individuals that have the same symptoms. Moreover, we propose two various scenarios that handle the input data in a different way. *Scenario 1* requires only a set of individuals that have the same symptoms, and then the algorithm finds genomic regions, or their intersections, where variants are enriched. On the contrary, *Scenario 2* admits adding background (control) samples that are used to filter out variants that are too widely spread in the population. If control samples are available, *Scenario 2* is recommended.

The ability to determine genomic regions of interest was confirmed in a real study that focused on mutations in ZEB1 gene. Expectedly, a better performance was shown by *Scenario 2* that reports a rule that covers variants of interest on the second position of the ordered rule set. This increases the chance of discovering interesting pathogenic variants.

Contrary to the original framework *sem1R*, we have proposed a new acceleration approach that significantly reduces the runtime of the algorithm for this particular application without changing the results at all.

The new adjusted algorithm is written in C++ and is called *variant-sem1R* algorithm. The whole framework is published as an R package freely available at http://github.com/fmalinka/variant-sem1R.

8 MULTI OBJECTIVE SEMANTIC BICLUSTERING IN OMICS DATA

This chapter addresses and extends one of the main ideas that has been presented in Chapter 4. In that chapter, we proposed two different approaches to handle the *semantic biclustering* task that are called *bi-directional enrichment* and *rule and tree learning*. Here, we introduce a new method that combines both previously mentioned methods into one with an ambition to reduce their disadvantages.

8.1 BACKGROUND

Indeed, *bi-directional enrichment* method consists of two consecutive phases that are partially dependent; the first phase affects the second phase, however, information from the second phase does not affect processes of the first phase. The first step forms a set of coherent biclusters, the following phase finds out the descriptions of these biclusters. Intuitively, this scheme has an evident drawback: when the first phase is finished, the second phase utilizes only the inputs that are provided. When the biclusters have been defined inappropriately in the first phase, the second phase can never reconstruct them. The optimal way would be to iterate or rather merge both the phases. However, this kind of iteration would be extremely time and computationally exhaustive regarding the number of ontological terms and the number of elements in the 2D binary matrix that increase the complexity as well. At the same time, the phase merge opens numerous non-trivial design issues.

In order to address the requirements for establishing the principle of feedback between the phases which would be feasible in a relatively short time, we come up with an idea to incorporate a type of heuristic, or oraculum, into the process of forming biclusters which might increase a chance to discover a better description of biclusters. Since the final description is in the form of conjunctions of ontology terms, we intercorporate their mutual semantic similarity into the process of yielding biclusters. This would help to reveal more semantically coherent biclusters and thus a more suitable description.

8.1.1 *Biclustering as an optimization task*

In general, the problem of clustering or biclustering can be posed as an optimization task. The objective/objectives to be optimized can reflect various cluster or bicluster characteristics. A good example of such objectives can be the homogeneity of clusters/biclusters. However, the meaning of cluster/bicluster homogeneity varies on the type of clusters/biclusters that we are seeking. It also depends on how the original research task is formulated or which type of data we are using. Especially in the bioinformatics field, various types of biclusters as for example *biclusters with constant values* or *biclusters with correlations* are considered. We briefly mentioned some of them in Section 2.2. Simply put, for the binarized datasets that are used throughout this dissertation thesis, i.e., *DOT*, *DISC*, and *m2801*, we can consider the form of biclusters as *biclusters with constant values*. For other RNA-Seq datasets with normalized continuous values representing gene expression by quantifying the amount of messenger RNA transcripts, we can consider *biclusters with correlations*, for example.

As we have already addressed in Chapters 4 and 5, numbers of examples and ontology terms for *DOT*, *DISC*, and *m2801* datasets are enormous, see Table 10. Although some clustering/biclustering algorithms might be computationally efficient, they often get stuck at some local optima depending on chosen parameters [107]. A good example that illustrates the usual trend of stucking at the local optima might be the usage of K-means algorithm which is highly dependent on the choice of the initial cluster centers. To overcome the issue of local optima and simultaneously reaching the global optima, some genetic algorithms are widely used across computer science [107] since they yield high-quality outcomes for a hard combinatorial optimization problem. For this reason, genetic algorithms seem to be a reasonably applicable approach in that task. Besides, we chose to exploit the principle of genetic algorithms for their relatively easy implementation and easy adaptation for solving multi-objective optimization tasks.

The proper objectives to be optimized and the techniques to reach the global optima are discussed in more detail in the following sections.

8.1.2 *Multi objective optimization*

Incorporating a further objective being optimized comes from the simple idea to connect the two separated phases of semantic biclustering, i.e., the phase of bicluster formulation and the phase of finding their descriptions. If the biclusters were also sufficiently semantically coherent and not only coherent in their values (gene expression, genetic variants, etc.), then the ontology description would be more understandable and easily interpretable since each bicluster would be determined by elements having more similar properties. On the other hand, semantically incoherent biclusters would report a description having less similar terms, e.g., nonsimilar ontology terms associated with genes of different biological functions. Furthermore, finding a description for coherent groups of elements would generate shorter rules. Consequently, a hypothesis validation would be much easier. These assumptions give rise to the idea of extending single objective optimization to multiple objective optimization.

Integrating biological knowledge for searching biclusters using some evolutionary inspired approaches is not an untouched scientific topic.

In [121], the authors present a scatter search metaheuristic for biclustering that optimizes three objectives. The authors proposed a fitness function to evaluate a bicluster B as follows:

$$f(B) = M_1 \times f_1(B) + M_2 \times f_2(B) + M_3 \times f_3(B) \tag{35}$$

where f_1 measures a size of the bicluster B, f_2 evaluates the patterns found in the bicluster B, and f_3 evaluates the semantic similarity of terms in the bicluster B from the biological point of view. Furthermore, weights represented by M_1, M_2, and M_3 show a relevance of the corresponding functions f_1, f_2, and f_3.

The transformation of multi-objective optimization tasks to single-objective optimization tasks using Eq. 35 is a very straightforward approach. On the other hand, the parameters M_1, M_2, and M_3 have to be defined apriori before the algorithm starts. This supposes that these parameters are known a priori or it brings a necessity to estimate their relevance manually or by any other technique.

To eliminate the defining relevance of the functions a priori, we solve the multi-objective optimization problem via a different approach that will be mentioned later in Section 8.2.

8.1.3 *Semantic similarity*

As we mentioned in Eq. 35 in the previous section, the function f_3 measures the bicluster coherency from the biological point of view. Here, because ontologies are available, we approximate the biological relevance by ontology-based semantic similarity measures. There is plenty of various semantic measures which use ontologies to estimate the similarity as *edge-based* or *node-based* approaches [69]. For the sake of simplicity, we firstly focus on gene-pairwise measures of terms from Gene ontology reviewed in [123] and originally presented in [127]. *SimUI* is a graph-based approach that estimates biological relevance by counting a number of common ontology terms and normalized by a number of terms that are in relationship "a more general term" with the compared initial terms. Although the *SimUI* is introduced as a similarity measure for estimating the gene similarity in Gene ontology, generally it is applicable to any other ontologies that are in the form of directed acyclic graph. We present *SimUI* measure for two terms t_1 and t_2 as follows:

$$\mathrm{SimUI}(t_1, t_2) = \frac{|mg(t_1) \cap mg(t_2)|}{|mg(t_1) \cup mg(t_2|} \tag{36}$$

where mg represents a function that returns all terms in the given ontology that are more general than the corresponding term t_1 or t_2. To fully understand the concept of "more general term", we note that the necessary definitions including the definition of ontology have been introduced in Section 5.1.

Although there are many other similarity measures [68] or already implemented toolkits (e.g. *The Semantic Measures Library Toolkit* online available at `https://www.semantic-measures-library.org/`), we decided to use *SimUI* measure because of its efficient and easy incorporation into the original algorithm *sem1R*. Note that the core of the *sem1R* algorithm is used also for the currently described algorithm solving the multi-objective optimization problem. For more details see Section 8.2.7.

8.2 METHODS

In this section, we describe the proposed algorithm in more detail. The algorithm consists of two separate phases, where the first phase deals with multi-objective optimization problem for forming biclusters and the second phase addresses the process of hypothesis or model induction.

8.2.1 *Bicluster forming*

To avoid specifying manually the particular values of the selected objectives and thus their relevance (weights), we aim our research to develop a method that does not require an external user intervention in this manner. A well-known genetic algorithm dealing with the multiple optimization problem that is used in our research is called Nondominated Sorting Genetic Algorithm II (*NSGA-II*) [37]. *NSGA-II* yields the final best solution in the format of nondominated Pareto optimal solutions. However, the main characteristic of these kinds of algorithms is a large number of produced Pareto optimal solutions [107] since the objectives are usually conflicting and therefore there is no one optimal solution. The problem of identifying "the best" solution from the whole Pareto optimal set has been addressed in several papers [13, 35] where the authors suggest approaches to focus the search on "the best" solutions from the region of interest, also known as *knee of the Pareto curve*. An example of identifying knees in a three-objective problem is depicted in Figure 36.

This problem and the proposed solution are discussed in Section 8.2.7. Now, go back to the bicluster forming phase and thus the application of NSGA-II.

8.2.2 *NSGA-II algorithm*

We outlined the fundamental steps of NSGA-II as the following:

1. Initialize a population randomly.
2. Sort the population based on a non-dominated sorting (see [37]).
3. Do binary tournament selection.
4. Do recombination and mutation to create the offspring population.
5. Combine parent and children population.

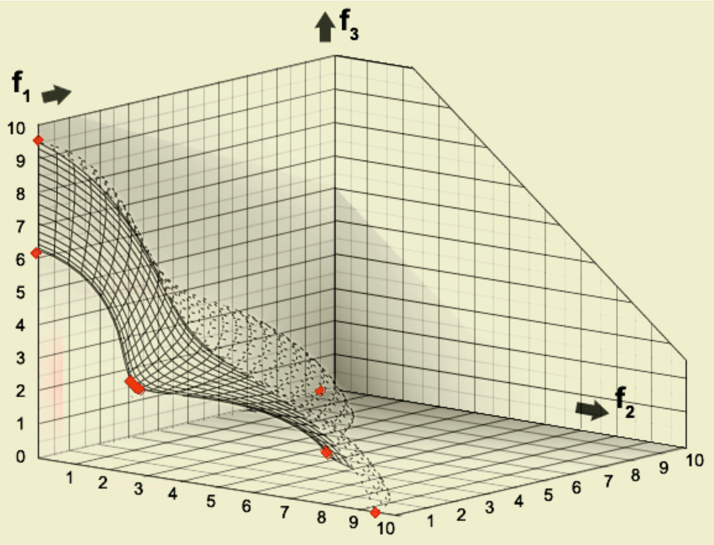


Figure 32: Three-objective problem with knees labeled by red color. Dashed lines represent the Pareto area that contains a large number of solutions in comparison with the number of knees. The figure is taken from [13].

6. Select the best individuals that go to the next generation by non-dominated sorting.

To fully accommodate the *semantic biclustering* problem, we firstly define an individual and its encoding into a chromosome representation. Moreover, we introduce three objectives to be optimized in more detail.

8.2.3 *Conventional Encoding (representation)*

In evolutionary inspired techniques, one of the fundamental steps that need to be solved is to encode a given solution of a task into a chromosome. In [114], the authors represent a bicluster with a chromosome as a fixed-sized binary vector which is divided into two parts. The first part of the whole vector is dedicated to determining rows of a bicluster, the second part focuses on its column dimension. A binary element in the vector is set to 1 if the corresponding row and/or column is present in the bicluster. Otherwise, the element is set to 0. An example of such type of bicluster representation is depicted in Figure 33.

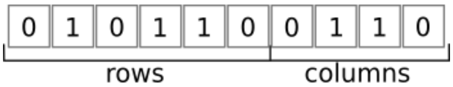


Figure 33: An example of chromosome representing a bicluster.

Computing the outer product of the two vectors, column and row vectors, gives arise to a matrix where 1's represent the elements that belong to the constructed bicluster. An example is presented below.

Example 13. *Suppose two vectors from Figure 33, let* $r = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$ *and* $c = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$ *Then, the bicluster* B *is determined as follows:*

$$B = r \times c^T = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where 1's in the matrix represent the elements that belong to the bicluster. Note that biclusters are usually illustrated in their sorted form, i.e., the form where columns/rows are shuffled to create a matrix where the elements belonging to the bicluster are next to each other. In this example, we keep the order of columns/rows in the original unchanged format.

8.2.4 *Extended encoding version*

Since the previous conventional encoding allows to handle only one bicluster - thus one individual - at the moment, we extend the original encoding with the ability to determine more biclusters encoded together as one individual/vector. This extension brings a new opportunity to formulate more biclusters with similar properties. For more details see Section 8.2.5. On the other hand, it requires more computational resources since the length of the chromosome is generally larger.

To encode k biclusters into one individual, each of the k biclusters is encoded in the same way as in the conventional approach. However, the corresponding biclusters are appended sequentially into one fixed-sized chromosome. In comparison to the previous encoding version, the extended version prolongs the chromosome k times as is depicted in Figure 34.

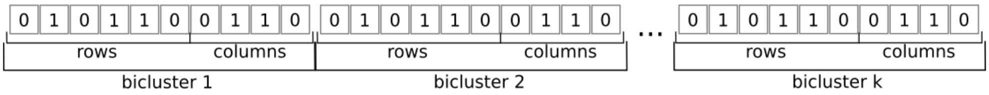


Figure 34: An example of chromosome that represents k various biclusters.

For the *extended encoding version*, the binary matrix is computed as in the previous encoding version but for each pair of rows and columns individually. Then, a pairwise logical *OR* operator is applied to the corresponding matrices. This yields the binary matrix indicating all elements that belong to any of k biclusters. An example is appended below.

Example 14. *Suppose two vectors* r *and* c *from Example 13. Furthermore, suppose vectors* $r2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ *and* $c2 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$ *Then, the final matrix is determined as follows:*

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \%OR\% \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

where %OR% is a pairwise logical OR operator.

Finally, we note that the conventional encoding version is the example of the extreme case of *extended encoding* for $k = 1$. In that case, the chromosomes are identical.

8.2.5 *Pareto set of multiple objectives*

A further aspect that has to be accommodated to NSGA-II algorithm is a definition of *tested problem*. Here, the tested problem means to gather a list of steps that have to be done to formulate a bicluster using NSGA-II algorithm. We formulate these steps as follows:

1. specify objectives to be optimized and the corresponding evaluation functions to approximate these objectives (collectively called *fitness function*),
2. set up an approach to handle the evaluation functions in the encoding of solutions in the chromosome.

As we outline in Section 8.1.2, we consider three objectives to be optimized simultaneously, i.e., *a size of bicluster* — to discover the appropriate size of the bicluster (not so big – too general, or not so small – too specific), *a coherence of bicluster* — to consider even the noise in biclusters introduced by the elements, and finally *a semantic similarity* — how the elements in the bicluster are biologically related.

To approximate the size of bicluster B of the original (input) binary matrix M with r rows and c columns, we introduce the function as follows:

$$f_{cor}(B) = -\sum_{i=1}^{r}\sum_{j=1}^{c}(b_{ij} \times m_{ij}) \tag{37}$$

Here, we use the notation presented in this chapter in Section 8.2.3. A bicluster B is defined by row and column vectors where the outer product of these two vectors formed a binary matrix. 1's in the matrix denote that the elements belong to the bicluster (see Example 13). The function f_{cor} returns a number of elements that are common for both matrices, for the bicluster B and for the original input matrix M. Here, we do not use the expected meaning of the expression "size of bicluster". Conventionally, the size is given as a multiplication of numbers of rows and columns. However, we focus on the number of 1's because of the subsequent *Model induction* phase where the set of targeted examples is given by the elements of the matrix with its value equal to 1. Moreover, since NSGA-II supposes to minimize objective functions, the minus sign was added into the formula.

The second function that captures the bicluster incoherence computes a number of elements with the different value of the bicluster B and the original input matrix M.

$$f_{xor}(B) = \sum_{i=1}^{r} \sum_{j=1}^{c} |(b_{ij} - m_{ij})| \tag{38}$$

Finally, the third function evaluates the semantic similarity of elements that occur in the bicluster B. All ontology terms associated with elements of the bicluster are separated based on their ontology membership and assigned as an element in a set $\mathcal{O}$. The final similarity measure is given as an average value of the semantic similarity across terms in the particular ontology. The term similarity across different ontologies is not considered since the similarity is equal to zero indeed. The formula is shown below. We note that $\overline{SimUI}$ is an average value of $SimUI$ across terms in elements of the set $\mathcal{O}$.

$$f_{sim}(B) = \frac{\sum_{s1 \in \mathcal{O}} \sum_{s2 \in \mathcal{O}, s1 \neq s2} \overline{SimUI}(s1, s2)}{2 \times \binom{|\mathcal{O}|}{2}} \tag{39}$$

In summary, we optimize the size and coherence of values in a bicluster calculating f_{cor} that reflects the number of correctly classified elements in the bicluster and f_{xor} that reflects incorrectly classified elements. To maximize the objective f_{cor}, we append the minus sign since all objectives in NSGA-II are minimized. Using these two functions simultaneously, it enables to form biclusters with reasonable size and their coherence. Moreover, we take into account the semantic similarity of terms that leads to form a bicluster that is biologically coherent.

For the sake of simplicity, we firstly considered the *conventional encoding*, i.e., a chromosome represents one solution and it encodes one bicluster. Let b be a bicluster, then the objectives are computed easily as follows:

$$cor = f_{cor}(b)$$

$$xor = f_{xor}(b)$$

$$sim = f_{sim}(b)$$

For the *extended encoding* version, the situation is a bit complicated. Intuitively, it is possible to extend the list of objectives and thus suggest optimizing the objectives by functions f_{cor}, f_{xor}, and f_{sim} for each individual bicluster encoded in the chromosome. However, this leads to $3 \times k$ objectives to optimize, where k is the number of biclusters encoded in the chromosome. In the case of Pareto optimization, the final Pareto set is usually large in our case. To simplify that, we suggest reporting only 3 objectives where the corresponding values of the objectives are computed as the average values of the corresponding objectives across each bicluster in the chromosome. Simply put, the objectives report an average property over k biclusters encoded in the chromosome.

8.2.6 *Improving the initialization of population*

In the original NSGA-II algorithm, the initial population is generated randomly from an uniform distribution. In our concept of biclustering, this means that values of rows/columns encoding a bicluster are randomly set up to 1 or 0 with the same probability.

