KNOWLEDGE DISCOVERY IN A SERVICE-ORIENTED DATA MINING ENVIRONMENT

Vid Podpečan

Doctoral Dissertation
Jožef Stefan International Postgraduate School
Ljubljana, Slovenia, January 2013

Evaluation Board:
prof. dr. Sašo Džeroski, Chairman, Jožef Stefan Institute, Ljubljana, Slovenia
prof. dr. Marko Bohanec, Member, Jožef Stefan Institute, Ljubljana, Slovenia
prof. dr. Filip Železný, Member, Czech Technical University, Prague, Czech Republic

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Vid Podpečan

KNOWLEDGE DISCOVERY IN A SERVICE-ORIENTED DATA MINING ENVIRONMENT

Doctoral Dissertation

ODKRIVANJE ZAKONITOSTI IZ PODATKOV V OKOLJU SPLETNIH SERVISOV

Doktorska disertacija

Supervisor: prof. dr. Nada Lavrač

Ljubljana, Slovenia, January 2013

Contents

Abstract

The thesis addresses the development of novel knowledge discovery scenarios in a modern data mining platform by utilising principles of service-oriented architecture with web services, interactive scientific workflows, knowledge discovery ontologies and automated construction of data mining workflows.

We present the developed Orange4WS platform which upgrades Orange, a mature open-source data mining toolkit. Orange4WS enables seamless integration of web services by implementing a widget code generator and provides tools for web service development, which adhere to the contract-first design principle. These tools are used to develop web services from different domains, including systems biology, data mining, text mining, and natural language processing. Furthermore, Orange4WS integrates the knowledge discovery ontology, an ontology which defines relationships among the components of knowledge discovery scenarios, and a workflow planner which enables automated construction of data mining workflows. The applicability of the Orange4WS platform is demonstrated and evaluated on several use cases.

Two advanced data analysis methodologies from the domain of systems biology were developed using Orange4WS: the SegMine methodology and a methodology for contrasting subgroup discovery.

The SegMine methodology enables semantic analysis of gene expression data by integrating a semantic subgroup discovery, interactive hierarchical clustering, and probabilistic link discovery using the Biomine system. Components of the SegMine methodology integrate publicly available Gene Ontology (GO), Kyoto Encyclopedia of Genes and Genomes (KEGG), gene-gene interaction data and several other public databases. SegMine enables advanced data interpretation and formulation of research hypotheses by integrating the analysis of experimental data with publicly available knowledge. The methodology is implemented in Orange4WS as a set of interactive workflow components and evaluated on two data sets, a well-known dataset from a clinical trial in acute lymphoblastic leukemia (ALL), and a dataset about senescence in human mesenchymal stem cells (MSC). The experiments on mesenchymal stem cells data set resulted in the formulation of three new scientific hypotheses.

The developed contrasting subgroup discovery allows for mining of subgroups that cannot be discovered using classical subgroup discovery. The methodology proposes a three-step approach where subgroup discovery in the first and the last step is complemented by the intermediate contrast definition step. The steps of the methodology, the differences with standard subgroup discovery, and the examples of set-theoretic functions for defining contrast classes are presented and illustrated on a simple use case. The methodology was applied to a systems biology domain. More specifically, the results of applying the methodology on a time series gene expression dataset for virus-infected Solanum tuberosum (potato) plants are presented and evaluated by a domain expert. The proposed methodology is implemented as a set of interactive

workflow components in the Orange4WS platform.

The thesis also contributes to open-source scientific software. The Orange4WS platform, the implementations of the SegMine methodology and contrasting subgroup discovery, as well as other Orange4WS use cases are available to general public. This enables experiment reproducibility, as well as workflow adaptation and enhancement.

Povzetek

Disertacija obravnava razvoj novih scenarijev odkrivanja znanja v modernem okolju za podatkovno rudarjenje z uporabo principov servisno orientirane arhitekture, spletnih servisov, interaktivnih delotokov, ontologij ter avtomatske gradnje delotokov za podatkovno rudarjenje.

Razvili smo orodje Orange4WS, ki nadgrajuje Orange, obstoječe odprtokodno orodje Orange za podatkovno rudarjenje. Orange4WS omogoča enostavno uporabo spletnih servisov s pomočjo generatorja programske kode komponent delotokov ter ponuja orodja za razvoj spletnih servisov po principu razvoja od zgoraj navzdol. Ta orodja so uporabljena za razvoj spletnih servisov v domenah sistemske biologije, rudarjenja podatkov, rudarjenja besedil in procesiranja naravnega jezika. Orange4WS vključuje tudi ontologijo odkrivanja znanja, ki določa relacije med komponentami v scenarijih za odkrivanje znanja, ter načrtovalca delotokov, ki omogoča samodejno gradnjo delotokov za rudarjenje podatkov. Delovanje orodja Orange4WS je prikazano in ovrednoteno na več primerih uporabe.

Z uporabo orodja Orange4WS sta bili razviti dve napredni metodologiji za analizo podatkov s področja sistemske biologije: SegMine in odkrivanje kontrastnih podskupin (contrasting subgroup discovery). Metodologija SegMine omogoča semantično analizo podatkov o izraženosti genov s povezavo algoritma za semantično odkrivanje podskupin s postopkom interaktivnega hierarhičnega razvrščanja v skupine ter s sistemom Biomine za verjetnostno odkrivanje povezav. Komponente metodologije SegMine uporabljajo prosto dostopne vire kot npr. ontologijo genov (Gene Ontology, GO), enciklopedijo genov in genomov (Kyoto Encyclopedia of Genes and Genomes, KEGG), bazo interakcij med geni (Entrez) ter številne druge javno dostopne podatkovne baze. SegMine omogoča razlago podatkov ter postavljanje znanstvenih hipotez z združevanjem eksperimentalnih podatkov in javno dostopnega znanja. Metodologija je implementirana v orodju Orange4WS kot množica interaktivnih komponent delotokov ter ovrednotena na dveh naborih podatkov: znanem naboru o kliničnem testiranju akutne limfoblastne levkemije (ALL) ter naboru podatkov o senescenci človeških zarodnih celic (MSC). Ekspertna analiza podatkov o zarodnih celicah z metodologijo SegMine je vodila v oblikovanje treh novih znanstvenih hipotez.

Predstavljena je tudi metodologija za odkrivanje kontrastnih podskupin, katerih ni mogoče najti z uporabo klasičnih metod odkrivanja podskupin. Metodologija predlaga tristopenjski pristop, v katerem sta odkrivanje podskupin v prvem in zadnjem koraku dopolnjena z določitvijo kontrastnih razredov v vmesnem koraku. Koraki metodologije, razlike s klasičnim odkrivanjem podskupin ter primeri funkcij iz teorije množic za definiranje kontrastnih razredov so predstavljeni in ponazorjeni na preprostem primeru. Metodologija je uporabljena v domeni sistemske biologije, predstavljeni pa so rezultati njene uporabe na časovni vrsti s podatki o izraženosti genov z virusom okuženih rastlin krompirja (Solanum tuberosum) ter ekspertna analiza

rezultatov. Metodologija je implementirana v orodju Orange4WS kot nabor interaktivnih komponent delotokov.

Disertacija doprinaša tudi k razvoju odprtokodne programske opreme v znanosti. Orodje Orange4WS, implementaciji metodologij SegMine in odkrivanja kontrastnih podskupin ter ostali primeri uporabe orodja Orange4WS so dostopni širši javnosti, kar omogoča ponovitve eksperimentov ter prilagajanje in dopolnjevanje delotokov.

Abbreviations

ALL	=	acute lymphoblastic leukemia
API	=	application programming interface
CRISP-DM	=	cross industry standard process for data mining
CSD	=	contrasting subgroup discovery
DAVID	=	database for annotation, visualization and integrated discovery
DMOP	=	data mining optimization ontology
DMWF	=	data mining workflow ontology
GEO	=	gene expression omnibus
GO	=	gene ontology
GSEA	=	gene set enrichment analysis
HTTP	=	hypertext transfer protocol
IDA	=	intelligent discovery assistant
JRE	=	Java runtime environment
KDD	=	knowledge discovery in databases
KEGG	=	Kyoto encyclopedia of genes and genomes orthology
mHG	=	minimum hyper-geometric
MSC	=	mesenchymal stem cell
OWL	=	web ontology language
PAGE	=	parametric analysis of gene set enrichment
PMC	=	PubMed Central
PMML	=	predictive model markup language
QoS	=	quality of service
RDBMS	=	relational database management system
REST	=	representational state transfer
RPC	=	remote procedure call
RSD	=	relational subgroup discovery
RSS	=	rich site summary
SaaS	=	software as a service
SD	=	subgroup discovery
SEGS	=	search for enriched gene sets
SOA	=	service-oriented architecture
SOAP	=	simple object access protocol
SQL	=	structured query language
URL	=	universal resource locator
WADL	=	web application description language
WCF	=	Windows communication foundation
WS-BPEL	=	web services business process execution language
WSDL	=	web service description language
WSRF	=	web services resources framework
WebGL	=	web graphics library
XML	=	extended markup language

XMPP	=	extensible messaging and presence protocol
XSD	=	XML schema definition

1 Introduction

This chapter summarises the motivation and introduces the general background of the relevant topics. It gives a brief overview of the developed solutions which are presented in more detail in the subsequent chapters. The research hypothesis and the main goals are also presented, followed by a list of scientific contributions. The chapter concludes by outlining the structure of the thesis.

1.1 Data mining and knowledge discovery

The rate at which humans generate new data is enormous. As the advances in electronics and computer technologies have enabled almost unlimited storage resources, virtually every bit of new data is stored, preserved, and made available. The Internet, an electronic network spread around the planet, hosts an almost unimaginable amount of human-generated data (Google's engineers already claimed 1 trillion of indexed unique URLs as of 2008). Apart from the static data in the form of internet pages, documents and databases, it also offers many sources of live data streams, such as RSS feeds, blogs, and various online social platforms, which constantly produce new data. Besides the Internet, which is mostly available to the general public, there are numerous corporate, private and personal networks, data storages and databases where huge amounts of specific data and information are being stored. A vast amount of experimental data from various scientific domains, such as systems biology, physics, astronomy, geology and others, also exists. For example, the detectors of the Large Hadron Collider experiments yield a stream of filtered data at about 300MB/s (The LCG TDR Editorial Board, 2005) which has to be stored and analysed. Microarray experiments in systems biology, which are also the topic of interest in this thesis, yield the expression values of tens of thousands of genes in each experiment. Most biological data and knowledge resources get integrated in publicly available databases. For example, the Biomine system (Eronen and Toivonen, 2012; Sevon et al., 2006), which enables search and discovery of non-trivial connections between biological entities, integrates 9 major databases into a large probabilistic graph consisting of more than 1 million nodes. However, not only experimental data are collected and stored in structured databases. The PubMed database[1], for example, contains more than 21 million citations (as of June 2012) for biomedical literature from MEDLINE, life science journals, and online books, many of which include links to full-text content (for example, the PubMed Central[2] (PMC) online library contains more than 2.4 million publicly available full-text articles).

Modern computer technology and software design not only allow for data acquisition and storage, but also for large scale data analysis. The efforts in this direction have resulted in numerous high-performance data mining libraries, online resources (such as web services) for data search and analysis, cloud services for data storage and analysis, open standards for data exchange and descriptions of data analysis models (e.g., PMML[3]), and ontologies which describe

[1] `http://www.ncbi.nlm.nih.gov/pubmed/`

[2] `http://www.ncbi.nlm.nih.gov/pmc/`

[3] Predictive Model Markup Language (PMML) is an XML-based markup language for describing predictive

the semantics of data, tools and services with the aim to enable their interoperability. One of the goals of this thesis is to provide an environment capable of effectively integrating data, computing, and semantic resources, which will enable the user to employ the available resources and create novel scenarios for data mining and knowledge discovery.

The term *data mining* denotes the activity of extracting new, valuable and nontrivial information from large volumes of data (Cios et al., 2010; Fayyad et al., 1996). Most commonly, the aim is to find patterns or build models using specific algorithms from various scientific disciplines including artificial intelligence, machine learning, database systems and statistics. According to this definition, the data mining tasks can be classified into two categories:

1. *predictive data mining* where the goal is to build an executable model from data which can be used for classification, prediction or estimation, and

2. *descriptive data mining* where the goal is to discover interesting patterns and relationships in data.

Although the term *data mining* is sometimes still considered a buzzword (Bouckaert et al., 2010), it is now an established concept denoting the computer-assisted application of specific algorithms for extracting patterns and/or models from data in a general multi-step process of extracting knowledge from data. According to the established definition (Fayyad et al., 1996), the term *knowledge discovery in databases (KDD)* denotes the overall process of discovering useful knowledge from data in which data mining is one particular step. However, this activity has been given different names in different communities. For example, Fayyad et al. (1996) list the following terms which denote the same: knowledge extraction, information discovery, information harvesting, data archaeology, data pattern processing and data mining.

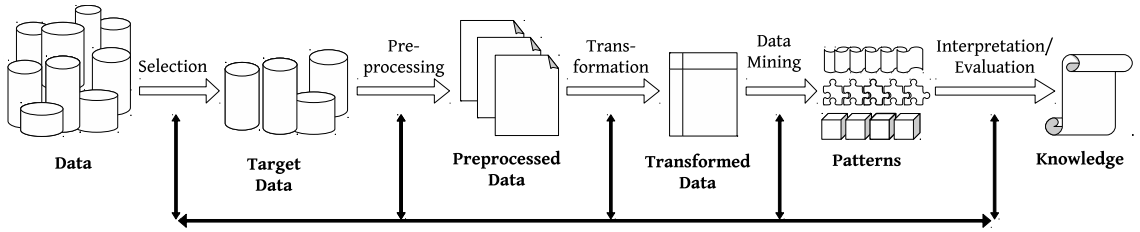

Figure 1.1: Overview of the steps of the KDD process (adapted from Fayyad et al. (1996)).

The KDD process is illustrated in Figure 1.1. Its major steps can be briefly summarised as follows (Fayyad et al., 1996):

1. **Data selection**. The first step starts with collecting the necessary and relevant knowledge about the domain and setting the goals to be achieved. This information is then used in the preparation of a dataset which includes selecting an appropriate (sub)set of data samples and/or variables.

2. **Data cleaning and preprocessing**. The second step consists of modelling and removing noise and outliers and handling of missing data fields. Temporal information about the data can also be taken into account, as well as various issues regarding the database system.

3. **Data transformation**. This step involves finding useful features representing the data with respect to the given goal, and applying dimensionality reduction and transforma-

and data mining models.

tion techniques to reduce the number of variables under consideration thus reducing the complexity of the given data analysis task.

4. **Data mining**. This is the most elaborate step as it consists of choosing the function of data mining, choosing the right data mining algorithm and its application. Choosing the function includes deciding the purpose of the resulting data mining model, such as classification, regression, clustering and summarisation. The selection of the data mining algorithm encompasses the decision which models and parameters are appropriate and matching with the criteria of the process (i.e., trade-off between the predictive power of the data mining model and its understandability).

5. **Evaluation, interpretation and application**. In the last step the discovered patterns are evaluated and their validity and relevance are assessed. Redundant and irrelevant patterns are removed while the remaining, relevant patterns are studied and interpreted, typically with the aid of computer-assisted visualisation techniques. Application of the discovered knowledge (i.e., new interesting patterns, models, and other induced information) includes resolving potential conflicts with existing knowledge, taking actions based on the obtained knowledge, such as aiding, modifying and improving existing processes and procedures, especially those involving human experts, and storing, documenting and reporting to interested parties.

Similarly, the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology (Shearer, 2000), the leading methodology used by data analysts according to user polls[1] defines a process model consisting of six major phases to accomplish the task of expert data analysis: business understanding, data understanding, data preparation, modelling, evaluation and deployment. Note that there exist several other knowledge discovery process models which are not discussed here (Anand et al., 1998; Cabena, 1998; Cios et al., 2010).

In the remainder of this thesis, the terms *data mining platform*, *data mining environment*, and *data mining toolkit* denote software capable of performing most of the tasks of the KDD process as described above. Note, however, that knowledge discovery in databases is not a monolithic process model. It continues to evolve by incorporating modern solutions designed to cope with the ever-increasing amount of data, software, and semantics. In the following, we summarise some of the key research issues related to next generation data mining and knowledge discovery as identified at the National Science Foundation Symposium on Next Generation of Data Mining and Cyber-Enabled Discovery for Innovation (NGDM) (Agrawal et al., 2007; Kargupta et al., 2008). The presented issues can be loosely grouped into the following topics: data and semantics, privacy protection in data mining, ubiquitous, distributed and high performance data mining and platforms, interpretation of data and data mining results.

- **Data and semantics**. (a) Next generation data mining should enable routine analysis of rich sources of data, like literary texts, video and user generated data like blogs, in a straightforward manner. (b) New methods are needed to include multiple sources of supporting data in the data mining process and to provide results that are meaningful to the domain expert. (c) Researchers should be encouraged to specify their data models and schemata using semantic languages, such as OWL, reusing appropriate published ontologies where possible and publishing any new ontologies developed. (d) Research is needed to develop integrated frameworks, such as inductive databases that unify the processes of

[1] `http://www.kdnuggets.com/polls/2007/data_mining_methodology.htm`

storing, retrieving, mining and managing scientific data and knowledge. (e) New data mining algorithms should exploit existing background knowledge and semantics.

- **Privacy protection in data mining**. (a) Research is needed to address the privacy concerns when mining personal information on the web, query logs and click stream data. (b) Metrics that bridge the gap between legal definitions and technological protections need to be developed, and we need to understand the different implications of data mining on privacy in different domains. (c) New technology is needed to understand and maintain privacy as data mining expands to mobile (spatio/temporal) data and complex relationships (e.g., social networks). (d) We need to improve the performance/security tradeoffs in privacy-preserving data mining.

- **Ubiquitous, distributed and high performance data mining and platforms**. (a) Develop appropriate middleware to mine very large data, geographically distributed data, and ubiquitous data. (b) Develop algorithms and analytic libraries that exploit a high level of parallelism that will be available in emerging systems. (c) Develop service-oriented data mining architectures. (d) Develop data mining platforms and middleware that take advantage of emerging wide area, high performance networks.

- **Interpretation of data and data mining results**. (a) Multidisciplinary teams should be involved both in incorporating domain knowledge into the learning process, and in the interpretation and evaluation of data mining models in the context of specific applications. (b) We need to develop new interactive discovery tools and visualisation techniques that take advantage of human perceptual abilities. (c) Domain experts need more advanced visual interfaces for exploratory data analysis both in traditional and non-traditional applications.

In this thesis, we are concerned with the development of a modern data mining platform employing principles of service-oriented architecture, general data mining ontologies and specific domain ontologies, and complex knowledge discovery scenarios from various scientific disciplines. Some of the research goals pursued in this thesis can be aligned to the recognised challenges for the next generation data mining and knowledge discovery tools discussed above.

In the next section, we briefly review the principles and design of service-oriented architectures. We then put them into the context of data mining and knowledge discovery software plaftorms in Section 1.3.

1.2 Service-oriented architectures

Service-oriented architecture (SOA) is a term related to the design of software systems that has been extensively used[1] in the last decade. In its abstract form, it denotes a set of principles and techniques for developing distributed software using components or units known as *services* with the aim of achieving application interoperability and reuse of IT assets (Erl, 2005; Newcomer and Lomow, 2005). In this context, services are individual, autonomous software units (building blocks) which conform to a common set of principles that allow them to evolve independently while maintaining commonality and standardisation (Erl, 2005).

Services *encapsulate* processing logic within some context which can be a specific process step, sub-process, entire process or even logic provided by other services. As the services are required

[1]Similarly to the term *data mining*, the term *service-oriented architecture* has been abused as a buzzword (Jahn, 2006) to attract attention.

to interact, they rely on *service descriptions* which establish names and locations of services, as well as their data exchange requirements. Service descriptions ensure *loose coupling* of services, but a suitable communication framework is also required. One commonly used framework is *messaging* where services exchange autonomous communication units. Overall, service-oriented architectures address the following issues (Erl, 2005) regarding services: (a) how the services are designed, (b) how the service descriptions are designed, (c) how the messages are designed, and (d) how the relationship between services is defined. The key guiding principles regarding these issues are as follows.

- **Loose coupling**. This is the most important service characteristic which refers to different elements of a service, its implementation and usage. Three types of coupling are considered (Newcomer and Lomow, 2005):
  - *Technology coupling*, which is related to how much a service depends on particular technology, product and development platform. A general guideline here is to avoid relying on features that are proprietary to a specific vendor, and to use open technology.
  - *Interface coupling*, which refers to the coupling betweeen service requesters and service providers, that is, dependencies that the service provider imposes on the requester. The requester should not require any information about the internals of the services.
  - *Process coupling*, which refers to how much the service is tied to a particular process. Ideally, the service can be reused across many different processes and applications.
- **Autonomy**. Services have control over their logic and can execute their functionality without relying on external resources. Design-time as well as runtime autonomies are taken into account.
- **Service contract**. Services follow a common communication agreement in order to ensure reusability and recomposability. This is applied to three areas of service contracts:
  - *data model standardisation*: standardised data models are used to avoid transformation and improve interoperability,
  - *policy standardisation*: terms of usage for a service are expressed in a consistent manner using standardised policy expressions that are based on industry standard vocabularies,
  - *functional expression standardisation*: service's operations, input and output message names and their corresponding type names are defined using standardised naming conventions.
- **Abstraction**. To the outside, world services only expose what is described in the service contract. This corresponds to the principle of *information hiding* meaning that all unnecessary details about service's internals are hidden and the service contract is limited to what is required to effectively utilise the service. Four types of abstractions are taken into account: functional abstraction, logic abstraction, technology information abstraction, and quality of service abstraction.
- **Reusability**. Services are designed in a manner so that their solution logic is independent of any particular business process or technology thus promoting reuse. Note that a common pitfall is to create overly complex service logic with extra capabilities that will suit possible future service usage scenarios instead of reusable service's core logic.

- **Composability**. Services can be coordinated and assembled to form composite services. This promotes the creation of new solutions by reusing existing services.
- **Statelessness**. Services minimise information related to some activity in order to achieve scalability and to reduce the consumption of resources. This applies to state information on context data, session data, and business data.
- **Discoverability**. Services can be found and assessed via discovery mechanisms which rely on common language-based service discovery protocols. Several service discovery protocols exist including WS-Discovery, Universal Description Discovery and Integration (UDDI), XMPP Service Discovery and others.
- **Granularity**. Services are designed in a way which provides optimal scope of functionality. The key design factors are: performance, message size, transaction, and business function.

In principle, a service-oriented architecture can be implemented using different technologies. Some of the most well-known are the following:

- SOAP, a protocol for exchanging structured information, which relies on XML for the message format and HTTP protocol (most typically) for message transmission,
- RPC (remote procedure call), an inter-process communication technique that allows for executing a procedure in another address space,
- WCF (Windows communication foundation), Microsoft's runtime and API in the .NET Framework for building connected, service-oriented applications which support SOAP messaging, but also standard XML data,
- REST (Representational state transfer) (Fielding, 2000), software architecture style which is less strongly typed than SOAP and whose language is based on the use of nouns and verbs with the emphasis on readability,
- Web services, a software system designed to support interoperable machine-to-machine interaction over a network (McCabe et al., 2004) which has a public interface described in a machine-processable format called WSDL (Web Services Description Language) (Liu and Booth, 2007).

While it is possible to implement service-oriented architecture using any of the described (and other) technologies, web services are by far the most important and widely used. Moreover, modern texbooks on SOA (Erl, 2005; Newcomer and Lomow, 2005) consider Web services as a key to modern SOA realisations.

The key technologies which enable web service software are XML, SOAP, and WSDL. WSDL is especially important as it enables human- and machine-readable descriptions of web services and automated client-side code generation. The latest specification of the WSDL language (2.0) additionally offers support for REST-based (RESTful) web services, which further extends the potential use. Besides the described key standards, which form the web services framework, numerous web service specifications, collectively referred to as *WS-**, have emerged. They cover domains such as *messaging* (e.g., WS-Notification for event-driven programming and WS-Addressing for allowing web services to communicate addressing information), *metadata exchange* (e.g., WS-Policy which allows for advertising the policies and specify policy requirements and WS-Discovery which defines a discovery protocol to locate services on a local network), *security* (e.g., WS-Security which specifies how integrity and confidentiality can be enforced on

messages to provide end-to-end security and WS-Trust which extends WS-Security with issuing, renewing, and validating of security tokens), *reliable messaging* (e.g., WS-ReliableMessaging which describes a protocol that allows SOAP messages to be reliably delivered in the presence of failures) and *resource specifications* (e.g., WS- Resource which enable and standardise interfaces to give the appearance of statefulness).

With respect to the design and implementation of web services two major approaches exist: *bottom up* (implementation-first or contract-last) and *top down* (contract-first or WSDL first). The bottom up approach is the easiest to use as all major programming language platforms such as the Java EE platform (Chinnici et al., 2006) and .NET Framework (Bahree et al., 2007), offer constructs (annotations and directives) which automate the development and can generate web service contracts automatically. While this approach is undoubtedly the simplest and allows for massive production of web services from existing code, it contradicts some of the guiding principles for the development of a service-oriented solution. For example, the reusability and service contract principles are seriously compromised as the service's contracts can change on every modification of the parameters of the underlying functions represented by the service. Performance and versioning of services are also affected.

On the other hand, the top-down or contract-first approach does not rely on automated web service contract generation, but rather on automated program code stubs generation. This approach requires more effort, but is consistent with the principles of service-oriented design and leads to cleanly designed and reusable web services. The contract-first service design process can be divided into five steps of which only the last two are language and platform specific: (1) conceptual design of new services, (2) implementation of the data contract using XML schema language (typically XSD), (3) implementation of the service contract, typically expressed as a WSDL file, (4) code stubs generation using automated tools, and (5) implementation of the actual code using the generated stubs as a skeleton.

1.3 Data mining plaftorms and construction of scientific workflows

Data mining software has evolved from implementations of individual machine learning algorithms (e.g., the C4.5 algorithm (Quinlan, 1993)) to complex software systems which are often integrated into even bigger enterprise software solutions. For example, Oracle RDBMS (Microsoft Corporation, 2008) and Microsoft SQL Server (Mozes, 2011) offer selected data preprocessing and selection tools, data transformation methods, data mining algorithms, and visualisation components, which are optimised for large data processing. The rapid development of data mining platforms in the last two decades was initiated also with the advent of platform independent programming language environments, most notably the Java platform, and high-level scripting languages, such as Python. However, widespread popularity among data analysts as well as non-experts can be mostly contributed to the development of user interfaces, especially to the visual programming using interactive workflows.

Visual programming (Burnett, 2001) is an approach to programming where a procedure or a program is constructed by arranging program elements graphically instead of writing the program as text. The capabilities of general purpose visual programming languages and systems (Johnston et al., 2004) far exceed the capabilities (and needs) of modern data mining software platforms where the visual programming features are limited to the execution of a series of computational and data manipulation steps and various interactive visualisations, and more advanced programming concepts are not required. It should also be noted that scientific

workflows, which are in the focus of our interest, differ from traditional business process workflows. Scientific workflows are simpler than business process workflows as they are focused to a very specific scientific application, and are typically not executed using complex orchestration languages such as WS-BPEL (Alves et al., 2007).

Scientific workflows play an important role in the emerging e-Science technologies where the goal is to enable scientists to collaborate and conduct scientific experiments and discover knowledge using distributed resources (computing services, data and devices). A prominent branch of scientific workflows are life science workflows, especially systems biology workflows which are the focus of rapid development and have achieved widespread use through systems biology workflow management systems. These workflow management tools rely on growing repositories of publicly accessible web services implementing different, highly specific functions and providing access to large databases. With respect to web services, the latest efforts are focused on organising, documenting, annotating, and monitoring. A prominent example is the Biocatalogue[1] project (Bhagat et al., 2010), a curated catalogue (registry) of Life Science Web Services. It provides an open platform for service registration, annotation and monitoring which offers access to 2,345 web services from 174 providers (as of December 2012). These services can be orchestrated from software platforms, e.g., Taverna (Hull et al., 2006), a well-known suite of tools used to design and execute scientific workflows, or Orange4WS (Podpečan et al., 2012), the platform developed in the scope of this thesis, which is presented in detail in Chapter 2. Taverna offers not only a workflow execution platform, but also in-built connectivity with *myExperiment*[2] (De Roure et al., 2009; Goble et al., 2010), a social web site for sharing research subjects, especially life science scientific workflows. It should be noted, however, that platforms, such as Taverna, Triana (Majithia et al., 2004) or Kepler (Altintas et al., 2004), while implementing extensive support for web services and various middleware architectures, do not offer data mining and machine learning algorithms, but are focused on domains such as life sciences and image and signal processing.

Web service technologies, principles of service-oriented architecture and semantic web which provides the key technologies needed to ensure interoperability of such semantically-aware services, have only recently penetrated the data mining software development communities. For example, the recent work by de Bruin (2010) investigated how these principles can improve the knowledge discovery process with respect to performance and ease of design and use, and how service orientation can speed up the creation and execution of knowledge discovery experiments. The FAEHIM (Ali et al., 2005) project focused on integrating algorithm implementations of the well-known Weka data mining toolkit (Hall et al., 2009) as web services into the Triana workflow environment while Weka4WS (Talia et al., 2008) extends Weka to support remote execution of the data mining algorithms through WSRF[3] (Banks, 2006) web services. Other well known data mining workflow environments, such as KNIME (Berthold et al., 2009) or RapidMiner (Mierswa et al., 2006), have only recently implemented support for web services.

The problem of semantic web service composition has been addressed by several authors in the last decade because in many business domains, the composition of web services is required to deliver new functionalities. Several different composition techniques have been proposed some of which also address quality of service (QoS) requirements such as response time, price, availability, and security. In general, the creation of composite web services can be categorised into one of the two major groups, workflow composition and AI planning (Rao and Su, 2005). Workflow

[1] `http://www.biocatalogue.org/`

[2] `http://www.myexperiment.org/`

[3] WSRF is a family of technical specifications concerned with modelling and accessing stateful resources using Web services.

compositions can be further divided into static and dynamic according to the availability of the abstract process model. On the other hand, composition using AI planning is based on the assumption that a web service can be specified by its preconditions and effects in the planning context (the preconditions and effects are the input and the output parameters of the service). A detailed overview of the related work on web service composition is presented in (Charif and Sabouret, 2006; Rao and Su, 2005; Zeshan and Mohamad, 2011). The publication included in Chapter 2 presents the most relevant approaches to automated workflow construction while the data mining and knowledge discovery domain formalisation is discussed in great details in (Panov, 2012). Recent work on the formalisation of data mining stem from the proposals for the unification of the field (Džeroski, 2006) which address some of the issues of the next generation data mining presented in Section 1.1. For example, the proposed ontologies OntoDM and OntoDT (Panov, 2012; Panov et al., 2009) provide definitions of basic data mining entities, but also allow for the definition of more complex entities and the possibility to represent arbitrary complex data types. The work by Žáková et al. (2011), which served as a basis for the integration of a knowledge discovery ontology into Orange4WS, addresses the automated construction of knowledge discovery workflows by developing a knowledge discovery ontology. The ontology is used for the formal conceptualisation of knowledge types and algorithms but also to formalise the task of workflow composition.

Validation, repeatability and availability of experimental results and data are addressed by Vanschoren et al. (2012) and Vanschoren and Blockeel (2009) by proposing and implementing[1] an open experiment database which promotes a collaborative approach to experimentation where data mining experiments are being stored and organised in order to be available for reuse, validation, and comparison.