However, as we observed, using this approach converges to satisfactory solutions disproportionately slowly. The reason for the slow convergence is evident. Especially for a sparse matrix, the original initialization method generates individuals of the population which focus on a few 1's or, in the extreme case, on none of them. To avoid searching in the space of unpromising solutions and simultaneously do not prune the search space too much to keep the diversity of the population, we evolved a simple heuristic initialization approach that satisfies these requirements and speeds-up the convergence.

The heuristic initialization approach has the following steps:

1. Split the population into 2 halves.

2. Individuals in the first half are generated randomly from an uniform distribution (the same way as in the NSGA-II).

3. Individuals in the second half are generated randomly but with a nonuniform distribution. The value of rows (resp. columns) encoding a bicluster is set up to 1 if a random value $p_r \in [0, 1]$ is smaller than a value p_h. Otherwise, the row (resp. column) is set up to 0. The p_h is computed as a number of 1's in the corresponding row (resp. column) divided by a number of elements in the row (resp. column). This eliminates rows (resp. columns) with a few or none of 1's and favorizes rows (resp. columns) with many 1's; these are potentially interesting.

In summary, a set of Pareto optimal solutions has been found by the multi-objective optimization algorithm NSGA-II where an individual solution corresponds to a bicluster (the original encoding) or a set of biclusters (the *extended encoding* version). To achieve sufficient results that reflect biological relevance and measured gene expression, we suppose to use the following functions: f_{cor}, f_{xor}, and f_{sim}.

After the Pareto set of biclusters is established, the process of model construction follows. This phase is described in the following section.

8.2.7 *Model induction*

Since the extremely fast method for multiple rules induction called *sem1R* has been developed in our previous work described in Chapter 5, currently it is time feasible to run the *sem1R* algorithm and induce a rule set for each particular bicluster in the Pareto set. This was not possible until the rapid algorithm exploiting biological ontologies has been developed. However, this brings a new problem of how to work with all induced rules; how to combine them and build a global model. For this reason, a new procedure that would build an arbitrary global model needs to be defined.

To establish a convenient approach, we follow a generic Lego framework [86] that utilizes local pattern discovery techniques for global modeling. A graphical scheme illustrating particular phases of the framework is depicted in Figure 35 where local patterns are discovered in a data source by local patterns discovery techniques, e.g., using *subgroup discovery* techniques. Then, the local patterns are preselected using a pattern set selection technique and upon these patterns, a global model is built.

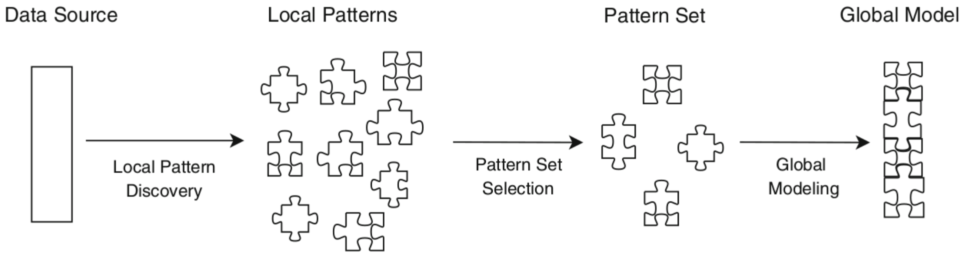


Figure 35: Lego framework stems from [53].

To adapt the generic Lego framework to the specific semantic biclustering task, we redefine and adjust the original meaning of each Lego phase from Figure 35 as the following:

1. **Local Pattern Discovery** discovers a set of local patterns from data. To generate a set of rules that describe local patterns in omics data in general, we firstly generate a Pareto optimal set of biclusters that are coherent enough not only by its values (in our work typically gene expression profiles) but semantically as well. Since the multi-objective optimization is applied, the output Pareto optimal biclusters would be easier to be described by ontological terms because particular elements of biclusters are "sufficiently" similar when semantic is taken into account. For example, each bicluster can contain only functionally similar genes regarding Gene Ontology. Afterwards, the *sem1R* algorithm is run for each bicluster to induce a set of rules describing local patterns.

2. **Pattern Set Selection** reduces a set of local patterns. Since some discovered biclusters might be very similar and then the rules describing them might have the same form, we simply propose to remove the identical rules from the pattern set.

3. **Global Modeling** constructs the final classification model over the pattern set that describes the input omics data. To build an accurate and easily interpretable model, we transform the induced rules and the input data into the *attribute-value* representation. This enables us to use a well-established machine learning framework, e.g. WEKA [176], for model construction.

   The machine learning *attribute-value* representation is expressed by 2-dimensional matrix with a finite number of rows and columns, where a row generally represents an example and a column represents an attribute, also called a feature. In considering the semantic biclustering task, the attribute-value matrix is a binary matrix where an example corresponds to an element of the input binary matrix and an attribute corresponds to a rule that is a part of the pattern set. If the given example is covered by a rule, then the value 1 is assigned to the corresponding position in the attribute-value matrix, otherwise zero is assigned. We note that in the case of WEKA, the attribute-value matrix binds an extra column that represents a membership to the target class. Here, the target class contains such elements which their value is equal to one in the input binary matrix. If an element belongs to the target class, then the value 1 is assigned to the last column of the corresponding row position. Otherwise, 0 is assigned to that position.

8.3 EXPERIMENTS

To evaluate the overall performance of the proposed algorithm on the datasets that are established throughout the thesis, we decided to test the performance of the algorithm in two separate levels. The separation brings a better option to explain the impact of the algorithm regarding the ability to induce a model which should be able to generalize sufficiently and prevent overfitting. We test the following:

1. **Bicluster forming** An ability to reveal homogeneous biclusters that are simultaneously semantically coherent.

2. **Global model construction** An ability to induce a set of ontology rules describing biclusters and an ability to construct the global model that prevents overfitting.

At this moment, for practical reasons, we disregard DOT and m2801 datasets from the experimental evaluation because the expected runtime of constructing a predictive model is longer than for DISC dataset. This is evident from the experiment conclusions presented in Chapters 4 and 5. Furthermore, the predictive accuracy for DISC is lower than for the other dataset as is shown in Table 7. Then, any improvements in the predictive capabilities of the new model could be more noticeable.

Since the ability to build a good predictive classification model is associated with an ability to predict correctly previously unknown data, we follow the evaluation procedure presented in Section 4.3 where datasets are splitted into train and test datasets. Because AUC is used as the main criterion for the prediction accuracy of the global model, the same optimization criterion, i.e. AUC, was chosen for the first phase which forms biclusters. ACC and F1-score were disregarded in our experimental protocol, although the framework enables to used them as the optimization criterion.

Algorithm	Parameter	Value
NSGA-II	popsize	240
	ngen	2500
	kbics	1,2,3,4,5
	pcross	0.7
	pmut	0.01
sem1R	objective	auc
	nrules	1,2,3,4,5
	minLevel	2
	ruleDepth	4

Table 17: Parameters of NSGA-II and *sem1R* used in the experiment. The meaning of parameters can be found in [37] or [103], respectively.

Parameters of the algorithms that were used in all experiments are depicted in Table 17. The parameters of NSGA-II were chosen to get a sufficient number of diverse biclusters in feasible runtime. For *sem1R*, the parameters were chosen to obtain easily interpretable hypotheses in a reasonable runtime. Basic characteristics of biclusters from the first step, as the bicluster forming step, are shown in Figure 36. Figure 36 A shows a distribution of values representing a portion of overlapping elements between biclusters. For example, 0% of overlapping clusters means that there is no common element in any bicluster appearing in a Pareto set. In opposite, 100% overlapping clusters means that all biclusters in the Pareto set are identical. This distribution is measured for a *kbics* parameter (the number of biclusters in chromosome) which values are equal to a numeric range 1-5. Here, the *kbics* parameter helps to discover biclusters with more various elements across the biclusters of the final Pareto set.

Figures 36 B-D show a distribution of scores of *fcor*, *fxor*, and *fsem* functions as optimization criterions, respectively. In summary, *kbics*= 1 leads to bigger and significantly overlapping biclusters with higher semantic similarity.

However, highly overlapping biclusters contain redundant elements for which the algorithm must find their description. At least, finding a description for the same elements does not bring a piece of new helpful knowledge in the context of classification. In other words, building a global model which captures only a small portion of the total aspects of the input data does not help to improve the classification accuracy at all. Simply, many other relevant elements are omitted. This proposition is supported by experimental results which are depicted in Figure 37 A.

Figure 37 A shows AUC on the test data for four different machine learning algorithms that were used for building a global classification model. The algorithms are the following: J48, JRip, OneR, and Random tree. All of them are implemented and presented in WEKA. The other shown parameters are *kbics* and *nrule* parameters. The first one represents a number of rules that are encoded into one solution in the bicluster forming phase. The second one determines a number of rules that are induced for one bicluster in the global model constructing phase. For all machine learning models, we use the default parameters.

OneR algorithm [176] is a simple classification algorithm which selects the rule with the smallest total error so the global model is defined by only one rule. As is shown in Figure 37 A, even the rule with the smallest total error does not report a sufficient capability of generalization. The resulting AUC on the test dataset very slightly goes beyond 0.5 in a few cases. The smallest AUC is equal to 0.43 and the highest is equal to 0.51. In summary, it can be stated that OneR is not a suitable method in that task, since only one rule has not enough generalization ability. For inducing more sophisticated hypotheses having a potential to improve the overall AUC, a more complex machine learning model has to be used.

The other machine learning algorithms such as J48, JRip, and Random Tree were chosen for comparison with the results presented in Chapter 4. As in the previous case of OneR algorithm, the results for J48, JRip, and Random Tree are depicted in Figure 37 A. From these algorithms, the worst AUC is achieved by JRip where the highest AUC is 0.55 for *kbics* equal to 1 and *nrule* equal to 3. In comparison with the results of JRip in Table 5, the new method inspired by multiple objective optimization does not bring better results. On the other hand, J48 algorithm reached the overall highest AUC 0.69 for *kbics* equal to 3 and *nrule* equal to 4. This AUC outperformed the corresponding results of our original work presented in Table 5. Moreover, Random Tree algorithm achieved better results, in general, compared to the corresponding results in Table 5.

Besides the classification accuracy, the other important aspect of machine learning algorithms is their runtime, especially for time-exhaustive tasks. For this reason, we measured the runtime of the proposed algorithm for each important algorithm step separately. Comparison of the cumulative runtime is shown in Figure 37 B. *NSGA-II POS* (resp. *NSGA-II NEG*) denotes a step of bicluster forming using positive (resp. negative) examples. *sem1R POS* (resp. *sem1R NEG*) denotes a step for finding descriptions of biclusters formed in the previous *NSGA-II POS* (resp. *NSGA-II NEG*) step. *model preparing* denotes a step that converts C++ data structures into ARFF, a native format of the data mining tool WEKA. Finally, *model building* denotes a step of building a specific machine learning model. Clearly, the most time-consuming phase is for description finding, i.e., *sem1R POS* and *sem1R NEG*. From the measured parameters, *kbics* has the most negative impact on the run-time because the total number of biclusters, which must be described using the *sem1R* algorithm, is increased by *kbics* times. In comparison with the runtimes of the previous work depicted in Table 8 and 9, the algorithm which is presented here does not outperform neither *bi-directional enrichment* nor *rule and tree learning* approaches.

The time complexity of building a machine learning model is associated with the number of examples/instances and a number of features. The numbers of features that are used to build the global model according to *nrule* parameter are shown in Figure 37 C. Note that redundant features, i.e., rules which are in equal form, are removed from the feature set. For this reason, the values represented by the dashed orange line are not three times greater than the corresponding values lying on the green line.

8.4 CONCLUSION

Considering the main characteristics of the original *bi-directional enrichment* and *rule and tree learning* algorithms that were presented in Chapter 4 and taking into account their disadvantages, we developed a new algorithm having the potential to improve the overall classification accuracy and interpretability of the results. Although we tested the performance of the new algorithm using only Drosophila imaginal disc dataset, we can conclude that the new algorithm does not significantly outperform both *bi-directional enrichment* and *rule and tree learning* algorithms at once. In spite of the better interpretability of results, the classification accuracy is not significantly higher than for the methods from our previous research. Additionally, runtimes dramatically grow up. This is caused by a large number of biclusters for which the process of induction is started. In spite of the negative conclusions, we consider the approach of combining multi-objective optimization with the rule learning algorithm *sem1R* very interesting still having the potential to overcome the current state-of-the-art results.

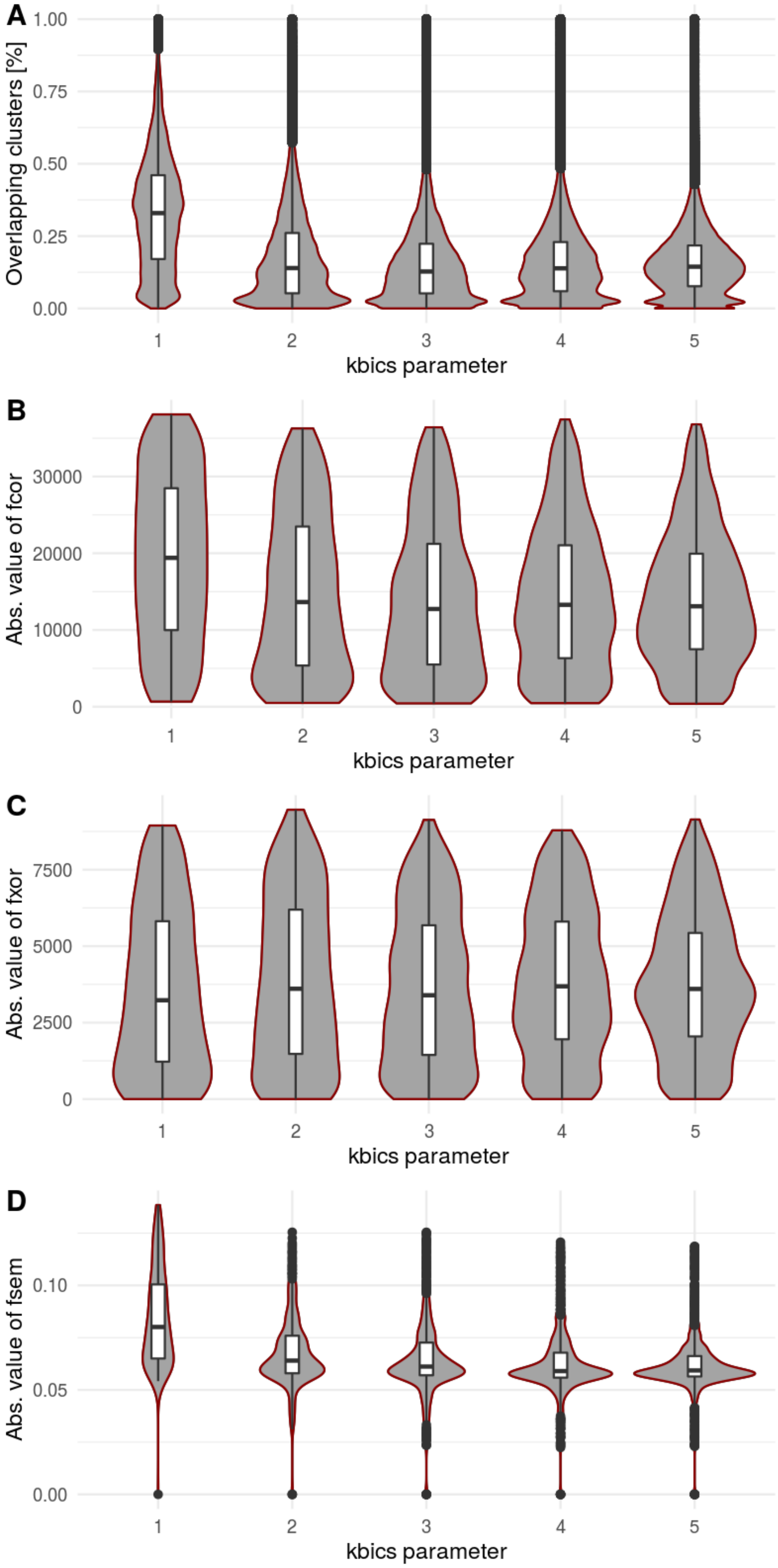


Figure 36: Violin plots showing a percentage of overlaps in biclusters, absolute values of fcor, fxor, and fsem functions for different values of *kbics* parameter in DISC dataset.

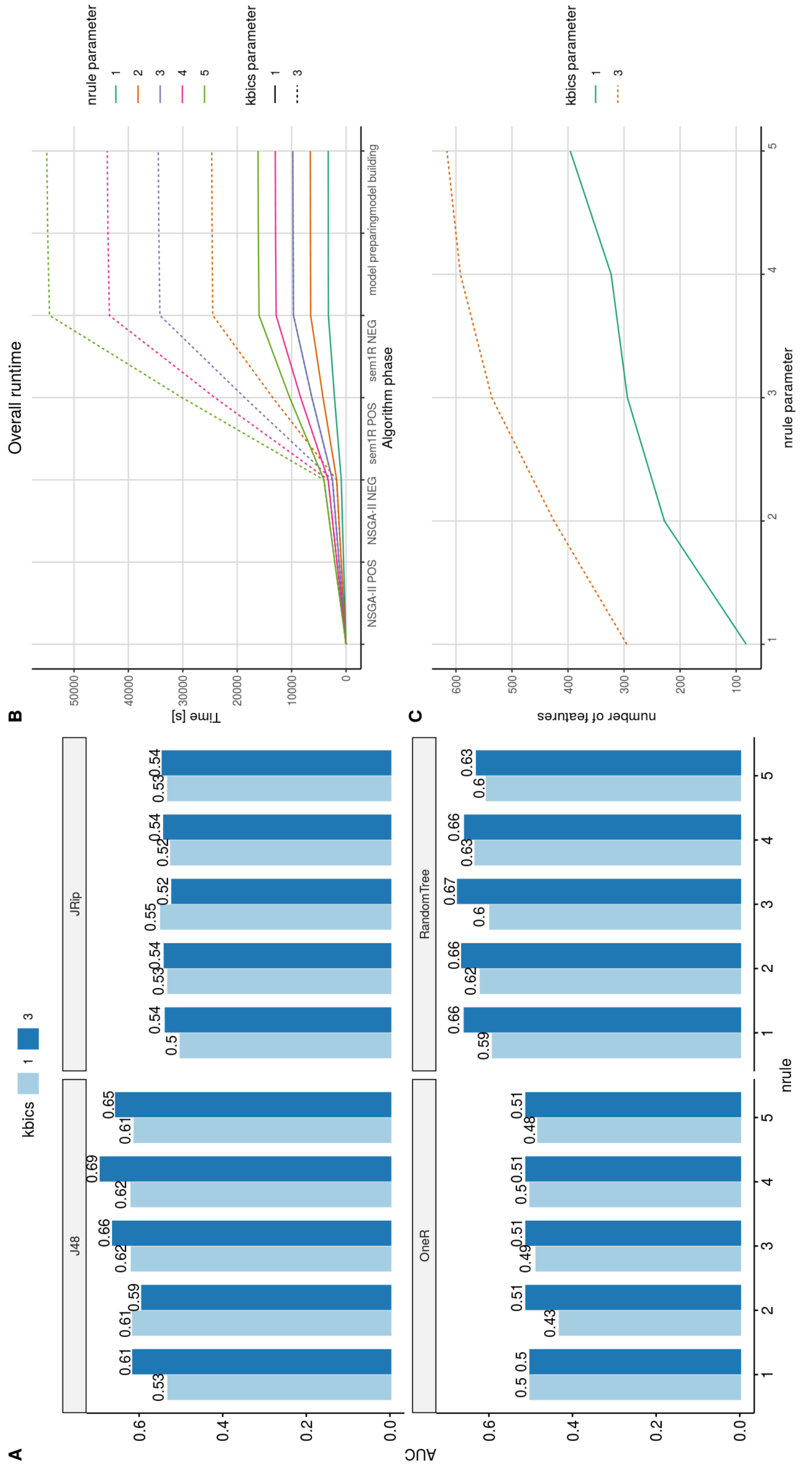


Figure 37: **A** Comparison of the algorithm performance on the test dataset. **B** Cumulative runtimes of the algorithm for various parameters. **C** Numbers of features according to nrule parameter. All depicted results are related to DISC dataset.

9 TOWARDS LARGE UNTARGETED LC-MS DATASETS FOR SEMANTIC BICLUSTERING: BATCH ALIGNMENT VIA RETENTION ORDERS

Although the previous chapters and also most of our work have been focused on mining patterns from biological data and their following interpretations or evaluations, here, we have changed the focus to a data preprocessing phase. In this case, we use untargeted metabolomics data using liquid chromatography–mass spectrometry technique. One of the reasons why we address the issue of preprocessing such kind of data is their potential to gather an enormous number of features and samples collectively formed into a matrix. This extensive matrix is a good candidate for finding biclusters using semantic biclustering approach. However, a suitable preprocessing method for handling liquid chromatography-mass spectrometry data has to be invented before adjusting semantic biclustering onto this specific application. In this chapter, we introduce such a method. Unfortunately, creating a suitable large-scale dataset is time-consuming and therefore the concrete application of semantic biclustering approach on liquid chromatography–mass spectrometry data has not been mentioned in this thesis. This issue is left for future research.

This chapter has been created with the cooperation of scientists from *Czech Centre for Phenogenomics*. I give my thanks to them. Note that this work is being considered for publication as [104].

Our present work introduces two algorithms that address the problem of aligning and combining individually preprocessed batches in multi-batch LC–MS data, taking into account the existence of retention order swaps. These algorithms help minimize information loss during the preprocessing of individual batches. The first algorithm consists of two phases, constructing a global feature alignment and a subsequent correction step that merges retention order swaps. The second algorithm incorporates these two phases into one using a decomposition into subsequences of length k, known as k-mers. Algorithms were tested on six sets of simulated and six sets of real datasets.

9.1 BACKGROUND

Untargeted metabolomics is a widely used strategy in disease biomarker discovery, metabolic profiling, and metabolic pathway studies. Unlike targeted metabolomics where only a predefined set of known metabolites are the focus of analysis, untargeted metabolomics emphasizes the study of the global metabolome by measuring ions from thousands of metabolites within a wide mass range [99, 171]. One of the most common techniques in untargeted metabolomics is liquid chromatography–mass spectrometry (LC-MS), offering high sensitivity and broad metabolite coverage [46].

Monitoring and investigating changes in metabolites and identifying evolving biomarkers over time, which is made possible by large-scale multi-batch LC–MS experiments, provide indispensable insights in such studies as cancer development and aging [154, 172]. As large-scale LC–MS studies comprise hundreds of samples, it is infeasible for them to be measured in a single LC-MS run, and for this reason such experiments are commonly divided into several batches and span long time periods. Over time, runs become susceptible to dramatic mass, retention time, and intensity shifts due to sensitivity to random effects and external factors [15, 174]. Obtained results may then show a large variance even when repeated on the same analytical platform or machine. Although problems such as intensity drift and retention time shift can be more pervasive in multi-batch experiments [130], certain LC–MS problems, such as elution order and retention time swap, are pronounced also on a run-to-run basis and accordingly in single-batch experiments, and hence not specific to large-scale multi-batch studies. Elution order and retention time swaps (hereafter collectively denoted as *retention order swap* in this thesis) are well-known characteristics of LC–MS data and prevalent in untargeted experiments [92, 93, 149, 152]. Although several algorithms are available to tackle the resulting side issues, several aspects of the problem remain unaddressed. In [148, 149], the authors have demonstrated that existing algorithms either hold the incorrect assumption that elution order is preserved across runs or fail to account for problems that arise from elution order swaps after alignment and data processing (e.g. 'distortions reversing the elution order'). Since retention order swaps occur in single-batch experiments as well as across different batches in multi-batch experiments, this places a high demand on improving existing sequence alignment algorithms that are commonly used to address the peak correspondence problem to also take into account the effects of retention order swaps [149].

LC-MS datasets are analyzed as a single batch in most preprocessing pipelines (e.g. XCMS [146] and MZmine [81]) where the user selects a set of parameters that are fitting for all samples. In large-scale multi-batch experiments, issues such as between-batch variation and retention time shift make it especially difficult to find a set of parameters that are equally fitting for samples in all batches. Such an approach may also result in information loss in some batches. As an example, when signal-to-noise ratio (S/N) varies among batches, choosing a large S/N can result in skipping small peaks in some batches with lower background noise, whereas a low S/N may result in picking lots of background noise in others. Preprocessing batches separately, with batch-specific parameters can alleviate said problems. However, for further downstream analysis (e.g. in MetaboAnalyst [22, 23]), algorithms for aligning and combining batches are required.

Conventional LC-MS alignment algorithms aim to solve the correspondence problem across all samples of a multi-batch experiment at once. In other words, such algorithms do not take the multi-batch design into account and do not align the batches. Examples of such algorithms are OBI-Warp [134] and MetAlign [100]. Several other alignment algorithms are examined in [149]. However, none of these algorithms aim to find corresponding peaks among individual batches in multi-batch experiments.

To the best of our knowledge, probably the first and most significant attempt to monitor and correct between-batch variability in large-scale untargeted LC-MS metabolomics data has been introduced by [15]. The authors similarly to us work with feature alignments between batches. To reduce between-batch variability, features are merged according to their mutual correspondence specified by user-defined parameters. However, the method proposed by [15] does not implicitly allow the user to preprocess each batch individually with a specific set of parameter values that are tuned according to individual batch property. That is to say, [15] exploits one aggregated multi-batch feature matrix. Evidently, the necessity to set the global parameters that fit all batches might be problematic. For this reason, the proposed methods avoid this multi-batch processing. Moreover, we extend their work by proposing a method for estimating retention order swaps, deletions, and insertions in misalignments between batches in authentic datasets.

9.2 MATERIALS AND METHODS

In this section, we first define the notations and clarify the terms used in this work to avoid misunderstanding as many terms, such as peak and feature, are used interchangeably in literature and the intended meaning might be unclear to the reader [147]. We then present two approaches to combining LC-MS batches in a pairwise fashion and further formulate a heuristic to accommodate multiple alignments.

9.2.1 *Notation and Definitions*

We define the LC–MS *batch experiment* B as a set of m samples; $B = \{S_1, S_2, \cdots, S_m\}$. As a common approach, the raw data of such unprocessed batch experiment is transformed onto a 2D matrix of processed data $\mathbb{L}_B = \{(f, s) : f \in F, s \in B\}$, where F is the set of features that are associated with the samples in B. It is not possible to interpret the experiment and validate results without defining associations between features and samples. Assuming that $\mathbb{L}_B$ consists of n features, m samples, and each element $\mathbb{L}_{B i,j} \in \mathbb{R}_{\geqslant 0}$ represents the intensity of feature i in sample j, then the transformation from B to $\mathbb{L}_B$ can be done via one of the many available LC–MS preprocessing tools which typically match chromatographic peaks across samples by solving a peak correspondence problem ensuing peak detection and alignment.

A feature is formally defined as a tuple of two values: m/z and retention time. We also introduce the binary relation $\leqslant_{RT}$ to denote the mutual order of features given by their retention time order. Accordingly, assume a set of features F with a totally (linearly) ordered set

$\leqslant_{RT}$ that encodes the more general relation. Note that the feature is associated with one or more chromatographic peaks in the raw data. This means that mass-to-charge ratio (m/z) and retention time (rt) values of a feature are given by the aggregation (e.g. median) of corresponding values from associated peaks. From this point onward, we will use the term *feature*, instead of peak, to emphasize that we work with already preprocessed LC-MS data, denoted as $\mathbb{L}_B$.

An LC–MS experiment may comprise several batches; each measured by the LC–MS instrument at different time points. Experiments with multiple batches include information about the time (date) of the run for each batch. We can think of every batch as an experiment on its own which can be defined as a set of samples according to our earlier definition of batch experiment B. Thus, we define the set of batches of a multi-batch experiment as a totally ordered set $\mathbb{E} = (E, \leqslant_T)$, where E represents a set of batch experiments and $\leqslant_T$ is a binary relation over the times when the batches were measured.

9.2.2 *Peak Correspondence*

Peak correspondence can be formulated as the problem of matching detected chromatographic peaks across samples and occasionally within samples for adjacent peaks. We consider peaks to be in *correspondence* when their m/z values lie within a predefined threshold of one another and elute in a predetermined overlapping time window.

The same problem with similar consequences also needs to be resolved on a feature-level basis—we refer to it as the *feature correspondence problem*. According to the relation $\leqslant_{RT}$, each two features are comparable and as a result, features in $\mathbb{L}_B$ can be ordered sequentially. The feature correspondence problem can be, thus, transformed into a sequence alignment problem by reorganizing features into a sequence of features, where the feature correspondence method takes this ordered sequence as input. Smith-Waterman [150] and Needleman-Wunsch [120] sequence alignment algorithms align protein and nucleotide sequences on the basis of substitutions, insertions, and deletions, where the latter two introduce gaps in the aligned sequence. Adapting these algorithms for LC–MS data, where feature transpositions, and likewise elution and peak order swaps, are present, would not yield an ideal alignment without alternative scoring functions [132] or additional computational steps, such as warping functions [134].

Conventional sequence alignment methods, which normally approximate the correspondence problem, do not reflect the real properties and characteristics of LC–MS data and thus do not take into account retention order swaps.

9.2.3 *Algorithms*

We propose two algorithms for aligning and aggregating batches into a 2D matrix for subsequent downstream analysis. Both algorithms employ dynamic programming and approximate the correspondence problem with respect to sequence alignment. The first proposed algorithm separates the process of alignment construction and retention order correction from each other, whereas the second algorithm integrates these two steps together. As both algorithms operate on the principle of pairwise alignment, we employ a heuristic approach inspired by progressive ([47]) and iterative alignment methods ([41, 71]) to accommodate multiple alignments.

9.2.3.1 *Pairwise rtcorrectedAlignment*

Taking cues from global alignment, Algorithm 9 operates in two phases: 1. global alignment is performed (Needleman-Wunsch algorithm); 2. retention order swaps are corrected via the rtCorrection function.

The *globalAlignment* function constructs a pairwise alignment using the global approach and supposes two linearly ordered feature sets on input. Source feature set src is aligned to target feature set tgt, both feature sets referring to input batches. This function returns an alignment and assigns it to the *align* variable. Since every two features are comparable due to the relation $\leqslant_{RT}$, a linearly ordered sequence of features can be used as the input for the conventional dynamic programming approach. Features in sequences are considered as *matched* if their corresponding m/z values are sufficiently similar, otherwise features are considered *unmatched*.

Algorithmus 9 : Pairwise rtcorrectedAlignment algorithm

```
input  : $src$, $tgt$, $n_{RT}$ // two feature sets referring batches, window
         size parameter
output : $matrix$ // feature matrix in $\mathbb{L}_B$ format

Function rtCorrection($align$, $n_{RT}$):
    $i \leftarrow 0$
    foreach $elm \in align$ do
        $w \leftarrow$ getSubset($elm$, $i$, $n_{RT}$) // correct. window
        if $elm$ in $w$ then
            // merge matched features in the window
            $alignment \leftarrow$ mergeFeatures($align$, $i$, $n_{RT}$)
        $i \leftarrow i + 1$
    end
    return $alignment$
End Function

// alignment of source and target feature sets
$align \leftarrow$ globalAlignment($src$, $tgt$)
// retention time correction with $n_{RT}$ value
$alignment \leftarrow$ rtCorrection($align$, $n_{RT}$)
// building the feature matrix from the alignment
$matrix \leftarrow$ buildMatrix($alignment$)
return $matrix$
```

Since conventional sequence alignment approaches primarily do not solve the issue of retention order swaps, the constructed align-

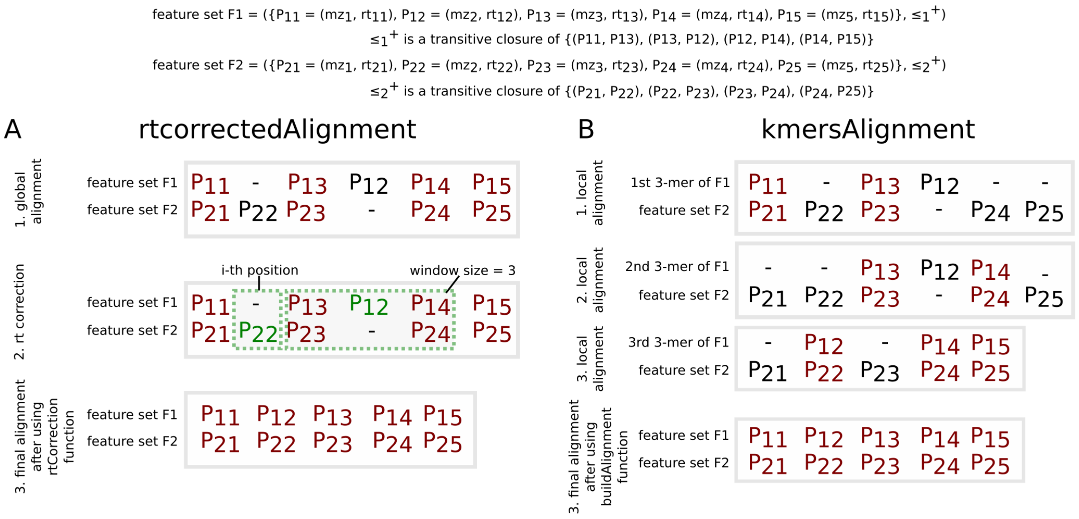


Figure 38: An example demonstrating the modus operandi of the two proposed approaches; *rtcorrectedAlignment* and *kmersAlign*. Both algorithms align feature sets F1 and F2 taking into account retention order swaps. **A** shows two phases of the *rtcorrectedAlignment* algorithm, i.e. making a global alignment and then correction by *rtCorrection* function with window size $n_{RT} = 3$. **B** demonstrates the usage of *kmersAlign* algorithm. Three 3-mers of F1 are locally aligned to F2 and then combined to form one final alignment.

ment needs to be corrected via the *rtCorrection* function, which requires two inputs: a global alignment ($align$) and window size parameter (n_{RT}). The correction algorithm is described in Algorithm 9 in lines 1-11. For each element[1] of the input alignment at unique position i, a subset w that contains elements at positions $i+1, \cdots, i+n_{RT}$ of the ordered alignment is considered. If the element at position i occurs in the subset w, then features with sufficiently similar m/z values are merged into one feature at the position i. Otherwise, i is incremented by one. The process of merging features is represented by the function *mergeFeatures* and the corrected alignment is appended to the end of the $alignment$ vector. These steps are repeated until end of the alignment is reached. Finally, function *buildMatrix* transforms the final alignment onto a 2D matrix in $\mathbb{L}_B$ format.

An illustrative example of the pairwise *rtcorrectedAlignment* algorithm is depicted in Figure 38 A where the initial global alignment of two feature sets, F1 and F2, and the final resulting alignment achieved by the *rtCorrection* function with window size $n_{RT} = 3$ are shown.

9.2.3.2 *Pairwise kmersAlignment*

Global alignment is susceptible to generating incorrect alignments, especially when sequences contain repeated subsequences. Furthermore, choosing an appropriate threshold for the retention order correction parameter n_{RT} is an equivocal task which can dramatically impact the final alignment. We have developed the pairwise *kmersAlign-*

[1] Element of the alignment can be seen as a pair of corresponding features of the alignment.

ment algorithm, which avoids the necessity of separating alignment and retention order correction phases, to address the aforementioned problems. This algorithm takes retention order swaps into account and does not assume that they are rare in multi-batch LC–MS data.

kmersAlignment (Algorithm 10) requires as input two linearly ordered feature sets, $\mathtt{src}$ and $\mathtt{tgt}$ both refering to input batches, and a $w_{\mathtt{size}}$ parameter. First, all possible *k-mers* of the linearly ordered feature set $\mathtt{src}$ are generated by the *createKmers* function where the length of the k-mer is determined based on the input $w_{\mathtt{size}}$ parameter. Each *k-mer* is then aligned with the linearly ordered target $\mathtt{tgt}$ feature set using the Smith-Waterman local alignment algorithm represented by the *localAlignment* function. As each *k-mer* partially shares a common fragment with the previous one, usage of *k-mers* allows the algorithm to generate and evaluate all possible variations of features near the given k-mer and take into account feature transpositions. All alignments are assigned to the *aligns* variable.

At a later stage, the algorithm combines all different alignments into one which comprises only features that are matched (in correspondence) at least in one local alignment. If a feature can be assigned to several different positions in $\mathtt{tgt}$ with regards to the used k-mer then the feature is assigned to a position according to the majority voting rule. For an ambiguous decision scenario where majority voting rule does not apply, the feature is assigned to a position in the alignment that is closest to its original feature position in the $\mathtt{src}$ feature set. Ultimately, unmatched features are appended to their corresponding position in the alignment. The process of combining various alignments into one final alignment is represented by *buildAlignment* function. Similar to Algorithm 9, function *buildMatrix* transforms the final alignment onto a 2D matrix in $\mathbb{L}_B$ format.