Up to date, the most advanced data mining software platform is the virtual laboratory software for interdisciplinary collaborative research in data mining and data-intensive sciences, developed in the context of EU-FP7 Collaborative Project e-LICO[2]. The e-LICO software integrates both Taverna and RapidMiner, and provides an Intelligent Discovery Assistant (IDA). IDA integrates Data Mining Workflow Ontology (DMWF) which models data mining operators, goals, tasks and workflows. DMWF is used by the integrated planner to allow for automated construction of data mining workflows for a given task. IDA also integrates meta-mining supported by Data Mining Optimisation (DMOP) Ontology to rank the automatically generated workflows and provides an API interface to data mining software where it can be used as a plugin (e.g., RapidMiner and Taverna).

Finally, the latest efforts in data mining software platform development are focused on software and platform independent solutions which are based on principles of service-oriented architecture, web services, cloud computing, and data stream processing. An example of the next generation data mining platform is ClowdFlows (Kranjc et al., 2012a,b), a cloud-based data mining workflow environment where the (cloud) client application runs in a web browser, but the application software is provided as Software as a service (SaaS).

1.4 Hypothesis and goals

This thesis is concerned with developing a modern software platform, based on principles and practices of service-oriented software development, scientific workflows and visual programming. Using the developed software tools, we aim to construct and implement novel knowledge discov-

[1] `http://expdb.cs.kuleuven.be`

[2] `http://www.e-lico.eu/`

ery scenarios.

The main research hypothesis of this thesis is that new generation knowledge discovery methodologies and environments, which utilise principles of service-oriented software development, can effectively handle the ever increasing amounts and complexity of heterogeneous data and information sources, distributed software and hardware resources, and semantic annotations, which existing knowledge discovery environments and tools are unable to deal with. Given a pool of distributed components, data and knowledge resources, semantic annotations and reasoning algorithms, such environments should be able to provide novel knowledge discovery scenarios which were not possible or not considered until now. Moreover, by utilising the latest developments in knowledge discovery and data mining ontologies and planning, such new generation environments should simplify the knowledge discovery processes by making them available to non-experts and scientists from very different domains of research, e.g., biologists, natural language processing experts, medical experts, etc.

This thesis addresses the research goals related to the development of a novel knowledge discovery platform, as well as the goals related to the developed knowledge discovery scenarios.

The goals related to the development of the Orange4WS platform (Orange for web services), presented in Chapter 2 were as follows.

1. Development of an open-source knowledge discovery platform utilising principles of service-oriented computing.

2. Transparent integration of web services as workflow components.

3. Implementation of software tools for the development of new services and for converting legacy code into web services, which encourages good software development practises for designing web services (e.g., WSDL-first design (Erl, 2005)),

4. Integration of a knowledge discovery ontology to describe workflow components (data, knowledge and services) in an abstract and machine-interpretable way, and its use by a planner that enables automated composition of data mining workflows.

5. Development of novel knowledge scenarios which demonstrate the usefulness of the presented knowledge discovery methodology and the practical capabilities of the implemented reference platform.

The research goals related to the development of the SegMine methodology for semantic analysis of microarray data, presented in detail in Chapter 3, were as follows.

1. Development of the SegMine methodology, which helps biologists in interpreting microarray data by finding groups of genes characterised with some semantic descriptions, and by discovering links between them, which may lead to the formation of new hypotheses, based on the experimental data and biological knowledge available in public databases.

2. Integration of the Biomine system and the SEGS algorithm in a service-oriented workflow environment.

3. A publicly available implementation of the SegMine methodology as interactive workflow components in the Orange4WS environment.

4. Validation of the SegMine methodology by comparative analysis of the experimental results on a well-known dataset from a clinical trial in acute lymphoblastic leukemia (ALL).

5. Experimental evaluation of the Segmine methodology on data on senescence in human mesenchymal stem cells (MSC).

Finally, the research goals related to the development of contrasting subgroup discovery (CSD) methodology and its implementation were as follows.

1. Development of theoretical foundations for contrasting subgroup discovery by proposing a multi-step process combining subgroup discovery and definition of contrast sets.
2. Proposing appropriate set theoretic functions to define contrasts on subgroups.
3. Implementation of a web service providing the SEGS algorithm using the GoMapMan gene ontology (Baebler et al., 2010) which is an extension of the plant ontology MapMan (Thimm et al., 2004).
4. Development and implementation of the contrasting subgroup discovery methodology in Orange4WS using its workflow and service-oriented capabilities.
5. Evaluation of the methodology on experimental time labelled gene expression data.

1.5 Scientific contributions

This thesis contributes to the fields of knowledge discovery, data mining, and systems biology. The contributions related to the developed Orange4WS platform include:

- a novel service-oriented knowledge discovery framework and its implementation in the service-oriented data mining environment Orange4WS, and
- publicly available implementation of the Orange4WS platform.

The work related to the design and development of the Orange4WS platform was published in the following publications:

Podpečan, V.; Juršič, M.; Žáková, M.; Lavrač, N. Towards a service-oriented knowledge discovery platform. In: Podpečan, V.; Lavrač, N.; Kok, J. N.; de Bruin, J. (eds.) *Proceedings of the 2nd workshop on service-oriented knowledge discovery (SoKD'09)*. 25–38 (Unpublished proceedings, 2009). `http://www.ecmlpkdd2009.net/wp-content/uploads/2008/09/service-oriented-knowledge-discovery_2.pdf` (Accessed: January, 2013).

Podpečan, V.; Žáková, M.; Lavrač, N. Workflow Construction for Service-Oriented Knowledge Discovery. In: Margaria, T.; Steffen, B. (eds.) *Proceedings of the 4th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2010), Part I*. **6415**, 313–327 (Springer, Berlin, 2010).

Podpečan, V.; Zemenova, M.; Lavrač, N. Orange4WS environment for service-oriented data mining. *The Computer Journal* **55**, 82–98 (2012).

Žáková, M.; Železný, F.; Podpečan, V.; Lavrač, N. Advancing data mining workflow construction: A framework and cases using the Orange toolkit. In: Podpečan, V.; Lavrač, N.; Kok, J. N.; de Bruin, J. (eds.) *Proceedings of the 2nd workshop on service-oriented knowledge discovery (SoKD'09)*. 39–51 (Unpublished proceedings, 2009). `http://www.ecmlpkdd2009.net/wp-content/uploads/2008/09/service-oriented-knowledge-discovery_2.pdf` (Accessed: January, 2013).

The contributions related to the SegMine methodology, its evaluation and implementation are as follows:

- development of the SegMine methodology for semantic analysis of gene expression data which offers improved hypothesis generation and data interpretation for biologists,
- three novel research hypotheses that improve understanding of the underlying mechanisms in senescence of human mesenchymal stem cells and identification of candidate marker genes, and
- publicly available implementation of the SegMine methodology in the form of interactive workflow components for Orange4WS.

The most important contributions related to the SegMine methodology were published in the following publications:

Lavrač, N.; Kralj Novak, P.; Mozetič, I.; Podpečan, V.; Motaln, H.; Petek, M.; Gruden, K. Semantic subgroup discovery: using ontologies in microarray data analysis. In: He, B.; Pan, X.; Kim, Y.; Worrell, G. (eds.) *Engineering the future of biomedicine: Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 5613–5616 (Institute of Electrical and Electronics Engineers, New York City, New York, 2009a).

Lavrač, N.; Mozetič, I.; Podpečan, V.; Kralj Novak, P.; Motaln, H.; Petek, M. Gene analytics: discovery and contextualization of enriched gene sets. In: Nürnberger, A.; Berthold, M. R.; Kötter, T.; Thiel, K. (eds.) *Proceedings of the Workshop on Explorative Analytics of Information Networks*. 39–49 (Unpublished proceedings, 2009b). `http://www.ecmlpkdd2009.net/wp-content/uploads/2009/08/explorative-analytics-of-information-networks.pdf` (Accessed: January, 2013).

Mozetič, I.; Lavrač, N.; Podpečan, V.; Kralj Novak, P.; Motain, H.; Petek, M.; Gruden, K.; Toivonen, H.; Kulovesi, K. Bisociative Knowledge Discovery for Microarray Data Analysis. In: Ventura, D.; Pease, A.; Pérez, R.; Ritchie, G.; Veale, T. (eds.) *Proceedings of the International Conference on Computational Creativity (ICCC-X)*. 190–199 (Department of Informatics Engineering, University of Coimbra, Lisbon, Portugal, 2010).

Mozetič, I.; Lavrač, N.; Podpečan, V.; Kralj Novak, P.; Motaln, H.; Petek, M.; Toivonen, H.; Kulovesi, K. Semantic subgroup discovery and cross-context linking for microarray data analysis. In: Berthold, M. R. (ed.) *Bisociative Knowledge Discovery: An Introduction to Concept, Algorithms, Tools, and Applications*. 379–389 (Springer, 2012).

Podpečan, V.; Lavrač, N.; Mozetič, I.; Kralj Novak, P.; Trajkovski, I.; Langohr, L.; Kulovesi, K.; Toivonen, H.; Petek, M.; Motaln, H.; Gruden, K. SegMine workflows for semantic microarray data analysis in Orange4WS. *BMC Bioinformatics* **12**, 416 (2011).

Finally, the contributions related to the contrasting subgroup discovery methodology are as follows:

- development of a novel methodology for contrasting subgroup discovery and its evaluation on the Solanum tuberosum time labelled gene expression data, and

- publicly available implementation of the methodology in the form of interactive workflow components for Orange4WS.

The contrasting subgroup methodology and related work were published in the following publications:

Langohr, L.; Podpečan, V.; Mozetič, I.; Petek, M.; Gruden, K. Contrast mining from interesting subgroups. In: Berthold, M. R. (ed.) *Bisociative Knowledge Discovery: An Introduction to Concept, Algorithms, Tools, and Applications*. 390–406 (Springer, Berlin, 2012a).

Langohr, L.; Podpečan, V.; Petek, M.; Mozetič, I.; Gruden, K.; Lavrač, N.; Toivonen, H. Contrasting subgroup discovery. *The Computer Journal* (2012b). In press.

1.6 Dissemination of the developed software, workflows, and results

The implementation of the Orange4WS platform, as well as implementations of the SegMine methodology and contrasting subgroup discovery, were released as open source software and are available to general public. Home pages of Orange4WS[1] and SegMine[2] provide the sources as well as platform-dependent installers and all required software dependencies. Contrasting subgroup discovery is not provided as a separate software package as the implemented workflow components are provided in Orange4WS and SegMine (contrasting subgroup discovery and SegMine share some of the workflow components).

The supplementary material for the SegMine methodology also includes video tutorials on composing and executing Orange4WS workflows as well as downloadable Orange4WS workflows ready to be loaded into Orange4WS and executed on provided data. Two video tutorials provide step-by-step analyses of two data sets, the data set on acute lymphoblastic leukemia[3] (Chiaretti et al., 2004) and the data set on human mesenchymal stem cells[4] (Wagner et al., 2008).

1.7 Organisation of the thesis

The rest of the thesis is structured as follows. Chapter 2 presents Orange4WS, the developed knowledge discovery platform which extends an existing data mining toolkit (Orange) and utilises principles of service-oriented architecture. The main part of this section is the journal publication (Podpečan et al., 2012) which presents the related work, the platform itself, automated workflow composition, as well as three use cases.

The next two chapters present in detail two knowledge discovery methodologies, which were developed in the Orange4WS plaftorm. In Chapter 3, the SegMine methodology for semantic analysis of gene expression data is presented in detail in a journal publication (Podpečan et al., 2011), which introduces the related work, the methodology, and two applications of the methodology. In the first application a comparative analysis of SegMine and DAVID (Huang et al., 2009a,b) on a well-known dataset from a clinical trial in acute lymphoblastic leukemia (ALL) is discussed, while the second evaluates the SegMine methodology on a dataset about senescence

[1] `http://orange4ws.ijs.si/`
[2] `http://segmine.ijs.si/`
[3] `http://segmine.ijs.si/media/video/SegMine-ALL_dataset.mp4`
[4] `http://segmine.ijs.si/media/video/SegMine-Wagner_dataset.mp4`

in human mesenchymal stem cells (MSC). The publication concludes with a description of the reference implementation of SegMine in Orange4WS.

In Chapter 4, the journal publication (Langohr et al., 2012) on contrasting subgroup discovery (CSD) methodology is presented. The definition of the problem is given along with a more general introduction to subgroup discovery, contrast set mining and the related approaches. The CSD methodology is then explained on an illustrative example, followed by an application in biology where a time labelled Solanum tuberosum (potato) gene expression data set is analysed using the proposed methodology.

Chapter 5 concludes the thesis and summarises the presented work while pointing out possible directions for further research.

Finally, appendices A, B, C, and D provide the following additional material. Appendix A documents the most important features of Orange4WS and discusses the availability of the software. Appendix B presents our implementation of the SegMine methodology and describes all of its Orange4WS workflow components. Appendix C presents our implementation of Contrasting subgroup discovery and provides descriptions of its Orange4WS workflow components. Finally, Appendix D lists the results of the SEGS algorithm on the dataset where gene expression profiles from late senescent passages of MSC from three independent donors were compared to the MSC of early passages. These experimental results served as a basis for expert data analysis with SegMine, which resulted in the construction of three novel scientific research hypotheses.

2 The Orange4WS Platform

This chapter presents the developed service-oriented knowledge discovery platform Orange4WS. The platform is based on Orange (Demšar et al., 2004), a user-friendly data mining toolkit which is implemented in a three-layer architecture:

1. efficient C++ implementations of data structures, algorithms and procedures (Orange core),

2. extensive collection of wrappers, algorithms, procedures and visualisations in the Python language, and

3. Orange Canvas, which provides a user-friendly workflow composition environment.

Orange4WS extends and upgrades Orange in multiple ways. In the following, we list the most important Orange4WS features and extensions which are then discussed in detail in the publication included in this chapter.

- Orange4WS provides new tools enabling the development of SOAP web services which support the contract-first (also known as WSDL first) development process (Erl, 2005). These tools allow for the development of simple, stateless services, as well as more complex job processing services which maintain state and past results.

- Orange4WS upgrades the Orange Canvas by providing fully automatic integration of web services as workflow components. This includes automated widget code construction and messaging, and a number of local Orange4WS widgets which enable web service integration such as data transformation, data serialisation and deserialisation etc. Simple, unconditional looping is also supported through the *Emitor* and *Collector* widgets that enable processing of data sequences by emitting unprocessed data and collecting the results.

- Orange4WS integrates the knowledge discovery ontology (KD ontology), the core part of which currently contains more than 150 concepts and 500 instances.

- Orange4WS enables automated workflow construction by employing a planner which uses the KD ontology.

A screenshot of the Orange4WS platform running a natural language processing workflow is shown in Figure 2.1. The platform and means for automated workflow construction are presented in the following publication:

Podpečan, V.; Zemenova, M.; Lavrač, N. Orange4WS environment for service-oriented data mining. *The Computer Journal* **55**, 82–98 (2012)

In addition, Appendix A presents the most relevant technical details of the Orange4WS platform, including software availability and user's manual, which outlines the most common

procedures supported by Orange4WS (e.g., importing a new web service, developing a new widget etc.).

The author's contributions are as follows. Orange4WS was designed and implemented by Vid Podpečan. Monika Zemenova developed a knowledge discovery ontology and a system for automatic construction of data mining workflows using an existing fast-forward planning algorithm. Nada Lavrač contributed the idea of developing a service-oriented data mining environment, and contributed to the scientific workflows implemented in Orange4WS.

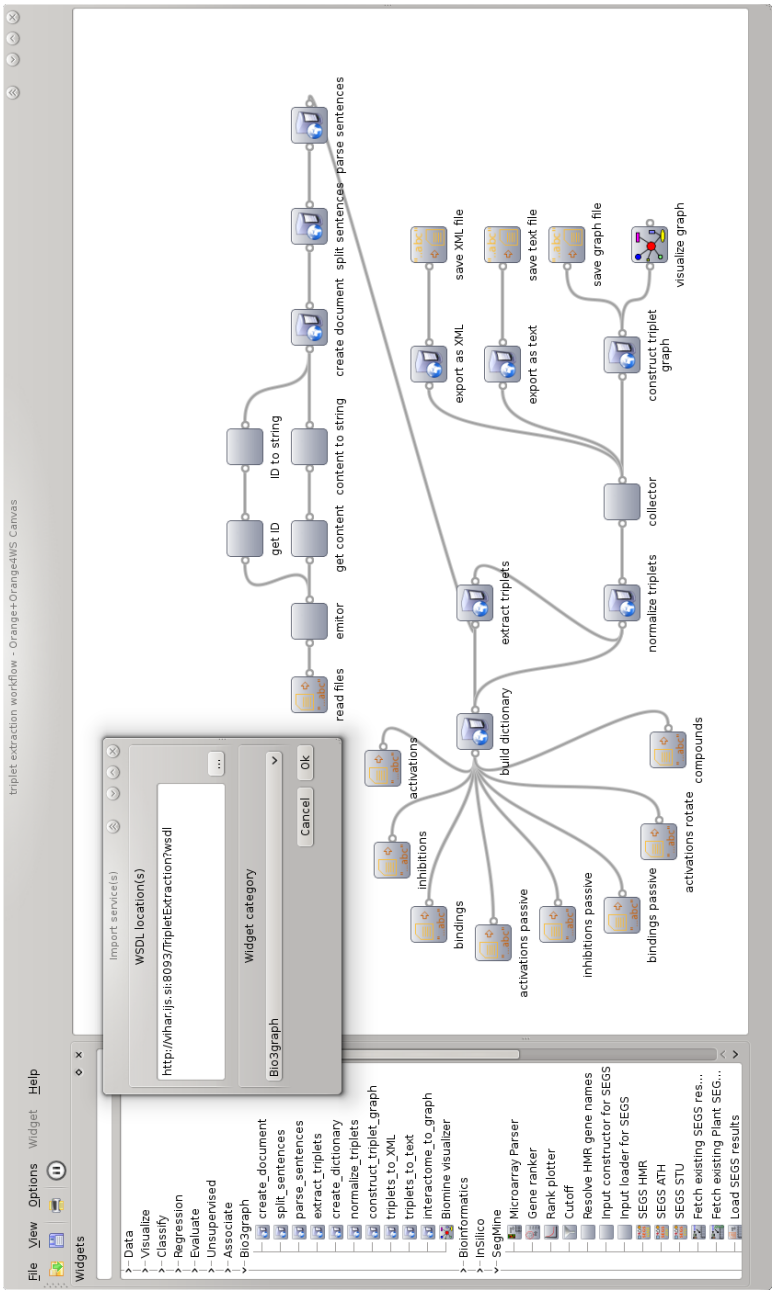

Figure 2.1: A screenshot of the Orange4WS platform running a natural language processing workflow for the extraction of triplets from biological literature. A sub-window for importing new web services into Orange4WS is also shown.

The Computer Journal Advance Access published August 7, 2011

© The Author 2011. Published by Oxford University Press on behalf of The British Computer Society. All rights reserved.
For Permissions, please email: journals.permissions@oup.com
doi:10.1093/comjnl/bxr077

Orange4WS Environment for Service-Oriented Data Mining

VID PODPEČAN[1], MONIKA ZEMENOVA[2] AND NADA LAVRAČ[1]

[1]*Jožef Stefan Institute, Ljubljana, Slovenia*
[2]*IZIP Inc., Prague, Czech Republic*
[*]*Corresponding author: vid.podpecan@ijs.si*

Novel data-mining tasks in e-science involve mining of distributed, highly heterogeneous data and knowledge sources. However, standard data mining platforms, such as Weka and Orange, involve only their own data mining algorithms in the process of knowledge discovery from local data sources. In contrast, next generation data mining technologies should enable processing of distributed data sources, the use of data mining algorithms implemented as web services, as well as the use of formal descriptions of data sources and knowledge discovery tools in the form of ontologies, enabling automated composition of complex knowledge discovery workflows for a given data mining task. This paper proposes a novel Service-oriented Knowledge Discovery framework and its implementation in a service-oriented data mining environment Orange4WS (Orange for Web Services), based on the existing Orange data mining toolbox and its visual programming environment, which enables manual composition of data mining workflows. The new service-oriented data mining environment Orange4WS includes the following new features: simple use of web services as remote components that can be included into a data mining workflow; simple incorporation of relational data mining algorithms; a knowledge discovery ontology to describe workflow components (data, knowledge and data mining services) in an abstract and machine-interpretable way, and its use by a planner that enables automated composition of data mining workflows. These new features are showcased in three real-world scenarios.

Keywords: data mining; knowledge discovery; knowledge discovery ontology; e-science workflows; automated planning of data mining workflows

Received 20 December 2010; revised 30 May 2011
Handling editor: Yannis Manolopoulos

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

1. INTRODUCTION

Fast-growing volumes of complex and geographically dispersed information and knowledge sources publicly available on the web present new opportunities and challenges for knowledge discovery systems. Principled fusion and mining of distributed, highly heterogeneous data and knowledge sources requires the interplay of diverse data processing and mining algorithms, resulting in elaborate data mining workflows. If such data mining workflows were built on top of a service-oriented architecture, the processing of workflow components (e.g. data mining algorithms) can be distributed between the user's computer and remote computer systems. Therefore, as the use of data mining algorithms (implemented as services) is no longer limited to any particular data mining environment, platform or scenario, this can greatly expand the domains where data mining and knowledge discovery algorithms can be employed. As an example, state-of-the-art data mining and knowledge discovery methods can become widely available in bioinformatics, business informatics, medical informatics and other research areas. Moreover, existing domain-specific services can become seamlessly integrated into service-oriented data mining environments.

There is another important aspect that makes data mining difficult for non-expert users. While the mutual relations of specialized algorithms used in the workflows and principles of their applicability are easily mastered by computer scientists, this cannot be expected from all end-users, e.g. life scientists. A formal capture of the knowledge of data mining tasks, and input–output characteristics of data mining algorithms is thus needed, which can be captured in the form of ontologies of relevant services and knowledge/data types, to serve as a basis for intelligent computational support in

knowledge discovery workflow composition. A formal capture of knowledge discovery tasks can then be used to improve repeatability of experiments and to enable reasoning on the results to facilitate their reuse.

This paper proposes a novel Service-oriented Knowledge Discovery (SoKD) framework, and its implementation that address the challenges discussed earlier. Building such a framework has been recognized as an important aspect of third-generation data mining [1]. A practical implementation of the proposed third-generation knowledge discovery platform, named Orange4WS (Orange for Web Services), has been conceived as an extension of the existing data mining platform Orange [2].

The third-generation data mining paradigm shift implies the need for a substantially different knowledge discovery platform, aimed at supporting human experts in scientific discovery tasks. In comparison with the current publicly available data mining platforms (best known examples being Weka [3], KNIME [4], RapidMiner [5] and Orange [2]), the Orange4WS platform provides the following new functionalities: (a) user-friendly composition of data mining workflows from local and distributed data processing/mining algorithms applied to a combination of local and distributed data/knowledge sources, (b) simplified creation of new web services from existing data processing/mining algorithms, (c) a knowledge discovery ontology of knowledge types, data mining algorithms and tasks and (d) automated construction of data mining workflows based on the specification of data mining tasks, using the data mining ontology through an algorithm that combines planning and ontological reasoning. This functionality is based on—and extends—a rich collection of data processing and mining components as well as data and information sources provided by local processing components as well as remote web services.

While each individual extension of the existing data mining technologies is not scientifically ground-breaking, the developed Orange4WS environment as a whole is a radically new data mining environment from many perspectives. From the machine learning and data mining perspective, the uniqueness of this platform is in the incorporation of propositional data mining as well as relational data mining algorithms (implemented in Prolog) in a unique data mining framework. On the other hand, from the Artificial Intelligence perspective, a unique feature of the proposed SoKD framework is the use of the developed knowledge discovery ontology of data types and data mining algorithms for automated data mining workflow construction using a fast-forward planning algorithm. From the e-Science perspective, Orange4WS substantially improves the existing environments that support manual construction of scientific workflows (such as Taverna [6] and Triana [7]) by incorporating advanced propositional and relational data mining algorithms as well as by supporting automated workflow construction. Finally, from the web services perspective, simplified creation of new web services from existing data processing/mining algorithms is a valuable extension of existing web-service-based environments. In the presented work, some of these unique features of Orange4WS are show-cased in three complex data mining scenarios, presented in Section 6.

The paper is structured as follows. Section 2 presents a motivating use case for developing and using a service-oriented knowledge discovery platform. Section 3 presents our approach to developing a novel SoKD framework and its implementation that upgrades the existing data mining system Orange into a new SoKD platform Orange4WS.[1] Sections 4 and 5 upgrade the implemented solution by introducing a knowledge discovery ontology of annotated types of data and knowledge resources, data mining algorithms and data mining tasks, and a facility for automated data mining workflow planning based on these annotations. Section 6 presents three use cases illustrating the advantages of the new platform. The Weka use case in Section 6.1 demonstrates that Weka algorithms can easily be integrated as services into the Orange4WS platform. The relational data mining use case in Section 6.2 shows how to combine propositional and relational data preprocessing and mining algorithms in a single environment. Section 6.3 illustrates a complex systems biology use case, which combines (a) a complex relational subgroup discovery system SEGS that uses biological ontologies and background knowledge for learning, and (b) a complex reasoning and visualization environment Biomine that includes data from numerous biological databases. Section 7 presents the related work. Section 8 concludes with a summary and plans for further work.

2. A SAMPLE TEXT MINING USE CASE

This section presents a motivating use case for developing and using a service-oriented knowledge discovery platform, including a user-friendly workflow editor. The use case is built upon text mining web services, available from LATINO[2] text mining library, which provides a range of data mining and machine learning algorithms, with the emphasis on text mining, link analysis and data visualization.

The goal of this use case is to produce a compact and understandable graph of terms, which could potentially give insights into relations between biological, medical and chemical terms, relevant to the subject of a user-defined query. A manually constructed Orange4WS workflow of processing components is shown in Fig. 1.

The use case demonstrates the need for a service-oriented platform able to combine publicly available data repositories (PubMed) with third-party data analysis tools (LATINO), specialized algorithms (Pathfinder) and powerful local visualization components (Orange graph visualizer).

[1] The Orange4WS software environment is available under the GPL licence at http://orange4ws.ijs.si.

[2] http://sourceforge.net/projects/latino.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

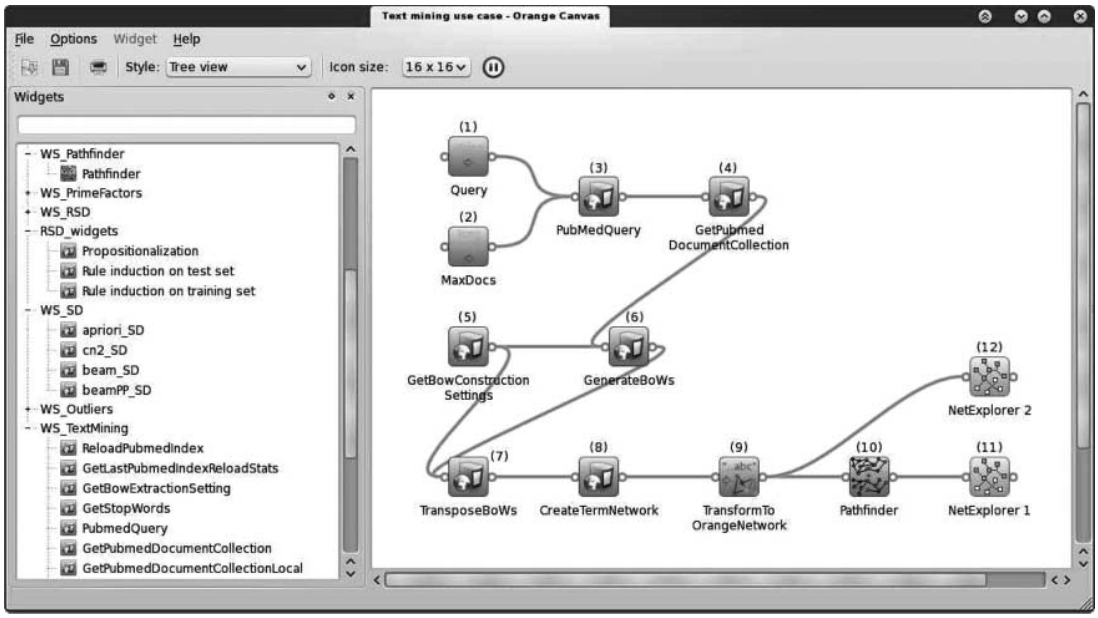

FIGURE 1. An Orange4WS workflow of text mining services in the Orange workflow execution environment. Components numbered 3, 4, 5, 6, 7, 8 and 10 are web services; components 1, 2 and 9 are Orange4WS supporting widgets; components 11 and 12 are instances of the native Orange graph visualizer.

PubMed search web services is queried with a user-defined query string and a parameter defining the maximal number of documents returned (components 1, 2 and 3). It returns a collection of IDs of relevant documents. Then, the obtained IDs are used to collect titles, abstracts and keyword of these documents (component 4). Next, bag-of-words (BoW) sparse vectors are created from the collection of words (component 6). To simplify the setting of parameters for unexperienced users, there is a service providing a suitable set of default values that can be used as an input to the web service that constructs BoW vectors (component 5). BoW vectors are then transposed (component 7) and a network of words/terms is created (component 8) in the .net format of the well-known Pajek social network analysis tool.[3] The resulting graph of terms in the .net format is then transformed into Orange's native data structure for representing graphs (component 9), and simplified using a sparse variant of the Pathfinder algorithm that is implemented as a web service (component 10). Finally, the original and pruned graph are visualized using the Orange's native `Network explorer` (components 11 and 12).

This Orange4WS workflow, implementing a complex text mining scenario, was designed and constructed manually in the Orange's user-friendly workflow editor. In Section 5, we will demonstrate how this workflow can be constructed automatically using a workflow planner and an ontology, which provides information about workflow operators and their input and output knowledge types.

3. THE ORANGE4WS PLATFORM

This section briefly describes the structure and design of the proposed software platform. We explain and comment our decisions concerning the selection of technologies and software tools used. The main part of this section describes the design of the Orange4WS platform and the accompanying toolkit for producing new web services.

3.1. Technological background

Our goal was to develop a simple, user-friendly software platform that is able to seamlessly integrate web services and local components in terms of workflow composition, originating from different communities (propositional data mining, relational data mining, text mining, systems biology, etc.), including also a knowledge discovery ontology to support the automatization of workflow construction.

The proposed software platform, named Orange4WS, is built on top of two open-source scientific-community-driven projects: (a) the Orange data mining framework [2] that provides a range of preprocessing, modeling and data exploration techniques and a user-friendly workflow execution environment, and (b) the Python Web Services project[4] (more

[3] User manual of the Pajek software tool for the analysis and visualization of large social networks is available at http://pajek.imfm.si/doku.php.

[4] http://pywebsvcs.sourceforge.net/.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

specifically, the Zolera SOAP infrastructure) that provides libraries for the employment and development of web services using the Python programming language by implementing various protocols, including SOAP, WSDL, etc.