An example making use of the *kmersAlignment* algorithm is shown in Figure 38 B where three 3-mers are aligned to the reference feature set F2 and then combined into one final alignment.

9.2.4 *Biological data*

Each real data set contains data from plasma analysis from 16-week old WT mice (strain C57Bl/6NCrl) and KO (same genetic background as WT) mice, where one gene was ablated. The name of the dataset represents the ablated gene. There are always two biological groups in the datasets. Information regarding the number of samples and features in each dataset is shown in Table 18. Biological difference between groups of the mice is above scope of this article.

9.3 RESULTS

To evaluate and test our proposed algorithms, we used a set of real and synthetic datasets. Since it is impossible to accurately determine the true number of retention order swaps in real LC–MS data, the utilization of synthetic data with known qualities and properties al-

Algorithmus 10 : Pairwise kmersAlignment algorithm

```
input  : src, tgt, $w_{size}$ // two feature sets referring batches, k-mer parameter
output : matrix // feature matrix in $\mathbb{L}_B$ format
// create a set of k-mers from src
allkmers ←createKmers(src, $w_{size}$)
aligns ← ∅ // empty set of alignments
// local alignemnts of each k-mer and tgt
foreach kmer ∈ allkmers do
    align ← localAlignment(kmer, tgt)
    aligns ← aligns ∪ align
end
// combine alignments into the final one
alignment ← buildAlignment(aligns, tgt)
// building the feature matrix from the alignment
matrix ← buildMatrix(alignment)
return matrix
```

		# of features	# of samples (WT/KO)
1	Klk8	3,459	24 (12/12)
2	Tmem60 C18	5,138	23 (8/15)
3	Tmem60 C30	8,562	23 (8/15)
4	Trim9	3,860	25 (14/11)
5	Wiz C18	8,737	20 (6/14)
6	Wiz C30	403	20 (6/14)

Table 18: First column shows the number of features in the final feature matrix generated by xcms for each dataset. Second column shows the total number of samples followed by the number of WT and KO samples in brackets.

lowed us to assess the quality of the final alignment and better compare and study the limitations and differences of the two algorithms under various settings.

9.3.1 *Synthetic data*

We used authentic LC-MS datasets from real experiments as the basis for generating synthetic datasets that accurately reflect the characteristics of LC-MS data and demonstrate realistic feature distributions across samples. Each authentic LC-MS dataset was randomly divided into two sub-datasets where both fragments had the same number of samples. Each sub-dataset was then pre-processed by the XCMS package and the results were used as references for generating its synthetic variants. The following operations were used to introduce noise in a reference sub-dataset to create a synthetic version of it: *insert feature*, *insert feature with existing m/z value*, *delete feature*, *swap feature*, and *do_nothing*. For each feature in the reference sub-dataset, in the order given by the relation $\leqslant_{RT}$, one operation is randomly selected from the uniform distribution and applied to the feature, where *insert feature* inserts a feature with a random m/z value, unique across the reference dataset, drawn from the uniform distribution in the interval of m/z values present in the real reference dataset; *insert feature with existing m/z* inserts a randomly selected m/z value already present in the reference dataset; *delete feature* skips the feature and does not insert it into the synthetic dataset; *swap feature* swaps the current feature with a feature whose position is given by a rounded number drawn from the normal distribution; and finally, *do_nothing* copies the current feature from reference data to the synthetic dataset without modifications.

To model synthetic datasets after real LC-MS datasets and control the degree of similarity between each reference dataset with its synthetic variants, we define the probabilities for each operation in advance. Probabilities of operations *insert feature*, *insert feature with existing m/z value*, *delete feature*, *swap feature* and *do_nothing* are denoted as p_{ins}, p_{inse}, p_{del}, p_{swap}, and $p_{nothing}$, respectively, with the sum of all probabilities being equal to one. The process of generating synthetic datasets from both sub-datasets of each of the six real LC-MS datasets (*Klk8 C18*, *Tmem60 C18*, *Tmem60 C30*, *Trim9 C30*, *Wiz C18*, and *Wiz C30*) was repeated ten times for each combination of probabilities of the aforementioned operations. To avoid unrealistic combinations that are unrepresentative of the properties of real LC-MS data, we restricted the interval of probabilities to [0,0.3] for each operation. Standard deviation of normal distribution reflecting a distance of feature swaps was set to 9.99 according to Klk8 dataset. Ultimately, all synthetic datasets were concurrently aligned by *rtcorrectedAlignment* and *kmersAlignment* against their corresponding reference real LC-MS dataset and the quality of the final alignments were evaluated. A schematic diagram representing the evaluation process is depicted in Figure 41 A.

To evaluate the performance and applicability of both proposed algorithms on synthetic data, we established a *distance score* D. This score quantifies the amount of differences in the two distributions of m/z values. In the case of synthetic experiments, we compare the m/z distributions of the expected ideal alignment and the alignment by proposed algorithms. This means that mishandling of retention order swaps are reflected in the proposed distance score. To bring the formal definition of D, we extend our formalism by multisets. Suppose two multisets $S1 = (M1, m1)$ and $S2 = (M2, m2)$ where S1 and S2 represents feature sets. M1 and M2 are sets of m/z values that are presented in the corresponding feature set. m1 (resp. m2) is a function from M1 (resp. M2) to the set of the positive integers, giving the number of occurrences. Given this, D is computed as follows:

$$D(S1, S2) = \sum_{x \in M1 \cup M2} (|m1(x) - m2(x)|)$$

We measured the performance of *rtcorrectedAlignment* and *kmersAlignment* algorithms and for each probability setting compared the results of the algorithms. This comparison on six different datasets is depicted in Figure 39 where each violin plot shows the distribution of differences in *kmersAlignment* and *rtcorrectedAlignment* distance scores for equal parameter settings (i.e. distance scores for the same probability setting for each algorithm were subtracted from each other). Details regarding outliers are provided in Supplementary Material of the original article [104]. In all cases, *kmersAlignment* was run with the same default parameters, i.e. $w_{size} = 5$. The default value of w_{size} parameter was determined in order to finish the run of the algorithm in a reasonable time and with an adequate score. To show the effect of w_{size} parameter on the performance of the algorithm and run time, the distance score D and run time of kmersAlignment was measured with w_{size} equal to 5, 10, 50, and 100 for the six datasets. The results in the form of violin plots are provided in Supplementary Material of the original article [104]. In summary, $w_{size} = 5$ reached a promising distance score and concurrently run times are the fastest. For this reason, the shortest runtime and good performance, we choose $w_{size} = 5$ as the default parameter.

We generally consider the performance of *rtcorrectedAlignment* on our synthetic datasets to be overfitted as the value of window size parameter n_{RT} was chosen for each dataset according to an approximated minimum distance score across all examined combinations of probabilities. The distance score was measured for different window sizes in the interval [0,300] for each combination of probabilities. These are reported in Figure 40 where curves represent general trends of distance scores according to the window size parameter n_{RT} for each dataset. The lowest point on each curve estimates the optimal window size for a particular dataset.

We used a linear mixed model for the estimation of effects p_{del}, p_{ins}, p_{swap} and p_{inse} and their double interactions on the difference of *kmersAlignment* and *rtcorrectedAlignment* scores. The model was applied on all synthetic datasets using the following formula:

$$\begin{aligned} F = p_{del} + p_{ins} + p_{swap} + p_{inse} + p_{del} * p_{ins} + \\ p_{swap} * p_{inse} + p_{del} * p_{swap} + p_{inse} * p_{ins} + \\ p_{del} * p_{inse} + p_{ins} * p_{swap} + (1|ID) \end{aligned}$$

where $(1|ID)$ reflects the random effect of different datasets. A positive value for F expresses that *rtcorrectedAlignment* has outperformed *kmersAlignment*, whereas a negative F expresses the dominance of kmersAlignment. The R package lmerTest ([91]) was used for this analysis.

Our linear mixed-effects analysis uncovered a strong inverse effect on F from the interaction p_{del}:p_{swap}, where with the increasing probability of p_{del} and p_{swap} the value of F becomes significantly smaller (p-value = 3.18E-8). We observed the same effect for the interaction p_{ins}:p_{swap} (p-value < 2E-16). On the other hand, the p_{ins}:p_{inse} in-

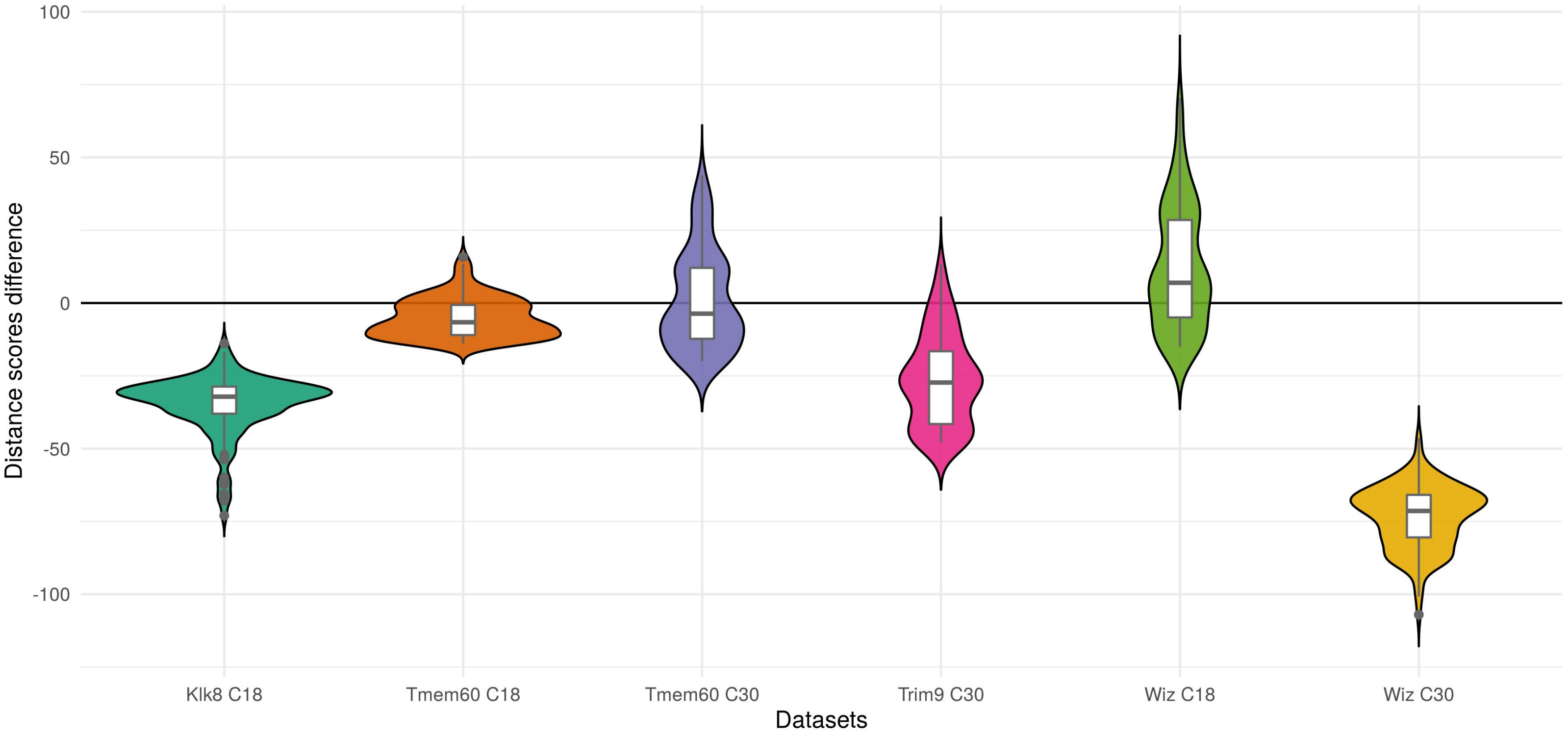


Figure 39: Distribution of differences between distance scores of *kmersAlignment* and *rtcorrectedAlignment* (with optimal window size n_{RT} used) for six synthetic LC-MS datasets each generated ten times using the same probabilities. Negative values denote that *kmersAlignment* achieves a better distance score compared to *rtcorrectedAlignment* and positive distance scores demonstrate the opposite case for a particular probability setting. Only in the case of *Wiz C18* dataset the *rtcorrectedAlignment* algorithm achieved a higher score in more than half of the combinations of probabilities compared to *kmersAlignment*. Outliers at the bottom of the violin plots are associated with a high value of $p_{swap} = 0.3$. On the other hand, outliers at the top share the same value of $p_{swap} = 0$.

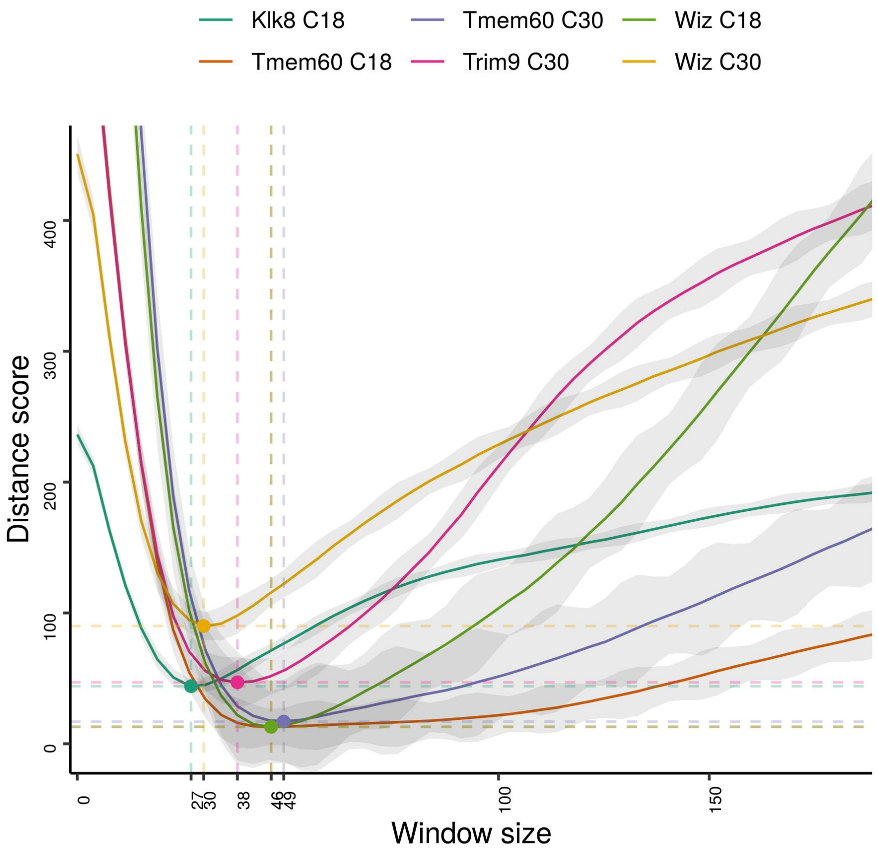


Figure 40: The distance score D according to the window size parameter of *rtcorrectedAlignment* algorithm computed for different combinations of probabilities. Curves illustrate the trends in each dataset, and were computed by LOESS smoothing. Dashed lines represent the minimum distance score for each curve and the corresponding window size.

teraction was shown to have a positive correlation with F, i.e. with a higher probability of p_{ins} and p_{inse} the value of F becomes significantly greater (p-value < 4.3E-5).

9.3.2 *Real data*

We used six independent sets of untargeted LC-MS experiments (*Klk8 C18*, *Tmem60 C18*, *Tmem60 C30*, *Trim9 C30*, *Wiz C18*, and *Wiz C30*) to assess the performance and applicability of our proposed algorithms on real-world datasets while also examining how well a 2D matrix constructed (aligned and combined) by the two algorithms from individually preprocessed batches would correspond to the same dataset preprocessed as a single batch. In all cases, the data were from LC-MS experiments that did not comprise multiple batches but were instead measured as a single batch. Every original one-batch experiment was first preprocessed by XCMS and the resulting 2D matrix of $\mathbb{L}_B$ format was taken as ground truth for the corresponding experiment. Afterwards, every original experiment was randomly divided into two parts (i.e. two sub-experiments or pseudo-batches) using stratified bootstrapping (the only criterion was the equal distribution of samples from biological groups between batches). This procedure was repeated ten times for every experiment. The two resulting sub-experiments were preprocessed by XCMS (using the same parameters used to preprocess the original experiment, i.e. ground truth) and the final 2D matrices (and linearly ordered feature sets) from the batches were aligned and aggregated into a final 2D matrix of $\mathbb{L}_B$ format by *rtcorrectedAlignment* and *kmersAlignment*. The evaluation process is graphically depicted in Figure 41 B. Ultimately, distance scores between the original matrix of one-batch experiments and the corresponding alignments/matrices by *rtcorrectedAlignment* and *kmersAlignment* were calculated. Distance score D here measures the ability of the algorithm to reconstruct the original one-batch experiment from its randomly divided and separately preprocessed parts. In other words, distance score D assesses how well the aligned sub-experiments correspond to the ground truth matrix, where a distance score of zero represents identical results. Furthermore, the distance score here reflects not only mishandling of retention order swaps but also inserted and deleted features. For this reason, here distance scores are generally much higher than in the case of synthetic experiments.

The results from all ten iterations of each real experiment were averaged and are reported in Table 19. For the *rtcorrectedAlignment* algorithm, we are additionally reporting the results for cases when window size parameter n_{RT} was set to 0, 27, 30, 38, 46, 49, 100, and 150. Except 0, 100, and 150, the rest of these numbers stem from our extensive experiments done on synthetic datasets shown in Figure 40. Here, 0 plays the role of baseline, and only a global alignment is considered. In every scenario, *kmersAlignment* either outperformed *rtcorrectedAlignment* or was extremely similar in performance to it except in the case of the Wiz C30 experiment where it was shown that the uncorrected experiment (i.e. without the application of *rtCorrection* function) was most similar to the ground truth.

9.3.3 *Batch property estimation*

The existence of retention order swaps in untargeted LC-MS data is one of the essential assumptions that we hold in this paper. However, it is important to also estimate the number of retention order swaps because they affect the alignment and an increase in the number of swaps complicate algorithm development. To estimate the number of retention order swaps in batches, we use the procedure already outlined in Section 9.3.2.