In contrast with other freely available workflow environments such as Weka, Taverna, Triana, KNIME, RapidMiner, etc., the Orange4WS framework offers a rather unique combination of features: (a) a large collection of data mining and machine learning algorithms, efficiently implemented in C++ (Orange core); (b) a three-layer architecture: C++, Python, as well as Orange and Orange4WS Widgets; (c) a collection of very powerful yet easy to use data visualization widgets; (d) incorporation of propositional as well as selected relational data mining algorithms, (e) simplicity of workflow composition in the Orange canvas and (f) automated workflow construction using a knowledge discovery ontology and a planner. Moreover, by using an interpreted high-level programming language (Python), it is possible to avoid the compile-test-recompile development cycle. Also, high-level interpreted languages are a perfect choice for rapid software development using emerging web technologies such as RESTful web services[5] or WEB APIs.[6]

3.2. Platform design

Apart from the Orange core in C++ and its interface to the Python programming language, the Orange framework enables *visual programming* achieved by graphically composing processing components into *workflows*. Workflows are—essentially—executable visual representations of complex procedures. They enable repeatability of experiments as they can be saved and reused. Moreover, workflows make the framework suitable also for non-experts due to the representation of complex procedures as sequences of simple steps.

Workflow construction in Orange is supported by the *Orange Canvas*, an interactive graphical user interface component. It enables graphical construction of workflows by allowing workflow elements called *Orange Widgets* to be positioned in a desired order, connected with lines representing flow of data, adjusted by setting their parameters and finally executed.

An Orange Widget is defined by its inputs, outputs and the graphical user interface. Inputs and outputs are defined by the so-called typed channels, which specify the name of the channel, multiplicity (inputs only), data type, and a handler function (inputs only), which is invoked when the input data are available. For example, one of the most common inputs (outputs) is the Orange `ExampleTable`, a data structure used to store tabular and/or sparse data.

[5]A RESTful web service is a simple web service implemented using HTTP and the principles of REST [8].

[6]A Web API is a defined set of HTTP request messages along with a definition of the structure of response messages, most commonly expressed in JSON or XML.

Orange4WS extends and upgrades Orange on three levels. First, it provides tools that ease the employment of web services from the Python interpreter. Second, it upgrades the Orange Canvas with the ability to use web services as workflow components. Note that this level also provides a number of local Orange4WS widgets that are required for web service integration such as data transformation, data serialization and deserialization etc. Third, it enables automatic workflow construction by integrating a knowledge discovery ontology and a planner.

The functionality of Orange4WS is provided by several components (modules). The most important modules are: web service widget code generator, web service types extractor, web services stubs importer and the subsystem for automated workflow construction. The latter offers a number of supporting modules and functions as well as a general knowledge discovery ontology (KD ontology) that enables automated workflow planning. A high-level overview of the design of Orange4WS showing the main components and their interaction is shown in Fig. 2. The structure of the subsystem for automated workflow planning is discussed in more details in Section 5.

The `Web service stubs importer` module provides the base functionality that is required by the majority of other components. It dynamically loads web service consumer classes (web service client) generated by the Zolera SOAP infrastructure library using the provided link to the WSDL description of the service. These classes provide a high-level access to all methods provided by a given SOAP web service.

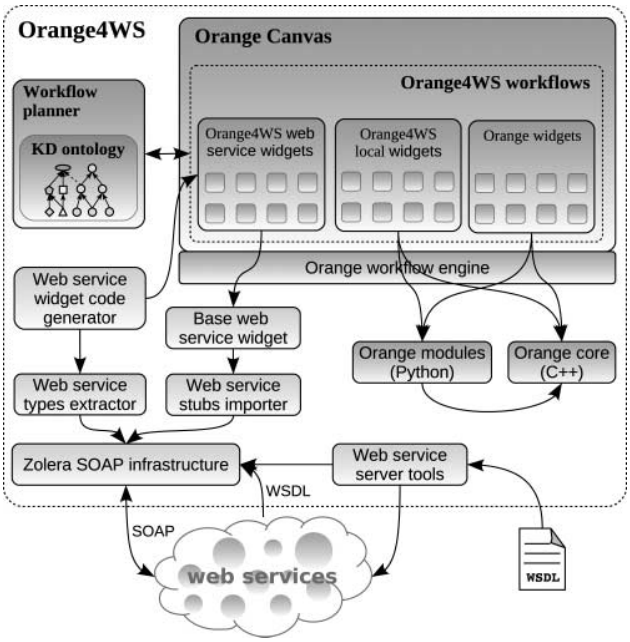

FIGURE 2. The structural design of the Orange4WS platform. A more detailed structure of the workflow planner component is shown in Fig. 8.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

The role of the `Web service types extractor` module is to extract all type information for a given web service client instance, which was imported by the `Web service stubs importer` module. All web service functions and their input and output parameters are analyzed in a recursive manner, and full type as well as multiplicity[7] information is extracted. Simple data types are mapped to equivalents from the Python language, while complex types are mapped to objects, respectively.

The `Web service widget code generator` implements one of the main functionalities of Orange4WS: fully automated creation of widgets from web services. It relies on the modules, described earlier, to import generated web service consumer code and to define web service widget's inputs and outputs according to the extracted types. For a given web service, a widget is generated for each of its methods, and each input (output) parameter of a given method is mapped to one input (output) typed channel. Every web service widget is a subclass of the `BaseWebServiceWidget` class that takes care of the execution of the corresponding method, error detection and reporting, user notification, etc.

Since the main design goals of Orange4WS are simplicity and automatization, all technical details of creating new Orange4WS widgets from web services are summarized as a single user-interface command `import web service`. It invokes the web service widget code generator, which implements all required steps to enable access to the web services through a collection of Orange4WS widgets representing its methods. The details of actual invocation of a given web service method are thus hidden and can be summarized from the user's perspective as a normal widget operation: (1) receiving data, (2) widget internal processing and (3) outputting processed data. Essentially, the Orange Canvas is not aware of a non-local nature of web service widgets. Such simplicity is essential as the platform is intended to be used by scientists from very different domains, including bioinformatics, natural language processing, etc.

3.3. Composition and execution of workflows

One of the most important features of Orange, also inherited by Orange4WS, is an easy-to-use interactive workflow construction in Orange Canvas. Workflows components (widgets) represented with icons can be dragged to the appropriate position on the Canvas, while their inputs and outputs can be connected visually by drawing lines. The `Signal manager`, Orange's main workflow management component, enables or disables the connectivity of inputs and outputs according to their types. It also prevents the user from creating loops while connecting widgets by detecting cycles in the corresponding directed graph. If a widget supports the adjustment of its parameters, this can be done from widget's user interface, which can also enable data and results visualization as well as other interactive features. Finally, a constructed workflow can be saved into an XML format that corresponds to a predefined XML schema. This ensures repeatability of scientific experiments as well as user collaboration.

Orange4WS extends the manual composition of workflows in Orange with the ability to construct workflows automatically. Automated workflow construction is treated as a planning task where available workflow components represent operators while their input and output knowledge types represent preconditions and effects. The `Workflow planner` that is used to discover workflows satisfying the specified knowledge discovery task queries the developed knowledge discovery ontology where the available operators are annotated. The discovered workflows are available in the Orange's XML format, and can be loaded, instantiated and executed in Orange4WS. Section 5 discusses this feature of Orange4WS in details.

Orange's approach to workflow execution differs from the conventional workflow execution engines [9]. Because Orange workflows tend to be simple and as interactive as possible, the execution is provided on per-widget basis. As such, workflow components are treated as standalone steps in interactive analysis of data. Essentially, Orange does not provide a central workflow execution engine. Instead, the decision on how and when a widget is to be executed is left to the widget itself. Widgets are basically GUI wrappers around data analysis and visualization algorithms [2] implemented in Orange (note that Orange4WS extends Orange with web service widgets). In comparison with the Taverna workflow management system [10], this allows for rich and complex workflow components enabling user interaction and visualizations but also decreases the overall complexity of workflows (note that this is a well-known tradeoff between the complexity of workflows and the complexity of their components).

Essentially, there are two types of widgets: *flow-through* widgets and *on-demand* widgets. Flow-through widgets are executed as soon as all required input data are available. On the other hand, on-demand widgets are executed only when the user request their execution (all required input data must also be present). This type of execution is usual in the case of rich and complex widgets that require user interaction prior to the actual execution.

Orange4WS workflows are executed in the same manner as Orange workflows only with the following differences. First, Orange4WS provides components that simulate unconditional looping. The `Emitor` and `Collector` widgets enable processing of data sequences by emitting unprocessed data and collecting the results, respectively. Second, unlike Orange where the majority of widgets are of the on-demand type, all auto-generated Orange4WS web service widgets are flow-through. This corresponds to the base principle of

[7] Parameter multiplicity can be one of the following: required (1..1), optional (0..1), zero or more (0..*), one or more (1..*).

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

service-oriented design according to which a web service should encapsulate only one well-defined functionality that should not require complex user interaction. However, using the supporting modules and tools Orange4WS provides, any kind of web service widget can be developed. For example, an on-demand-type web service widget with progress polling was developed to interact with the computationally complex web service implementing the SEGS algorithm [11] (Section 6.3 discusses this service in more detail). Finally, the actual flow of data in Orange4WS workflows depends on the types of web services. In the case of location unaware web services, the results of the execution are always sent back to the caller (Orange4WS), while in the case of location aware web services,[8] Orange4WS only coordinates the execution while the actual data are not transmitted.

3.4. Creation of new web services

A separate part of our service-oriented knowledge discovery platform, also shown in Fig. 2 as the *Web service server tools* component, is a package of tools that ease the creation of new web services. These tools closely follow the general *WSDL first* design principle [12]. This principle promotes clearly designed, interoperable and reusable services by separating the design of interfaces from the actual logic. Essentially, our tools extend the Python language framework by using the Python Web Services package, enhanced with multiprocessing capabilities, security, logging and other related functionalities. By using these tools, any code can easily be transformed into a SOAP web service and used as an ingredient for Orange4WS workflow composition (or in any other workflow environment capable of using web services). Moreover, the provided tools support the creation of simple request/response stateless services as well as more complex batch (job) processing services, which can be used for time-consuming server-side processing. Such batch processing services also store results which can be retrieved later.

We have successfully created web services for the Relational subgroup discovery algorithm [13] implemented in Prolog. As a result, this relational data mining algorithm is available as a processing component in a propositional workflow-enabled environment. Also, the SEGS algorithm [11], a highly computationally complex rule discovery algorithm that uses biological ontologies as background knowledge, was transformed into a web service that greatly improved its processing capability, availability and also its ontology update mechanisms, which are now automated. Section 6 provides more details on these web services.

4. KNOWLEDGE DISCOVERY ONTOLOGY

To enrich the proposed knowledge discovery platform with semantics, we have developed the *Knowledge Discovery ontology* (the KD ontology, for short). The ontology defines relationships among the components of knowledge discovery scenarios, both declarative (various knowledge representations) and algorithmic. The primary purpose of the KD ontology is to enable the workflow planner to reason about which algorithms can be used to produce the results required by a specified knowledge discovery task and to query the results of knowledge discovery tasks. In addition, the ontology can also be used for automated annotation of manually created workflows facilitating their reuse.

An illustrative part of the top-level structure of the ontology is shown in Fig. 3. The three core concepts are: `Knowledge`, capturing the declarative elements in knowledge discovery; `Algorithm`, which serves to transform knowledge into (another form of) knowledge; `Knowledge discovery task`, which describes a task that the user wants to perform mainly by specifying the available data and knowledge sources and the desired outputs. The ontology is implemented in semantic web language OWL-DL.[9] The primary reasons for this choice were OWL's sufficient expressivity, modularity, availability of ontology authoring tools and optimized reasoners. The core part of the KD ontology currently contains around 150 concepts and 500 instances and is available online.[10] The structure of workflows is described using OWL-S.[11]

In the following sections, we describe `Knowledge` and `Algorithm` concepts in more detail and provide information on the annotation of algorithms available locally in the Orange4WS toolkit and in the LATINO library.

4.1. Knowledge

All the declarative components of the knowledge discovery process such as datasets, constraints, background knowledge, rules, etc. are instances of the `Knowledge` class. In data mining, many knowledge types can be regarded as sets of more elementary pieces of knowledge [14], e.g. first-order logic theories consist of formulas. This structure is accounted for through the property `contains`, so e.g. a first-order theory `contains` a set of first-order formulas.

Moreover, some knowledge types may be categorized according to the expressivity of the language in which they are encoded. For this purpose, we have designed a hierarchy of language expressivity (see Fig. 3, `Expressivity`). We further distinguish knowledge types that play special roles in knowledge discovery, e.g. the `Dataset` class, defined as `Knowledge`, that contains `Examples`. `Expressivity` can also be defined for datasets to distinguish between propositional datasets and relational datasets.

All the other types of knowledge such as pattern sets, models and constraints are clustered under the class `NonLogical-`

[8] Location aware web services only exchange references to the actual data that are usually stored on shared data storage resources.

[9] http://www.w3.org/TR/owl-semantics/.

[10] http://krizik.felk.cvut.cz/ontologies/2008/kd.owl.

[11] http://www.w3.org/Submission/OWL-S/.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

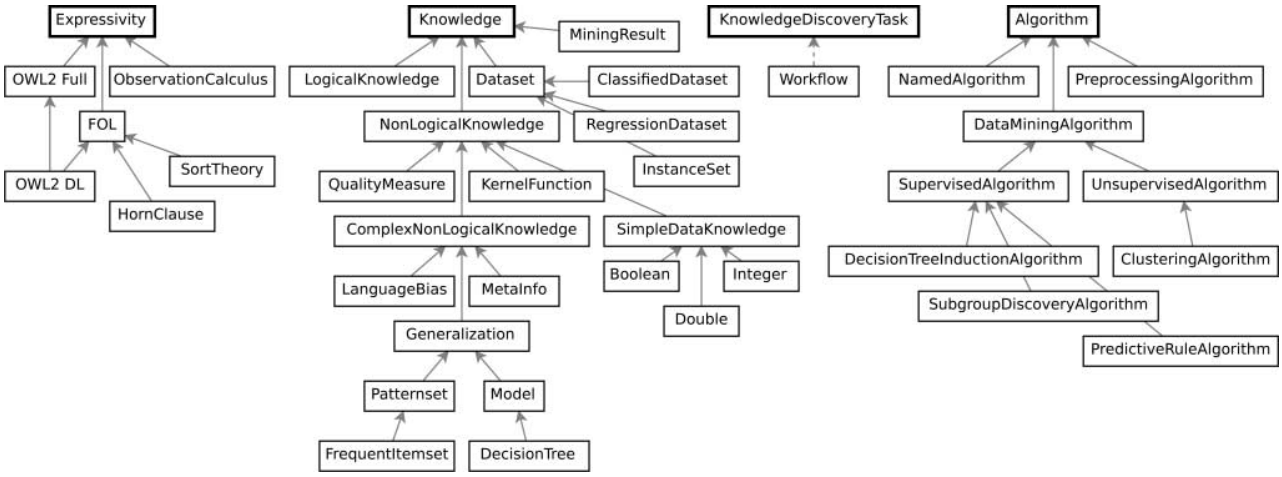

FIGURE 3. Part of the top level structure of the KD ontology (the whole ontology contains more than 150 concepts and 500 instances). Subclass relations are shown through solid arrows. The relation between `KnowledgeDiscoveryTask` and `Workflow`, shown through a dashed arrow, is defined as `forTask` relation. The `Workflow` class is a specialization of the OWL-S class `CompositeProcess`. The `Algorithm` class is a specialization of the OWL-S class `Process`, while the `NamedAlgorithm` class is a specialization of the OWL-S class `AtomicProcess`. The top-level classes shown in bold are subclasses of the predefined OWL class `Thing`.

`Knowledge`. It contains concept `Generalization`, which describes knowledge produced by data mining algorithms. The `Generalization` class currently contains two subclasses, `Patternset` and `Model` which can be distinguished by the property of decomposability and also by the type of algorithms used to produce them.

4.2. Algorithms

The notion of an algorithm involves all executable routines used in a knowledge discovery process, ranging from inductive algorithms to knowledge format transformations. Any algorithm turns a knowledge instance into another knowledge instance, e.g. inductive algorithms will typically produce a `Generalization` instance out of a `Dataset` instance. The `Algorithm` class is a base class for all algorithms, such as the *APriori* algorithm for association rule induction implemented in Orange or the *GenerateBows* algorithm implemented in the LATINO text mining library for constructing the bag of words representation of a collection of documents. For this work, we have refined the hierarchy of fully defined classes, such as `DecisionTreeAlgorithm` or `DataPreprocessingAlgorithm` for fine-grained categorization of data mining algorithms according to their functionality. This fine-grained hierarchy allows for the formulation of additional user constraints on the workflows. For example, constraints can refer to some particular category of data mining algorithms, e.g. `DiscretizationAlgorithm`, `FormatChangingAlgorithm`, `ClusteringAlgorithm`, etc.

Each algorithm configuration is defined by its input and output knowledge specifications and by its parameters. The `Algorithm` class is a specialization of the OWL-S class `Process` and an algorithm configuration is an instance of its subclass `NamedAlgorithm`.[12] Both the input knowledge and the parameters are instances of `AlgorithmParameter` and are defined using the `input` property. The output knowledge specifications are instances of `AlgorithmParameter` and defined using the `output` property. The parameter instances are then mapped to the appropriate `Knowledge` subclasses using the `isRangeOf` property.

4.3. Annotating algorithms

The KD ontology was used to annotate most of the algorithms available in the Orange toolkit. More than 60 algorithms have been annotated so far. We have also annotated the components of the LATINO text mining library according to their WSDL descriptions, using the approach described by Kalyanpur *et al.* [15]. As an example, we present a definition of the `GenerateBows` algorithm. `GenerateBows` is defined as an algorithm that can be applied to a collection of documents and produces a bag of words representation of these documents. The settings are quite complex; therefore, they are provided as a single input object. The definition in the description logic notation using the extended ABox syntax [16] is shown in Fig. 4.

The locally available Orange4WS algorithms were annotated manually, since no systematic description of these algorithms,

[12] The `DataMiningAlgorithm` class represents categories of data mining algorithms, e.g. subgroup discovery algorithm or decision tree induction algorithms, while the `NamedAlgorithmClass` represents concrete algorithms, such as *CN2* for subgroup discovery or *C4.5* for decision tree algorithms.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

$$
\begin{aligned}
\{\texttt{GenerateBows}\} &\sqsubseteq \texttt{NamedAlgorithm}\\
&\sqcap \exists\,\texttt{output}\cdot\{\texttt{GenerateBows-O-Bows}\}\\
&\sqcap \exists\,\texttt{input}\cdot\{\texttt{GenerateBows-I-Docs}\}\\
&\sqcap \exists\,\texttt{input}\cdot\{\texttt{GenerateBows-I-Settings}\}\\
\{\texttt{GenerateBows-I-Docs-Range}\} &\equiv \texttt{isRangeOf}\cdot\{\texttt{GenerateBows-I-Docs}\}\\
&\equiv \texttt{DocumentCollection}\\
\{\texttt{GenerateBows-O-Bows-Range}\} &\equiv \texttt{isRangeOf}\cdot\{\texttt{GenerateBows-O-Bows}\}\\
&\equiv \texttt{BowSpace}
\end{aligned}
$$

FIGURE 4. A definition of the `GenerateBows` method in the description logic notation using the extended ABox syntax.

e.g. in PMML[13] or WSDL[14] was available. The algorithms available in LATINO were also annotated manually based on their WSDL descriptions. The annotated algorithms also served as case studies to validate and extend the KD ontology, while the development of a procedure for semi-automatic annotation is a subject of future work.

5. AUTOMATED WORKFLOW CONSTRUCTION

The focus of this section is on automatic construction of abstract workflows of data mining algorithms. The mapping to concrete computational resources, particular data sets and algorithm parameters are not taken into account during abstract workflow construction. Each generated workflow is stored as an instance of the `Workflow` class and can be instantiated with a specific algorithm configuration either manually or using a predefined default configuration. We treat automatic workflow construction as a planning task, in which algorithms represent operators, and their input and output knowledge types represent preconditions and effects. However, since the information about the algorithms, knowledge types and the specification of the knowledge discovery task is encoded through the KD ontology, we implemented a planning algorithm capable of directly querying the KD ontology using the Pellet[15] reasoner. The main motivation for using Pellet was its ability to deal with literals, its availability in Protégé,[16] which we used for ontology development, and processing of SPARQL-DL [17] queries.

Our work was originally motivated mainly by complex relational data mining tasks, where the number of alternative workflows, which can be produced, is quite small, due to use of complex knowledge types and specialized algorithms [18]. This is also the case for the motivating text mining scenario from Section 2. The LATINO web services, which were annotated as specified in Section 4.3, can now be used in the process of automated workflow construction. Our planner was able to automatically (re)construct the workflow, presented in Section 2, according to the given instance of `KnowledgeDiscoveryTask` that specified the input data and the desired output. Note, however, that the *Pathfinder* algorithm is not present in the automatically generated workflow, as the corresponding web service is not yet annotated in the KD ontology. Figure 5 shows the automatically generated abstract workflow for the text mining scenario as well as an executable instantiation of the same workflow in the Orange Canvas inside Orange4WS.

As we have extended the KD ontology with annotations of algorithms available in the Orange and Orange4WS toolkits, we encountered the problem of having sets of algorithms, which—on the basis of their inputs and outputs—subsume each other or are even equivalent. For tasks such as inducing association rules from a propositional dataset, this led to producing a large number of workflows, a lot of which were very similar. In this work, we alleviate this problem by exploiting the algorithm subsumption hierarchy.

5.1. Exploiting algorithm hierarchy

The planning algorithm used to generate abstract workflows automatically is based on the Fast-Forward (FF) planning system [19]. We have implemented the basic architecture of the FF planning system consisting of the enforced hill climbing algorithm and the relaxed GRAPHPLAN. Since the planning problem in workflow construction contains no goal ordering, no mechanisms for exploiting goal ordering were implemented.

The planner obtains neighboring states during enforced hill-climbing by matching preconditions of available algorithms with currently satisfied conditions. Each matching is conducted during the planning time by posing an appropriate SPARQL-DL query to the KD ontology. In the original version of the planner [18], there are no mechanisms for exploiting the algorithms hierarchy. In this work, we have enhanced the algorithm in two ways: a hierarchy of algorithms based on defined classes and input/output specifications is computed, and in searching for neighboring states the planner exploits the algorithm hierarchy.

A hierarchy of algorithms is inferred before the actual planning. It needs to be recomputed only when a new algorithm is added to the ontology. The hierarchy of algorithms is based on the inputs and outputs of the algorithms and on the defined algorithm classes such as `PreprocessingAlgorithm`. It holds that $A_j \sqsubseteq A_i$ if for every input I_{ik} of A_i there is an input I_{jl} of algorithm A_j such that range of $I_{ik} \sqsubseteq I_{jl}$. An algorithm $A_i \equiv A_j$ if $A_j \sqsubseteq A_i$ and $A_i \sqsubseteq A_j$. The subsumption relation on algorithms is used to construct a forest of algorithms with roots given by the explicitly defined top-level algorithm classes, e.g. `DataPreprocessingAlgorithm`.

The planning algorithm was adapted so that in the search for the next possible algorithm, it traverses the forest structure instead of only a list of algorithms and considers a set of equivalent algorithms as a single algorithm. Currently, only constraints on repetition of some kind of algorithms (defined by a class or set of classes in the KD ontology)

[13] http://www.dmg.org/pmml-v4-0.html.
[14] www.w3.org/TR/wsdl.
[15] http://clarkparsia.com/pellet/.
[16] http://protege.stanford.edu/.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

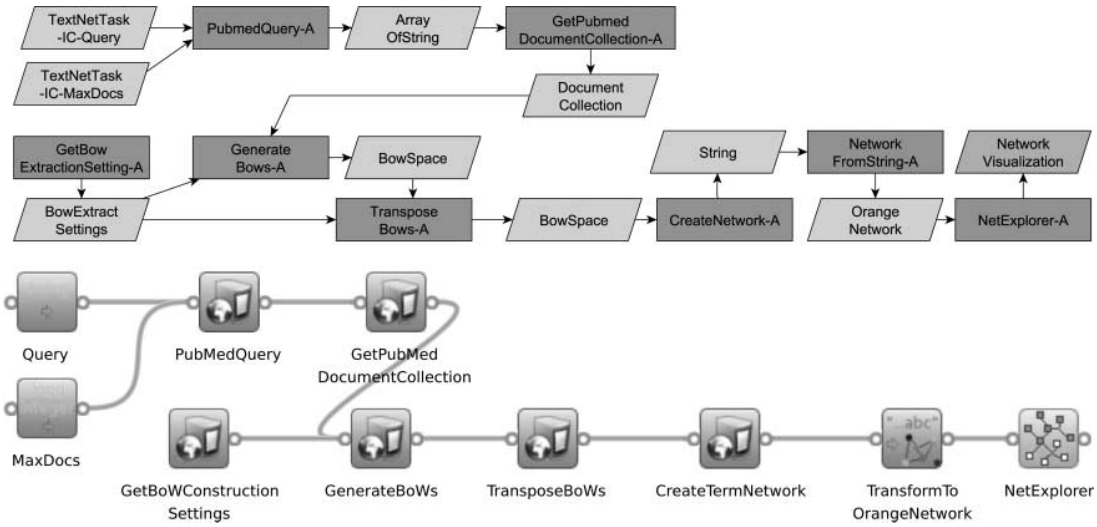

FIGURE 5. A schema of automatically generated abstract workflow and its executable instantiation in the Orange4WS environment. The underlying knowledge discovery task is a text-mining scenario of Section 2 for the analysis of a graphs of terms, obtained by querying the PubMed database using a publicly accessible web service.

```
task - instance of KnowledgeDiscoveryTask
maxSteps - max length of the workflow
constr - additional constraints
generateWorkflows(task, maxSteps, constr):
 classify KD ontology;
 algs := {instances of NamedAlgorithm};
 algforest := inferAlgorithmHierarchy(algs);
 workflows := runPlanner(task, algforest, maxSteps);
 atomicW := expandWorkflows(workflows, algforest);
 filteredW := filterWorkflows(atomicW, constr);
```

FIGURE 6. A skeleton of the procedure for automatic workflow composition using the KD ontology.

in a linear part of the workflow are built into the planner. Additional constraints on workflows are used only for filtering the generated workflows during post-processing (procedure `filterWorkflows`). Workflows for all the members of an equivalence set are generated using the `expandWorfklows` procedure. The information about algorithms subsumption is also used when presenting the workflows to the user. The whole procedure for workflow generation is outlined in Fig. 6.

The generated workflows are presented to the user through interactive visualization, which enables the user to browse the workflows from the most abstract level to any specific combination of algorithm instances. Workflows consisting of the smallest number of steps are presented first. An example of a set of workflows generated for discovering association rules in Orange4WS is shown in Fig. 7.

The set of generated workflows shown in Fig. 7 illustrates the use of the algorithm hierarchy for workflow presentation. Since there are four discretization, four sampling, five ranking and six continuization algorithms, it would be infeasible to present all the generated workflows without using the algorithm hierarchy. Automatic selection of a relevant subset of workflows is non-trivial and is the subject of future work.

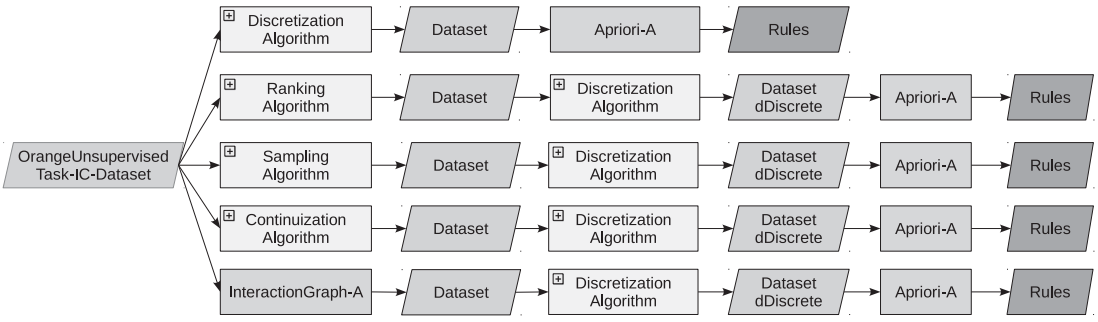

FIGURE 7. A set of automatically generated abstract workflows for discovering association rules in Orange4WS.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

5.2. Integrating annotations and planning into Orange4WS

We have developed a framework for integrating our methodology into the Orange4WS platform, so that the workflows, which were constructed manually using the Orange4WS GUI and which contain only algorithms represented in the KD ontology, can be automatically annotated using the KD ontology. The annotated workflows can then be used for querying and reasoning. All the information required for the Orange4WS representation is preserved in the annotation. Therefore, Orange4WS workflows can be recreated from the annotations and executed again in the Orange4WS toolkit. On the other hand, workflows generated by the planner using KD annotations of Orange4WS algorithms can be converted to the Orange4WS representation and executed in Orange4WS.

An overview of the framework is shown in Fig. 8. The Orange2Onto module, which acts as an interface between Orange4WS and the ontology representation, does not work directly with the internal representation of Orange4WS, but works with the OWS format used in the standard Orange distribution to store workflows in the XML format.

In order to formally capture the mapping between the internal Orange4WS representation and the representation of algorithms using the KD ontology, the Orange-Map (OM) ontology was developed defining templates for mapping of algorithms, data and parameters. The OM ontology is then used for converting the automatically generated workflows into the Orange representation. In order to facilitate the creation of the mapping for new algorithms, the mapping can be specified using an XML file. The corresponding instances in the ontology are then generated automatically.

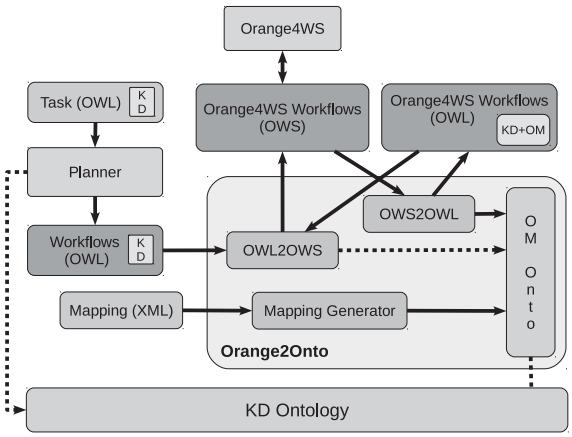

FIGURE 8. An overview of the framework for integration of annotations and planning into Orange4WS.

Annotation of a new algorithm available in Orange4WS thus requires the following steps:

(1) create instances of `AlgorithmParameter` for all inputs and outputs;
(2) create an instance of `NamedAlgorithm`;
(3) for each instance of `AlgorithmParameter` create a class defining its range (if not yet defined, add the necessary subclasses of `Knowledge` - this should be required only when a new type of algorithm is added);
(4) create an XML file defining a mapping between the algorithm representation in Orange and in the KD ontology;
(5) generate a mapping using the OM ontology by means of the provided tools.

Annotations of Orange4WS workflows containing algorithms not annotated using the KD ontology can also be created automatically. The missing information about input/output types of the algorithms is then either deduced from the links with annotated algorithms or considered to be a form of `Knowledge` expressed as a string. The annotations of such workflows can therefore be used for querying and for repeating the experiments; however, the generated annotation of the unknown algorithm is not suitable for planning.

The procedures for converting the Orange4WS representation to OWL and vice versa were implemented in Python using JPype[17] cross-language bridge to enable access to the Jena[18] ontology API implemented in Java.