Number of insertions or deletions are easy to highlight as they are simply the number of features which have no corresponding features in the other dataset. To estimate the number of retention order swaps between a sub-experiment and the full experiment, we used the *rtCorrection* function (Algorithm 9). Value of the window size parameter n_{RT} was selected according to the minimal distance score D between

	kmersAlignment	rtcorrectedAlignment							
		$n_{RT} = 0$	$n_{RT} = 27$	$n_{RT} = 30$	$n_{RT} = 38$	$n_{RT} = 46$	$n_{RT} = 49$	$n_{RT} = 100$	$n_{RT} = 150$
Klk8 C18	578.4	1760.9	689.3	669.7	637.6	630.6	629.9	691.7	768.8
Tmem60 C18	2,085.5	3,397.8	2,547.9	2,491.3	2,380.6	2,303.6	2,280.2	2,149.2	2,123.7
Tmem60 C30	3,090.9	5,285.5	4,122.0	4,023.3	3,833.5	3,714.2	3,681.0	3,429.9	3,347.3
Trim9 C30	1,244.7	2,298.5	1,360.3	1,336.0	1,294.3	1,274.5	1,269.7	1,257.9	1,341.9
Wiz C18	1,524.5	4,256.4	2,130.9	2,043.2	1,886.4	1,779.9	1,746.5	1,540.5	1,521.9
Wiz C30	1,239.0	906.6	1,332.8	1,328.9	1,321.0	1,314.0	1,312.0	1,287.8	1,279.6

Table 19: Similarity of each one-batch experiment to its two-batch variant (divided into two parts by stratified bootstrapping). Distance scores (averaged over ten iterations of division) are reported for the uncorrected sub-experiments and *kmersAlignment* algorithm and *rtcorrectedAlignment* algorithm based on the optimum window size (found by applying the algorithm with the window size parameter starting from one up to the number of features in the aligned sub-experiments) and additionally window sizes found to be applicable to most datasets according to our experiments on synthetic datasets.

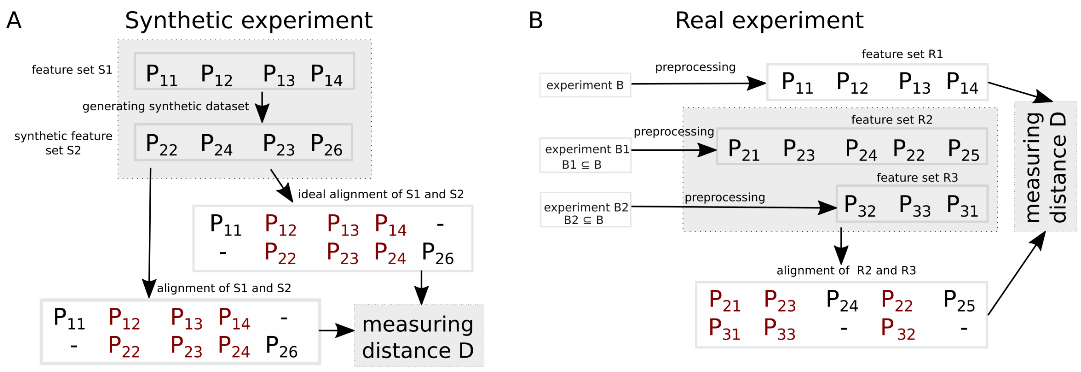


Figure 41: A scheme for evaluation of the two types of experiments used in our study: synthetic and real experiment. Note that features are notated in the same way as in Figure 38. **A** shows the process of generating synthetic feature set S2 from the original S1 by introducing one swap, one deletion, and one insertion. Distance D was then measured for the ideal alignment and the alignment that was reconstructed by any of the proposed algorithms. In this case, distance D is equal to zero; alignments are identical. **B** illustrates the comparison of the alignment of two individually preprocessed sub-experiments B1 and B2 and the preprocessed full experiment B. Here, distance D is equal to one.

the sub-experiment (batch) and the full original experiment. For each experiment, we have found the average number of features that are between two closest swapped feature. Table 20 demonstrates these average number of features between swapped features (μDist) in addition to the probabilities of deletions, insertions, swaps, and insertions with m/z values that are already present in the experiment.

Experiment	μDist	p_{ins}	p_{del}	p_{swap}	p_{inse}
Klk8 C18	12.84	0.04	0.08	0.26	0.02
Tmem60 C18	35.00	0.12	0.21	0.22	0.03
Tmem60 C30	55.84	0.09	0.14	0.27	0.03
Trim9 C30	16.39	0.04	0.14	0.25	0.02
Wiz C18	25.01	0.06	0.12	0.27	0.03
Wiz C30	168.92	0.08	0.33	0.27	0.01

Table 20: Average number of features between swapped features, probability of insertion, deletion, swap, and insertion of the same features for features in six authentic LC-MS datasets.

9.4 DISCUSSION

We have developed *rtcorrectedAlignment* and *kmersAlignment* algorithms for aligning and combining batches in multi-batch LC-MS experiments. Primarily, this tackles the need for finding global parameter values for data preprocessing for all samples in multi-batch experiments because each batch is preprocessed separately with more appropriate and sensitive parameters. It is easier to find global parameter values for the samples belonging to the same batch since retention time and m/z shift is smaller within batches. Furthermore, these algorithms allow batches within a large-scale multi-batch experiment to be continuously preprocessed and evaluated as new data appears from the runs over time. This resolves the need for preprocessing older batches which have already been preprocessed while reducing the amount of computing resources required.

Our results show that *kmersAlignment* is a more robust algorithm compared to *rtcorrectedAlignment*; it constructs more reliable 2D matrices in $\mathbb{L}_B$ format and works effectively in the accuracy and runtime with the default parameter $w_{size} = 5$, thus rendering it much easier to operate. In all of tested synthetic datasets, the default parameter $w_{size} = 5$ performs sufficiently with the faster run of the algorithm since the time complexity of the kmersAlignment algorithm is $\mathcal{O}(K \times w_{size} \times |tgt|)$ where K is the total number of k-mers in the source feature set, tgt denotes the target feature set. Especially due to the nature of time complexity, we recommend to use smaller values of w_{size} parameter or values which are closer to the default value. On the other hand, the results of the *rtcorrectedAlignment* algorithm are highly dependent on the value of the window size parameter n_{RT}, estimation of which is a non-trivial task as we have shown. In some of the tested synthetic datasets, the distance score increases steeply with only small changes in the windows size parameter n_{RT}. This demonstrates that even a small change in the window size value can lead the *rtcorrectedAlignment* algorithm to introduce a considerable amount of noise to the final 2D matrix $\mathbb{L}_B$. On the other hand, *rtcorrectedAlignment* achieves a smaller distance score D in datasets where retention order swaps are absent or scarce. This behavior was examined via an analysis with linear mixed models which showed the dominance of the *kmersAlignment* for interactions where p_{swap} is present, precisely for p_{ins}:p_{swap} and p_{del}:p_{swap}. Although finding the proper value of n_{RT} might be challenging, we conclude that it is better to use a higher value than a smaller one in general. This is supported by the curves in Figure 40 where the corresponding curves on the left side from the optimal points grow more steeply in contrast to the right side.

Our experiments on real datasets confirm the superiority of the *kmersAlignment* algorithm, where in most cases it achieved a better score than *rtcorrectedAlignment* even when the optimal window size parameter n_{RT} leading to the best score was selected. Also, we note that in a real untargeted LC-MS multi-batch experiment it is difficult to find the best value for the window size parameter n_{RT} because any reference solution is missing. However, in comparison with synthetic experiments, changes in the window size parameter n_{RT} of *rtcorrectedAlignment* do not impact distance score as dramatically. Although, in the case of *Wiz C30*, the *kmersAlignment* algorithm achieves a lower distance score compared to *rtcorrectedAlignment* when the rtCorrection function is applied, a non-corrected alignment of *rtcorrectedAlignment* gets the lowest distance score. This trend has not been observed in any of the other real datasets and the anomaly in this particular dataset, i.e. Wiz C30, can be explained by the distribution of features along the retention time axis. The distribution of features in the two Wiz C30 batches is considerably different in comparison to the rest of the experiments. A significantly different number of features across batches in a particular interval of retention time decreases the ability of the algorithm to compensate for retention order swaps because it shrinks the number of possibly affected features according to the retention time axis. This issue cannot be easily resolved by establishing the size of the correction window according to retention time, since the values on the retention time axis can be shifted across batches. The decision for leveraging the order of features and not their precise retention times for solving the correspondence problem is supported by ([5]) who claim that retention order is better conserved across instruments than retention time is. Accordingly, we suggest that when using proposed algorithms, feature distribution and the total number of features for each batch are examined and controlled so that they are as similar as possible.

Both developed algorithms, *kmersAlignment* and *rtcorrectedAlignment*, were written in C++ and are freely available as an open-source R package (*metaboCombineR*): http://www.github.com/fmalinka/metaboCombineR.

10 CONCLUSIONS

Conventional biclustering is a complex task with a considerable number of possible applications. One of the research fields where biclustering has its own substantiation is biology. In that field, the biclustering technique is currently well-established, which is evidenced by the fact that for gene expression data the biclustering was first introduced in 2000 and tens of well-known biclustering tools have been developed since then [178]. Nowadays, incorporating background knowledge into the data analysis is, in general, a desirable trend in bioinformatics. For this reason, a combination of conventional biclustering and methods dealing with background knowledge was the main motivation for this thesis. This combination we refer to as semantic biclustering.

The main idea of semantic biclustering is to find biologically interesting and easily interpretable biclusters which are described by predictive rules. The assumption of an easily interpretable and human-readable form of pattern description is accomplished by a rule in the form of a conjunction of ontological terms. More sophisticated forms of pattern description have been also considered. However, first-order logic, which is used in inductive logic programming for example, has been eventually decided as hardly interpretable because of predicates or function symbols appearing in the hypothesis. In addition, another disadvantage of inductive logic programming is runtime of the algorithms, which is unacceptably slow, especially for high-throughput genomic datasets.

The first *bi-directional enrichment* and *rule and tree learning* approaches introduced reported promising results in the beginning of our research. Therefore, the research has continued in this direction extending the proposed algorithms. However, the *rule and tree learning* show some particular disadvantages which make the hypothesis interpretation complicated. The *rule learning* approach does not guarantee that the hypothesis is non-redundant. Moreover, the *tree learning* approach produces a complex form of hypotheses that contain redundant terms too, similar to the previous approach.

To avoid the redundancy in a rule, we developed a new refinement operator which utilizes the existence of relationships over terms in ontologies. This operator is incorporated into the rule learning algorithm CN2 and it is published as the R package called *sem1R*. In comparison with the traditional refinement operator, the proposed operator dramatically speeds-up the runtime of the algorithm since it prunes a rule space safely. Two reduction procedures of the operator guarantee that a potential best rule, the rule with the highest score of chosen evaluation criteria, will not be excluded from the rule evaluation process. Furthermore, we proposed a new method for finding biclusters that combines a multi-objective optimization approach with the presented ontology-based refinement operator.

Besides the application of the *sem1R* in omics data, the thesis focuses on more specific use cases too. The first algorithm adaptation is called *variant-sem1R* and is determined to find genomic regions which are shared over a cohort of patients. This serves to discover a list of potential pathogenic variants causing rare diseases. The second application shows a concrete scenario of using the *sem1R* package in analysis of E3-ubiquitin ligase in the gastrointestinal tract with respect to tissue regeneration potential.

Besides the main contributions which are the algorithms for semantic biclustering and their specific versions for two different biological research areas, another algorithm for preprocessing a large number of LC-MS experiments was introduced. Since the output of the LC-MS algorithm fits the requirements on the inputs of our semantic biclustering packages, the semantic biclustering seems to be a promising approach for analysis of LC-MS data. However, this issue has not been addressed by this dissertation thesis because of the lack of time which is necessary for measuring an enormous number of samples. In addition, some other aspects must be discussed in more depth, e.g. the form of ontology, discretization, etc. This task, we let open for future research.

Finally, we would like to mention some issues that we faced during the development of semantic biclustering employing gene expression data. As a current challenge, we consider imbalances of specificity of associated ontological terms, especially in sample/condition-related ontologies. Generally speaking, the ontologies associated with samples of gene expression data are usually not eminently large, or ontological terms associated with the samples are relatively general. In this case, such terms of sample ontologies occur in induced rules less frequently or not at all. The nonexistence of sample terms in rules leads to biclusters which are formulated through all samples, and thus might complicate their biological interpretation, or be unwanted in specific cases. To solve this issue and introduce terms of sample ontologies into rules and hypotheses some additional effort is required by a researcher, such as hyperparameter tuning, choosing a different evaluation function, or appending more relevant ontologies. Furthermore, the reason for the less specificity of associated terms in the sample ontologies or their relevant subparts might be: 1) a total number of samples is dramatically different in comparison to thousands or ten thousands genes, and 2) for a specific domain some sample ontologies are manually created which is time-exhaustive, e.g. Dresden Ovary Table. On the other hand, there is a relatively new RNA sequencing method for profiling gene expression in cells that is capable of measuring thousands of samples and some well-defined ontologies are currently available too. The capability to measure a tremendous number of samples with different annotations enables us to eliminate a discrepancy between the size of gene and sample dimensions and then induce rules with a more balanced ratio of gene and sample ontological terms. According to the results that we have shown in Chapter 6, the single-cell RNA-seq method [30] seems to be a promising and interesting application of our semantic biclustering algorithms.

Furthermore, negative ontology terms - in the sense of propositional logic - play an important role in predictive accuracy. For this reason, some specific version of the new refinement operator which would be able to deal with negative terms could be developed. In that case, inducing rules with negative terms might speed-up the process of rule refining once more. This may constitute the object of future studies.

Part I

MANUALS FOR R PACKAGES

A SEM1R PACKAGE

The following text stems from the author's handbook available at GitHub https://github.com/fmalinka/sem1r.

sem1R is a machine learning algorithm that finds interesting, hidden, and non-trivial semantic patterns in omics data. The algorithm produces a set of prediction rules that form data into clusters or biclusters, this depends on the type of ontologies used (column, row, or both). Here, we make distinctions between two types of ontologies: an ontology describing rows (e.g. genes) and columns (e.g. samples). Practically, for gene expression data, where rows represent genes and column represent samples, we recommend to use Gene ontology or any pathway ontologies as a row ontology. Choosing a proper column ontology depends on the type of experiment, e.g. OBO Foundry provides almost two hundreds ontologies and many of them are domain specific so some anatomical ontologies can be used as well. An example of gene expression dataset that addresses simultaneously column and row ontologies is DOT (Dresden Ovary Table) at http://tomancak-srv1.mpi-cbg.de/DOT/main.html.

The sem1R is based on rule learning methods, where two reduction procedures make the algorithm extremely fast and efficient in comparison with the traditional CN2 approach. Additionally, it is easy to use, because all important methods are included into this R package.

A.1 GETTING STARTED

The algorithm is implemented in C++ and provided as R package. The following instructions will show you how to install all prerequisites and the sem1R package as well into your local machine. Afterwards, we will demonstrate the sem1R on real gene expression dataset.

A.1.1 *Prerequisites*

We required to use R in version 3.4. All prerequisites R packages that are needed for the sem1R package are the following: Rcpp (>= 0.12.6), RcppProgress, RcppArmadillo (>= 0.7.800.2.0), and BH (>= 1.72.0-3). All of these packages come from CRAN, so install them by install.packages function in R.

Or, for easier installation, we recommend to install 'devtools' that can download and install the project instantly from GitHub using only one command.

For installing 'devtools' package run R and type the following:

```
install.packages("devtools")
```

A.1.2 *Installing*

If you choose installation via devtools, you would go to the terminal, run R and then the following commands:

```
library(devtools)
install_github("fmalinka/sem1R")
```

All prerequisites packages should be installed automatically.

Or for non-devtools users, download the sem1R package and extract it to your arbitrary folder. Run the terminal, go to the folder and install all prerequisites using install.packages R's function. Finally, build the package.

```
cd /my/path/to/package
R CMD build .
```

The package in tar.gz format will be named as 'sem1R_[version].tar.gz'. The concrete name depends on the package version. Then, install the sem1R package.

```
R CMD INSTALL sem1R\_[version].tar.gz
```

And finally, check whether the sem1R package has been installed. Run R and load the package.

```
library(sem1R)
```

A.2 RUNNING THE EXAMPLE

Running example that we present here comes from Dresden Ovary Table (DOT) located at http://tomancak-srv1.mpi-cbg.de/DOT/main.html. Since the original data matrix is to complex for a brief algorithm exhibition, we will work just with a submatrix of the original matrix in this tutorial. All necessary files for this example are stored at *i*nst folder.

A.2.1 *Data matrix*

A file *d*otmatrix.csv contains binary information about gene expression of the fruit fly adult ovary in many locations. The matrix is two-dimensional where rows represent genes and columns represent samples (locations). Each dimension has own identifier, i.e. genes are described by FBgn (FlyBase) identifiers and columns by your notation. Ones in the matrix mean "expressed" and zeros mean "non-expressed" in the given positions. Obviously, process of binarization has to be done if your data are not in the binary format. Look below how the matrix looks like.

```
            X1.8.somatic.cells X1.9.germline.cells X1.10.terminal.filament
FBgn0033019                  1                   1                       1
FBgn0263251                  1                   1                       1
FBgn0037224                  1                   1                       1
FBgn0038013                  1                   1                       1
FBgn0037358                  1                   1                       1
```

A.2.2 *Ontologies*

Ontologies are the second type of input that has to be given to *sem1R* algorithm. Ontology has to be in OBO format (`https://owlcollab.github.io/oboformat/doc/GO.format.obo-1_4.html`) and relationships of terms must be acyclic (usually OBO ontologies are acyclic). For many other interesting ontologies look at OBO Foundry (http://obofoundry.org). In our running example, we provide two type of different ontologies. Gene ontology, located at *i*nst/extdata/go-basic-reduced.obo, aims to rows of the data matrix and DOT ontology (http://tomancak-srv1.mpi-cbg.de/cgi-bin-public/ovary_annotation_hierarchy.pl), located at *i*nst/extdata/dotOntology.obo, focuses on the columns.

A.2.3 *Connection between the data matrix and the ontologies*

Now, the last step is to establish an annotation, a connection between our data matrix and all given ontologies. Firstly, we look at the rows which are described by the FBGN identifiers. Result of mapping from FBGN identifiers to Gene ontology terms id is provided at *i*nst/extdata/initRowDot_reduced.csv file. File showing a mapping from data matrix columns to DOT ontology can be found at *i*nst/extdata/initColDot.csv file.

A.2.4 *Run sem1R*

Finally, let's run the example!