6. USE CASES ILLUSTRATING THE UTILITY OF ORANGE4WS

This section presents three use cases from different domains, which illustrate some of the capabilities of the Orange4WS implementation. The presented workflows were not constructed automatically since not all workflow components and services were annotated in the KD ontology. Although the use cases presented here are simple, they give an overview of what our implementation is capable of, and illustrates the potential of web services technology for knowledge discovery.

6.1. Use case illustrating the availability of WEKA algorithms

A data mining practitioner would ideally like to have all the standard data mining algorithms at his disposal. While some of these are already provided in the Orange data mining toolkit[19] [2], data mining practitioners might also like to have

[17] http://jpype.sourceforge.net/.
[18] http://jena.sourceforge.net/.
[19] Implementations of classic data mining algorithms in Orange typically include several improvements, but some additions are not well documented, which is undesirable.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

the classical Weka algorithms [3] available as well. Workflow tools, which are based on the Java technology (e.g. KNIME, RapidMiner, Taverna), typically include the Weka core (i.e. algorithm implementations), and manually written wrappers. In Orange4WS, this is simply achievable through Weka web services already available on the internet, or created with our tools described in Section 3.4. The advantage of a web-service-based approach is twofold. First, through web services, the computation is distributed among servers hosting the services. Second, the latest versions of underlying software libraries are provided automatically to all clients given that the services are updated regularly.

A collection of Weka web services has been made available by A. Bosin.[20] There are eight services available: `atributeRank`, `attributeSelect`, `datasetFilter`, `datasetDiscretize`, `modelTest`, `modelApply`, `classifierBuild` and `clustererBuild`. Although these services currently have poor semantics (they operate using string representations of native WEKA data types), major functionality of Weka is available (attribute evaluation, data filtering, model building and testing) and can be used in the construction of data mining workflows.

This simple but illustrative use case implements the following processing steps: (1) loading the data from a local file, (2) ranking of attributes to manually select few best, (3) partitioning the data into the training and testing set, (4) building a classifier and evaluating it on the test set and (5) reporting the results to the user. This is accomplished by connecting 16 processing entities, 6 of which are web services, 3 are native Orange widgets while the rest are the supporting widgets provided by Orange4WS (data transformation and creation of integral data types). Note, however, that annotating the semantics of these services would enable reasoning and automatic planning of such workflows, and incorporation into larger and more complex scenarios. The workflow, created and executed within the Orange4WS platform, is shown in Fig. 9.

For illustrational purposes, we tested the created workflow with the *voting* dataset. Seven most important attributes were chosen and stratified random sampling was used to partition the data into training (75% of all instances) and test (25% of all instances) data. Weka's J48 decision tree induction algorithm was used to build a decision tree model, which was then applied to the test data. The `modelTest` web service provided Weka's model evaluation output, which was finally visualized locally with a data viewer component.

6.2. Relational data mining use case

This use case is built upon the propositionalization-based approach to relational subgroup discovery. The implementation of the relational subgroup discovery algorithm RSD, developed by Železný and Lavrač [13], is used to illustrate the use

[20] http://www.dsf.unica.it/~andrea/webservices.html.

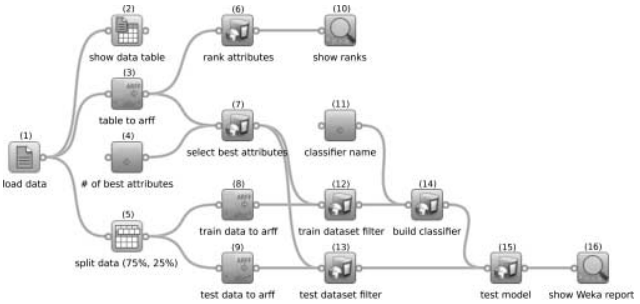

FIGURE 9. A workflow of Weka data mining services and local processing elements constructed within the Orange4WS platform. Components number 6, 7, 12, 13, 14, 15 are Weka web services; components 1, 2, and 5 are native Orange widgets. Other components are the supporting widgets provided by Orange4WS.

of our platform in a relational data mining scenario. The input to the RSD algorithm consists of a relational database containing (a) one main relation defined by a set of ground facts (training examples), each corresponding to a unique individual and having one argument specifying the class, and (b) background knowledge in the form of a Prolog program including functions, recursive predicate definitions, syntactic and semantic constraints, defined for the purpose of first-order feature construction.

Relational data mining and inductive logic programming are relatively separate research areas from standard propositional data mining. The main reason is the background of this research in logic programming, typically requiring a Prolog execution environment. Also, the data representation formalism is different (Prolog clauses), and taking into account relational background knowledge into the learning process requires a conceptually different approach from propositional learning, which only accepts tabular data as the input to a data mining algorithm. Consequently, standard data mining environments do not deal with relational data mining, and only once a service-oriented approach is considered, the two data mining frameworks can be handled within the same data mining environment.

The implementation of RSD, although efficient and stable, requires a YAP Prolog interpreter and specific implementation-related knowledge. Therefore, in order to be used in the Orange4WS environment, web services were created, which expose its abilities to the outside world. More specifically, using our tools for service development described in Section 3.4, we created a service for propositionalization and rule induction, respectively. In this use case, however, only the propositionalization service was used as we combined it with other, classic propositional data mining algorithms, also available as services. We employed the CN2-SD subgroup discovery algorithm [20], the SD algorithm [21], which implements beam search, and the APRIORI-SD algorithm [22]. It is worth noting that all three implementations are able to

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

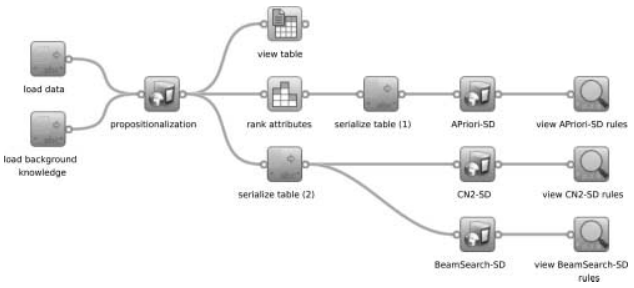

FIGURE 10. A workflow combining propositionalization of relational data, feature ranking, and subgroup discovery. Workflow components for `propositionalization`, `APRIORI-SD`, `CN2-SD` and `BeamSearch-SD` are web services, respectively.

produce results in the PMML[21] format, which makes them compatible with processing components outside Orange4WS.

The workflow of this use case, shown in Fig. 10, is illustrated on the Trains dataset [23], which is well-known in the area of relational data mining and Inductive Logic Programming. Altogether, 125 features were generated from the given relational data. As this was too much for the APRIORI-SD algorithm, feature selection was performed to obtain 10 best features (the other two algorithms were able to handle the complete feature set). For example, the highest ranked feature *f8* is as follows:

```
f8(Train) :- hasCar(Train,Car),
             carShape(Car,rectangle),
             carLength(Car,short),
             hasSides(Car,not_double).
```

Two example subgroups (one for each class), that are generated by the CN2-SD algorithm are shown as follows.

```
class = eastboundTrain IF f8  = true  AND
                          f82 = false AND
                          f25 = false AND
                          f40 = false

class = westboundTrain IF f121 = false AND
                          f5   = true  AND
                          f62  = false AND
                          f65  = false
```

6.3. Complex real-life systems biology use case

This use case is built upon two tools used in systems biology: the SEGS algorithm [11] and the Biomine system [24]. The combination of these systems, both of which make use of publicly available databases such as GO, Entrez, KEGG, PubMed, UniGene, OMIM and KEGG, enables novel scenarios for knowledge discovery from biological data.

In data mining terms, the SEGS (Search for Enriched Gene Sets) algorithm [11] is a specialized semantic subgroup discovery algorithm capable of inducing descriptions of groups of differentially expressed genes in terms of conjunctions of first-order features constructed from ontological relations available in public biological ontologies. The novelty of SEGS is that the method does not only test existing gene sets for differential expression but it also generates new gene sets that represent novel biological hypotheses. In short, in addition to testing the enrichment of individual GO and KEGG terms, this method tests the enrichment of newly defined gene sets constructed by the intersection and conjunctions of GO ontology terms and KEGG pathways.

The two new operators, *interact()* and *intersect()*, can yield to the discovery of gene sets that cannot be found by any other currently available gene set enrichment analysis software. They can be formalized as follows. If S is a gene set and ENTREZ is a database of gene–gene interactions, then the new interacting geneset INT(S) is defined as

$$\mathrm{INT}(S) = \{g : \exists g\prime \in S : \exists \mathrm{ENTREZ}(g, g\prime)\}. \quad (1)$$

Additionally, if S_1 is a term from the *molecular function* domain of the GO ontology, and S_2 belongs to the *cellular component* domain, and S_3 belongs to the *biological process* domain, and K is a KEGG pathway, then the gene set S defined by the *intersect()* operator is constructed as follows:

$$S_{S_1,S_2,S_3,K} = \{g : g \in \{S_1 \cap S_2 \cap S_3 \cap K\}\}. \quad (2)$$

As a result, the SEGS algorithm is able to discover complex rules that cannot be found by any other gene set enrichment analysis method or tool.

In the scope of the Biomine project, data from several publicly available databases were merged into a large graph (currently, ∼2 million nodes and 7 million edges) and a method for link discovery between entities in queries was developed. In the Biomine framework, nodes correspond to entities and concepts (e.g. genes, proteins, GO terms), and edges represent known, probabilistic relationships between nodes. A link (a relation between two entities) is manifested as a path or a subgraph connecting the corresponding nodes. The Biomine graph data model consists of various biological entities and annotated relations between them. Large, annotated biological data sets can be readily acquired from several public databases and imported into the graph model in a relatively straightforward manner. Currently used databases are: EntrezGene, GO, HomoloGene, InterPro, MIM, STRING, SwissProt, Trembl and UniProt.

The Biomine project provides the Biomine search web service (more specifically, a web API based on the HTTP

[21]The Predictive Model Markup Language (PMML) is an XML-based markup language that enables applications to define models related to predictive analytics and data mining and to share those models between PMML-compliant applications.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

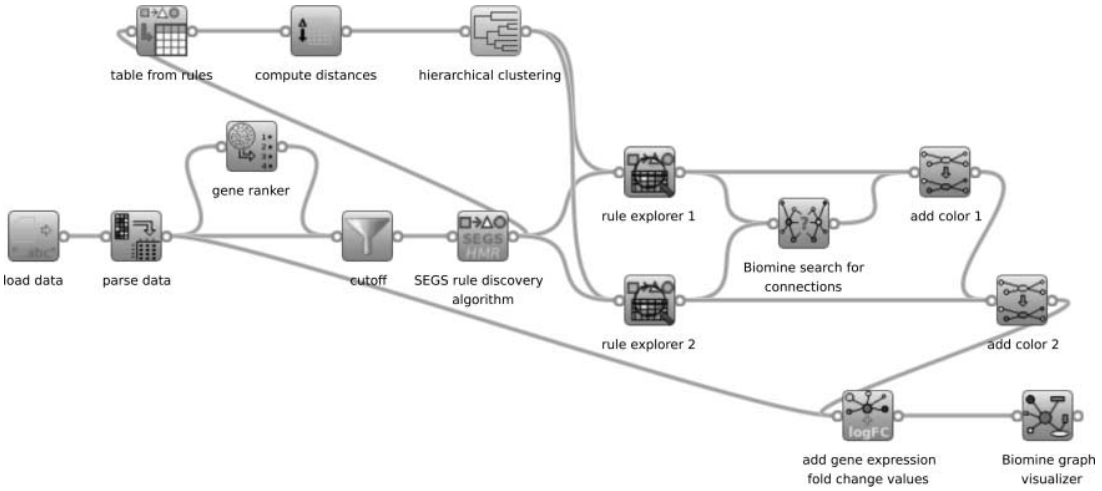

FIGURE 11. A workflow implementing the knowledge discovery scenario using the SEGS algorithm and the Biomine system. The component for computing rule distance and the interactive widget for hierarchical clustering are provided by Orange, other components are part of Orange4WS. The SEGS rule discovery algorithm is available as a SOAP web service while the Biomine search service is based on JSON.

protocol and JSON[22]), interactive web application and a powerful platform independent graph visualizer, implemented as a Java applet. The presented use case employs the Biomine search web service as well as the graph visualizer, which runs locally as an Orange4WS widget.

The original implementation of the SEGS algorithm was transformed into a SOAP 1.1 compatible web service using our tools described in Section 3.4. This greatly improved its flexibility and portability since the actual processing is now performed on more powerful server-side hardware employing massive parallel processing, and can be accessed from any workflow tool capable of calling web services. Moreover, publicly available databases, used by SEGS, can now be regularly updated by an automated update mechanisms. For space limitations, we do not provide a complete description of the SEGS service because it has a lot of input parameters but rather a short description of the provided functions and sample results.

As the SEGS algorithm has a large time complexity, the corresponding web service is designed as a partially stateful service. The SEGS service is actually a batch (job) processing service that stores the results of rule discovery; so they can be retrieved later using an unique user identifier. Apart from this, no consumer-specific context is stored or shared, and the invocations have no correlation to prior interactions. The service is able to report progress, and stores computed results indefinitely. It offers three functions: `runSEGS`, `getProgress` and `getResult`. The `getResults` function returns constructed rules, evaluated with the SEGS's built-in gene set enrichment tests (currently, Fisher's exact test, GSEA and PAGE).

A typical general scenario of knowledge discovery from gene expression data by using the SEGS algorithm and the Biomine system consists of the following steps:

(1) raw data preprocessing (normalization, missing values removal, merging, etc.);
(2) gene ranking (most typically, the Relief ranker or *t*-test is used);
(3) rule discovery using the SEGS algorithm;
(4) postprocessing of obtained SEGS rules (e.g. clustering);
(5) employing the Biomine system to discover interesting links, thus providing insights into the underlying biological processes.

The presented scenario, implemented as a workflow in the Orange4WS toolkit, is shown in Fig. 11. It is composed of local Orange4WS widgets, Orange components (clustering, example distances computation) and web services (the SEGS algorithm, Biomine search). First, the data are loaded and parsed, and the present genes are ranked. Then, the cutoff is applied to remove genes that seem not be involved in the observed biological processes. The resulting list of genes is fed to the SEGS algorithm to discover and evaluate rules composed of GO ontology terms, KEGG pathways as well as term interactions. The induced rules (if any) are sent to interactive hierarchical clustering component. The rules as well as clusters can be displayed in a user-friendly HTML browser where the user can select an interesting cluster or individual rule to be sent to the Biomine system.

The Biomine search web service returns the most reliable subgraph, which can be visualized using the provided interactive graph visualizer component. Such graphs offer non-trivial

[22]JSON is an acronym for JavaScript Object Notation, a lightweight text-based open standard designed for human-readable data interchange.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

```
class of differentially expressed genes :-
  leukocyte differentiation AND
  interact(primary immunodeficiency)

  leukocyte differentiation AND
  membrane AND
  interact(natural killer cell mediated cytotoxicity)

  lymphocyte differentiation AND
  interact(fc epsilon RI signaling pathway) AND
  interact(fc gamma R-mediated phagocytosis)
```

FIGURE 12. The top three rules describing the class of differentially expressed genes from a classical acute lymphoblastic leukemia (ALL) dataset. The rules are composed of terms from the GO ontology and KEGG pathways.

insights into biological relations that are of interest to domain experts, and can potentially reveal previously unknown links (literature search is also included in Biomine).

For illustrative purposes, the presented knowledge discovery scenario was tested on a sample microarray dataset, a classical acute lymphoblastic leukemia (ALL) dataset [25]. The top three rules (according to the *P*-value obtained by permutation testing) that describe the class of differentially expressed genes are shown in Fig. 12. The rules are composed of terms from the GO ontology and KEGG pathways, respectively.

7. RELATED WORK

This section presents the work related to the key components of our framework: knowledge discovery domain formalization for workflow construction and reuse, workflow editing and execution environment and service-oriented architecture for knowledge discovery.

Construction of analytic workflows has been the topic of substantial research and development in the recent years. The best known systems include the Triana [7] workflow environment for P2P and Grid containing a system for integrating various types of middleware toolkits, and the Taverna [6] environment for workflow development and execution (primarily used in bioinformatics). However, these two system currently do not provide means for automatic workflow construction. Although Triana and Taverna are not specialized to support data mining tasks, there are projects aimed to incorporate general data mining components into these two sotfware systems. In the context of the DataMiningGrid project [26], which used Triana as a front end, generic and sector-independent data mining tools and services for the grid were developed. Similarly, a number of systems biology related data mining web services have become available in the myExperiment Virtual Research Environment[23] which can be used in Taverna (or any other tool capable of using web services).

On the other hand, the specialized data mining platforms Weka [3], KNIME [4], RapidMiner [5] and Orange [2] have mostly failed to recognize and adopt the web services computing paradigm, and the need for unification and formalization of the field of data mining. Currently, only RapidMiner offers some support for service-oriented computing through the Web Extension component, while none integrates an ontology of data, algorithms and tasks.

There has been some work on workflows for distributed data mining using a service-oriented architecture, e.g. Guedes *et al.* [27] and Ali *et al.* [28]. These systems focus on demonstrating the feasibility of a service-oriented approach for distributed data mining with regard to parallelization and distributed data sources, while none of these approaches enable automated data mining workflow construction.

Also relevant for our work is Weka4WS [29], a framework that extends the Weka toolkit to support distributed data mining on the Grid. The Weka4WS user interface supports the execution of both local and remote data mining tasks but only native Weka components and extensions are available, and the framework does not support arbitrary web services that can be found on the internet.

There exist several systems using a formal representation of data mining (DM) operators for automated workflow composition and ranking, including IDEA [30], NExT [31] and KDDVM [32], which focus solely on propositional data mining, and do not offer a general scientific workflow environment for data mining, whereas our approach allows also for the inclusion of complex relational data mining and text mining algorithms in a general workflow-based data mining environment.

Other efforts to provide a systematic formalization of the data mining tasks include projects MiningMart [33], DataMiningGrid [26], and a system described by Li *et al.* [34]. The first two focus on mining propositional patterns from data stored in a relational database. None of the systems provide means for automated workflow construction.

Another, very practically oriented approach to the generalization data mining algorithm implementations was introduced by Zaki *et al.* [35]. The proposed Data Mining Template Library is built using the principle of generic programming.[24] The library is generic with respect to the algorithm, data source and format, data structure, storage management and pattern to be mined. Nevertheless, this approach focuses solely on frequent pattern mining, and only provides generic templates in implementation-specific programming language instead of a general and independent ontology.

Parallel to our work, the OntoDM [36] ontology is currently being developed, adopting a principled top-down approach aimed at achieving maximal generality of the developed ontology. Given the complexity of the domain subject to be

[23] http://www.myexperiment.org/

[24] The Generic Programming paradigm focuses on finding suitable abstractions so that a single, generic algorithm can cover many concrete implementations.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

modeled, the ontology is currently not sufficiently refined for the purpose of automated workflow construction. Also, unlike our ontology, OntoDM is not compatible with OWL-S. Recent work aimed at the development of a data mining ontology includes also [37, 38], where the work by Hilario *et al.* [37] has been influenced also by the knowledge discovery ontology described in this paper.

Solutions to the problem of web service composition in the context of planning are also relevant for our work. The work of Lecue *et al.* [39] relies on computing a *causal link matrix* for all the available services. In contrast, we work with a more general, non-linear notion of a plan. Work by Sirin *et al.* [40], Klusch *et al.* [41] and Liu et al. [42] translate an OWL description to a planning formalism based on PDDL. While the work presented in [41] and [42] use classical STRIPS planning, Sirin *et al.* [40] employ Hierarchical Task Network (HTN) planning. HTN is not applicable in our framework as it is not constrained to tree-based task decomposition. The approach presented by Liu *et al.* [42] and Klusch *et al.* [41] uses a reasoner in the pre-processing phase; we take a step further by integrating the reasoning engine directly with the planner. Planning directly in description logics is addressed by Hoffmann [43]. Currently, the algorithm can only deal with DL-Lite descriptions with reasonable efficiency.

8. CONCLUSIONS

This paper proposes a third-generation knowledge discovery framework and its implementation in a service-oriented data mining platform named Orange4WS. Based on the Orange data mining toolkit, which supports the execution of workflows of processing components, our new platform upgrades its capabilities by transparent integration of web services. As web services are an extremely versatile and powerful concept that is becoming more and more popular, we believe their use in data mining and knowledge discovery will increase rapidly. We have added semantic capabilities to the framework by proposing a methodology for integrating semantic annotation and planning into our data mining platform by means of the developed KD ontology. We have developed a planner, which exploits the hierarchy of algorithms annotated using the KD ontology.

In summary, the described service-oriented knowledge discovery paradigm shift, implemented in the Orange4WS platform, was achieved through the integration of latest achievements in the field of service-oriented approaches to knowledge discovery, knowledge discovery ontologies and automated composition of scientific workflows. This paradigm shift can potentially lead to the development of a novel intelligent knowledge discovery process model for data mining, extending the current CRISP-DM data mining methodology.[25] This paradigm shift will enable the orchestration of web-based data mining services and fusion of information of various formats, as well as design of repeatable data mining and information fusion workflows used in novel life science, bioinformatics and e-science applications.

Similarly to all other service-based solutions, a potential drawback of the presented platform is that the execution of workflows depends on the availability and reliability of remote services. As a result, the enactment of a selected workflow is not entirely under the control of the user, and there is no guarantee of successful completion of experiments. Also, the presented platform is still conventional in the sense that it does not support Web 2.0 collaborative work functionalities. Finally, our platform is platform-independent but system independence is not addressed. Note that this would require a complete reimplementation of the user interface and local processing components using web technologies only. Such reimplementation would allow for employing Orange4WS on any system equipped with a modern web browser, including mobile devices.

In future work, we will explore adding means for semantic web service discovery and their semi-automatic annotation. The planner will also be a subject of future improvements as we aim to incorporate the ability of satisfying user-defined constraints and preferences. We will add support for web service libraries other than ZSI such as the WSO2 web service framework (based on Apache Axis2/C), lightweight SOAP client SUDS and the *pysimplesoap* library, which will greatly expand the range of supported web services.

Finally, the proposed SoKD framework and its implementation in the Orange4WS platform will enable also for meta-mining of data mining workflows, which is a challenging topic of future research.

[25] http://www.crisp-dm.org/

REFERENCES

[1] Finin, T. *et al.* (2007). National Science Foundation Symposium on Next Generation of Data Mining and Cyber-Enabled Discovery for Innovation (NGDM'07). Final Report.

[2] Demšar, J., Zupan, B., Leban, G. and Curk, T. (2004) Orange: From Experimental Machine Learning to Interactive Data Mining. In Boulicaut, J.-F., Esposito, F., Giannotti, F. and Pedreschi, D. (eds), *PKDD*, Lecture Notes in Computer Science 3202, pp. 537–539. Springer.

[3] Witten, I.H., Frank, E. and Hall, M.A. (2011) *Data Mining: Practical Machine Learning Tools and Techniques* (3rd edn). Morgan Kaufmann, Amsterdam.

[4] Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K. and Wiswedel, B. (2007) KNIME: The Konstanz Information Miner. In Preisach, C., Burkhardt, H., Schmidt-Thieme, L. and Decker, R. (eds), *GfKl*, Studies in Classification, Data Analysis, and Knowledge Organization, pp. 319–326. Springer.

[5] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M. and Euler, T. (2006) YALE: Rapid Prototyping for Complex Data Mining

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

Tasks. In Eliassi-Rad, T., Ungar, L.H., Craven, M. and Gunopulos, D. (eds) *KDD*, pp. 935–940. ACM.

[6] Roure, D.D., Goble, C.A. and Stevens, R. (2009) The design and realisation of the myExperiment virtual research environment for social sharing of workflows. *Future Gener. Comput. Syst.*, **25**, 561–567.

[7] Taylor, I., Shields, M., Wang, I. and Harrison, A. (2007) The Triana workflow environment: architecture and applications. *Workflows e-Sci.*, **1**, 320–339.

[8] Fielding, R.T. (2000) Architectural styles and the design of network-based software architectures. PhD Thesis, University of California, Irvine CA 92697, USA.

[9] Zupan, B., Leban, G., Demšar, J. and Curk, T. (2003) Widgets and Visual Programming. Technical Report. Bioinformatics Laboratory, Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia.

[10] Hull, D., Wolstencroft, K., Stevens, R., Goble, C.A., Pocock, M.R., Li, P. and Oinn, T. (2006) Taverna: a tool for building and running workflows of services. *Nucleic Acids Res.*, **34**, 729–732.

[11] Trajkovski, I., Lavrač, N. and Tolar, J. (2008) SEGS: search for enriched gene sets in microarray data. *J. Biomed. Inf.*, **41**, 588–601.

[12] Erl, T. (2005) *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

[13] Zelezný, F. and Lavrač, N. (2006) Propositionalization-based relational subgroup discovery with RSD. *Mach. Learn.*, **62**, 33–63.

[14] Dzeroski, S. (2006) Towards a General Framework for Data Mining. In Dzeroski, S. and Struyf, J. (eds), *KDID*, Lecture Notes in Computer Science 4747, pp. 259–300. Springer.

[15] Kalyanpur, A., Pastor, D.J., Battle, S. and Padget, J.A. (2004) Automatic Mapping of OWL Ontologies into Java. In Maurer, F. and Ruhe, G. (eds) *SEKE*, pp. 98–103.

[16] Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D. and Patel-Schneider, P.F. (eds) (2003) *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

[17] Sirin, E. and Parsia, B. (2007) SPARQL-DL: SPARQL Query for OWL-DL. In Golbreich, C., Kalyanpur, A. and Parsia, B. (eds), *OWLED*, CEUR Workshop Proceedings, Vol. 258. CEUR-WS.org.

[18] Žáková, M., Křemen, P., Železný, F. and Lavrač, N. (2008) Planning to Learn with a Knowledge Discovery Ontology. *Planning to Learn Workshop (PlanLearn 2008) at ICML 2008*. Helsinki, Finland.

[19] Hoffmann, J. and Nebel, B. (2001) The FF planning system: fast plan generation through heuristic search. *J. Artif. Intell. Res. (JAIR)*, **14**, 253–302.

[20] Lavrač, N., Kavšek, B., Flach, P.A. and Todorovski, L. (2004) Subgroup discovery with CN2-SD. *J Mach. Learn. Res.*, **5**, 153–188.

[21] Gamberger, D. and Lavrač, N. (2002) Expert-guided subgroup discovery: methodology and application. *J. Artif. Intell. Res. (JAIR)*, **17**, 501–527.

[22] Kavšek, B. and Lavrač, N. (2006) Apriori–SD: adapting association rule learning to subgroup discovery. *Appl. Artif. Intell.*, **20**, 543–583.

[23] Michie, D., Muggleton, S., Page, D. and Srinivasan, A. (1994) To the International Computing Community: A New East–West Challenge. Technical Report. Oxford University Computing laboratory, Oxford, UK.

[24] Sevon, P., Eronen, L., Hintsanen, P., Kulovesi, K. and Toivonen, H. (2006) Link Discovery in Graphs Derived from Biological Databases. In Leser, U., Naumann, F. and Eckman, B.A. (eds), *DILS*, Lecture Notes in Computer Science 4075, pp. 35–49. Springer.

[25] Chiaretti, S., Li, X., Gentleman, R., Vitale, A., Vignetti, M., Mandelli, F., Ritz, J. and Foa, R. (2004) Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood*, **103**, 2771–2778.

[26] Stankovski, V., Swain, M.T., Kravtsov, V., Niessen, T., Wegener, D., Kindermann, J. and Dubitzky, W. (2008) Grid-enabling data mining applications with DataMiningGrid: An architectural perspective. *Future Gener. Comput. Syst.*, **24**, 259–279.

[27] Guedes, D., Meira, W. and Ferreira, R. (2006) Anteater: a service-oriented architecture for high-performance data mining. *IEEE Internet Comput.*, **10**, 36–43.

[28] Ali, A.S., Rana, O.F. and Taylor, I.J. (2005) Web Services Composition for Distributed Data Mining. *ICPP Workshops*, pp. 11–18. IEEE Computer Society. Oslo, Norway.

[29] Talia, D., Trunfio, P. and Verta, O. (2005) Weka4WS: A WSRF-Enabled Weka Toolkit for Distributed Data Mining on Grids. In Jorge, A., Torgo, L., Brazdil, P., Camacho, R. and Gama, J. (eds), *PKDD*, Lecture Notes in Computer Science 3721, pp. 309–320. Springer.

[30] Bernstein, A., Provost, F.J. and Hill, S. (2005) Toward intelligent assistance for a data mining process: an ontology-based approach for cost-sensitive classification. *IEEE Trans. Knowl. Data Eng.*, **17**, 503–518.

[31] Bernstein, A. and Dänzer, M. (2007) The NExT System: Towards True Dynamic Adaptations of Semantic Web Service Compositions. In Franconi, E., Kifer, M. and May, W. (eds), *The Semantic Web: Research and Applications*, Chapter 52, Lecture Notes in Computer Science 4519, pp. 739–748. Springer, Berlin, Heidelberg.

[32] Diamantini, C., Potena, D. and Storti, E. (2009) Ontology-Driven KDD Process Composition. In Adams, N.M., Robardet, C., Siebes, A. and Boulicaut, J.-F. (eds), *IDA*, Berlin, Lecture Notes in Computer Science 5772, pp. 285–296. Springer.

[33] Morik, K. and Scholz, M. (2003) The MiningMart Approach to Knowledge Discovery in Databases. In Zhong, N. and Liu, J. (eds), *Intelligent Technologies for Information Analysis*, pp. 47–65. Springer.

[34] Li, Y. and Lu, Z. (2004) Ontology-based universal knowledge grid: enabling knowledge discovery and integration on the grid. *IEEE SCC*, pp. 557–560. IEEE Computer Society. Shanghai, China.

[35] Hasan, M.A., Chaoji, V., Salem, S., Parimi, N. and Zaki, M.J. (2005) DMTL: A Generic Data Mining Template Library. *Proc. Workshop on Library-Centric Software Design, Object-Oriented*

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

Programming, Systems, Languages and Applications Conf. (OOPSLA '05), San Diego, CA, USA, pp. 53–63. Rensselaer Polytechnic Institute.

[36] Panov, P., Džeroski, S. and Soldatova, L.N. (2008) OntoDM: An Ontology of Data Mining. *ICDM Workshops*, pp. 752–760. IEEE Computer Society. Pisa, Italy.

[37] Hilario, M., Kalousis, A., Nguyen, P. and Woznica, A. (2009) A Data Mining Ontology for Algorithm Selection and Meta-Mining. *Proc. 2nd Workshop on Service-Oriented Knowledge Discovery (SoKD '09): Third Generation Data Mining: Towards Service-Oriented Knowledge Discovery, ECML PKDD Conf.*, Bled, Slovenia, September 7–11, pp. 76–87.

[38] Diamantini, C., Potena, D. and Storti, E. (2009) KDDONTO: An Ontology for Discovery and Composition of KDD Algorithms. *Proc. 2nd Workshop on Service-Oriented Knowledge Discovery (SoKD '09): Third Generation Data Mining: Towards Service-Oriented Knowledge Discovery, ECML PKDD Conf.*, Bled, Slovenia, September 7–11, pp. 13–24.

[39] Lécué, F., Delteil, A. and Léger, A. (2007) Applying Abduction in Semantic Web Service Composition. *ICWS*, pp. 94–101. IEEE Computer Society. Salt Lake City, Utah, USA.

[40] Sirin, E., Parsia, B., Wu, D., Hendler, J.A. and Nau, D.S. (2004) HTN planning for web service composition using SHOP2. *J. Web Sem.*, **1**, 377–396.

[41] Klusch, M. and Gerber, A. (2005) Semantic Web Service Composition Planning with OWLS-XPlan. *Proc. 1st Int. AAAI Fall Symp. Agents and the Semantic Web*, pp. 55–62. Arlington, Virginia, USA.

[42] Liu, Z., Ranganathan, A. and Riabov, A. (2007) A planning approach for message-oriented semantic web service composition. *AAAI*, pp. 1389–1394. Proc. of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22–26, 2007, Vancouver, British Columbia, Canada.

[43] Hoffmann, J. (2008) Towards Efficient Belief Update for Planning-Based Web Service Composition. In Ghallab, M., Spyropoulos, C.D., Fakotakis, N. and Avouris, N.M. (eds), *ECAI*, Frontiers in Artificial Intelligence and Applications 178, pp. 558–562. IOS Press.

Downloaded from comjnl.oxfordjournals.org by guest on August 23, 2011

3 The SegMine Methodology

This chapter presents the SegMine methodology for semantic analysis of microarray data, which was developed and implemented in the Orange4WS platform.

High throughput platforms such as microarrays yield amounts of data which far exceed the analytical capabilities of humans. Obviously, efficient statistical and data mining methods have to be employed to extract useful information from the expression levels of tens of thousands of genes. With the availability of controlled vocabularies (ontologies), the Gene Ontology (Ashburner et al., 2000) (GO) being a notable example, the emphasis has shifted from identifying individual differentially expressed genes to the identification of categories of genes (genes assigned to groups according to the processes, functions and cellular components in which they are involved).

The GO ontology provides a hierarchical collection of terms which describe molecular functions, cellular components and biological processes, and a typical analytical approach is to perform some kind of statistical procedure (e.g., Fisher's exact test (Agresti, 1992), GSEA (Subramanian et al., 2007), and PAGE (Kim and Volsky, 2005)) to identify sets of genes which are significantly over- or under-expressed. In contrast with gene set enrichment analysis tools, such as DAVID (Huang et al., 2009a,b), GOrilla (Eden et al., 2009), GOMiner (Zeeberg et al., 2003) and others, the SEGS (Search for Enriched Gene Sets) algorithm (Trajkovski, 2007; Trajkovski et al., 2008) can discover gene sets which are constructed as combinations of GO terms, Kyoto Encyclopedia of Genes and Genomes Orthology (Kanehisa and Goto, 2000) (KEGG) terms, and terms describing gene-gene interactions in the Entrez (Maglott et al., 2005) database. However, SEGS cannot make use of other sources of biological knowledge and literature, which may provide important information on complex relations and interactions among genes, groups of genes and other biological entities.

The problem of integrating large biological databases is addressed by the Biomine system (Eronen and Toivonen, 2012; Sevon et al., 2006), which provides a search engine for link discovery and visualisation of heterogeneous biological databases. As of October 2012, Biomine integrates nine major databases: EntrezGene, Gene Ontology, HomoloGene, InterPro, MIM, STRING, SwissProt, Trembl, and UniProt which are merged into a single large graph. Moreover, Biomine provides probabilistic graph search algorithms to automatically extract the most relevant subgraphs, and can search for links between given query terms.

The main research idea of the SegMine methodology is to integrate an advanced algorithm for the analysis of experimental data with a link discovery engine which integrates major sources of biological knowledge. By such integration we aimed to achieve novel explanations of findings in experimental data and improved hypothesis generation and data interpretation. However, the implementation of such a methodology in an extensible and user-friendly way poses a number of problems. First, the SEGS algorithm is computationally very intensive and its requirements exceed typical desktop computer capabilities (due to the size of the ontologies used and because of the employed permutation testing technique). Second, the requirements of the Biomine system are also far beyond desktop computing, and moreover, the Biomine system is not open software, but is allowed to be used only through one of its interfaces. Furthermore, the proposed connec-

tion of the two systems is far from being straightforward and requires a number of additional data processing steps. Finally, additional algorithms and visualisations should also be available at various stages of the data processing pipeline.

As a solution, we have developed a workflow-based implementation of SegMine in Orange4WS which employs web services to enable access to SEGS and Biomine, and a number of supporting components, provided by Orange and Orange4WS to complement the data analysis process. A screenshot of the Orange4WS platform running the SegMine workflow is shown in Figure 3.1. The methodology is presented in the following publication:

Podpečan, V.; Lavrač, N.; Mozetič, I.; Kralj Novak, P.; Trajkovski, I.; Langohr, L.; Kulovesi, K.; Toivonen, H.; Petek, M.; Motaln, H.; Gruden, K. SegMine workflows for semantic microarray data analysis in Orange4WS. *BMC Bioinformatics* **12**, 416 (2011)

In addition, Appendices B and D include the following relevant material. Appendix B presents our implementation of the SegMine methodology by providing descriptions of Orange4WS workflow components, their inputs and outputs as well as graphical representations (where applicable). Appendix D lists the complete set of results of the SEGS algorithm on human mesenchymal stem cell gene expression data. In the expert analysis these results were used with the Biomine link discovery engine to discover connections which served as a basis for the formulation of new scientific hypotheses.

The author contributions are as follows. Vid Podpečan contributed to the development of the SegMine methodology and implemented it as a set of interactive workflow components in Orange4WS. Nada Lavrač and Igor Mozetič conceived and coordinated the computer science aspects of the study. Petra Kralj Novak originated the idea of connecting SEGS and Biomine. Igor Trajkovski developed and implemented SEGS in the context of his thesis and implemented its parallelisation. Laura Langohr implemented the Biomine's gene medoids algorithm which is available in SegMine. Kimmo Kuloves implemented the Biomine visualiser which allows for interactive exploration of Biomine graphs in SegMine. Hannu Toivonen conceived and coordinated the development of Biomine. Marko Petek contributed to the experimental comparison of SegMine and DAVID on the ALL dataset. Helena Motaln performed an expert analysis of the MSC dataset and formulated new hypotheses. Kristina Gruden coordinated the biological aspects of the study.

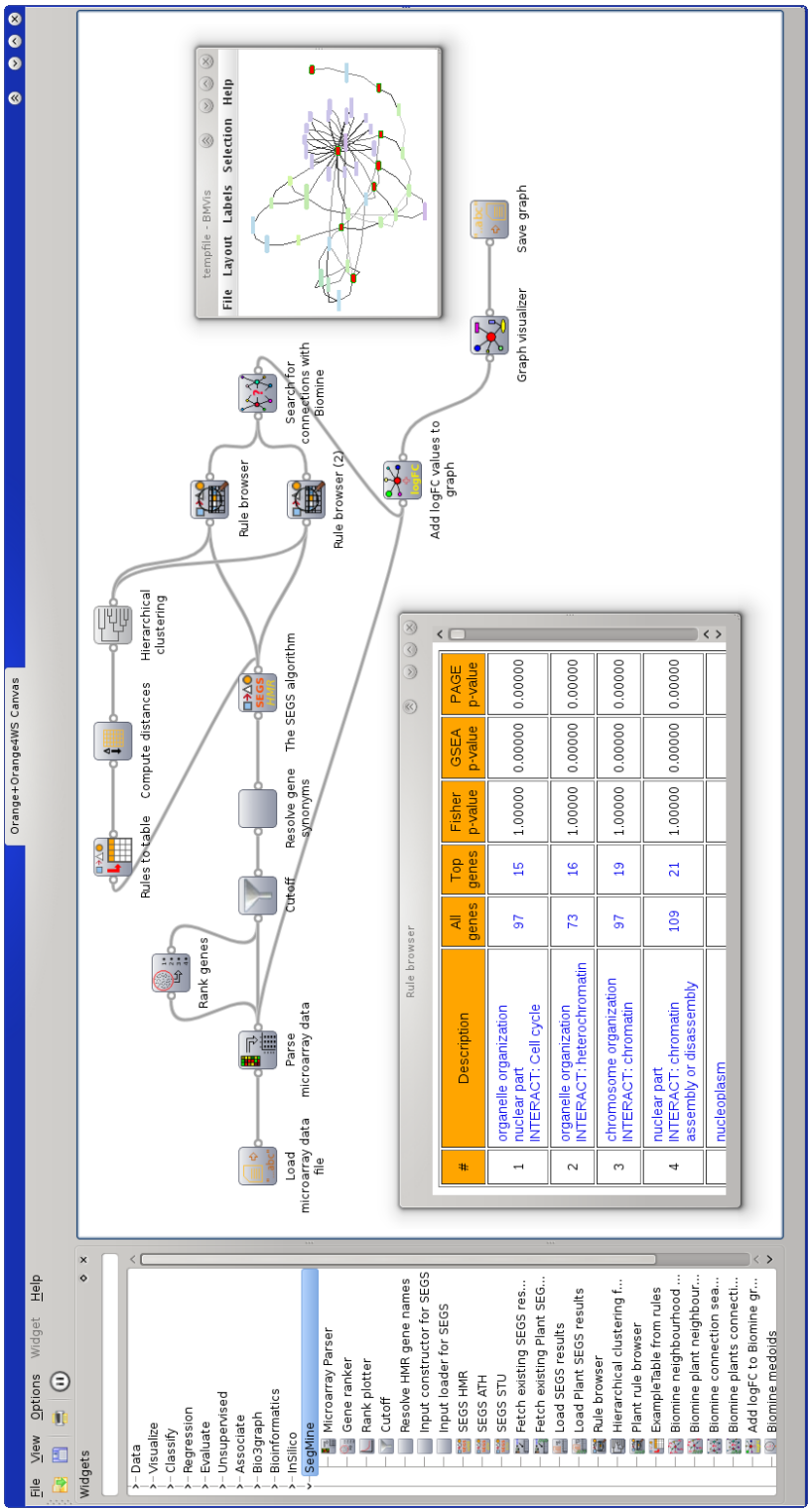

Figure 3.1: A screenshot of the Orange4WS platform running the SegMine workflow featuring the SEGS web service and the Biomine search engine. Two windows displaying the results of SEGS (an HTML table) and Biomine (an interactive graph) are also shown.

Podpečan et al. BMC Bioinformatics 2011, 12:416
http://www.biomedcentral.com/1471-2105/12/416

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

SegMine workflows for semantic microarray data analysis in Orange4WS

Vid Podpečan[1*], Nada Lavrač[1,2], Igor Mozetič[1], Petra Kralj Novak[1], Igor Trajkovski[3], Laura Langohr[4], Kimmo Kulovesi[4], Hannu Toivonen[4], Marko Petek[5], Helena Motaln[5] and Kristina Gruden[5]

Abstract

Background: In experimental data analysis, bioinformatics researchers increasingly rely on tools that enable the composition and reuse of scientific workflows. The utility of current bioinformatics workflow environments can be significantly increased by offering advanced data mining services as workflow components. Such services can support, for instance, knowledge discovery from diverse distributed data and knowledge sources (such as GO, KEGG, PubMed, and experimental databases). Specifically, cutting-edge data analysis approaches, such as semantic data mining, link discovery, and visualization, have not yet been made available to researchers investigating complex biological datasets.

Results: We present a new methodology, SegMine, for semantic analysis of microarray data by exploiting general biological knowledge, and a new workflow environment, Orange4WS, with integrated support for web services in which the SegMine methodology is implemented. The SegMine methodology consists of two main steps. First, the semantic subgroup discovery algorithm is used to construct elaborate rules that identify enriched gene sets. Then, a link discovery service is used for the creation and visualization of new biological hypotheses. The utility of SegMine, implemented as a set of workflows in Orange4WS, is demonstrated in two microarray data analysis applications. In the analysis of senescence in human stem cells, the use of SegMine resulted in three novel research hypotheses that could improve understanding of the underlying mechanisms of senescence and identification of candidate marker genes.

Conclusions: Compared to the available data analysis systems, SegMine offers improved hypothesis generation and data interpretation for bioinformatics in an easy-to-use integrated workflow environment.

Background

Systems biology aims at system-level understanding of biological systems, that is, understanding of system structures, dynamics, control methods, and design methods [1]. Biologists collect large quantities of data from in vitro and in vivo experiments with gene expression microarrays being the most widely used high-throughput platform [2]. Since the amount of available data exceeds human analytical capabilities, technologies that help analyzing and extracting useful information from such large amounts of data need to be developed and used.

The field of *microarray data analysis* has shifted emphasis from methods for identifying individual differentially expressed genes to methods for identifying differentially expressed gene categories (enriched gene sets). A gene set is *enriched* if the member genes are statistically significantly differentially expressed compared to the rest of the genes. One of the most popular controlled vocabularies (ontologies) used for over representation analysis was developed by the Gene Ontology (GO) Consortium [3].

A typical approach to gene set enrichment is to perform Fisher's exact test [4] to identify gene sets annotated by the GO ontology terms which are statistically significantly over-represented. Examples of other approaches include Gene Set Enrichment Analysis (GSEA) [5], GSEA-P [6], Parametric Analysis of Gene set Enrichment (PAGE) [7], and other methods [8-11]. A comparison of several software and web tools (OntoExpress, CLASSIFI, GoMiner, EASEonline, GeneMerge,

* Correspondence: vid.podpecan@ijs.si
[1]Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
Full list of author information is available at the end of the article

BioMed Central

© 2011 Podpečan et al; licensee BioMed Central Ltd. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/2.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

FuncAssociate, GOTree Machine, etc.) has been performed by Khatri and Draghici [12].

In contrast with the existing gene set enrichment methods, the SEGS (**S**earch for **E**nriched **G**ene **S**ets) semantic subgroup discovery algorithm [13], which is a part of the SegMine methodology, constructs candidate gene sets as combinations of GO terms, Kyoto Encyclopedia of Genes and Genomes Orthology [14] (KEGG) terms, and terms describing gene-gene interactions in the Entrez [15] database. Furthermore, the generalized variant of SEGS called g-SEGS [16] is not limited to the domain of systems biology, and allows for semantic subgroup discovery on any domain using supplied domain ontologies. One way to construct biologically meaningful interpretations from a large amount of experimental data is to present and visualize it using correlation networks. A notable example is ONDEX [17], a database system that combines methods from semantic database integration and text mining with methods for graph-based analysis. It can be applied to the interpretation of gene expression results. Reactome [18], Biocyc [19], BioLayout [20] and MapMan [21] are examples both of curated knowledge bases of metabolic reactions and pathways, and of computational tools to aid in the interpretation of microarrays and similar large-scale datasets. These tools offer powerful techniques for data exploration, but they often are limited to a few types of data and rely on the user to notice relevant connections. In contrast, the Biomine system [22], which is an integral part of SegMine, is a search engine for link discovery and visualization of heterogeneous biological databases. Biomine currently integrates and indexes information from eight major databases (Entrez Gene, UniProt, Gene Ontology, OMIM, NCBI HomoloGene, InterPro, STRING, and KEGG), merged into a single large graph. Moreover, Biomine provides probabilistic graph search algorithms to automatically extract the most relevant subgraphs, and can search for links between given query sets. Due to the complexity of data analysis, bioinformaticians rely more and more on tools that enable composition and reuse of workflows. Several tools exist to support creation, management, and execution of advanced scientific workflows, such as the Taverna workbench [23], the Weka data mining platform [24], KNIME [25], Orange [26], the Kepler scientific workflow system [27], and the Triana problem solving environment [28]. However, workflow environments originating from systems biology have virtually no support for advanced machine learning and data mining techniques, while data mining tools have very limited abilities for making use of the available rich resources of systems biology web services, databases, ontologies and other resources. In contrast, the Orange4WS (Orange for web services) knowledge discovery platform, where the SegMine methodology was implemented, was constructed by integrating data mining software with the wealth of knowledge and services available on the web, including systems biology resources.

The SegMine methodology and its implementation as reusable Orange4WS workflows are the main scientific contributions of this paper. SegMine allows for holistic interpretation of experimental data in the context of general biological knowledge available in public databases. The experimental results from two microarray datasets (a classical acute lymphoblastic leukemia dataset [29] and a dataset on senescence in mesenchymal stem cells [30]) show that SegMine subsumes the results of a state-of-the-art gene set enrichment tool, and can be instrumental in supporting formulation of new hypotheses.

To summarize, this paper presents a new microarray data analysis methodology and its implementation in a newly developed service-oriented workflow environment. It substantially advances previous work in the areas of microarray data analysis [20,21], link discovery and visualization [17,22], and workflow environments [23-26].

Results and Discussion

This section describes the key results of the presented work. The developed methodology is presented first. Next, the results of the experimental evaluation of the methodology are presented and discussed. Finally, the implementation of the working environment, and the implementation of the methodology itself are described.

The SegMine methodology

The SegMine methodology aids biologists in interpreting microarray data, in finding groups of genes with semantic descriptions, and in discovering links between them. This leads to better understanding of the underlying biological phenomena and may lead to the formation of new hypotheses, based on the experimental data and biological knowledge available in public databases.

The methodology is based on semantic subgroup discovery with the SEGS algorithm, which is complemented by link discovery and visualization using Biomine services. Several additional steps (e.g. hierarchical clustering, ranking of genes) and features (e.g. resolution of gene synonyms, graph coloring) have been implemented to make the proposed methodology operational and more flexible. A schematic overview of the SegMine methodology is presented in Figure 1.

Steps of the SegMine methodology

The SegMine methodology for semantic analysis of microarray data consists of four main steps, which are outlined below. Note that the first two steps can be partially aligned with the general framework for gene set

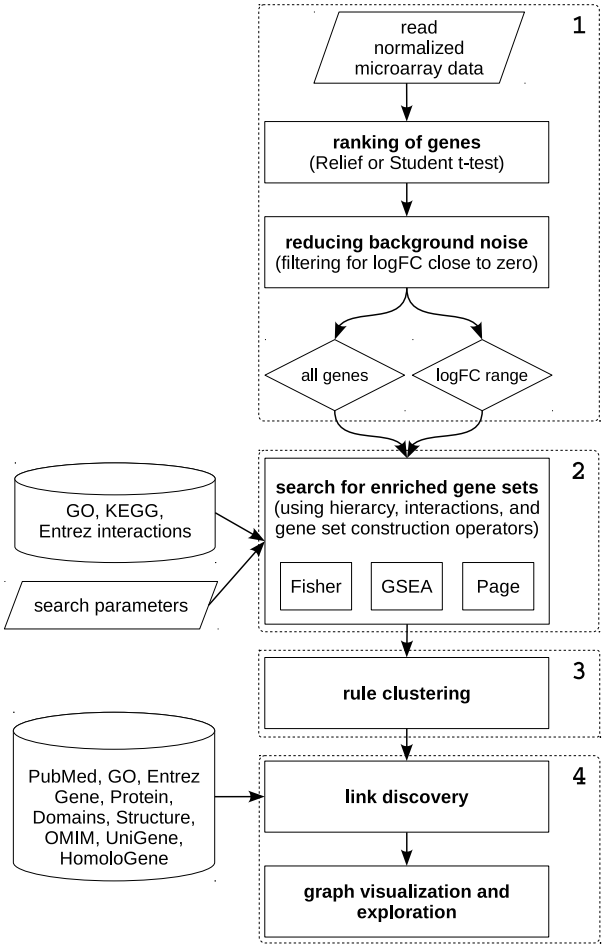

Figure 1 An overview of the SegMine methodology. The four main steps of the SegMine methodology: data preprocessing, Identification of differentially expressed gene sets, clustering of rules describing differentially expressed gene sets, and link discovery, graph visualization and exploration. The data preprocesing step (1) takes normalized microarray data as the input, and results in a ranked list of genes. Identification of differentially expressed gene sets (2) is performed by the SEGS algorithm, which makes use of the GO and KEGG ontologies and Entrez interactions to construct gene sets using SEGS operators, hierarchy information, and solution space search parameters. Rules composed of ontology terms describing gene sets that SEGS found to be statistically significant according to three enrichment tests are sent to the agglomerative hierarchical clustering component (3), which enables grouping of similar and separation of different rules. Finally, link discovery and graph visualization (4) is provided by Biomine, which can perform neighbourhood search as well as search for connections between two query sets. Note that SegMine supports the construction of Biomine queries composed of individual genes, gene sets, ontology terms, rules composed of these terms or even whole clusters.

enrichment analysis as proposed by Ackermann et al. [31].

1. Data preprocessing This step corresponds to the *gene-level statistics* and *transformation* modules of the enrichment analysis framework [31] and is composed of three stages.

In the first stage, SegMine takes class-labeled microarray data that are first loaded and validated as input, and expression fold change (*logFC*) is computed. At this point, different options are available for treating repeated measurements and missing data.

Second, the genes are ranked using the ReliefF [32] algorithm or Student's t-test. Note that other gene-level statistics and methods that result in ranking may also be used, such as fold change, signal-to-noise ratio, correlation networks or Support Vector Machines [31,33-35].

Third, different filtering options can be applied to select a subset of genes. As genes with little variability across samples are often inherently uninteresting, filtering for genes with low $|logFC|$ is generally recommended to reduce background noise. Note that the suitable $|logFC|$ cutoff point needs to be determined for each dataset separately. Finally, separation of up- and down-regulated genes is also supported.

2. Identifying differentially expressed gene sets The second step in the SegMine methodology includes the *gene set statistics* and *significance assessment* steps from [31].

The ranked list of genes generated by step one is used as input to the SEGS algorithm [13], which discovers relevant gene groups, described by logical rules formulated as conjunctions of ontology terms from GO, KEGG and Entrez. The rules semantically explain differentially expressed gene groups in terms of gene functions, components, processes, and pathways as annotated in biological ontologies.

SEGS has four main components: (1) the background knowledge (the GO ontology, KEGG pathways annotations, and Entrez interactions), (2) the SEGS hypothesis language (the GO, KEGG and interaction terms, and their conjunctions), (3) the hypothesis generation and pruning procedure utilizing hierarchy relations and solution space search parameters, and (4) statistical evaluation of the hypotheses. Note that SEGS only makes use of the *is_a* and *part_of* hierarchical relations in GO.

The SEGS algorithm introduces two new operators, *interact()* and *intersect()*, which can lead to discovery of gene sets that cannot be found by any other currently available gene set enrichment analysis software. If S is a gene set and *Entrez* is a database of gene-gene interactions, then the new interacting gene set $INT(S)$ is defined as:

$$INT(S) = \{g : \exists g' \in S : \exists Entrez(g, g')\}, \quad (1)$$

where $Entrez(g, g')$ is a known interaction of genes g and g' from the Entrez gene interaction database. Additionally, let F be a term from the molecular function

branch of the GO ontology, C a term from the cellular component branch, P a term from the biological process branch, and K a KEGG orthology term. Let F', C', P', and K', be the sets of genes annotated by these terms. The new gene set S can then be constructed as the intersection of the sets of annotated genes:

$$S_{F,C,P,K} = \left\{g : g \in \{F\prime \cap C\prime \cap P\prime \cap K\prime\}\right\} \quad (2)$$

The constructed gene sets that are found to satisfy the specified solution space search parameters must be tested for potential enrichment. Currently, SEGS incorporates three different tests commonly used in gene set enrichment analysis: Fisher's exact test, the GSEA method, and parametric analysis of gene set enrichment (PAGE).

The p-values of all three methods may be combined into a single value by taking into account user-defined weights, according to the following formula, which allows for controlling preferences for enrichment tests:

$$p = \frac{\sum w_i * p_i}{\sum w_i} \quad (3)$$

Note that the aggregate p-value is not the p-value in the classical sense but is only used to identify gene sets that have small p-values on several tests.

The significance of gene sets is assessed using permutation testing, but other methods for correcting p-values for multiple hypothesis testing, such as Bonferroni correction or false discovery rate (FDR), can be applied.

3. Rule clustering The aim of the third step is to reduce the complexity of the results produced by SEGS. Often, several groups of rules found by the SEGS algorithm are composed of very similar gene sets rendering the analysis more difficult due to duplicate information.

Therefore, SegMine incorporates interactive agglomerative hierarchical clustering of SEGS rules to simplify the exploration of large sets of rules, and to provide a natural summarization of the results. Hierarchical clustering of rules is performed according to the similarity of gene sets that are found to be significantly enriched. Several different metrics are available for the computation of similarities, for example, Euclidean, Manhattan, Relief and Hamming. Additionally, agglomerative hierarchical clustering (provided by Orange), supports various linkage criteria for computing clusters including Ward's linkage, complete linkage, single linkage, and average linkage.

4. Link discovery and graph visualization The last step of the SegMine methodology is provided by the Biomine system, which incorporates several public databases into a single large graph. Biomine implements advanced probabilistic graph search algorithms that can discover the parts of the graph most relevant to the given query. An important integral part of Biomine is the interactive graph visualization component, which supports one click links to the original data sources.

In the Biomine graph data model, nodes of the graph correspond to different concepts (such as gene, protein, domain, phenotype, biological process, tissue), and semantically labelled edges connect related concepts (e. g. gene BCHE encodes protein CHLE, which in turn has the molecular function 'beta-amyloid binding'). The main goal of Biomine is to enable the discovery of new, indirect connections between biological concepts. Biomine evaluates, extracts and visualizes connections between given nodes.

All components of the results from steps 1-3 can be used to formulate queries to the Biomine link discovery engine. SegMine supports the construction of queries composed of individual genes, gene sets, terms from the GO ontology, KEGG pathways, rules composed of these terms, or even whole clusters of gene sets, which are then sent to the Biomine query engine. Biomine is able to find a connecting subgraph between these elements using other entities from a number of public biological databases including Entrez Gene, UniProt, Gene Ontology, OMIM, NCBI HomoloGene, InterPro, STRING, and KEGG pathways. Links in such subgraphs help biologists to uncover unexpected indirect relations and biological mechanisms potentially characteristic for the underlying biological system. Moreover, subgraphs produced by Biomine also present known biological facts, relations, and literature citations in an organized and structured way. Finally, Biomine allows addition of experimental results (e.g., gene expression *logFC* values) to subgraphs, which facilitates the interpretation of discovered links in the context of experimental results.

Experiments

This section presents two applications of the proposed methodology and its implementation with experimental microarray data. Two microarray datasets were used for the validation and evaluation of the SegMine methodology: a well-known dataset from a clinical trial in acute lymphoblastic leukemia (ALL), and a dataset about senescence in human mesenchymal stem cells (MSC).

Acute lymphoblastic leukemia

The aim of the first experiment was to validate the SegMine methodology, and perform a comparative analysis of the results using the well-known DAVID [36,37] tool (**D**atabase for **A**nnotation, **V**isualization and **I**ntegrated **D**iscovery). Because DAVID does not provide probabilistic search in large graphs that is provided in SegMine through Biomine services, only the results of the _rst step of the SegMine methodology, namely the sets of

differentially expressed genes found by the SEGS algorithm, were used in the comparison.

Experimental setup

Comparative analysis was performed on a well-known dataset from a clinical trial in acute lymphoblastic leukemia (ALL) [29], which is a typical dataset for medical research, with several samples available for each class (95 arrays for B-type cells and 33 arrays for T-type cells). This dataset serves as an appropriate reference for such evaluations, as it has also been a model dataset for other microarray data analysis tools [8-10,38,39].

In order to enable direct comparison of the results both tools were set to use the same parameters. The GO ontology, KEGG pathways and Entrez gene-gene interaction database (note that the BIND interaction database was used in DAVID, as DAVID is not able to use Entrez). were used as the background knowledge.

In DAVID, the broadest ontology terms were filtered using the GO FAT filter which attempts to filter the broadest terms (term specificity is based on the number of child terms). On the other hand, a manually created list of terms was used in SegMine.

The top 1000 ranked genes from the data were provided as the input while the remainder (8001) were treated as the background. The resulting enriched terms from DAVID and rules of terms from SegMine were filtered according to the corrected p-value of 0.05. Using DAVID, p-values are obtained with Fisher's exact test and Bonferroni correction. The p-values in SegMine are aggregated by combining p-values of Fisher's exact test, PAGE and GSEA methods, which are corrected using permutation testing. All weights for the aggregation of p-values were equal in our experiments.

As shown in Figures 2 and 3, thirteen terms obtained by DAVID remained after p-value filtering. On the other hand, using SegMine, more rules of terms were found, although only the top 100 are shown. The gene sets covered by DAVID and SegMine were compared using the following formula:

$$v_{i,j} = \frac{|S_i \cap D_j|}{|D_j|} \tag{4}$$

where S_i is the set of genes covered by the i-th SegMine rule, and D_j is the set of genes covered by the j-th DAVID term, respectively.

The values $v_{i,j} \in [0,...,1]$ indicate how well the j-th DAVID term is covered by i-th SegMine rule. Note that the exclusion of general terms in SegMine is of key importance for the validity of this measure. If some general terms were found to be enriched by SegMine, according to the above formula they could completely cover gene sets found by DAVID.

Both DAVID and SegMine identified similar enriched gene sets describing differences in gene expression between B-ALL and T-ALL cells, such as lymphocyte differentiation and activation, cell adhesion molecules and KEGG processes in which lymphocyte-specific genes play a major role. Almost all significantly enriched DAVID gene sets were covered by one or more SegMine rules, with the exception of gene set 6 (*lipid biosynthetic process*), which was covered only partially by several SegMine rules (see Figures 2 and 3). The main advantage of the results produced by SegMine is that by combining ontology terms the description of the regulated process is more specific. Many GO terms that were found as enriched by DAVID appear several times in the result of SegMine in conjunction with interacting gene sets.

For example, *lymphocyte differentiation* from the GO ontology appears in 17 SegMine rules in conjunction with different GO and KEGG terms. Such rules can be interpreted as an enrichment of a gene set that includes not only genes described by the first term (*lymphocyte differentiation*) but also interacting genes described by the second term, for example, *Fc gamma R-mediated phagocytosis*.

Additionally, several gene sets obtained by SegMine were not identified by DAVID (Figure 2), for example, rules 25, 33, 41 and 43, which describe *positive regulation of lymphocyte activation* interacting with *peptide binding*, *leukocyte activation* interacting with *T cell receptor complex*, *positive regulation of leukocyte activation* interacting with *phosphoprotein binding* and *positive regulation of leukocyte activation* interacting with *peptide binding*. These rules suggest a different regulation of a set of receptor-interacting genes (or gene products) in the two different lineages of ALL cells.

The comparison shows that SegMine is able to discover the same biological knowledge as DAVID. Moreover, SegMine provides more expressive results in the form of rules, that is, conjunctions of terms. Such rules describe gene sets that are more specific than gene sets reported from other gene set enrichment analysis tools such as DAVID (see Figures 2 and 3), and therefore more suitable for generation of new (more specific) hypotheses. They are evaluated with not only one, but three enrichment tests. Also, the corrected p-values of the available tests can be combined into a single, aggregated value by specifying custom weights controlling user preferences for different gene set enrichment tests.