First of all, load the R library and create a new class sem1R. Then, we load the example data containing all necessary files described above.

```r
library(sem1R)
mysem1R <- new(sem1R)
myExample <- getDatasetExample()
```

Now, we load the data matrix to the sem1R class. Be sure, that the data matrix is a 'matrix' R type and has named rows and columns. It is important! Note that public methods of the class are call by $ symbol.

```r
mysem1R$setDataset(myExample$datamatrix)
```

Then, we load all ontologies. For this, use createCOLOntology or createROWOntology methods, it depends on your matrix design generally. The first argument of these methods is name of ontolgy, the second argument set up path to the corresponding obo file, and the last one is a list of vectors representing the connection between rows/-columns and ontologies. For the proper format look at one of the examples (myExample$colOntologyDesc or myExample$rowOntologyDesc). When you have more than one ontology, just call the corresponding method one again. However, the name of ontology must be unique!

```
mysem1R$createCOLOntology("DOT", myExample$colOntologyPath,
    myExample$colOntologyDesc)
mysem1R$createROWOntology("GO", myExample$rowOntologyPath,
    myExample$rowOntologyDesc)
```

Now, we set all algorithm parameters (see R manual).

```
mysem1R$filterTh <- 50
mysem1R$objective <- "auc"
mysem1R$ruleDepth <- 3
mysem1R$nrules <- 2
mysem1R$featureSelectionMethod <- 0
mysem1R$minLevel <- 2
```

If you want to check out correctness of the connection of data matrix and the ontologies, call 'mysem1R$checkRowDescription()' or 'mysem1R$checkColDescription()'.

Finally, run the algorithm and save the results!

```
myhypothesis <- mysem1R$findDescription()
```

When it ends ...

```
[sem1R SETTINGS]
filter threshold: 50
rule depth: 3
significance threshold: 6.635
objective function: auc
number of rules: 2
featureSelectionMethod: 0
ruleFormat: both
0%   10   20   30   40   50   60   70   80   90   100%
[----|----|----|----|----|----|----|----|----|----|
**************************************************|
```

... your final rule set will be printed on STDOUT.

```
******************************************************************
************************** FINAL RULESET *************************
===== RULE 1=====
 STATS: score 0.536225 t-score: 2182.28 POSITIVE: 23351 NEGATIVE: 11649
 RULE: GO:0044763 AND GO:0043229
 DETAILS:
ID: GO:0044763
NAME: single-organism cellular process
DEF: "Any process that is carried out at the cellular level, occurring
      within a single organism." [GOC:jl]
```

```
level: 2

ID: GO:0043229
NAME: intracellular organelle
DEF: "Organized structure of distinctive morphology and function,
      occurring within the cell. Includes the nucleus, mitochondria,
      plastids, vacuoles, vesicles, ribosomes and the cytoskeleton.
      Excludes the plasma membrane." [GOC:go_curators]
level: 2

COVERED:
  POSITIVE:
X1.8.somatic.cells, X1.9.germline.cells, X1.11.cap.cells,
X1.13.follicle.stem.cells, X1.15.interfollicular.stalk.cells,
X1.18.germline.stem.cells, X1.20.presumptive.nurse.cells,
X2.26.oocyte, X4.29.oocyte, X5.30.oocyte, X3.32.nurse.cells,
X5.34.nurse.cells, X2.35.somatic.cells, X4.37.somatic.cells,
X2.39.follicle.cells, X3.40.follicle.cells, X5.42.follicle.cells,
X3.44.interfollicular.stalk.cells, X5.46.interfollicular.stalk.cells,
X3.48.anterior.follicle.cells, X4.49.border.cells,
X2.58.posterior.follicle.cells, X4.60.posterior.follicle.cells,
X5.62.centripetally.migrating.follicle.cells,
X2.64.anterior.restriction, X2.66.nurse.cells_nuclear.foci,
X2.69.cytoplasmic.foci, X3.71.anterior.restriction,
X5.74.anterior.restriction, X3.75.posterior.restriction,
X5.77.posterior.restriction, X5.81.cortical.enrichment,
X3.82.nurse.cells_nuclear.foci, X5.84.nurse.cells_nuclear.foci,
X4.86.nurse.cells_perinuclear, X3.105.basal.restrictrion,
X4.106.basal.restrictrion, X3.108.apical.restriction,
X5.110.apical.restriction, X4.112.cytoplasmic.foci,
X5.113.cytoplasmic.foci, X3.115.cytoplasmic.foci,
X5.117.cytoplasmic.foci, X3.119.cytoplasmic.foci,
X4.120.cytoplasmic.foci, X2.128.nuclear.foci, X4.130.nuclear.foci,
X4.139.anterior.follicle.cell, X5.140.squamous.follicle.cells,
X4.143.cortical.enrichment, X2.145.cortical.enrichment,
X4.147.cortical.enrichment, X5.149.follicle.cells.overlaying.the.oocyte,
X2.164.perinuclear, X4.166.perinuclear, X3.168.oocyte.nucleus,
X5.170.oocyte.nucleus,
  NEGATIVE:
X1.8.somatic.cells, X1.10.terminal.filament, X1.12.escort.cells,
X1.14.follicle.cells, X1.17.posterior.follicle.cells,
X1.19.cystoblast, X1.21.presumptive.oocyte, X4.29.oocyte,
X2.31.nurse.cells, X5.34.nurse.cells, X3.36.somatic.cells,
X5.38.somatic.cells, X4.41.follicle.cells,
X2.43.interfollicular.stalk.cells, X4.45.interfollicular.stalk.cells,
X3.48.anterior.follicle.cells, X5.50.border.cells,
X3.59.posterior.follicle.cells,
X5.62.centripetally.migrating.follicle.cells,
X2.65.posterior.restriction, X2.67.nurse.cells_perinuclear,
X2.70.apical.restriction, X5.74.anterior.restriction,
X4.76.posterior.restriction, X4.80.cortical.enrichment,
X4.83.nurse.cells_nuclear.foci, X3.85.nurse.cells_perinuclear,
X3.105.basal.restrictrion, X5.107.basal.restrictrion,
X4.109.apical.restriction, X4.112.cytoplasmic.foci,
X2.114.cytoplasmic.foci, X4.116.cytoplasmic.foci,
X3.119.cytoplasmic.foci, X5.121.cytoplasmic.foci,
X4.130.nuclear.foci, X4.139.anterior.follicle.cell,
X3.142.cortical.enrichment, X5.144.cortical.enrichment,
X4.147.cortical.enrichment,
X5.149.follicle.cells.overlaying.the.oocyte, X3.165.perinuclear,
X3.168.oocyte.nucleus, X5.170.oocyte.nucleus,
===== =====
```

And the structure of returned hypothesis is the following:

```
str(myhypothesis[[1]])
```

```
List of 9
 $ ruleID     : int 1
 $ score      : num 0.536
 $ tscore     : num 2182
 $ nCoveredPOS: int 23351
 $ nCoveredNEG: int 11649
 $ rules      : chr [1:2] "GO:0044763" "GO:0043229"
 $ details    : chr [1:6] "ID: GO:0044763" "NAME: single-organism
                            cellular process"
                          "DEF: \"Any process that is carried out
                                at the cellular level, occurring
                                within a single organism.\"
                                [GOC:jl]" "ID: GO:0043229" ...
 $ coveredPOS : chr [1:23351] "FBgn0039115,X1.8.somatic.cells"
                              "FBgn0022238,X1.8.somatic.cells"
                              "FBgn0262601,X1.8.somatic.cells"
                              "FBgn0029134,X1.8.somatic.cells" ...
 $ coveredNEG : chr [1:11649] "FBgn0026737,X1.8.somatic.cells"
                              "FBgn0003087,X1.8.somatic.cells"
                              "FBgn0031873,X1.8.somatic.cells"
                              "FBgn0003514,X1.8.somatic.cells" ...
```

where ruleID represents order of the induced rule, score represents quality of the rule depends on the type of evaluation function, tscore represents chi-square score of the rule, positive and negative covered is a number of examples covered by the rule, rules represents a conjunction of ontological terms, details provides additional information about the terms in conjunction, and finally covered represents covered examples expressed by their position in the matrix.

B METABOCOMBINER PACKAGE

The following text stems from the author's handbook available at GitHub https://github.com/fmalinka/metabocombiner.

metaboCombineR is an R package written in C++ that focuses on large-scale untargeted LC-MS experiments. The package allows to preprocess each batch of samples separately using an arbitrary preprocessing program, such as XCMS. The main pro is a possibility to handle and observed results of experiments during the time and program parameter tuning should be easier since within-batch variability is smaller than between-batch variability. The package contains two different approaches called *k*mersAlignment and *r*tcorrectedAlignment that were both published in [104].

B.1 GETTING STARTED

B.1.1 *Prerequisites*

Only one external R package is required: Rcpp (>= 0.12.16). For compiling vignette, we suggest rmarkdown package.

B.1.2 *Installing*

The simplest way to install metaboCombineR package is via *d*evtools that allows to download and install the project instantly from GitHub using only one command. For the package installation, run *R* and type:

```
library(devtools)
install_github("fmalinka/metaboCombineR")
```

B.2 RUNNING THE EXAMPLE

For an illustration, we prepared four authentic real datasets. For loading them to workspace, type the following:

```
library(metaboCombineR)
data(metaboExp1)
data(metaboExp2)
data(metaboExp3)
data(metaboExp4)
```

Presented datasets are in 2-dimensional matrix format where rows represent features and each row has it own name which m/z value is prefixed by *M* and rt by *T*. Columns represent samples. Names of all samples/features must be filled by *c*olnames/rownames function through R. To see an example of dataset format, type *h*ead(metaboExp1).

MetaboCombineR enables to define assignments of samples to some groups, typically as treatment and control groups. To provide this information, add a vector of assignments as the first row to the input *d*ata.frame. The row must be named as *g*roup. If the group label is present in a dataset it must be present in all other datasets. Inconsistencies are not allowed. An example of dataset with three samples (two treatments and one control) and 2 features is depicted below.

```
            X064.EPK83_m_Mzb1_ESI.mzML  X064.EPK88_m_Mzb1_ESI.mzML
group                treatment                  treatment
M57.0813T1428.187     -0.321542424867644         0.286250559905367
M57.2355T1428.090      0.408813652123444        -1.0100177456997

            X064.EPK94_m_Mzb1_ESI.mzML
group                control
M57.0811T1428.187     1.17078411764221
M57.2355T1428.090    -0.153473421445439
```

To combine all of these experiments into one table, call *r*unMetaboCombiner function, where the first argument is supposed to be a list of experiments, *m*zprecision argument defines a number of digits considered for peaks, and *a*lgorithm selects one of the proposed algorithms. The *a*lgorithm argument supposes only two values on input: kmer and rtcor. The default algorithm is kmersAlignment. *w*indowsize argument represents the size of the window for *r*tcorrectedAlignment algorithm. For *k*mersAlignment algorithm, the same argument represents the k-mer value.

```
mytableKmer <- runMetaboCombiner(list(metaboExp1, metaboExp2,
    metaboExp3, metaboExp4), mzprecision = 2, algorithm="kmer",
    windowsize = 5)

mytableRtcor <- runMetaboCombiner(list(metaboExp1, metaboExp2,
    metaboExp3, metaboExp4), mzprecision = 2, algorithm="rtcor",
    windowsize = 50)
```

Then, the result matrix is stored in *m*ytableKmer/*m*ytableRtcor variable, respectively.

BIBLIOGRAPHY

[1] Jesus S Aguilar-Ruiz. “Shifting and scaling patterns from gene expression data.” In: *Bioinformatics* 21.20 (2005), pp. 3840–3845.

[2] Adrian Alexa and Jorg Rahnenfuhrer. *topGO: topGO: Enrichment analysis for Gene Ontology*. R package version 2.4.0. 2010.

[3] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. “Gene ontology: tool for the unification of biology.” In: *Nature genetics* 25.1 (2000), pp. 25–29.

[4] Geraldine A Van der Auwera, Mauricio O Carneiro, Christopher Hartl, Ryan Poplin, Guillermo Del Angel, Ami Levy-Moonshine, Tadeusz Jordan, Khalid Shakir, David Roazen, Joel Thibault, et al. “From FastQ data to high-confidence variant calls: the genome analysis toolkit best practices pipeline.” In: *Current protocols in bioinformatics* 43.1 (2013), pp. 11–10.

[5] Eric Bach, Sandor Szedmak, Céline Brouard, Sebastian Böcker, and Juho Rousu. “Liquid-chromatography retention order prediction for metabolite identification.” In: *Bioinformatics* 34.17 (2018), pp. i875–i883.

[6] Albert-Laszlo Barabasi and Zoltan N Oltvai. “Network biology: understanding the cell’s functional organization.” In: *Nature reviews genetics* 5.2 (2004), pp. 101–113.

[7] Nick Barker, Johan H Van Es, Jeroen Kuipers, Pekka Kujala, Maaike Van Den Born, Miranda Cozijnsen, Andrea Haegebarth, Jeroen Korving, Harry Begthel, Peter J Peters, et al. “Identification of stem cells in small intestine and colon by marker gene Lgr5.” In: *Nature* 449.7165 (2007), pp. 1003–1007.

[8] Riccardo Bellazzi and Blaz Zupan. “Predictive data mining in clinical medicine: current issues and guidelines.” In: *International journal of medical informatics* 77.2 (2008), pp. 81–97.

[9] Amir Ben-Dor, Benny Chor, Richard Karp, and Zohar Yakhini. “Discovering local structure in gene expression data: the order-preserving submatrix problem.” In: *Journal of computational biology* 10.3-4 (2003), pp. 373–384.

[10] Christopher E Berndsen and Cynthia Wolberger. “New insights into ubiquitin E3 ligase mechanism.” In: *Nature structural & molecular biology* 21.4 (2014), p. 301.

[11] Joep Beumer and Hans Clevers. “Cell fate specification and differentiation in the adult mammalian intestine.” In: *Nature Reviews Molecular Cell Biology* (2020), pp. 1–15.

[12] Jiří Borovec and Jan Kybic. "Binary pattern dictionary learning for gene expression representation in drosophila imaginal discs." In: *Asian Conference on Computer Vision*. Springer. Cham, Switzerland: Springer, 2017, pp. 555–569.

[13] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. "Finding knees in multi-objective optimization." In: *International conference on parallel problem solving from nature*. Springer. 2004, pp. 722–731.

[14] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[15] Carl Brunius, Lin Shi, and Rikard Landberg. "Large-scale untargeted LC-MS metabolomics data correction using between-batch feature alignment and cluster-based within-batch signal intensity drift correction." In: *Metabolomics* 12.11 (2016), pp. 1–13.

[16] Andrew Butler, Paul Hoffman, Peter Smibert, Efthymia Papalexi, and Rahul Satija. "Integrating single-cell transcriptomic data across different conditions, technologies, and species." In: *Nature biotechnology* 36.5 (2018), pp. 411–420.

[17] Laurence Calzone, Nathalie Chabrier-Rivier, François Fages, and Sylvain Soliman. "Machine learning biochemical networks from temporal logic properties." In: *Transactions on Computational Systems Biology VI*. Berlin, Heidelberg: Springer, 2006, pp. 68–94.

[18] Matteo Cereda, Thanos P Mourikis, and Francesca D Ciccarelli. "Genetic redundancy, functional compensation, and cancer vulnerability." In: *Trends in cancer* 2.4 (2016), pp. 160–162.

[19] Hung-Chia Chen, Wen Zou, Yin-Jing Tien, and James J Chen. "Identification of bicluster regions in a binary matrix and its applications." In: *PloS one* 8.8 (2013), e71680.

[20] Yizong Cheng and George M Church. "Biclustering of expression data." In: *Ismb*. Vol. 8. 2000, pp. 93–103.

[21] Shunsuke Chikuma, Mitsuhiro Kanamori, Setsuko Mise-Omata, and Akihiko Yoshimura. "Suppressors of cytokine signaling: potential immune checkpoint molecules for cancer immunotherapy." In: *Cancer science* 108.4 (2017), pp. 574–580.

[22] Jasmine Chong, Othman Soufan, Carin Li, Iurie Caraus, Shuzhao Li, Guillaume Bourque, David S Wishart, and Jianguo Xia. "MetaboAnalyst 4.0: towards more transparent and integrative metabolomics analysis." In: *Nucleic acids research* 46.W1 (2018), W486–W494.

[23] Jasmine Chong, Mai Yamamoto, and Jianguo Xia. "MetaboAnalystR 2.0: from raw spectra to biological insights." In: *Metabolites* 9.3 (2019), p. 57.

[24] Qinjun Chu, Dan Shen, Long He, Hongwei Wang, Chunlan Liu, and Wei Zhang. "Prognostic significance of SOCS3 and its biological function in colorectal cancer." In: *Gene* 627 (2017), pp. 114–122.

[25] Peter Clark and Robin Boswell. "Rule induction with CN2: Some recent improvements." In: *European Working Session on Learning*. Springer. 1991, pp. 151–163.

[26] Peter Clark and Tim Niblett. "The CN2 induction algorithm." In: *Machine learning* 3.4 (1989), pp. 261–283.

[27] William W Cohen. "Fast effective rule induction." In: *Proceedings of the twelfth international conference on machine learning*. 1995, pp. 115–123.

[28] Gene Ontology Consortium. "Gene ontology consortium: going forward." In: *Nucleic acids research* 43.D1 (2015), pp. D1049–D1056.

[29] Gene Ontology Consortium. "Expansion of the Gene Ontology knowledgebase and resources." In: *Nucleic acids research* 45.D1 (2016), pp. D331–D338.

[30] Tabula Muris Consortium et al. "Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris." In: *Nature* 562.7727 (2018), pp. 367–372.

[31] Marta Costa, Simon Reeve, Gary Grumbling, and David Osumi-Sutherland. "The Drosophila anatomy ontology." In: *Journal of biomedical semantics* 4.1 (2013), p. 32.

[32] David Croft, Antonio Fabregat Mundo, Robin Haw, Marija Milacic, Joel Weiser, Guanming Wu, Michael Caudy, Phani Garapati, Marc Gillespie, Maulik R Kamdar, et al. "The Reactome pathway knowledgebase." In: *Nucleic acids research* 42.D1 (2013), pp. D472–D477.

[33] R Keira Curtis, Matej Orešič, and Antonio Vidal-Puig. "Pathways to the analysis of microarray data." In: *TRENDS in Biotechnology* 23.8 (2005), pp. 429–435.

[34] Piero Dalerba, Tomer Kalisky, Debashis Sahoo, Pradeep S Rajendran, Michael E Rothenberg, Anne A Leyrat, Sopheak Sim, Jennifer Okamoto, Darius M Johnston, Dalong Qian, et al. "Single-cell dissection of transcriptional heterogeneity in human colon tumors." In: *Nature biotechnology* 29.12 (2011), pp. 1120–1127.