Senescence in stem cells

In the second experiment SegMine was applied to the analysis of senescence in human mesenchymal stem cells (MSC). To date, the underlying molecular mechanisms or candidate marker genes that could reflect a

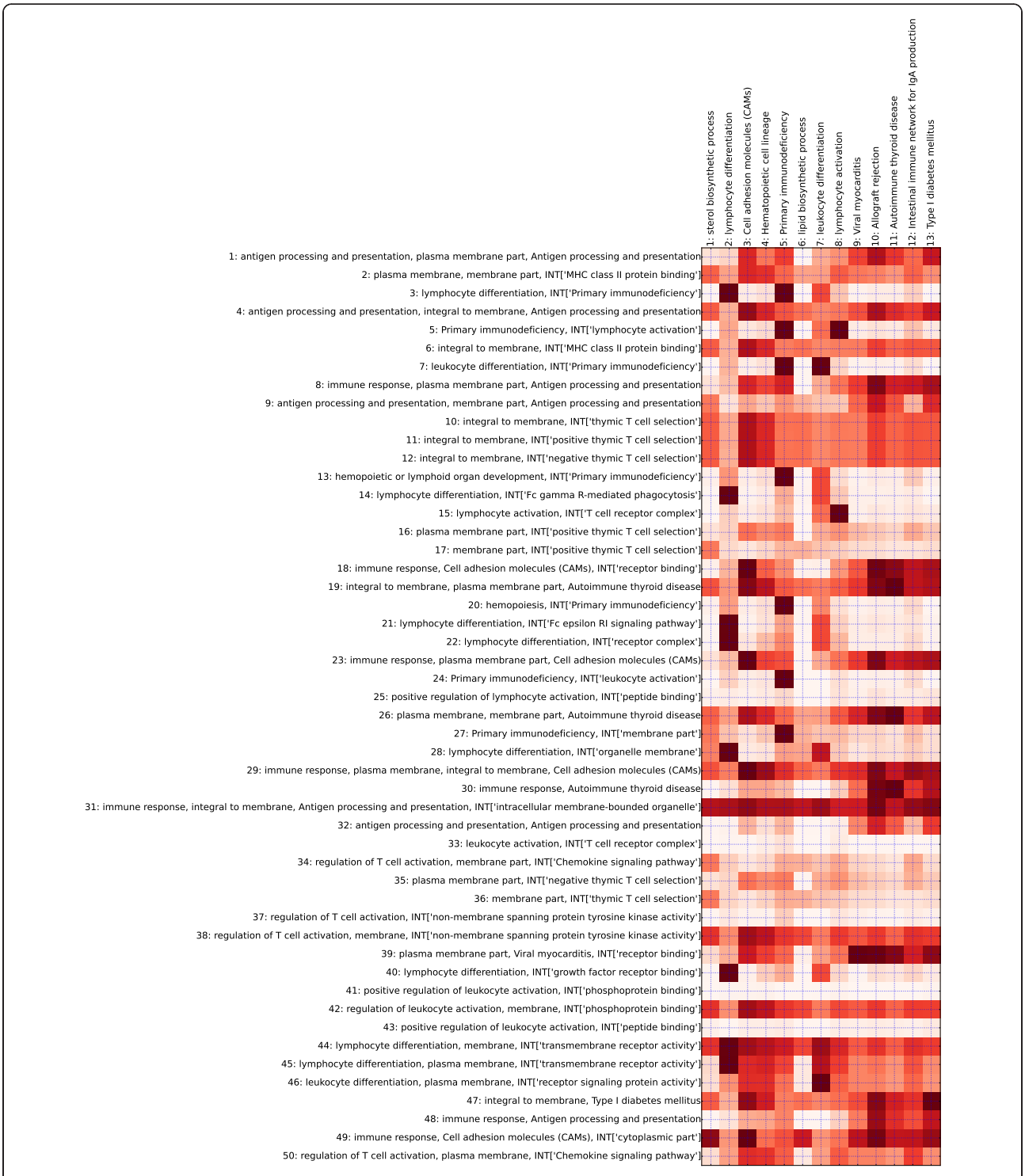

Figure 2 Comparison of SegMine and DAVID. The first part of the comparison of the results of SegMine and DAVID on the ALL dataset. Columns are terms found to be enriched by DAVID, while rows are rules of terms found to be enriched by SegMine. Only the first half of the 100 rules of terms obtained by SegMine is shown. All results are statistically significant with $p \leq 0.05$. Darker red shades of matrix cells indicate higher overlapping of corresponding gene sets. Note that rows of the matrix that consist of lightly shaded cells represent gene sets identified as significantly enriched by SegMine but not by DAVID, e.g., 25, 33, 41 and 43.

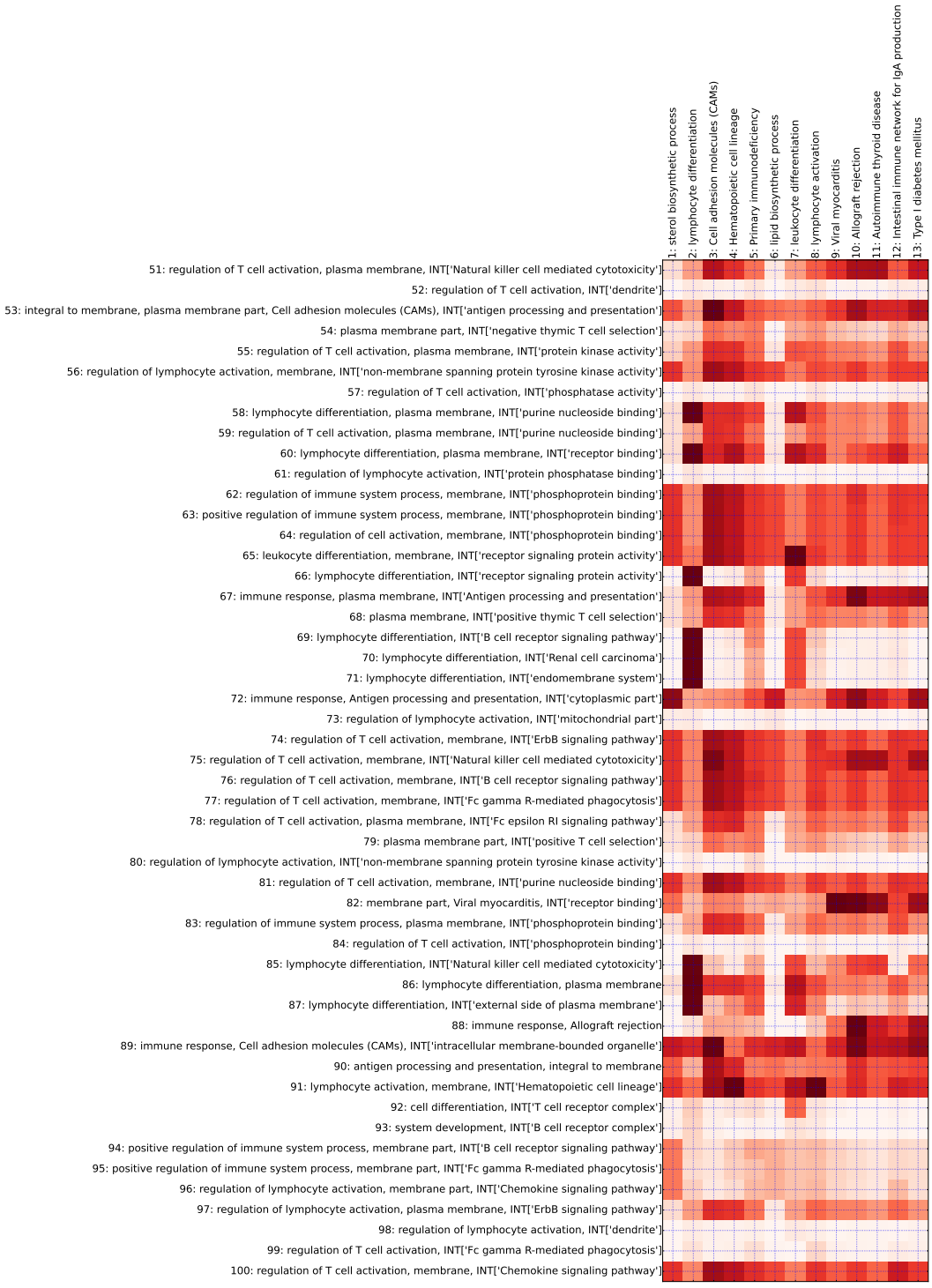

Figure 3 Comparison of SegMine and DAVID. The second part of the comparison of the results of SegMine and DAVID on the ALL dataset. Columns are terms found to be enriched by DAVID, while rows are rules of terms found to be enriched by SegMine. The second half of the 100 rules of terms obtained by SegMine is shown. All results are statistically significant with $p \leq 0.05$. Darker red shades of matrix cells indicate higher overlapping of corresponding gene sets. Note that rows of the matrix that consist of lightly shaded cells represent gene sets identified as significantly enriched by SegMine but not by DAVID, e.g., 57, 61, 73, 80, 84, and 98.

degree of cellular aging in MSC are still not known or explained. However, the increasing use of MSC as cellular therapeutics necessitates standardized isolation and reliable quality control assessment of cell preparations. Therefore, we focused on the analysis of a dataset where gene expression profiles from late senescent passages of MSC from three independent donors were compared to the MSC of early passages [30]. We were able to formulate three novel research hypotheses that could improve understanding of mechanisms in senescence and identification of candidate marker genes. One of our hypotheses, derived from the 2008 dataset, may even substantiate a recent proposition independently derived from additional senescence gene expression data [30,40] in 2010. Even though the hypotheses still need to be verified by additional laboratory experiments these results confirm that SegMine is a very useful tool for exploratory analysis of gene expression data and formulation of new research hypotheses.

Several analyses of microarray data from senescent cells have already been performed [30,40,41]. In these analyses, the senescence candidate marker genes were typically drawn from a list of top differentially expressed genes, that is, their selection depended mainly on their gene expression (*logFC* and p-values). In contrast, SegMine also considers functional properties, as well as direct and indirect connections to related genes and proteins. We have taken the following SegMine steps to analyze the MSC data, published by Wagner et al. [30]:

1. all regulated genes that have absolute *logFC* values lower than 0.3 were filtered out,
2. only SegMine rules with the corrected p-value $p \leq 0.05$ were considered,
3. hierarchical clustering of rules using Ward linkage criteria was used to produce nine rule clusters (Figure 4),
4. several Biomine queries between the source (clusters 1, 2, 3) and target (cluster 9) genes were formulated,
5. the resulting Biomine subgraphs were thoroughly inspected prior to focusing on (a) *gene hubs* (nodes with a large number of edges) where the majority of edges were of the type *interacts with*, and (b) *outlier genes*, which are represented with nodes having few edges with very low weights, or isolated nodes (see Figures 5 and 6).

First we turned attention to the gene enrichment and clustering of rules (steps 2 and 3 above). Comparing these to the originally published results [30], we noticed that our results lack rules annotated with cytoskeletal parts, vacuole or lysosome terms, which had a low number of genes annotated to them in the original study. These compartments are obviously not recognized as important by SegMine. On the other hand, SegMine analysis revealed annotations that were strongly overrepresented in Wagner's analysis. We believe that these processes (cell cycle, DNA metabolism and chromatin organization) are indeed crucial for senescence progression. Wagner's group recently approached the same set of senescence associated gene clusters [40] with an improved analysis of additional senescence gene expression data. Similarly to the SegMine clusters, their recent publication does not mention the above unimportant compartments that appeared in [30].

The nine clusters of rules produced in step 3 were further analyzed to find links between distant clusters (step 4 above). In particular, Biomine was queried to discover links between genes from the source clusters 1-3, and the genes of the target cluster 9, respectively. Subgraphs, discovered by Biomine in step 5 were carefully inspected, and the following gene hubs were identified:

1. BRCA1 and SMAD2 genes from the cluster 1 vs. cluster 9 query (Figure 5).
2. SMAD1, SMAD2 genes, and SMARCD1, SMARCE1 genes from the cluster 2 vs. cluster 9 query.
3. MCM10 gene from the cluster 3 vs. cluster 9 query (Figure 6).

Four identified gene hubs (BRCA1 - breast cancer 1, early onset; SMAD2 - SMAD family member 1; SMARCD1 - SWI/SNF related, matrix associated, actin dependent regulator of chromatin, subfamily D, member 1; and MCM10 - minichromosome maintenance complex component 10) were evaluated for the presence of direct links to previously published senescent candidate marker genes. We found some of the senescent candidate marker genes, STAT1 [30], MCM3 [40], H2AFX, AURKA [41], RAD50, and MRE11 [42], to be linked (by the *interacts with* edge) to the BRCA1 gene hub (see Figure 5). Likewise, MCM3 and MCM6 [41] were found to be linked to the MCM10 gene hub (see Figure 6). None of those already identified (patented) senescence candidate marker genes could be recognized as a gene hub by SegMine analysis, as they all had only a limited number of direct links to other genes/proteins. Moreover, a published senescence candidate marker gene SFPQ [41] was even identified as an outlier gene, without any direct link with sufficiently high weight to be present in the Biomine subgraph.

It can be hypothesized that the gene hubs (BRCA1, MCM10, SMAD2, SMARC) identified by SegMine may represent additional senescent candidate marker genes. The results also show that the expression fold difference

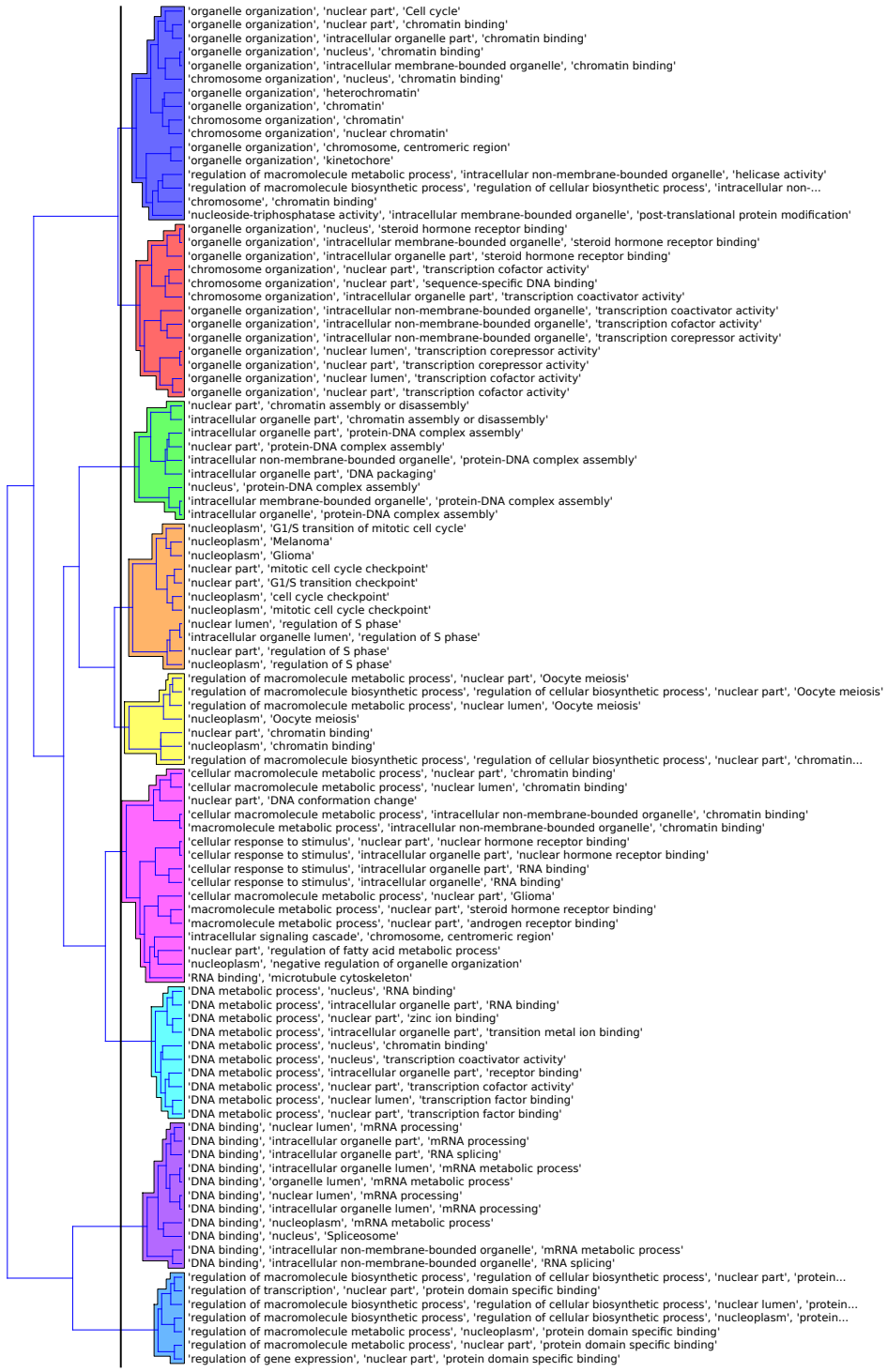

Figure 4 Hierarchical clusters of rules for the MSC dataset. Hierarchical clustering of the top 100 statistically significant rules ($p \leq 0.05$). SegMine rules were obtained from a dataset of senescence in human mesenchymal stem cells (MSC dataset). Euclidean distance and Ward's linkage criteria were used to compute the hierarchy.

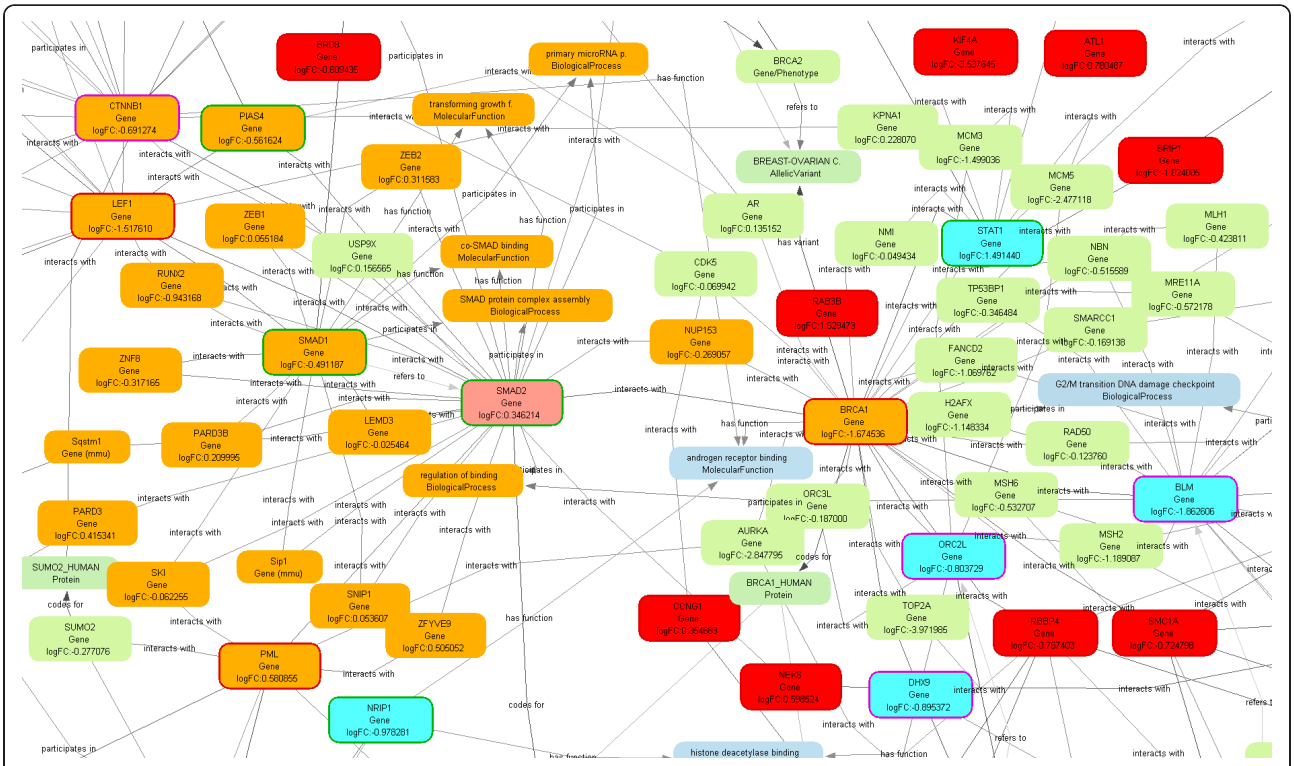

Figure 5 A SegMine subgraph, where a strongly linked gene hub BRCA1 was identified. A part of the subgraph obtained by link search between two clusters of rules describing differentially expressed gene sets. All genes covered by the rules from clusters 1 and 9 of hierarchical clustering obtained from the MSC dataset were used. The subgraph allows for the Identification of BRCA1 and SMAD2 gene hubs. Some of the senescent candidate marker genes such as STAT1, MCM3, H2AFX, AURKA, RAD50, MRE11 are linked to the identified BRCA1 gene hub.

of genes in gene hubs is not necessarily the highest. We believe that even small expression changes in SMARCD1, SMARCE1 and SMAD2 gene hubs may nonetheless have quite a substantial impact on the process of replicative senescence.

This assumption was confirmed by a literature survey for biological functions of gene hubs identified by SegMine. MCMs, including our newly identified MCM10, have long been known to regulate DNA synthesis by replicative fork formation and to influence proliferation during cells' progression toward senescence, when their expression is switched off [43]. Even BRCA1, a tumor suppressor notorious for its mutation-associated development of certain types of tumors, was recently found to be associated with replicative senescence. The confirmed inactivation of the BRCA1 pathway in MSC was found to reduce their long-term proliferation ability and increase senescence associated beta-galactosidase activity [44]. This functional involvement of the BRCA1/2 and RAD50/MRE11 in replicative senescence, implicated first by Lansdorp [42], was now confirmed also by our SegMine analysis. Additionally, SegMine identified SMADs gene hubs (signal transducers and transcriptional modulators), including SMAD2, which has been known to mediate the TGFb signalling pathway [45] involved in the long-term MSC cultivation resulting in doubling time increase and senescence. Besides senescence, SMADs involved in the TGFb pathways were confirmed to regulate adipogenesis [45]. Similarly, potential involvement of SMARCs genes in adipogenesis was confirmed by profiling of mature differentiated adipocytes vs. proliferatively active adipoblast [46].

As a consequence, we believe that our four identified gene hubs may represent even better senescent gene markers than the patented cell quality markers identified solely by their high expression difference in senescent cells and which were nevertheless found to be connected to our gene hubs (note that only those gene hubs that have edges of type *interacts with* with high probabilities were selected and displayed). Furthermore, SegMine allows visualization of links between genes, enabling clear and easy identification of top processes influencing cellular senescence. Lastly, least, the identification of gene hubs, not necessarily the ones with highest differential expression, allowed us to formulate three new hypotheses (which have yet to be confirmed in future laboratory experiments).

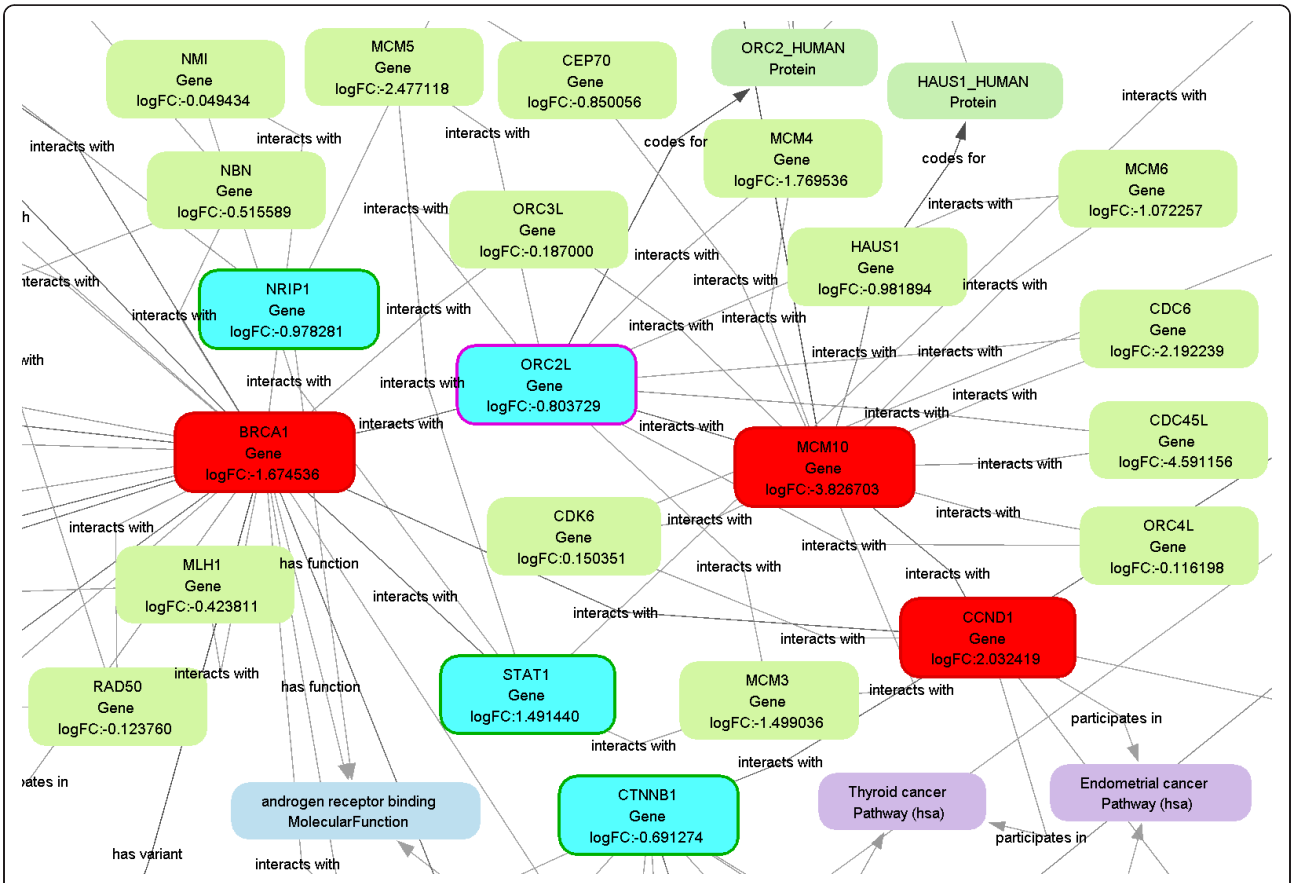

Figure 6 A SegMine subgraph with the MCM10 gene identified as a gene hub. A part of the subgraph obtained by link search between two clusters of rules describing differentially expressed gene sets. All genes covered by the rules from clusters 3 and 9 of hierarchical clustering obtained from the MSC dataset were used. The subgraph allows for the Identification of the MCM10 gene hub which is linked to MCM3 and MCM6 senescent candidate marker genes.

Hypothesis 1: Progression to senescence protects cells from entering tumorigenic transition

This hypothesis is Wagner's recent original proposition substantiated with our SegMine results. It was proposed [47] that a central pathway in senescence might provide a purposeful program to protect the organism from tumorigenesis by somatic cells that have accumulated DNA mutations after a certain number of cell divisions. We believe that an additional piece of evidence was revealed to support this hypothesis. Besides the known fact that senescence is not an inevitable fate for all cells, we identified a novel senescence candidate marker gene, the BRCA1 gene hub. The fact that BRCA1 has so far been recognized mostly in tumor development provides additional support for this hypothesis.

Hypothesis 2: The Low quality of adipose tissue derived MSC is due to their enhanced tendency to senesce

This hypothesis speculates on a cause for the low quality of adipose derived MSC reported by numerous labs worldwide. Fat derived MSC cease to proliferate and begin to senesce quite early, sometimes even immediately after isolation. SMAD and SMARC gene hubs, identified by SegMine, were all proven in the past to be deregulated during adipogenic differentiation [45,46]. Yet in our analysis they appear also to be over-represented and deregulated in senescent cells; thus we assume that genes of the senescence pathway are most likely involved in adipose tissue homestasis as well. This hypothesis would explain why MSC isolated from the adipose tissue display enhanced permissiveness to senescence upon isolation, as compared to MSC derived from any other tissue.

Hypothesis 3: Autophagy may help cells to transiently override their commitment to senesce

Several genes from intracellular protein trafficking and autophagy (MAP1B, LYST, BECN1) were identified by SegMine as outlier genes. When used in Biomine queries they appeared in Biomine subgraphs as nodes with no edges or with edges having very low weights, meaning that knowledge about their links to other genes/proteins is not readily available. However, as cells use autophagy to overcome cell damage or nutrient

deprivation, this hypothesis is worth exploring, especially in the light of the SEGS clustering, which on the basis of gene-gene interactions already associated those genes into clusters.

While the above three hypotheses will need to be explored in laboratory experiments to validate their likelihood as contributing factors, the authors believe that SegMine's primary contribution is in providing a unique exploratory environment that allows new hypotheses to be formulated.

Implementation

In this section we discuss the implementation of the workflow environment named Orange4WS, and the implementation of the methodology itself. Note that the presented implementation of the SegMine methodology is only an example, as it can be implemented very differently in a different environment.

The Orange4WS data mining platform

The service-oriented data mining platform Orange4WS is an easy-to-use software tool that enables creation and execution of scientific workflows. It is built on top of two open-source projects:

- the Orange data mining framework [26] and
- the Python Web Services project [48].

Orange provides a range of preprocessing, modeling, and data exploration and visualization techniques as well as a user-friendly workflow execution environment. The Python Web Services project enables employment and development of web services using the Python programming language by implementing various protocols and formats including XML [49], SOAP [50] and WSDL [51].

In contrast to other freely available data mining workflow environments such as Weka, Taverna, Triana, KNIME and RapidMiner, the Orange4WS framework offers a unique combination of features: (a) a large collection of data mining and machine learning algorithms, efficiently implemented in C++; (b) a three-layer architecture: C++, Python, and interactive workflows; (c) a collection of very powerful yet easy-to-use data visualization components; (d) incorporation of propositional as well as selected relational data mining algorithms, and (e) simplicity of workflow composition.

As a result, Orange4WS provides a service-oriented data mining software platform, ready to be used for any task requiring data mining algorithms, web services, workflows, complex visualization, rapid prototyping, and other knowledge discovery scenarios. In comparison with the well known Taverna workbench, Orange4WS integrates a complete data mining environment (Orange) with a wide range of machine and data mining algorithms and visualization methods, as well as the ability to use web services and rapid prototyping in Python. Orange4WS offers a high level of abstraction when composing workflows, which contributes to their understandability and simplicity. Finally, Orange4WS also integrates a general knowledge discovery ontology and a planner enabling automated composition of data mining workflows, although this topic is beyond the scope of the work presented here, and therefore will not be discussed. Finally, Orange4WS also enables automated composition of data mining workflows by integrating a general knowledge discovery ontology and a planner, although this topic is beyond the scope of the work presented here, and therefore will not be discussed.

Composition and execution of workflows

One of the most important features of Orange, also inherited by Orange4WS, is an easy-to-use interactive workflow construction that is supported by the *Orange Canvas*, an interactive graphical user interface component.

It enables graphical construction of workflows by allowing interactive workflow elements called *Orange Widgets* to be positioned in a desired order, connected with lines representing flow of data, adjusted by setting their parameters, and finally executed. For example, Figure 7 shows the Orange4WS environment running a workflow of SegMine components (widgets).

The workflow management component enables or disables the connectivity of inputs and outputs according to their types. It also prevents the user from creating loops while connecting widgets by detecting cycles in the corresponding directed graph. If a widget supports the adjustment of its parameters, this can be done from the widget's user interface, which also enables data and results visualization as well as other interactive features. Finally, a constructed workflow can be saved into an XML format that corresponds to a predefined XML schema. This ensures repeatability of scientific experiments as well as support for user collaboration.

Orange4WS offers support for SOAP as well as RESTful web services, which can be used as workflow components. It provides modules that enable:

- loading web service consumer code,
- extracting information about web service input and output data types,
- fully automatic creation of widgets (workflow components) from web services, and
- support for creation of SOAP web services from existing software and algorithm implementations.

When successfully imported, a web service can be used as a normal Orange4WS widget. As a result,

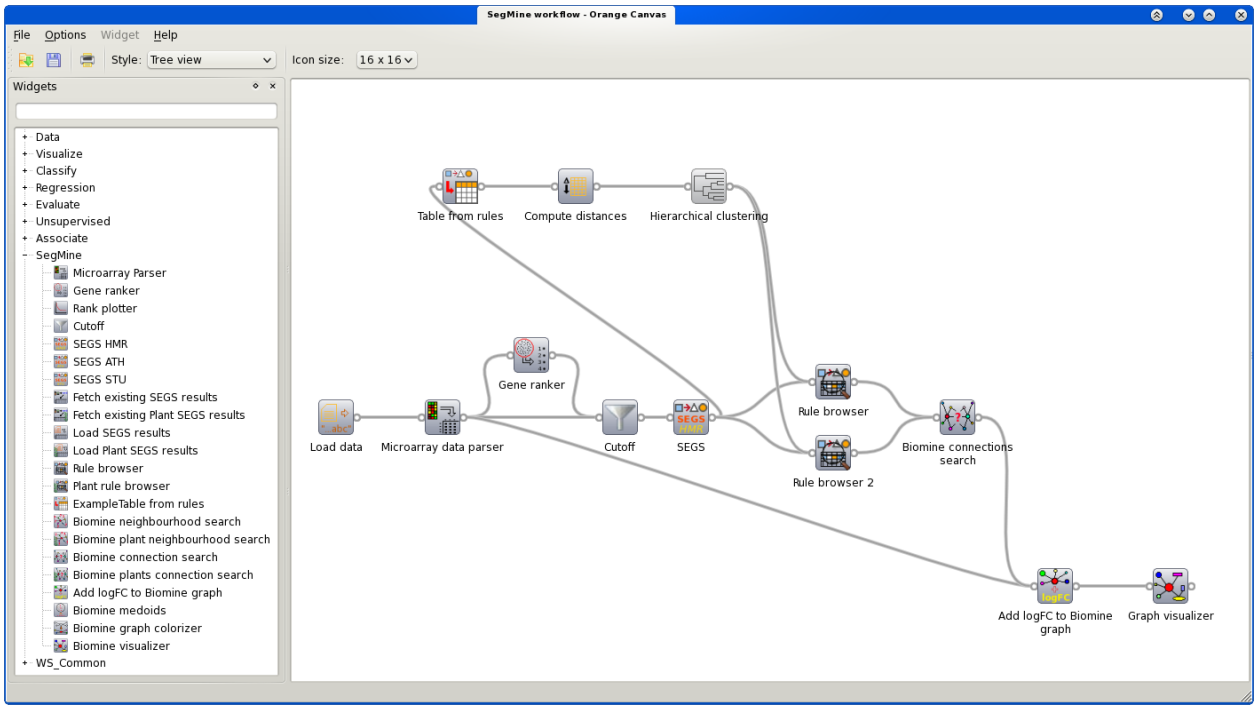

Figure 7 Orange4WS environment. A screenshot of Orange4WS running a workflow of SegMine components. The workflow exploits all four main components of SegMine: loading and preprocessing the data, search for enriched gene sets, hierarchical clustering, and link discovery and visualization. All available SegMine, as well as Orange4WS and Orange workflow components, are accessible by clicking on the corresponding items in a tree view on the right.

Orange4WS enables access to public databases such as PubMed, the BioMart project [52], EMBL-EBI data resources and analysis tools [53], life science web services indexed by BioCatalogue [54], etc.

SegMine as Orange4WS workflows

We have implemented the SegMine methodology as a collection of Orange4WS workflow components. According to the four steps of the methodology these components can be divided into four groups: (1) data preprocessing, (2) identification of enriched gene sets, (3) rule clustering, and (4) link discovery and visualization.

Data preprocessing

The following data preprocessing workflow components (widgets) are available: loading of microarray data from a text file, parsing of microarray data into an internal versatile data structure used by Orange and Orange4WS, resolution of gene synonyms according to the gene data provided by NCBI, ranking of genes using ReliefF algorithm or t-test, loading of precomputed gene ranks from a text file, plotting of gene ranks, and cutoff of ranked genes according to the *logFC* values.

Identification of differentially expressed gene sets

Our SegMine implementation offers the following widgets that enable and support identification of differentially expressed gene sets: the SEGS algorithm for different organisms (the current version supports *Homo sapiens*, *Mus musculus* and *Rattus norvegicus* experimental data by integrating corresponding annotations to the ontologies), which is available as a fully SOAP 1.1 compatible web service ready to be used in any service-oriented software supporting SOAP web services, e.g. The Taverna Workflow Management System; a rule browser component, which provides an HTML table visualization where the results are linked to the original data sources; construction of Orange's native data structure from the results of SEGS, which enables the use of data mining techniques and algorithms on the obtained enriched gene sets, and loading and saving the results of SEGS into local files and fetching the results from the server where the SEGS web service is currently running.

Rule clustering

Clustering of SEGS rules is provided by the widget for computing distances between rules using different metrics, and the hierarchical clustering widget, which provides different linkage criteria and supports interactive cluster assignment and visualization (see Figure 4). The rule browser component also links the rules to their clusters and provides unions as well as intersections of gene sets in each cluster.

Link discovery

The presented implementation offers several components that enable link discovery using Biomine services. First, it provides widgets for neighborhood and connections search as well as search for medoids in a group of genes, all of which query the Biomine web service using the JSON protocol. Second, it integrates the Biomine graph visualization component, which is run locally from Orange4WS as a Java applet. Finally, it implements widgets for adding information about gene expression values, and for coloring selected nodes in Biomine graphs.

Conclusions

This paper presents SegMine, a methodology for microarray data analysis combining cutting-edge data analysis approaches, such as semantic data mining, link discovery and visualization.

The methodology is implemented in reusable workflows within a new service-oriented data mining platform, Orange4WS. Providing a novel approach to the exploration of microarray datasets in the context of general knowledge is a step beyond the existing state-of-the-art transcriptomic analysis tools. The developed platform is flexible, enabling easy adaptation to the investigated dataset through different filtering options, through different SEGS and Biomine settings, and through different combinations of analysis and visualization tools. The advanced options additionally enable cross-domain link discovery, thus rendering the interpretation of the biological mechanisms underlying differential gene expression understandable to life scientists.

Novel hypotheses, based on the SegMine analysis of MSC microarray data, were presented. We confirmed the strength of SegMine's exploratory analysis, which links the deregulated genes to other related genes/proteins, and this was further supported by literature survey. We were able to formulate three novel research hypotheses that improve understanding of the underlying mechanisms in senescence and identification of candidate marker genes. This may pave the way to a reliable, functionally confirmed panel of senescence marker genes, which can be used as molecular signatures to distinguish between senescent and normal high quality MSC. Such specification of senescence-associated candidate marker genes, functionally evaluated and cross-validated in different MSC preparations, may ultimately result in more reliable quality control of cell preparations, which are increasingly used in cell based therapies.

In the future the presented work will be extended at several levels. While the SegMine methodology is fairly complete, it only provides means for the analysis of genomics data; we plant to extend the methodology to other types of omics data, such as proteomics and metabolomics. The Biomine system currently employs only basic text mining techniques, which will be improved and complemented with natural language processing tools in order to obtain more structured data from textual data sources such as open-access article databases. The SegMine implementation in Orange4WS will be extended with additional components such as visualization of enriched ontology terms similar to the one provided by the GOrilla tool [55]. The Orange4WS workflow environment will also be subject to improvements in order to adapt to the extensions of the methodology, and to improve the support for the publicly available systems biology web services and data and knowledge sources.

Availability

The Orange4WS platform is available at http://orange4ws.ijs.si.

Our reference implementation of the SegMine methodology for Orange4WS is available at http://segmine.ijs.si.

Acknowledgements

The work presented in this paper was supported by the European Commission under the FP7-ICT-2007-C FET-Open project BISON-211898, by the Slovenian Research Agency grants P2-0103, J4-2228, P4-0165, Slovenian Ministry of Higher Education, Science and Technology grant No. 4302-38-2006-4 and by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland.

Author details

[1]Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia. [2]University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia. [3]Ss. Cyril and Methodius University, 1000 Skopje, Macedonia. [4]University of Helsinki, P.O. Box 68, FI-00014 Helsinki, Finland. [5]National Institute of Biology, Večna pot 111, 1000 Ljubljana, Slovenia.

Authors' contributions

VP has implemented Orange4WS and SegMine, and contributed to the manuscript. NL and IM conceived and coordinated the computer science aspects of the study and contributed to the manuscript. PKN originated the idea of connecting SEGS and Biomine. IT parallelized and adapted SEGS to be incorporated in SegMine. LL implemented the gene medoids algorithm. KK implemented the Biomine visualizer. HT conceived and coordinated development of Biomine. MP participated in comparison of SegMine and DAVID on the ALL dataset. HM performed an expert analysis of the MSC dataset and formulated new hypotheses. KG coordinated the biological aspects of the study and contributed to the manuscript. All the authors have read and approved the final manuscript.

Received: 31 May 2011 Accepted: 26 October 2011
Published: 26 October 2011

References

1. Hiroaki K: *Foundations of systems biology* MIT Press; 2001.
2. Schena M, Heller RA, Theriault TP, Konrad K, Lachenmeier E, Davis RW: **Microarrays: biotechnology's discovery platform for functional genomics.** *Trends in biotechnology* 1998, **16(7)**:301-306.
3. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A,

Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G: **Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.** *Nature genetics* 2000, **25**:25-29.
4. Man MZ, Wang X, Wang Y: **POWER_SAGE: comparing statistical tests for SAGE experiments.** *Bioinformatics* 2000, **16(11)**:953-959.
5. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP: **Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles.** *Proceedings of the National Academy of Sciences of the United States of America* 2005, **102(43)**:15545-15550.
6. Subramanian A, Kuehn H, Gould J, Tamayo P, Mesirov JP: **GSEA-P: a desktop application for Gene Set Enrichment Analysis.** *Bioinformatics* 2007, **23(23)**:3251-3253.
7. Kim SY, Volsky DJ: **PAGE: Parametric Analysis of Gene Set Enrichment.** *BMC Bioinformatics* 2005, **6**:144.
8. Jiang Z, Gentleman R: **Extensions to gene set enrichment.** *Bioinformatics* 2007, **23(3)**:306-313.
9. Falcon S, Gentleman R: **Using GOstats to test gene lists for GO term association.** *Bioinformatics* 2007, **23(2)**:257-258.
10. Nettleton D, Recknor J, Reecy JM: **Identification of differentially expressed gene categories in microarray studies using nonparametric multivariate analysis.** *Bioinformatics* 2008, **24(2)**:192-201.
11. Zhang S, Cao J, Kong YM, Scheuermann RH: **GO-Bayes: Gene Ontology-based overrepresentation analysis using a Bayesian approach.** *Bioinformatics* 2010, **26(7)**:905-911.
12. Khatri P, Draghici S: **Ontological analysis of gene expression data: current tools, limitations, and open problems.** *Bioinformatics* 2005, **21(18)**:3587-3595.
13. Trajkovski I, Lavrac N, Tolar J: **SEGS: Search for enriched gene sets in microarray data.** *Journal of Biomedical Informatics* 2008, **41(4)**:588-601.
14. Kanehisa M, Goto S: **KEGG: Kyoto Encyclopedia of Genes and Genomes.** *Nucleic Acids Research* 2000, **28**:27-30.
15. Maglott D, Ostell J, Pruitt KD, Tatusova T: **Entrez Gene: gene-centered information at NCBI.** *Nucleic Acids Res* 2005, **33 Database**.
16. Petra Kralj Novak IT, Anže Vavpetič Lavrač N: **Towards semantic data mining with g-SEGS.** *Proceedings of the 13th International Multiconference INFORMATION SOCIETY (IS 2010), Volume A* 2010, 173-176.
17. Köhler J, Baumbach J, Taubert J, Specht M, Skusa A, Rüegg A, Rawlings CJ, Verrier P, Philippi S: **Graph-based analysis and visualization of experimental results with ONDEX.** *Bioinformatics* 2006, **22(11)**:1383-1390.
18. Matthews L, Gopinath G, Gillespie M, Caudy M, Croft D, de Bono B, Garapati P, Hemish J, Hermjakob H, Jassal B, Kanapin A, Lewis S, Mahajan S, May B, Schmidt E, Vastrik I, Wu G, Birney E, Stein L, D'Eustachio P: **Reactome knowledgebase of human biological pathways and processes.** *Nucleic Acids Research* 2009, , **37 Database:** 619-622.
19. Caspi R, Altman T, Dale JM, Dreher K, Fulcher CA, Gilham F, Kaipa P, Karthikeyan AS, Kothari A, Krummenacker M, Latendresse M, Mueller LA, Paley SM, Popescu L, Pujar A, Shearer AG, Zhang P, Karp PD: **The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases.** *Nucleic Acids Research* 2010, , **38 Database:** 473-479.
20. Theocharidis A, van Dongen S, Enright AJ, Freeman TC: **Network visualization and analysis of gene expression data using BioLayout Express(3D).** *Nature protocols* 2009, **4(10)**:1535-1550.
21. Usadel B, Nagel A, Thimm O, Redestig H, Blaesing OE, Palacios-Rojas N, Selbig J, Hannemann J, Piques MC, Steinhauser D, Scheible WR, Gibon Y, Morcuende R, Weicht D, Meyer S, Stitt M: **Extension of the Visualization Tool MapMan to Allow Statistical Analysis of Arrays, Display of Coresponding Genes, and Comparison with Known Responses.** *Plant Physiol* 2005, **138(3)**:1195-1204.
22. Sevon P, Eronen L, Hintsanen P, Kulovesi K, Toivonen H: **Link Discovery in Graphs Derived from Biological Databases.** In *DILS, Volume 4075 of Lecture Notes in Computer Science*. Edited by: Leser U, Naumann F, Eckman BA. Springer; 2006:35-49.
23. Hull D, Wolstencroft K, Stevens R, Goble CA, Pocock MR, Li P, Oinn T: **Taverna: a tool for building and running workflows of services.** *Nucleic Acids Research* 2006, , **34 Web-Server:** 729-732.
24. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH: **The WEKA data mining software: an update.** *SIGKDD Explorations* 2009, **11**:10-18.
25. Berthold MR, Cebron N, Dill F, Gabriel TR, Kötter T, Meinl T, Ohl P, Thiel K, Wiswedel B: **KNIME - the Konstanz information miner: version 2.0 and beyond.** *SIGKDD Explorations* 2009, **11**:26-31.
26. Demšar J, Zupan B, Leban G, Curk T: **Orange: From Experimental Machine Learning to Interactive Data Mining.** In *PKDD, Volume 3202 of Lecture Notes in Computer Science*. Edited by: Boulicaut JF, Esposito F, Giannotti F, Pedreschi D. Springer; 2004:537-539.
27. Altintas I, Berkley C, Jaeger E, Jones MB, Ludäscher B, Mock S: **Kepler: An Extensible System for Design and Execution of Scientific Workflows.** *SSDBM, IEEE Computer Society* 2004, 423-424.
28. Majithia S, Shields MS, Taylor IJ, Wang I: **Triana: A Graphical Web Service Composition and Execution Toolkit.** *Proceedings of the IEEE International Conference on Web Services (ICWS'04), IEEE Computer Society* 2004, 514-524.
29. Chiaretti S, Li X, Gentleman R, Vitale A, Vignetti M, Mandelli F, Ritz J, Foa R: **Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival.** *Blood* 2004, **103(7)**:2771-2778.
30. Wagner W, Horn P, Castoldi M, Diehlmann A, Bork S, Saffrich R, Benes V, Blake J, Pfister S, Eckstein V, Ho AD: **Replicative senescence of mesenchymal stem cells: a continuous and organized process.** *PloS one* 2008, **3(5)**.
31. Ackermann M, Strimmer K: **A general modular framework for gene set enrichment analysis.** *BMC Bioinformatics* 2009, **10**.
32. Robnik-Šikonja M, Kononenko I: **Theoretical and Empirical Analysis of ReliefF and RReliefF.** *Machine Learning* 2003, **53(1-2)**:23-69.
33. Mishra D, Sahu B: **Feature Selection for Cancer Classification: A Signal-to-noise Ratio Approach.** *International Journal of Scientific & Engineering Research* 2011, **2**.
34. Emmert-Streib F, Dehmer M, Liu J, Mühlhäuser M: **A systems approach to gene ranking from DNA microarray data of cervical cancer.** *International Journal of Mathematical and Computer Sciences* 2008, **4**.
35. Hwang T, Sun CHH, Yun T, Yi GSS: **FiGS: a filter-based gene selection workbench for microarray data.** *BMC bioinformatics* 2010, **11**.
36. Huang DW, Sherman BT, Lempicki RA: **Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources.** *Nature protocols* 2009, **4**:44-57.
37. Huang DW, Sherman BT, Lempicki RA: **Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists.** *Nucleic acids research* 2009, **37**:1-13.
38. Jeffery IB, Higgins DG, Culhane AC: **Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data.** *BMC Bioinformatics* 2006, **7**:359.
39. Csárdi G, Kutalik Z, Bergmann S: **Modular analysis of gene expression data with R.** *Bioinformatics* 2010, **26(10)**:1376-1377.
40. Schallmoser K, Bartmann C, Rohde E, Bork S, Guelly C, Obenauf AC, Reinisch A, Horn P, Ho AD, Strunk D, Wagner W: **Replicative senescence-associated gene expression changes in mesenchymal stromal cells are similar under different culture conditions.** *Haematologica* 2010, **95(6)**:867-874.
41. Noh H, Ahn HJ, Lee WJ, Kwack K, Kwon Y: **The molecular signature of in vitro senescence in human mesenchymal stem cells.** *Genes & Genomics* 2010, **32**:87-93.
42. Lansdorp PM: **Repair of telomeric DNA prior to replicative senescence.** *Mechanism of Ageing and Development* 2000, **118(1-2)**:23-34.
43. Karnani N, A D: **The effect of the intra-S-phase checkpoint on origins of replication in human cells.** *Genes & Development* 2010, **25(6)**:621-633.
44. Lecourt S, Vanneaux V, Leblanc T, Leroux G, Ternaux B, Benbunan M, Chomienne C, Baruchel A, Marolleau J, Gluckman E, Socie G, Soulier J, Larghero J: **Bone marrow microenvironment in fanconi anemia: a prospective functional study in a cohort of fanconi anemia patients.** *Stem Cells and Development* 2010, **19(2)**:203-208.
45. Tsurutani Y, Fujimoto M, Takemoto M, Irisuna H, Koshizaka M, Onishi S, Ishikawa T, Mezawa H, He P, Honjo S, Maezawa Y, Saito Y, Yokote K: **The roles of transforming growth factor-b and SMAD3 signalling in adipocyte differentiation and obesity.** *Biochemical and Biophysical Research Communications* 2011, **407**:68-73.
46. Urs S, Smith C, Campbell B, Saxton A, Taylor J, Zhang B, Snoddy J, Jones V, Moustaid-Moussa N: **Gene expression profiling in human preadipocytes and adipocytes by microarray analysis.** *Journal of Nutrition* 2004, **134(4)**:762-770.

47. Wagner W, Bork S, Lepperdinger G, Joussen S, Ma N, Strunk D, Koch C: **How to track cellular aging of mesenchymal stromal cells?** *Aging* 2010, **2(4)**:224-230.
48. Hobbs C: **Using ZSI.** *Tech. rep., Nortel Advanced Technology Research* 2007.
49. Bray T, Paoli J, Sperberg-McQueen CM, Maler E, Yergeau F: **Extensible Markup Language (XML) 1.0 (Fifth Edition).** *W3C Recommendation* 2008 [http://www.w3.org/TR/2008/REC-xml-20081126/].
50. Mitra N, Lafon Y: **SOAP Version 1.2 Part 0: Primer (Second Edition).** *W3C Recommendation* 2007 [http://www.w3.org/TR/2007/REC-soap12-part0-20070427/].
51. Booth D, Canyang K: **Web Services Description Language (WSDL) Version 2.0 Part 0: Primer.** *W3C Recommendation* 2007 [http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/].
52. Haider S, Ballester B, Smedley D, Zhang J, Rice PM, Kasprzyk A: **BioMart Central Portal - unified access to biological data.** *Nucleic Acids Research* 2009, , **37 Web-Server:** 23-27.
53. Mcwilliam H, Valentin F, Goujon M, Li W, Narayanasamy M, Martin J, Miyar T, Lopez R: **Web services at the European Bioinformatics Institute-2009.** *Nucleic Acids Research* 2009, **37(suppl_2)**:W6-10.
54. Bhagat J, Tanoh F, Nzuobontane E, Laurent T, Orlowski J, Roos M, Wolstencroft K, Aleksejevs S, Stevens R, Pettifer S, Lopez R, Goble CA: **BioCatalogue: a universal catalogue of web services for the life sciences.** *Nucleic Acids Research* 2010, , **38 Web-Server:** 689-694.
55. Eden E, Navon R, Steinfeld I, Lipson D, Yakhini Z: **GOrilla: a tool for discovery and visualization of enriched GO terms in ranked gene lists.** *BMC Bioinformatics* 2009, **10**:48.

doi:10.1186/1471-2105-12-416
Cite this article as: Podpečan *et al.*: SegMine workflows for semantic microarray data analysis in Orange4WS. *BMC Bioinformatics* 2011 **12**:416.

Submit your next manuscript to BioMed Central and take full advantage of:

- **Convenient online submission**
- **Thorough peer review**
- **No space constraints or color figure charges**
- **Immediate publication on acceptance**
- **Inclusion in PubMed, CAS, Scopus and Google Scholar**
- **Research which is freely available for redistribution**

Submit your manuscript at
www.biomedcentral.com/submit

BioMed Central

4 Contrasting Subgroup Discovery

This chapter presents contrasting subgroup discovery which allows for finding subgroups of objects that could not be found with classical subgroup discovery. The methodology was applied in the field of systems biology, and implemented as a collection of interactive workflow components in Orange4WS.

The goal of subgroup discovery (SD) (Klösgen, 1996; Wrobel, 1997) methods is to find population subgroups in data that are statistically interesting with respect to a given property of interest (i.e., class). SD algorithms induce rules of the form $\mathit{Conditions} \mapsto \mathit{Class}$ where *Conditions* is a conjuction of attribute values, and *Class* is a property of interest. As a result, these methods discover sets of objects that share attribute values, characteristic for the distinguished property of interest (class).

The proposed contrasting subgroup discovery methodology allows for the discovery of more general groups where the members are characteristic for some class, but the classes are not necessarily identical. To achieve this, CSD proposes a three-step mining process which consists of the following steps.

1. The first step is to apply a classical subgroup discovery method to discover statistically interesting groups of objects.

2. In the second step, two contrasting classes are defined. The definitions of contrast classes can include several different class attributes, as well as subgroup memberships obtained in the first step. The contrasting classes are constructed using set theoretic functions the selection of which depends on the problem being solved.

3. In the final step, classical subgroup discovery is performed again in order to analyse the subgroups with respect to the contrast classes.

This complex, multi-step approach requires several software components, which need to be interconnected. Therefore, the CSD methodology was implemented in Orange4WS as a set of several workflow components which can be easily composed into interactive workflows. In our experiments contrasting subgroup discovery was applied to a time labelled Solanum tuberosum gene expression dataset for virus-infected and non-infected plants, and several existing SegMine workflow components were reused in CSD workflows. Most notably, the SEGS web service was adapted for the GoMapMan ontology (Baebler et al., 2010), an extension of the plant ontology MapMan (Thimm et al., 2004). An example of a CSD workflow in Orange4WS applied to time-labelled gene expression data is shown in Figure 4.1. It features four instances of SEGS subgroup discovery service and widgets for the definition of contrasting classes (details are provided in the caption). The methodology is presented in the following publication:

Langohr, L.; Podpečan, V.; Petek, M.; Mozetič, I.; Gruden, K.; Lavrač, N.; Toivonen, H. Contrasting subgroup discovery. *The Computer Journal* (2012). In press

In addition, Appendix B presents our implementation of the CSD methodology in Or-

ange4WS by providing descriptions of Orange4WS workflow components[1], their inputs and outputs as well as graphical representations (where applicable).

The author's contributions are as follows. Laura Langohr originated and developed the theoretical framework of contrasting subgroup discovery and performed the experiments. Vid Podpečan implemented the CSD methodology as Orange4WS workflows, and extended the SEGS service with the GoMapMan ontology to allow for the analysis of plant gene expression data. Marko Petek interpreted and evaluated the results of the experiments on the Solanum tuberosum data. Igor Mozetič coordinated computer science aspects of the development of the GoMapMan ontology. Kristina Gruden coordinated the biological aspects of the study. Nada Lavrač and Hannu Toivonen coordinated the computer science aspects of the study and development of the software.

[1] Our implementations of Contrasting subgroup discovery and the SegMine methodology share several workflow components. For example, data preprocessing and clustering components are the same in both implementations while the SEGS algorithm and the Biomine search engine differ only in the background knowledge (ontologies and databases).

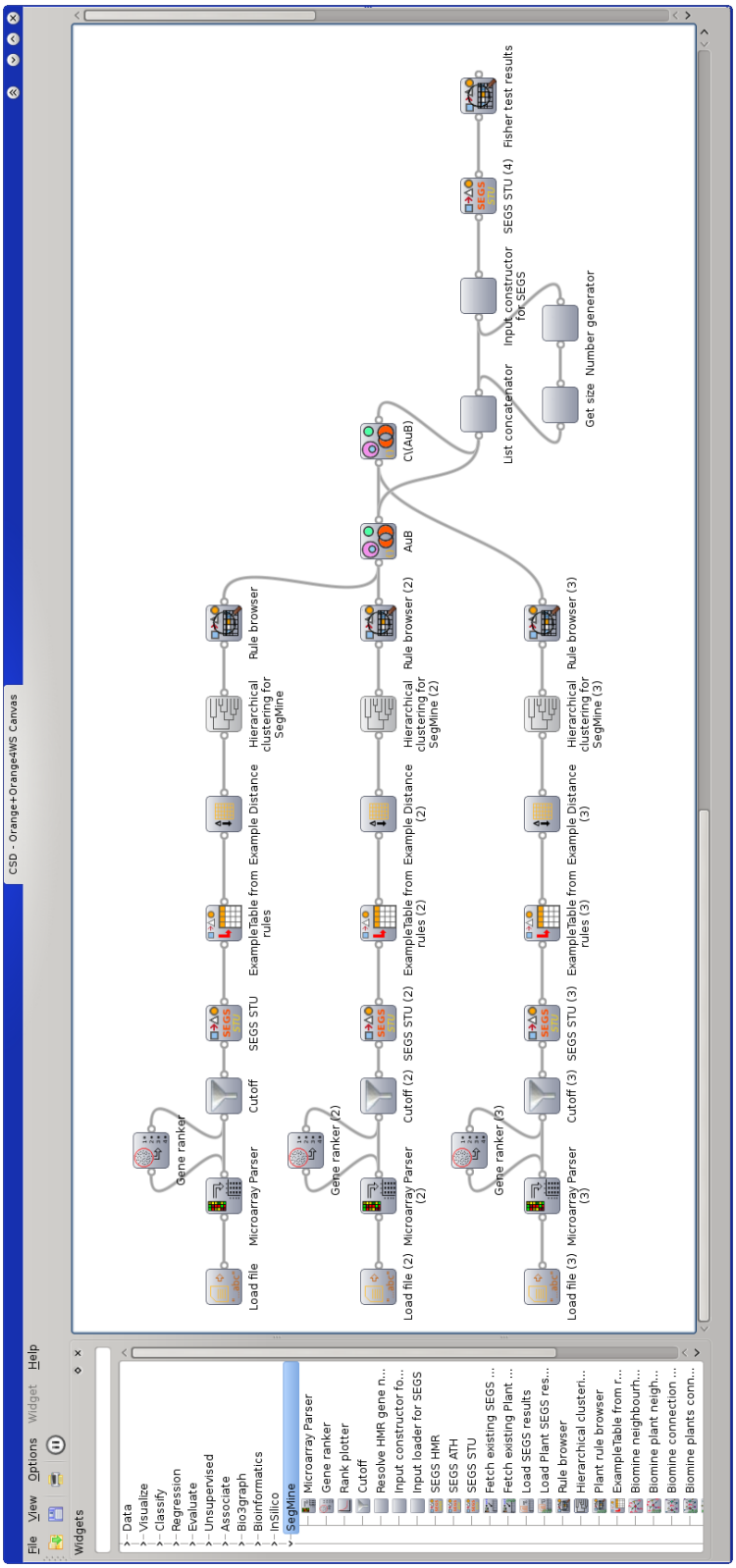

Figure 4.1: *Contrasting subgroup discovery workflow.* The Orange4WS workflow implementing contrasting subgroup discovery, applied to a time labelled gene expression dataset for virus infected Solanum tuberosum (potato) plants (one, three and six days after the infection). The SEGS algorithm is used for subgroup discovery while the contrast sets P_c and $\overline{P_c}$ are constructed to contrast the sixth day after infection with the other two time points as follows: $P_c = P_C \setminus (P_A \cup P_B)$, $\overline{P_c} = P_A \cup P_B$ where P_A, P_B and P_C are characteristic gene sets for the first, third and the sixth day after virus infection.

The Computer Journal Advance Access published September 22, 2012

© The Author 2012. Published by Oxford University Press on behalf of The British Computer Society. All rights reserved.
For Permissions, please email: journals.permissions@oup.com
doi:10.1093/comjnl/bxs132

Contrasting Subgroup Discovery

Laura Langohr[1]*, Vid Podpečan[2,3], Marko Petek[4], Igor Mozetič[2], Kristina Gruden[4], Nada Lavrač[2] and Hannu Toivonen[1]

[1]*Department of Computer Science and Helsinki Institute for Information Technology HIIT, University of Helsinki, Helsinki, Finland*
[2]*Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia*
[3]*International Postgraduate School Jožef Stefan, Ljubljana, Slovenia*
[4]*Department of Biotechnology and Systems Biology, National Institute of Biology, Ljubljana, Slovenia*
**Corresponding author: laura.langohr@cs.helsinki.fi*

Subgroup discovery methods find interesting subsets of objects of a given class. Motivated by an application in bioinformatics, we first define a generalized subgroup discovery problem. In this setting, a subgroup is interesting if its members are characteristic for their class, even if the classes are not identical. Then we further refine this setting for the case where subsets of objects, for example, subsets of objects that represent different time points or different phenotypes, are contrasted. We show that this allows finding subgroups of objects that could not be found with classical subgroup discovery. To find such subgroups, we propose an approach that consists of two subgroup discovery steps and an intermediate, contrast set definition step. This approach is applicable in various application areas. An example is biology, where interesting subgroups of genes are searched by using gene expression data. We address the problem of finding enriched gene sets that are specific for virus-infected samples for a specific time point or a specific phenotype. We report on experimental results on a time series dataset for virus-infected *Solanum tuberosum* (potato) plants. The results on *S. tuberosum*'s response to virus-infection revealed new research hypotheses for plant biologists.