[35] Indraneel Das. "On characterizing the "knee" of the Pareto curve based on normal-boundary intersection." In: *Structural optimization* 18.2-3 (1999), pp. 107–115.

[36] John Day-Richter, Midori A Harris, Melissa Haendel, Gene Ontology OBO-Edit Working Group, and Suzanna Lewis. "OBO-Edit—an ontology editor for biologists." In: *Bioinformatics* 23.16 (2007), pp. 2198–2200.

[37] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. “A fast and elitist multiobjective genetic algorithm: NSGA-II.” In: *IEEE transactions on evolutionary computation* 6.2 (2002), pp. 182–197.

[38] Inderjit S Dhillon. “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*. ACM. 2001, pp. 269–274.

[39] *Dresden Ovary Table*. http://tomancak-srv1.mpi-cbg.de/DOT/main. [Online; accessed 15-February-2016].

[40] Lubica Dudakova, Cerys J Evans, Nikolas Pontikos, Nathaniel J Hafford-Tear, František Malinka, Pavlina Skalicka, Ales Horinek, Francis L Munier, Nathalie Voide, Pavel Studeny, et al. “The utility of massively parallel sequencing for posterior polymorphous corneal dystrophy type 3 molecular diagnosis.” In: *Experimental eye research* 182 (2019), pp. 160–166.

[41] Robert C Edgar. “MUSCLE: multiple sequence alignment with high accuracy and high throughput.” In: *Nucleic acids research* 32.5 (2004), pp. 1792–1797.

[42] Derrick D Eichele and Kusum K Kharbanda. “Dextran sodium sulfate colitis murine model: An indispensable tool for advancing our understanding of inflammatory bowel diseases pathogenesis.” In: *World journal of gastroenterology* 23.33 (2017), p. 6016.

[43] Mohamed A El-Brolosy, Zacharias Kontarakis, Andrea Rossi, Carsten Kuenne, Stefan Günther, Nana Fukuda, Khrievono Kikhi, Giulia LM Boezio, Carter M Takacs, Shih-Lei Lai, et al. “Genetic compensation triggered by mutant mRNA degradation.” In: *Nature* 568.7751 (2019), pp. 193–197.

[44] Mohamed A El-Brolosy and Didier YR Stainier. “Genetic compensation: A phenomenon in search of mechanisms.” In: *PLoS genetics* 13.7 (2017), e1006780.

[45] P Facer, AE Bishop, RV Lloyd, BS Wilson, RJ Hennessy, and JM Polak. “Chromogranin: a newly recognized marker for endocrine cells of the human gastrointestinal tract.” In: *Gastroenterology* 89.6 (1985), pp. 1366–1373.

[46] Zhong-Ze Fang and Frank J Gonzalez. “LC–MS-based metabolomics: an update.” In: *Archives of toxicology* 88.8 (2014), pp. 1491–1502.

[47] Da-Fei Feng and Russell F Doolittle. “Progressive sequence alignment as a prerequisitetto correct phylogenetic trees.” In: *Journal of molecular evolution* 25.4 (1987), pp. 351–360.

[48] Daniel Finley and Vincent Chau. “Ubiquitination.” In: *Annual review of cell biology* 7.1 (1991), pp. 25–69.

[49] Julie Frantsve, Juerg Schwaller, David W Sternberg, Jeffery Kutok, and D Gary Gilliland. "Socs-1 inhibits TEL-JAK2-mediated transformation of hematopoietic cells through inhibition of JAK2 kinase activity and induction of proteasome-mediated degradation." In: *Molecular and cellular biology* 21.10 (2001), pp. 3547–3557.

[50] *Frequent Itemset Mining Implementations Repository*. http://fimi.ua.ac.be/. [Online; accessed 15-February-2016].

[51] Jerome H Friedman and Nicholas I Fisher. "Bump hunting in high-dimensional data." In: *Statistics and Computing* 9.2 (1999), pp. 123–143.

[52] John B Furness, Leni R Rivera, Hyun-Jung Cho, David M Bravo, and Brid Callaghan. "The gut as a sensory organ." In: *Nature reviews Gastroenterology & hepatology* 10.12 (2013), p. 729.

[53] Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrač. *Foundations of rule learning*. Heidelberg: Springer, 2012. ISBN: 978-3-540-75197-7.

[54] LH Gallo, J Ko, and DJ Donoghue. "The importance of regulatory ubiquitination in cancer and metastasis." In: *Cell Cycle* 16.7 (2017), pp. 634–648.

[55] Audrey P Gasch and Michael B Eisen. "Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering." In: *Genome biology* 3.11 (2002), pp. 1–22.

[56] *Gene Ontology Consortium*. http://geneontology.org/. [Online; accessed 15-February-2016].

[57] François Gerbe, Bénédicte Brulin, Leila Makrini, Catherine Legraverend, and Philippe Jay. "DCAMKL-1 expression identifies Tuft cells rather than stem cells in the adult mouse intestinal epithelium." In: *Gastroenterology* 137.6 (2009), pp. 2179–2180.

[58] Gad Getz, Erel Levine, and Eytan Domany. "Coupled two-way clustering analysis of gene microarray data." In: *Proceedings of the National Academy of Sciences* 97.22 (2000), pp. 12079–12084.

[59] Soudeh Ghafouri-Fard, Vahid Kholghi Oskooei, Iman Azari, and Mohammad Taheri. "Suppressor of cytokine signaling (SOCS) genes are downregulated in breast cancer." In: *World journal of surgical oncology* 16.1 (2018), pp. 1–9.

[60] Guri Giaever, Angela M Chu, Li Ni, Carla Connelly, Linda Riles, Steeve Véronneau, Sally Dow, Ankuta Lucau-Danila, Keith Anderson, Bruno Andre, et al. "Functional profiling of the Saccharomyces cerevisiae genome." In: *nature* 418.6896 (2002), pp. 387–391.

[61] Greg Gibson. "Rare and common variants: twenty arguments." In: *Nature Reviews Genetics* 13.2 (2012), pp. 135–145.

[62] José Luis Gómez-Skarmeta, Sonsoles Campuzano, and Juan Modolell. "Half a century of neural prepatterning: the story of a few bristles and many genes." In: *Nature Reviews Neuroscience* 4.7 (2003), pp. 587–598.

[63] Teofilo F Gonzalez. "On the computational complexity of clustering and related problems." In: *System modeling and optimization*. Springer, 1982, pp. 174–182.

[64] Alex Gregorieff, Daniel E Stange, Pekka Kujala, Harry Begthel, Maaike Van den Born, Jeroen Korving, Peter J Peters, and Hans Clevers. "The ets-domain transcription factor Spdef promotes maturation of goblet and paneth cells in the intestinal epithelium." In: *Gastroenterology* 137.4 (2009), pp. 1333–1345.

[65] Rohit Gupta, Navneet Rao, and Vipin Kumar. *Discovery of error-tolerant biclusters from noisy gene expression data*. 2011.

[66] Daniel Gusenleitner, Eleanor A Howe, Stefan Bentink, John Quackenbush, and Aedín C Culhane. "iBBiG: iterative binary bi-clustering of gene sets." In: *Bioinformatics* 28.19 (2012), pp. 2484–2492.

[67] Adam L Haber, Moshe Biton, Noga Rogel, Rebecca H Herbst, Karthik Shekhar, Christopher Smillie, Grace Burgin, Toni M Delorey, Michael R Howitt, Yarden Katz, et al. "A single-cell survey of the small intestinal epithelium." In: *Nature* 551.7680 (2017), pp. 333–339.

[68] Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. "Semantic similarity from natural language and ontology analysis." In: *Synthesis Lectures on Human Language Technologies* 8.1 (2015), pp. 1–254.

[69] Sébastien Harispe, David Sánchez, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. "A framework for unifying ontology-based semantic similarity measures: A study in the biomedical domain." In: *Journal of biomedical informatics* 48 (2014), pp. 38–53.

[70] John A Hartigan. "Direct clustering of a data matrix." In: *Journal of the american statistical association* 67.337 (1972), pp. 123–129.

[71] Makoto Hirosawa, Yasushi Totoki, Masaki Hoshida, and Masato Ishikawa. "Comprehensive study on iterative algorithms of multiple sequence alignment." In: *Bioinformatics* 11.1 (1995), pp. 13–18.

[72] Torgeir R Hvidsten, Astrid Lægreid, and Jan Komorowski. "Learning rule-based models of biological process from gene expression time profiles using gene ontology." In: *Bioinformatics* 19.9 (2003), pp. 1116–1123.

[73] Veronika Iatsiuk, František Malinka, Marketa Pickova, Jolana Tureckova, Jiri Klema, Frantisek Spoutil, Vendula Novosadova, Jan Prochazka, and Radislav Sedlacek. "Semantic clustering analysis of E3-ubiquitin ligases in GIT defines genes ontology clusters with tissue expression patterns." submitted.

[74] Helena Jambor, Vineeth Surendranath, Alex T Kalinka, Pavel Mejstrik, Stephan Saalfeld, and Pavel Tomancak. "Systematic imaging reveals features and changing localization of mRNAs in Drosophila development." In: *Elife* 4 (2015).

[75] Kristie Jenkins, Jing Jing Khoo, Anthony Sadler, Rebecca Piganis, Die Wang, Natalie A Borg, Kathryn Hjerrild, Jodee Gould, Belinda J Thomas, Phillip Nagley, et al. "Mitochondrially localised MUL1 is a novel modulator of antiviral signaling." In: *Immunology and cell biology* 91.4 (2013), pp. 321–330.

[76] Shintaro Kamizono, Toshikatsu Hanada, Hideo Yasukawa, Shigeru Minoguchi, Reiko Kato, Mayu Minoguchi, Kimihiko Hattori, Shigetsugu Hatakeyama, Masayoshi Yada, Sumiyo Morita, et al. "The SOCS box of SOCS-1 accelerates ubiquitin-dependent proteolysis of TEL-JAK2." In: *Journal of biological chemistry* 276.16 (2001), pp. 12530–12538.

[77] Minoru Kanehisa, Miho Furumichi, Mao Tanabe, Yoko Sato, and Kanae Morishima. "KEGG: new perspectives on genomes, pathways, diseases and drugs." In: *Nucleic acids research* 45.D1 (2016), pp. D353–D361.

[78] Minoru Kanehisa and Susumu Goto. "KEGG: kyoto encyclopedia of genes and genomes." In: *Nucleic acids research* 28.1 (2000), pp. 27–30.

[79] Minoru Kanehisa, Yoko Sato, Masayuki Kawashima, Miho Furumichi, and Mao Tanabe. "KEGG as a reference resource for gene and protein annotation." In: *Nucleic acids research* 44.D1 (2016), pp. D457–D462.

[80] Konrad J Karczewski, Laurent C Francioli, Grace Tiao, Beryl B Cummings, Jessica Alföldi, Qingbo Wang, Ryan L Collins, Kristen M Laricchia, Andrea Ganna, Daniel P Birnbaum, et al. "The mutational constraint spectrum quantified from variation in 141,456 humans." In: *Nature* 581.7809 (2020), pp. 434–443.

[81] Mikko Katajamaa, Jarkko Miettinen, and Matej Orešič. "MZmine: toolbox for processing and visualization of mass spectrometry based molecular profile data." In: *Bioinformatics* 22.5 (2006), pp. 634–636.

[82] Warren A Kibbe, Cesar Arze, Victor Felix, Elvira Mitraka, Evan Bolton, Gang Fu, Christopher J Mungall, Janos X Binder, James Malone, Drashtti Vasant, et al. "Disease Ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data." In: *Nucleic acids research* 43.D1 (2014), pp. D1071–D1078.

[83] Moon-Hong Kim, Moon-Sun Kim, Wonwoo Kim, Mi Ae Kang, Nicholas A Cacalano, Soon-Beom Kang, Young-Joo Shin, and Jae-Hoon Jeong. "Suppressor of cytokine signaling (SOCS) genes are silenced by DNA hypermethylation and histone deacetylation and regulate response to radiotherapy in cervical cancer cells." In: *PloS one* 10.4 (2015), e0123133.

[84] Jiří Kléma, František Malinka, and Filip Železný. "Semantic Biclustering: A New Way to Analyze and Interpret Gene Expression Data." In: *Bioinformatics Research and Applications* (2016), p. 332.

[85] Yuval Kluger, Ronen Basri, Joseph T Chang, and Mark Gerstein. "Spectral biclustering of microarray data: coclustering genes and conditions." In: *Genome research* 13.4 (2003), pp. 703–716.

[86] Arno Knobbe, Bruno Crémilleux, Johannes Fürnkranz, and Martin Scholz. "From local patterns to global models: the LeGo approach to data mining." In: *LeGo* 8 (2008), pp. 1–16.

[87] David Komander and Michael Rape. "The ubiquitin code." In: *Annual review of biochemistry* 81 (2012), pp. 203–229.

[88] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. "Supervised machine learning: A review of classification techniques." In: *Emerging artificial intelligence applications in computer engineering* 160 (2007), pp. 3–24.

[89] Miloš Krejník and Jiří Kléma. "Empirical evidence of the applicability of functional clustering through gene expression classification." In: *IEEE/ACM transactions on computational biology and bioinformatics* 9.3 (2012), pp. 788–798.

[90] Adrian Kuhn, Stéphane Ducasse, and Tudor Gîrba. "Semantic clustering: Identifying topics in source code." In: *Information and software technology* 49.3 (2007), pp. 230–243.

[91] Alexandra Kuznetsova, Per B. Brockhoff, and Rune H. B. Christensen. "lmerTest Package: Tests in Linear Mixed Effects Models." In: *Journal of Statistical Software* 82.13 (2017), pp. 1–26.

[92] Eva Lange, Clemens Gröpl, Ole Schulz-Trieglaff, Andreas Leinenbach, Christian Huber, and Knut Reinert. "A geometric approach for the alignment of liquid chromatography—mass spectrometry data." In: *Bioinformatics* 23.13 (2007), pp. i273–i281.

[93] Eva Lange, Ralf Tautenhahn, Steffen Neumann, and Clemens Gröpl. "Critical assessment of alignment procedures for LC-MS proteomics and metabolomics measurements." In: *BMC bioinformatics* 9.1 (2008), p. 375.

[94] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. "UpSet: visualization of intersecting sets." In: *IEEE transactions on visualization and computer graphics* 20.12 (2014), pp. 1983–1992.

[95] Heng Li. "Toward better understanding of artifacts in variant calling from high-coverage samples." In: *Bioinformatics* 30.20 (2014), pp. 2843–2851.

[96] Heng Li and Richard Durbin. "Fast and accurate short read alignment with Burrows–Wheeler transform." In: *bioinformatics* 25.14 (2009), pp. 1754–1760.

[97] Heng Li and Richard Durbin. "Fast and accurate long-read alignment with Burrows–Wheeler transform." In: *Bioinformatics* 26.5 (2010), pp. 589–595.

[98] Wei Li, Mario H Bengtson, Axel Ulbrich, Akio Matsuda, Venkateshwar A Reddy, Anthony Orth, Sumit K Chanda, Serge Batalov, and Claudio AP Joazeiro. "Genome-wide and functional annotation of human E3 ubiquitin ligases identifies MULAN, a mitochondrial E3 that regulates the organelle's dynamics and signaling." In: *PloS one* 3.1 (2008), e1487.

[99] Zhucui Li, Yan Lu, Yufeng Guo, Haijie Cao, Qinhong Wang, and Wenqing Shui. "Comprehensive evaluation of untargeted metabolomics data processing software in feature detection, quantification and discriminating marker selection." In: *Analytica chimica acta* 1029 (2018), pp. 50–57.

[100] Arjen Lommen. "MetAlign: interface-driven, versatile metabolomics tool for hyphenated full-scan mass spectrometry data preprocessing." In: *Analytical chemistry* 81.8 (2009), pp. 3079–3086.

[101] Claudio Lucchese, Salvatore Orlando, and Raffaele Perego. "A Unifying Framework for Mining Approximate Top-Binary Patterns." In: *Knowledge and Data Engineering, IEEE Transactions on* 26.12 (2014), pp. 2900–2913.

[102] Sara C Madeira and Arlindo L Oliveira. "Biclustering algorithms for biological data analysis: a survey." In: *IEEE/ACM transactions on computational biology and bioinformatics* 1.1 (2004), pp. 24–45.

[103] František Malinka, Filip Železný, and Jiří Kléma. "Finding semantic patterns in omics data using concept rule learning with an ontology-based refinement operator." In: *BioData mining* 13.1 (2020), pp. 1–22.

[104] František Malinka, Ashkan Zareie, Jan Prochazka, Radislav Sedlacek, and Vendula Novosadova. "Batch alignment via retention orders for preprocessing large-scale multi-batch LC-MS experiments." submitted.

[105] James Malone, Ele Holloway, Tomasz Adamusiak, Misha Kapushesky, Jie Zheng, Nikolay Kolesnikov, Anna Zhukova, Alvis Brazma, and Helen Parkinson. "Modeling sample variables with an Experimental Factor Ontology." In: *Bioinformatics* 26.8 (2010), pp. 1112–1118.

[106] J Kent Martin and DS Hirschberg. "On the complexity of learning decision trees." In: *International Symposium on Artificial Intelligence and Mathematics*. Citeseer. 1996, pp. 112–115.

[107] Ujjwal Maulik, Sanghamitra Bandyopadhyay, and Anirban Mukhopadhyay. *Multiobjective genetic algorithms for clustering: applications in data mining and bioinformatics*. Springer Science & Business Media, 2011.

[108] John Mayer, Robert Layfield, Helen C Ardley, and Philip A Robinson. "E3 ubiquitin ligases." In: *Essays in biochemistry* 41 (2005), pp. 15–30.

[109] Jason Merkin, Caitlin Russell, Ping Chen, and Christopher B Burge. "Evolutionary dynamics of gene and isoform regulation in Mammalian tissues." In: *Science* 338.6114 (2012), pp. 1593–1599.

[110] Ryszard S. Michalski. "On the Quasi-Minimal Solution of the Covering Problem." In: *Proceedings of the 5th International Symposium on Information Processing (FCIP-69)*. Bled, Yugoslavia: Vol. A3 (Switching Circuits), 1969, pp. 125–128.

[111] Pauli Miettinen, Taneli Mielikäinen, Aristides Gionis, Gautam Das, and Heikki Mannila. "The discrete basis problem." In: *IEEE transactions on knowledge and data engineering* 20.10 (2008), pp. 1348–1362.

[112] Pauli Miettinen and Jilles Vreeken. "Model order selection for boolean matrix factorization." In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011, pp. 51–59.

[113] George A Miller. "WordNet: a lexical database for English." In: *Communications of the ACM* 38.11 (1995), pp. 39–41.

[114] Sushmita Mitra and Haider Banka. "Multi-objective evolutionary biclustering of gene expression data." In: *Pattern Recognition* 39.12 (2006), pp. 2464–2477.

[115] Bhopal Mohapatra, Gulzar Ahmad, Scott Nadeau, Neha Zutshi, Wei An, Sarah Scheffe, Lin Dong, Dan Feng, Benjamin Goetz, Priyanka Arya, et al. "Protein tyrosine kinase regulation by ubiquitination: critical roles of Cbl-family ubiquitin ligases." In: *Biochimica Et Biophysica Acta (BBA)-Molecular Cell Research* 1833.1 (2013), pp. 122–139.

[116] Carrie B Moore, John R Wallace, Alex T Frase, Sarah A Pendergrass, and Marylyn D Ritchie. "BioBin: a bioinformatics tool for automating the binning of rare variants using publicly available biological knowledge." In: *BMC medical genomics* 6.2 (2013), pp. 1–12.

[117] Francesca Ester Morreale and Helen Walden. "Types of ubiquitin ligases." In: *Cell* 165.1 (2016), pp. 248–248.

[118] Ardhendu S Mukherjee and W Beermann. "Synthesis of ribonucleic acid by the X-chromosomes of Drosophila melanogaster and the problem of dosage compensation." In: *Nature* 207.4998 (1965), pp. 785–786.

[119] TM Murali and Simon Kasif. "Extracting conserved gene expression motifs from gene expression data." In: *Biocomputing 2003*. World Scientific, 2002, pp. 77–88.

[120] Saul B Needleman and Christian D Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins." In: *Journal of molecular biology* 48.3 (1970), pp. 443–453.

[121] Juan A Nepomuceno, Alicia Troncoso, Isabel A Nepomuceno-Chamorro, and Jesús S Aguilar-Ruiz. "Integrating biological knowledge based on functional annotations for biclustering of gene expression data." In: *Computer methods and programs in biomedicine* 119.3 (2015), pp. 163–180.

[122] Juan A Nepomuceno, Alicia Troncoso, Isabel A Nepomuceno-Chamorro, and Jesús S Aguilar-Ruiz. "Biclustering of Gene Expression Data Based on SimUI Semantic Similarity Measure." In: *International Conference on Hybrid Artificial Intelligence Systems*. 2016, pp. 685–693.

[123] Juan A Nepomuceno, Alicia Troncoso, Isabel A Nepomuceno-Chamorro, and Jesús S Aguilar-Ruiz. "Pairwise gene GO-based measures for biclustering of high-dimensional expression data." In: *BioData Mining* 11.1 (2018), p. 4.

[124] Rasmus Nielsen, Joshua S Paul, Anders Albrechtsen, and Yun S Song. "Genotype and SNP calling from next-generation sequencing data." In: *Nature Reviews Genetics* 12.6 (2011), pp. 443–451.

[125] Tobias Österlund, Marija Cvijovic, and Erik Kristiansson. "Integrative analysis of omics data." In: *Systems biology* 6 (2017), p. 1.

[126] René Peeters. "The maximum edge biclique problem is NP-complete." In: *Discrete Applied Mathematics* 131.3 (2003), pp. 651–654.

[127] Catia Pesquita, Daniel Faria, Hugo Bastos, António EN Ferreira, André O Falcão, and Francisco M Couto. "Metrics for GO based protein semantic similarity: a systematic evaluation." In: *BMC bioinformatics*. Vol. 9. S5. Springer. 2008, S4.

[128] Victoria Petri, Pushkala Jayaraman, Marek Tutaj, G Thomas Hayman, Jennifer R Smith, Jeff De Pons, Stanley JF Laulederkind, Timothy F Lowry, Rajni Nigam, Shur-Jen Wang, et al. "The pathway ontology–updates and applications." In: *Journal of biomedical semantics* 5.1 (2014), pp. 1–12.

[129] Robert Petryszak, Maria Keays, Y Amy Tang, Nuno A Fonseca, Elisabet Barrera, Tony Burdett, Anja Füllgrabe, Alfonso Muñoz-Pomer Fuentes, Simon Jupp, Satu Koskinen, et al. "Expression Atlas update—an integrated database of gene and protein expression in humans, animals and plants." In: *Nucleic acids research* 44.D1 (2015), pp. D746–D752.

[130] Katharina Podwojski, Arno Fritsch, Daniel C Chamrad, Wolfgang Paul, Barbara Sitek, Kai Stühler, Petra Mutzel, Christian Stephan, Helmut E Meyer, Wolfgang Urfer, et al. "Retention time alignment algorithms for LC/MS data must consider non-linear shifts." In: *Bioinformatics* 25.6 (2009), pp. 758–764.

[131] Beatriz Pontes, Raúl Giráldez, and Jesús S Aguilar-Ruiz. "Biclustering on expression data: A review." In: *Journal of biomedical informatics* 57 (2015), pp. 163–180.

[132] Amol Prakash, Parag Mallick, Jeffrey Whiteaker, Heidi Zhang, Amanda Paulovich, Mark Flory, Hookeun Lee, Ruedi Aebersold, and Benno Schwikowski. "Signal maps for mass spectrometry-based comparative proteomics." In: *Molecular & Cellular Proteomics* 5.3 (2006), pp. 423–432.

[133] Amela Prelić, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Bühlmann, Wilhelm Gruissem, Lars Hennig, Lothar Thiele, and Eckart Zitzler. "A systematic comparison and evaluation of biclustering methods for gene expression data." In: *Bioinformatics* 22.9 (2006), pp. 1122–1129.

[134] John T Prince and Edward M Marcotte. "Chromatographic alignment of ESI-LC-MS proteomics data sets by ordered bijective interpolated warping." In: *Analytical chemistry* 78.17 (2006), pp. 6140–6152.

[135] J Ross Quinlan. *C4.5. programs for machine learning*. San Mateo, Calif.: Morgan Kaufmann Publishers, c1993. ISBN: 1558602380.

[136] Dhivyaa Rajasundaram and Joachim Selbig. "More effort—more results: recent advances in integrative 'omics' data analysis." In: *Current opinion in plant biology* 30 (2016), pp. 57–61.

[137] Domingo S Rodriguez-Baena, Antonio J Perez-Pulido, and Jesus S Aguilar-Ruiz. "A biclustering algorithm for extracting bit-patterns from binary datasets." In: *Bioinformatics* 27.19 (2011), pp. 2738–2745.

[138] Stuart J. Russell, Peter. Norvig, and Ernest. Davis. *Artificial intelligence. a modern approach*. 3rd ed. Upper Saddle River: Prentice Hall, c2010. ISBN: 0136042597.

[139] Tasleem Samji, Soonwook Hong, and Robert E Means. "The membrane associated RING-CH proteins: a family of E3 ligases with diverse roles through the cell." In: *International scholarly research notices* 2014 (2014).

[140] Toshiro Sato, Johan H Van Es, Hugo J Snippert, Daniel E Stange, Robert G Vries, Maaike Van Den Born, Nick Barker, Noah F Shroyer, Marc Van De Wetering, and Hans Clevers. "Paneth cells constitute the niche for Lgr5 stem cells in intestinal crypts." In: *Nature* 469.7330 (2011), pp. 415–418.

[141] Lynn Marie Schriml, Cesar Arze, Suvarna Nadendla, Yu-Wei Wayne Chang, Mark Mazaitis, Victor Felix, Gang Feng, and Warren Alden Kibbe. "Disease Ontology: a backbone for disease semantic integration." In: *Nucleic acids research* 40.D1 (2011), pp. D940–D946.

[142] Jurian Schuijers and Hans Clevers. "Adult mammalian stem cells: the role of Wnt, Lgr5 and R-spondins." In: *The EMBO journal* 31.12 (2012), pp. 2685–2696.

[143] *Semantic Biclustering Project*. http://github.com/IDActu/semantic-biclustering. [Online; accessed 30-January-2017].

[144] Qizheng Sheng, Yves Moreau, and Bart De Moor. "Biclustering microarray data by Gibbs sampling." In: *Bioinformatics* 19.suppl_2 (2003), pp. ii196–ii205.

[145] Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, et al. "The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration." In: *Nature biotechnology* 25.11 (2007), p. 1251.

[146] Colin A Smith, Elizabeth J Want, Grace O'Maille, Ruben Abagyan, and Gary Siuzdak. "XCMS: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification." In: *Analytical chemistry* 78.3 (2006), pp. 779–787.

[147] Rob Smith, Andrew D Mathis, Dan Ventura, and John T Prince. "Proteomics, lipidomics, metabolomics: a mass spectrometry tutorial from a computer scientist's point of view." In: *BMC bioinformatics* 15.7 (2014), pp. 1–14.

[148] Rob Smith, John T Prince, and Dan Ventura. "A coherent mathematical characterization of isotope trace extraction, isotopic envelope extraction, and LC-MS correspondence." In: *BMC bioinformatics* 16.7 (2015), S1.

[149] Rob Smith, Dan Ventura, and John T Prince. "LC-MS alignment in theory and practice: a comprehensive algorithmic review." In: *Briefings in bioinformatics* 16.1 (2013), pp. 104–117.

[150] Temple F Smith, Michael S Waterman, et al. "Identification of common molecular subsequences." In: *Journal of molecular biology* 147.1 (1981), pp. 195–197.

[151] Hugo J Snippert, Laurens G Van Der Flier, Toshiro Sato, Johan H Van Es, Maaike Van Den Born, Carla Kroon-Veenboer, Nick Barker, Allon M Klein, Jacco Van Rheenen, Benjamin D Simons, et al. "Intestinal crypt homeostasis results from neutral competition between symmetrically dividing Lgr5 stem cells." In: *Cell* 143.1 (2010), pp. 134–144.

[152] Lloyd R Snyder and John W Dolan. *High-performance gradient elution: the practical application of the linear-solvent-strength model*. John Wiley & Sons, 2007.

[153] Arnaud Soulet, Jiří Kléma, and Bruno Crémilleux. "Efficient mining under rich constraints derived from various datasets." In: *International Workshop on Knowledge Discovery in Inductive Databases*. Springer. 2006, pp. 223–239.

[154] Sarika Srivastava. "Emerging insights into the metabolic alterations in aging using metabolomics." In: *Metabolites* 9.12 (2019), p. 301.

[155] Robert Stevens, Carole A Goble, and Sean Bechhofer. "Ontology-based knowledge representation for bioinformatics." In: *Briefings in bioinformatics* 1.4 (2000), pp. 398–414.

[156] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles." In: *Proceedings of the National Academy of Sciences* 102.43 (2005), pp. 15545–15550.

[157] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. "Yago: a core of semantic knowledge." In: *Proceedings of the 16th international conference on World Wide Web*. ACM. New York, N.Y: ACM Press, 2007, pp. 697–706.

[158] Kate D Sutherland, Geoffrey J Lindeman, David YH Choong, Sergio Wittlin, Luci Brentzell, Wayne Phillips, Ian G Campbell, and Jane E Visvader. "Differential hypermethylation of SOCS genes in ovarian and breast carcinomas." In: *Oncogene* 23.46 (2004), pp. 7726–7733.

[159] Martin Svatoš, Gustav Šourek, Filip Železný, Steven Schockaert, and Ondřej Kuželka. "Pruning Hypothesis Spaces Using Learned Domain Theories." In: *International Conference on Inductive Logic Programming*. Springer. Cham, Switzerland: Springer, 2018, pp. 152–168.

[160] Amos Tanay, Roded Sharan, and Ron Shamir. "Discovering statistically significant biclusters in gene expression data." In: *Bioinformatics* 18.suppl_1 (2002), S136–S144.

[161] Chun Tang, Li Zhang, Aidong Zhang, and Murali Ramanathan. "Interrelated two-way clustering: an unsupervised approach for gene expression data analysis." In: *Proceedings 2nd Annual IEEE International Symposium on Bioinformatics and Bioengineering (BIBE 2001)*. IEEE. 2001, pp. 41–48.

[162] Diethard Tautz. "Problems and paradigms: Redundancies, development and the flow of information." In: *Bioessays* 14.4 (1992), pp. 263–266.

[163] Xinchen Teng, Margaret Dayhoff-Brannigan, Wen-Chih Cheng, Catherine E Gilbert, Cierra N Sing, Nicola L Diny, Sarah J Wheelan, Maitreya J Dunham, Jef D Boeke, Fernando J Pineda, et al. "Genome-wide consequences of deleting any single gene." In: *Molecular cell* 52.4 (2013), pp. 485–494.

[164] Shulan Tian, Huihuang Yan, Michael Kalmbach, and Susan L Slager. "Impact of post-alignment processing in variant discovery from whole exome data." In: *BMC bioinformatics* 17.1 (2016), p. 403.

[165] William S Tobelaim, Claudia Beaurivage, Audrey Champagne, Véronique Pomerleau, Aline Simoneau, Walid Chababi, Mehdi Yeganeh, Philippe Thibault, Roscoe Klinck, Julie C Carrier, et al. "Tumour-promoting role of SOCS1 in colorectal cancer cells." In: *Scientific reports* 5.1 (2015), pp. 1–13.

[166] Ann M Turnley, Clare H Faux, Rodney L Rietze, Jason R Coonan, and Perry F Bartlett. "Suppressor of cytokine signaling 2 regulates neuronal differentiation by inhibiting growth hormone signaling." In: *Nature neuroscience* 5.11 (2002), pp. 1155–1162.

[167] Pradeep D Uchil, Angelika Hinz, Steven Siegel, Anna Coenen-Stass, Thomas Pertel, Jeremy Luban, and Walther Mothes. "TRIM protein-mediated regulation of inflammatory and innate immune signaling and its association with antiretroviral activity." In: *Journal of virology* 87.1 (2013), pp. 257–272.

[168] Miranda van Uitert, Wouter Meuleman, and Lodewyk Wessels. "Biclustering sparse binary genomic data." In: *Journal of Computational Biology* 15.10 (2008), pp. 1329–1345.

[169] Iven Van Mechelen, Hans-Hermann Bock, and Paul De Boeck. "Two-mode clustering methods: a structured overview." In: *Statistical methods in medical research* 13.5 (2004), pp. 363–394.

[170] Marie Verbanck, Sébastien Lê, and Jérôme Pagès. "A new unsupervised gene clustering algorithm based on the integration of biological knowledge into expression data." In: *BMC bioinformatics* 14.1 (2013), p. 1.

[171] Nawaporn Vinayavekhin and Alan Saghatelian. "Untargeted metabolomics." In: *Current protocols in molecular biology* 90.1 (2010), pp. 30–1.

[172] Hui Wang, Tujin Shi, Wei-Jun Qian, Tao Liu, Jacob Kagan, Sudhir Srivastava, Richard D Smith, Karin D Rodland, and David G Camp. "The clinical impact of recent advances in LC-MS for cancer biomarker discovery and verification." In: *Expert review of proteomics* 13.1 (2016), pp. 99–114.

[173] Kai Wang, Mingyao Li, and Hakon Hakonarson. "ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data." In: *Nucleic acids research* 38.16 (2010), e164–e164.

[174] Ron Wehrens, Jos A Hageman, Fred van Eeuwijk, Rik Kooke, Pádraic J Flood, Erik Wijnker, Joost JB Keurentjes, Arjen Lommen, Henriëtte DLM van Eekelen, Robert D Hall, et al. "Improved batch correction in untargeted MS-based metabolomics." In: *Metabolomics* 12.5 (2016), p. 88.

[175] Jacqueline K White, Anna-Karin Gerdin, Natasha A Karp, Ed Ryder, Marija Buljan, James N Bussell, Jennifer Salisbury, Simon Clare, Neil J Ingham, Christine Podrini, et al. "Genome-wide generation and systematic phenotyping of knockout mice reveals new roles for many genes." In: *Cell* 154.2 (2013), pp. 452–464.

[176] I. H. Witten, Eibe Frank, and Mark A. Hall. *Data mining. practical machine learning tools and techniques*. 3rd ed. Burlington: Morgan Kaufmann, c2011. ISBN: 9780123748560.

[177] Yang Xiang, Ruoming Jin, David Fuhry, and Feodor F Dragan. "Summarizing transactional databases with overlapped hyperrectangles." In: *Data Mining and Knowledge Discovery* 23.2 (2011), pp. 215–251.

[178] Juan Xie, Anjun Ma, Anne Fennell, Qin Ma, and Jing Zhao. "It is time to apply biclustering: a comprehensive review of biclustering applications in biological and biomedical data." In: *Briefings in bioinformatics* 20.4 (2019), pp. 1450–1465.

[179] Juan Xie, Anjun Ma, Yu Zhang, Bingqiang Liu, Sha Cao, Cankun Wang, Jennifer Xu, Chi Zhang, and Qin Ma. "QUBIC2: a novel and robust biclustering algorithm for analyses and interpretation of large-scale RNA-Seq data." In: *Bioinformatics* 36.4 (2020), pp. 1143–1149.

[180] Qi Yang, Nessan A Bermingham, Milton J Finegold, and Huda Y Zoghbi. "Requirement of Math1 for secretory cell lineage commitment in the mouse intestine." In: *Science* 294.5549 (2001), pp. 2155–2158.

[181] Hirohide Yoshikawa, Kenichi Matsubara, Geng-Sun Qian, Peta Jackson, John D Groopman, Jasper E Manning, Curtis C Harris, and James G Herman. "SOCS-1, a negative regulator of the JAK/STAT pathway, is silenced by methylation in human hepatocellular carcinoma and shows growth-suppression activity." In: *Nature genetics* 28.1 (2001), pp. 29–35.

[182] *ZOOMA*. https://www.ebi.ac.uk/spot/zooma/. [Online; accessed 30-April-2018].

[183] Monika Žáková and Filip Železný. "Exploiting term, predicate, and feature taxonomies in propositionalization and propositional rule learning." In: *European Conference on Machine Learning*. Springer. 2007, pp. 798–805.

[184] Filip Železný and Nada Lavrač. "Propositionalization-based relational subgroup discovery with RSD." In: *Machine Learning* 62.1 (2006), pp. 33–63.

[185] Zhong-Yuan Zhang, Tao Li, Chris Ding, Xian-Wen Ren, and Xiang-Sun Zhang. "Binary matrix factorization for analyzing gene expression data." In: *Data Mining and Knowledge Discovery* 20.1 (2010), pp. 28–52.

[186] Marinka Žitnik and Blaž Zupan. “Nimfa: A python library for nonnegative matrix factorization.” In: *The Journal of Machine Learning Research* 13.1 (2012), pp. 849–853.