Keywords: subgroup discovery; gene set enrichment

Received 30 November 2011; revised 15 August 2012
Handling editor: Einoshin Suzuki

1. INTRODUCTION

Subgroup discovery [1, 2] is a typical task in data mining for finding interesting subsets of objects. Classical subgroup discovery methods consider a set of objects interesting if they share a combination of attribute values that is characteristic for some class. In contrast, we aim to find subgroups of the following type: a set of objects is interesting if each of its members is characteristic for its own class, even if the classes are not identical. This allows finding patterns that could not be found with classical subgroup discovery.

For instance, in a dataset of bank customers, it may be the case that males tend to be characteristic in the sense that the combination of their education, occupation and location is characteristic for either high or low spenders. The setting proposed in this paper allows discovering males as an interesting subgroup, since being male implies that the person is characteristic for his class. Classical subgroup discovery methods would only be able to find separate subgroups for high spenders and low spenders, and would miss that males, in general, are characteristic for their classes.

This powerful effect is obtained by allowing the user to specify subsets of objects she wants to contrast in a flexible manner. First, these contrast sets can be defined using not only the original attributes, but also using information about characteristics with respect to classes (i.e. classical subgroup memberships). Secondly, contrast set definitions can use set-theoretic operations. For instance, an economist might be interested in contrasting different time points (e.g. before, during and after the financial crisis). She could then specify that she is interested in objects at a specific time point in contrast to all other time points. In such settings, classical subgroup discovery can contrast two time points, or several time points in a pairwise fashion. In the setting proposed here, and in the biological application that motivates our work, we are interested

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

in contrasting subgroups from several time points (or several phenotypes) at the same time. We call this generalized problem *the contrasting subgroup discovery problem*.

To find such generalized subgroups of objects, we propose an approach that consists of two subgroup discovery steps and an intermediate, contrast set definition step. In the first step, interesting subgroups are found in a classical manner, based on semantic and statistical properties of objects. In the banking example, we can use an existing subgroup discovery method to find classical subgroups for the classes of low and high spenders, and would do this for each time point separately. In the second step, the user defines two new classes of objects; these are the contrast classes for the third step. As mentioned above, the definitions of contrast classes can take into account several different class attributes (such as different time points) as well as subgroup memberships from the first step. In the third and final steps, a classical subgroup discovery method is used to find interesting subgroups of objects of the two contrast classes. As a result, the subgroups can contain objects that are characteristic for their class, regardless of their class.

In the next section, we give a brief overview of classical subgroup discovery and describe how subgroup discovery and contrast mining have been addressed in different applications before (Section 2). In Section 3, we then propose the problem of contrasting subgroup discovery more formally. We then show how well-known algorithms can be combined to solve the problem as outlined above (Section 4).

In the second half of the paper, we focus on an important application in biology. In Section 5, we describe a gene set enrichment problem where the goal is to analyze contrasting gene sets, and we give an instance of the proposed methodology to solve the problem. In Section 6, we apply it on a time-series dataset from virus-infected potato plants (*Solanum tuberosum*) and report experimental results. Finally, we conclude with some notes about the results and future work.

2. BACKGROUND

Discovering patterns in data is a classical problem in data mining and machine learning [3, 4]. To represent patterns in an explanatory form, they are often described by rules $X \mapsto Y$, where $\mapsto$ denotes an implication and the antecedent X and the consequent Y can represent sets of attribute values (e.g. terms), a class or sets of objects, depending on the problem at hand.

Next, we define the problem of subgroup discovery formally, review related work and discuss how our approach differs from other pattern discovery approaches.

2.1. Subgroup discovery

Subgroup discovery methods find rules of the form $Condition \mapsto Subgroup$, where the antecedent $Condition$ is a conjunction of attribute values and the consequent $Subgroup$ is a set of objects, which satisfy some class-related interestingness measure.

Subgroups defined by individual attribute values. Consider a set S of objects, annotated by a set T of attribute values (e.g. terms). Each attribute value $t \in T$ defines a subgroup $S_t \subset S$ that consists of all objects $s \in S$ where t is true, that is, all objects annotated by the attribute value t:

$$S_t = \{s \in S \mid s \text{ is annotated by } t\}. \quad (1)$$

EXAMPLE 2.1. Consider the bank customers of Table 1 which are annotated by the attributes *Occupation* and *Location* and assigned the class *high* or *low* for the class attribute *Spending* for two different time points, before and after the financial crisis, respectively. The attribute value $Location = village$ defines the subgroup $\{19, 20\}$ of two bank customers and $Occupation = education$ defines a subgroup of five bank customers $\{6, 9, 11, 16, 18\}$.

Subgroups defined by logical conjunctions of attribute values. Subgroups can be constructed by intersections, which are described by logical conjunctions of attribute values. Let $S_1, \ldots, S_k$ be k subgroups described by the attribute values $t_1, \ldots, t_k$. Then the logical conjunction of k attribute values defines the intersection of k subgroups:

$$t_1 \wedge t_2 \wedge \cdots \wedge t_k \mapsto S_1 \cap S_2 \cap \cdots \cap S_k. \quad (2)$$

TABLE 1. Bank customers *before* and *after* the financial crisis described by attributes Occupation and Location, and the class attribute Spending (adapted from [43]).

			Spending	
ID	Occupation	Location	Before	After
1	Industry	Big city	High	High
2	Industry	Big city	High	Low
3	Retail	Big city	High	Low
4	Finance	Big city	High	High
5	Doctor	Big city	High	High
6	Education	Big city	High	Low
7	Nurse	Big city	High	Low
8	Industry	Small city	High	High
9	Education	Small city	High	Low
10	Retail	Small city	High	Low
11	Education	Big city	Low	Low
12	Nurse	Big city	Low	Low
13	Unemployed	Big city	Low	Low
14	Retail	Small city	Low	Low
15	Doctor	Small city	Low	Low
16	Education	Small city	Low	Low
17	Unemployed	Small city	Low	Low
18	Education	Small city	Low	Low
19	Unemployed	Village	Low	Low
20	Unemployed	Village	Low	Low

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

Alternatively, we can write $T' \mapsto S_{T'}$, where T' is a set of attribute values $T' = \{t_1, \ldots, t_k\} \subset T$, whose conjunction defines the subgroup $S_{T'} = S_1 \cap S_2 \cap \cdots \cap S_k$.

EXAMPLE 2.2. The set $T' = \{education, small\ city\}$ defines a subgroup of three bank customers $\{9, 16, 18\}$ in Table 1.

An object can be a member of several subgroups. A subgroup might be a subset of another subgroup. In particular, in case the attribute values are organized in a hierarchy (or ontology), an object that is annotated by the attribute value t is also considered to be annotated by the ancestors of t in the hierarchy.

EXAMPLE 2.3. Consider the hierarchies in Fig. 1. All individuals working in the retail sector also work in the service and private sector.

An *ontology* is a representation of a conceptualization and is often represented by a hierarchy, where nodes represent concepts (e.g. occupations or locations) and edges a subsumption relation (e.g. 'is a' or 'part of') between concepts [5]. See, for example, Fig. 1, where *nurses* as well as *doctors* are part of the *health* sector, which is part of the *public* sector. Ontologies can be used to incorporate background knowledge about attribute values (such as concepts, terms or something else). Subgroup discovery methods often use hierarchies to restrict the search space (see, e.g. [6, 7]), but subgroup discovery does not require that the attribute values be organized in a hierarchy.

Class-related interestingness measure. For each subgroup, one has to measure whether the subgroup is interesting or not. Classical subgroup discovery methods look for groups that are specific for a class when compared with the rest of the objects.

Similarly, to attribute values, classes define subgroups (sets) of objects. Let $c \in T$ be a specific class. Then an object $s \in S$ belongs to the subgroup defined by c if and only if s is annotated by c.

EXAMPLE 2.4. Consider again the bank customers in Table 1. For the time point before, the financial crisis the class $Spending = high$ defines the subgroup $\{1, \ldots, 10\}$ and the class $Spending = low$ defines the subgroup $\{11, \ldots, 20\}$.

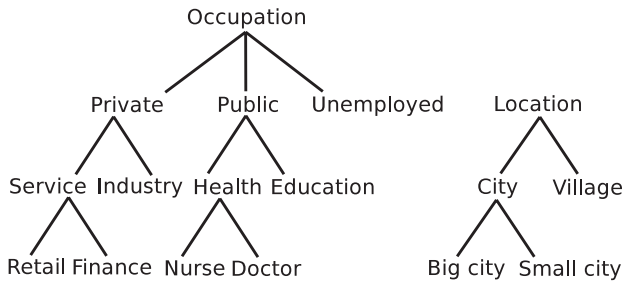

FIGURE 1. Example hierarchies of attribute values which are in this case terms (adapted from [43]).

In practice, a subgroup is often classified homogeneously. To formalize this idea, let

$$classes : \mathcal{P}(S) \to \mathbb{Z}_+ \times \mathbb{Z}_+ \tag{3}$$

be a function that gives the class distribution of a given set $S_{T'} \subset S$ of objects, that is, the number of objects in $S_{T'}$ annotated by c and the number of objects in $S_{T'}$ not annotated by c. (Here $\mathcal{P}(S)$ is the powerset of S.)

EXAMPLE 2.5. Consider the subgroup $S_{T'} = \{9, 16, 18\}$ of three bank customers described by $T' = \{education, small\ city\}$ and the class $Spending = high$ for the time point before the financial crisis in Table 1. The class distribution of $S_{T'}$ is $classes(S_{T'}) = (1, 2)$ as one object of $S_{T'}$ is annotated by $Spending = high$ and two objects by $Spending = low$. Similarly, the class distribution of $S \setminus S_{T'}$ is $classes(S \setminus S_{T'}) = (9, 8)$.

DEFINITION 2.1. *The classical* class-related interestingness measure *is a function*

$$\begin{aligned} f_c : \mathcal{P}(T) &\to \mathbb{R}, \\ T' &\mapsto g(classes(S_{T'}), classes(S \setminus S_{T'})), \end{aligned} \tag{4}$$

for some function g. That is, f_c is a function $g(\cdot)$ of the class distributions within and outside of the subgroup.

The exact definition of g varies from one problem variant to another, but the common denominator is that it is based on the class distributions alone.

Often the subgroups are analyzed by statistical tests, like Fisher's exact test, χ^2 test or the binomial probability. In our experiments, we use a p-value estimate obtained by *Fisher's exact test* [8] and a simple permutation test as class-related interestingness measure f_c. Without loss of generality, in the rest of the paper we assume that smaller values of f_c indicate more interesting subgroups.

Given a class attribute c with two possible classes $t_c, t_{\bar{c}} \in T$, the data are arranged in a contingency table for each subgroup $S_{T'} \subset S$, where $classes(S_{T'}) = \{n_{11}, n_{12}\}$, $classes(S \setminus S_{T'}) = \{n_{21}, n_{22}\}$ and $n = |S| = n_{11} + n_{12} + n_{21} + n_{22}$:

	t_c	$t_{\bar{c}}$
$S_{T'}$	n_{11}	n_{12}
$S \setminus S_{T'}$	n_{21}	n_{22}

Fisher's exact test then evaluates the probability of obtaining the observed distribution (counts n_{ij}), or a more extreme one, assuming that the marginal counts (t_c, $t_{\bar{c}}$, $S_{T'}$, $S \setminus S_{T'}$) are

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

fixed [8]. Therefore, first, the probability of observed quantities is calculated by

$$P(X = n_{11}) = \binom{n_{11}+n_{12}}{n_{11}} \binom{n_{21}+n_{22}}{n_{21}} \Big/ \binom{n}{n_{11}+n_{21}}. \quad (5)$$

Then the p-value is the sum of all probabilities for the observed or more extreme (that is, $X < n_{11}$) observations:

$$p = \sum_{i=0}^{n_{11}} P(X = i). \quad (6)$$

EXAMPLE 2.6. Consider the bank customers in Table 1, the time point before the financial crisis, the attribute value set $T' = \{village\}$ and the classes $t_c = high$ vs. $t_{\bar{c}} = low$ for the class attribute *Spending*. There are two bank customers living in a *village*: $S_t = S_{village} = \{19, 20\}$, of which none is annotated by $Spending = high$. Hence, Fisher's exact p-value is $p \approx 0.237$.

Permutation test. In our experiments, to address the multiple testing problem, we perform a simple permutation test that returns adjusted p-values (see Appendix for the details).

Subgroup discovery. We can now describe the problem of subgroup discovery formally.

DEFINITION 2.2. *The* subgroup discovery problem *is to output all sets $T' \subset T$ of attribute values for which $f_c(T') \leq \alpha$ for some given constant α.*

Equivalently, the subgroups defined by the sets of attribute values could be output, and in practice both, the sets of attribute values and subgroups are often shown as a result. An alternative formulation of the problem is to output the k best subgroups instead of using a fixed threshold.

EXAMPLE 2.7. Consider again the bank customers in Table 1. When using Fisher's exact test, the adjusted p-value as class-related interestingness measure $f_c(\cdot)$, and $\alpha = 0.3$ (for the sake of simplicity, we consider a relatively high threshold in this toy example), a subgroup discovery method finds four interesting subgroups for the time point before the financial crisis: $village \mapsto \{19, 20\}$, $unemployed \mapsto \{13, 17, 19, 20\}$, $unemployed \wedge city \mapsto \{13, 17\}$ and $unemployed \wedge village \mapsto \{19, 20\}$ as well as two interesting subgroups $education \mapsto \{6, 9, 11, 16, 18\}$ and $education \wedge city \mapsto \{6, 9, 11, 16, 18\}$ for the time point after the financial crisis.

2.2. Other pattern mining approaches

Other pattern mining approaches mentioned below can be classified as unsupervised and supervised. Unsupervised methods (frequent item set mining and association rule mining) take a dataset without class labels as input, while the input to supervised methods (the other methods listed below) is a class labeled dataset. Note that the supervised methods can take multiple classes into account by comparing two classes where one is a union of several (sub)classes [9].

Frequent item set mining aims to find frequent combinations of attribute values (items) such as $Occupation = industry \wedge Spending = high$ [10]. Similar to the approach presented here, some methods intersect transactions to find closed frequent item sets [11–13].

Emerging patterns are item sets for which the supports increase significantly from one class to another [14].

Association rules describe associations, such as $X \mapsto Y$, where the antecedent X and consequent Y are item sets (e.g. sets of terms) [15]. In categorical data, the antecedent and consequent are (attribute, attribute value) pairs such as $Occupation = industry \mapsto Spending = high$ [16, 17].

Exception rule mining aims to find unexpected association rules that differ from a highly frequent association rule [18]. That is, it finds unexpected association rules $X \wedge Z \mapsto Y$, where $X \mapsto Y'$ and $Z \not\mapsto Y'$. Here, X and Z are item sets or (attribute, attribute value) pairs, and Y and Y' are different (class attribute, class) pairs. Consider, for example, X as $Occupation = industry$, Z as $Location = city$, Y as $Spending = high$ and Y' as $Spending = low$.

Contrast set mining is an extension of association rule mining and aims to understand the differences between contrasting groups of objects [16, 17, 19, 20]. Contrast set mining and emerging pattern mining are formally equivalent and can be effectively solved by subgroup discovery methods [17, 21]. In contrast set mining two contrast classes are defined, while in subgroup discovery only one class and its complement are used.

Examples of contrast set mining methods are Search and Testing for Understandable Consistent Contrasts [16], Contrasting Grouped Association Rules [20] and Rules for Contrast Sets [22] all of which derive rules of attribute-value pairs for which the support differs meaningfully across groups.

In a setting where several different class attributes exist, these methods can be applied in a pairwise manner. For example, one could contrast two different levels of spending for different time points or different locations separately. That is, these methods find rules such as $Occupation = industry \wedge Spending = high$ for which the support is significantly larger within the individuals that are described by $Location = city$ than $Location = village$.

We also aim to understand the differences between several contrasting groups. However, in contrast to contrast set mining and the other approaches described here, our aim is to find interesting subgroups of objects which are characteristic for their class, regardless of their class. Next we describe the problem formally.

3. PROBLEM DEFINITION

We now formulate the problem of contrasting subgroup discovery in more exact terms. We replace the direct dependency

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

on the class distribution of the classical subgroup discovery by a contrasting, indirect one. In the classical, direct case, one is interested in sets of attribute values that are characteristic for a class. Our aim to understand phenomena in a setting where several different classes (for example, different time points) are given. That is, in the contrasting case, we want to find sets of attribute values that indicate objects that are characteristic for their class, but not necessarily the same one.

To formally define the task, we first introduce a notation P for the set of objects characteristic for their class:

$$P = \{s \in S \mid \text{there exists } T' \subset T \text{ such that } f_c(T') \leq \alpha \text{ and } s \in S_{T'}\}, \quad (7)$$

where (as before) T denotes the set of attribute values (e.g. terms), S the set of objects, $S_{T'}$ the set of objects annotated by the attribute value set $T' \subset T$, $f_c(\cdot)$ the class-related interestingness measure and α a given constant.

EXAMPLE 3.1. Consider again the bank customers in Table 1 and the subgroups found with a classical subgroup discovery method (see Example 2.7). Then the set of objects characteristic for their class is $P = \{13, 17, 19, 20\}$ for the time point before and $P = \{6, 9, 11, 16, 18\}$ for the time point after the financial crisis.

Now the user can define two contrast classes P_c, $\overline{P_c} \subset P$. The selection of these two contrast classes depends on the objective and is left to the user. They can, for example, take several classes (such as different time points) into account.

Let $c_1, \ldots, c_m$ be m class attributes and $P_1, \ldots, P_m$ be the sets of objects characteristic for each of the class attributes. Here, we define P_c and $\overline{P_c}$ in two different, exemplary ways. First, P_c can be defined as the set of objects occurring in interesting subgroups of *all* class attributes:

$$P_c = \bigcap_{i \in \{1,\ldots,m\}} P_i. \quad (8)$$

This is useful when one wants to find interesting subgroups that are common to all class attributes (for example, a specific time point in contrast to all other time points).

Secondly, P_c can be defined as the set of objects occurring *only* in interesting subgroups of the kth class attributes:

$$P_c = P_k \setminus \bigcup_{\substack{i \in \{1,\ldots,m\}, \\ i \neq k}} P_i. \quad (9)$$

This definition can be used to find interesting subgroups that are specific for one class attribute in contrast to all the other class attributes.

The contrast class $\overline{P_c}$ can be defined as the complement of P_c, that is,

$$\overline{P_c} = P \setminus P_c, \quad (10)$$

when one is interested in subgroups specific for the objects in P_c compared with all other objects of P. Or, if a user is interested in contrasting two specific time points even in a case where more time points exist. Then P_c would be defined as one of those time points and $\overline{P_c}$ as the other time point.

EXAMPLE 3.2. In the case of bank customers and the two classes before and after the financial crisis, we obtain the set of objects characteristic for each class attribute separately, that is, $P_1 = \{13, 17, 19, 20\}$ and $P_2 = \{6, 9, 11, 16, 18\}$. When specifying the contrast classes P_c and $\overline{P_c}$ as $P_c = P_1 \setminus P_2$ and $\overline{P_c} = P_2$ (Equations (9) and (10)), we contrast the time point before the financial crisis against the time point after the financial crisis and obtain $P_c = \{13, 17, 19, 20\}$ as well as $\overline{P_c} = \{6, 9, 11, 16, 18\}$ as also shown in Table 2. (Note that we could alternatively contrast the time point after against the time point before the financial crisis by defining $P_c = P_2 \setminus P_1$ and $\overline{P_c} = P_1$.)

Let us define function $characteristic(\cdot)$ that gives the number of objects characteristic for their class in the contrasting classes P_c and $\overline{P_c}$ for a given set $S_{T'}$:

$$characteristic : \mathcal{P}(S) \to \mathbb{Z}_+ \times \mathbb{Z}_+, \quad S_{T'} \mapsto (|S_{T'} \cap P_c|, |S_{T'} \cap \overline{P_c}|). \quad (11)$$

Now, the contrasting interestingness measure, as well as the contrasting subgroup discovery problem, can be formulated as follows.

DEFINITION 3.1. *A* contrasting interestingness measure *is a function*

$$f_i : \mathcal{P}(T) \to \mathbb{R}, \quad T' \mapsto g'(characteristic(S_{T'}), characteristic(P \setminus S_{T'})), \quad (12)$$

for some function g'.

That is, the contrasting interestingness measure analyzes whether a subgroup is interesting w.r.t. the two contrast classes, which both consist only of objects that are characteristic for their own class. This is in contrast to the classical class-related

TABLE 2. Contrast classes of bank customers.

ID	Occupation	Location	Contrast class
6	Education	Big city	$\overline{P_c}$
9	Education	Small city	$\overline{P_c}$
11	Education	Big city	$\overline{P_c}$
13	Unemployed	Big city	P_c
16	Education	Small city	$\overline{P_c}$
17	Unemployed	Small city	P_c
18	Education	Small city	$\overline{P_c}$
19	Unemployed	Village	P_c
20	Unemployed	Village	P_c

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

interestingness measure, which analyzes whether a subgroup is interesting w.r.t. the object's classes.

EXAMPLE 3.3. Consider again the bank customers and the two contrast classes $P_c = \{13, 17, 19, 20\}$ and $\overline{P_c} = \{6, 9, 11, 16, 18\}$ of Table 2. Then the attribute value *Occupation* = *education*, for instance, defines a set of five bank customers $\{6, 9, 11, 16, 18\}$, who are all in $\overline{P_c}$. Given the adjusted p-value as function g', we obtain $f_i(education) \approx 0.0079$.

DEFINITION 3.2. *The* contrasting subgroup discovery problem *is to output all sets $T' \subset T$ of attribute values for which $f_i(T') \leq \alpha'$ for some given constant α'.*

In other words, while classical subgroup discovery is related to the question of how to find sets of objects that are characteristic for a specific class, the problem of contrasting subgroup discovery is related to asking if sets of objects characteristic for (any) classes can be found.

The relationship between the classical and contrasting cases immediately implies that, for any subgroup found for the contrasting subgroup discovery problem, its objects are characteristic for their class. On the other hand, a set of attribute values may be a valid answer to the contrasting problem even if it is not for the classical problem.

That is exactly where the main conceptual contribution of this paper is. Contrast subgroup discovery allows finding subgroups of objects that could not be found with classical subgroup discovery.

4. METHOD

Given a set of objects described by attribute values (e.g. terms) and different classes of objects, our goal is to find interesting subgroups of objects characteristic for their class. Thereby we allow taking different class attributes into account.

To find such subgroups, we propose an approach that consists of three steps: First, interesting subgroups are found by a classical subgroup discovery method. Secondly, contrast classes on those subgroups are defined by set-theoretic functions. Thirdly, contrasting subgroup discovery finds interesting subgroups in the contrast classes. Next, we will describe each step in detail.

Classical subgroup discovery (*Step* 1). Given some objects that are annotated by attribute values, and assigned a class, a subgroup discovery method is applied. Thereby, we consider only one class attribute [e.g. Spending before the financial crisis with different classes (e.g. $Spending = high$ vs. $Spending = low$)], and apply a subgroup discovery method separately for each class attribute (e.g. separately for each time point). The subgroups are then analyzed by a statistical test, like Fisher's exact test followed by a permutation test. (See Example 2.7 for exemplary results of classical subgroup discovery.)

Construction of contrast classes (*Step* 2). Let $P_1, \ldots, P_m$ denote the objects characteristic for their class of m class attributes (e.g. for m different time points). Then the two contrast classes P_c and $\overline{P_c}$ are defined by two set-theoretic functions, for example, by Equations (9) and (10). (As stated before, the selection of a particular set-theoretic function depends on the objective and is left to the user.)

Contrasting subgroup discovery (*Step* 3). In this step, we apply a second subgroup discovery instance in order to analyze subgroups with respect to the constructed contrast classes. Given the objects in the two contrast classes P_c and $\overline{P_c}$, we find interesting subgroups of these objects by a second subgroup discovery instance. Again, the p-values are calculated, using a permutation test.

Assuming that both subgroup discovery instances (Steps 1 and 3) find all subgroups for which the classical interesting measures hold (Equation (4)), the proposed method does find all subgroups that satisfy the indirect interestingness measure (Equation (12)).

EXAMPLE 4.1. In the case of bank customers, we saw already in Example 3.3 that *education* is obtained with contrasting subgroup discovery when the two classes $P_c = \{13, 17, 19, 20\}$ and $\overline{P_c} = \{6, 9, 11, 16, 18\}$ are contrasted. In this contrasting subgroup, discovery the following subgroups are found to be interesting:

$$\begin{aligned} education &\mapsto \{6, 9, 11, 16, 18\},\\ education \wedge city &\mapsto \{6, 9, 11, 16, 18\},\\ education \wedge big\ city &\mapsto \{6, 11\},\\ education \wedge small\ city &\mapsto \{9, 16, 18\},\\ public &\mapsto \{6, 9, 11, 16, 18\},\\ public \wedge city &\mapsto \{6, 9, 11, 16, 18\},\\ public \wedge big\ city &\mapsto \{6, 11\} \end{aligned}$$

and

$$public \wedge small\ city \mapsto \{9, 16, 18\}.$$

In contrast, with a classical subgroup discovery method we obtain

$$\begin{aligned} village &\mapsto \{19, 20\},\\ unemployed &\mapsto \{13, 17, 19, 20\},\\ unemployed \wedge city &\mapsto \{13, 17\} \end{aligned}$$

and

$$unemployed \wedge village \mapsto \{19, 20\},$$

for the time point before the financial crisis and

$$education \mapsto \{6, 9, 11, 16, 18\}$$

and

$$education \wedge city \mapsto \{6, 9, 11, 16, 18\},$$

for the time point after the financial crisis.

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

Hence, some of the subgroups found by the contrasting subgroup discovery were already found by the classical subgroup discovery (for example, $education \wedge city$). Other subgroups found by the contrasting subgroup discovery are more specific than the one found by the classical subgroup discovery (for example, $education \wedge big\ city$). Again other subgroups found by the contrasting subgroup discovery were not found at all by the classical subgroup discovery (for example, $public \wedge big\ city$) as its members are not characteristic for either class (that is, some of them are assigned the class $Spending = high$ and some $Spending = low$).

Both, more specific and new subgroups might reveal new research hypotheses for the user. For example, $public \wedge big\ city$ defines in classical subgroup discovery a subgroup that is not interesting since its objects are characteristic for either *high* or *low* spending. In contrasting subgroup discovery, it defines a subgroup that is characteristic when the two contrasting classes are analyzed. That is, this subgroup's objects occur only in subgroups that are characteristic for the time point after the financial crisis, but not in one that is characteristic for the time point before the financial crisis. Hence, there has been some changes in those subgroups between the two points. This directs the user where to look for the causes of the differences between the time points. Other methods (and possibly data) are needed to find those causes.

5. AN APPLICATION IN BIOLOGY

Application areas of subgroup discovery include sociology [1, 2], marketing [23], vegetation data [24] and transcriptomics [25]. In bioinformatics, high-throughput techniques and simple statistical tests are used to produce rankings of thousands of genes. Life-scientists have to choose a few genes for further (often expensive and time consuming) experiments. In this context, subgroup discovery is known as gene set enrichment (see, e.g. [26, 27]).

A lifescientist might be interested in studying an organism in virus-infected and non-infected conditions at different time points or in different phenotypes of that organism. Here, our aim is to find enriched gene sets characteristic for their class, regardless of their class (for example, characteristic for either differently expressed or not). Further, we allow the user to specify subsets of objects she wants to contrast. The lifescientist could then specify that she is interested in objects at a specific time point in contrast to all other, or for a specific phenotype in contrast to all other phenotypes. With our proposed approach of contrasting subgroup discovery, we can then contrast several time points or phenotypes at the same time.

Using subgroup discovery terminology, we consider genes as objects, and their annotations by terms (e.g. by their molecular functions or biological processes) as attribute values. Table 3 aligns the terms used in the data mining and bioinformatics communities to provide a better understanding of the terminologies.

TABLE 3. Synonyms from different communities.

Subgroup discovery	Bioinformatics
Object or instance	Gene
Attribute value or feature value, e.g. a term in a hierarchy	Annotation or biological concept, e.g. a GO term
Class attribute	Gene expression under a specific experimental condition such as a specific time point or phenotype
Class or class attribute value, e.g. positive/negative	Differential/non-differential gene expression
Subgroup of objects	Gene set
Interesting subgroup	Enriched gene set

Next, we describe measures used for transforming the expression values of several samples (e.g. virus infected vs. non-infected plants) into a class attribute, called differential expression, how the constructed gene sets are analyzed for statistical significance and how enriched gene sets can be found. Finally, we discuss how our proposed method finds contrasting gene sets.

5.1. Measures of differential expression

After preprocessing the gene expression data (including microarray image analysis and normalization), the genes can be ranked according to their gene expression. The dataset of our experiments consists of four samples for both experimental conditions. That is, for each gene we have gene expression levels for four replicates of virus-infected and for four replicates of non-infected plants. Different methods can be used to transform several samples into one class attribute. Here, we will discuss two widely used ones.

Fold change (*FC*) is a metric for comparing the expression level of a gene g between two distinct experimental conditions, for example, virus-infected and non-infected [25]. FC is defined as the log ratio of the average gene-expression levels with respect to the two conditions [28]. Note that FC values do not indicate the level of confidence in the designation of genes as differently expressed or not.

The *t-test* is used to determine the statistical significance of the gene expression between two distinct experimental conditions [25] though, the power of the test is relatively low for small sample sizes [28]. A Bayesian t-test is advantageous if only a few (that is, two or three) samples are used, but no advantage is gained if more replicates are used [29]. In our experiments, we use four replicates and therefore will use the simple t-test.

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

5.2. Analysis of gene set enrichment

Given a list $L = \{g_1, \ldots, g_n\}$ of n genes in which all genes of S are ranked by their expression levels $e_1, \ldots, e_n$, we can analyze the enrichment of a gene set $S_{T'}$ compared with the other genes $S \setminus S_{T'}$ with statistical tests like Fisher's exact test [8]. Alternatively, gene set enrichment analysis (GSEA) [30] or parametric analysis of gene set enrichment (PAGE) [27] can be used. Both methods use the ranking of differential expressions, instead of a partition of the genes into two classes.

Fisher's exact test. When analyzing the gene set $S_{T'}$ compared with the other genes $S \setminus S_{T'}$ with Fisher's exact test, we need to divide the genes into two classes t_c and $t_{\bar{c}}$. Therefore, a cut-off is set in the gene ranking: genes in the upper part are defined as differentially expressed and the genes in the lower part are defined as not differentially expressed genes. Then the p-values are calculated and a permutation test is performed.

GSEA evaluates whether objects of $S_{T'}$ are randomly distributed throughout the list L or primarily found at the top or bottom of the list [26, 30]. An enrichment score (ES) is calculated, which is the maximum deviation from zero of the fraction of genes in the set $S_{T'}$ weighted by their correlation and the fraction of genes not in the set:

$$\mathrm{ES}(S_{T'}) = \max_{i \in \{1,\ldots,n\}} \left| \sum_{\substack{g_j \in S_{T'} \\ j \le i}} \frac{|e_j|^p}{n_w} - \sum_{\substack{g_j \notin S_{T'} \\ j \le i}} \frac{1}{n - n_w} \right|, \qquad (13)$$

where $n_w = \sum_{g_j \in S_{T'}} |e_j|^p$. If the ES is small, then $S_{T'}$ is randomly distributed across L. If it is high, then the genes of $S_{T'}$ are concentrated in the beginning or the end of the list L. The exponent p controls the weight of each step. We see that $\mathrm{ES}(S_{T'})$ reduces to the standard Kolmogorov–Smirnov statistic if $p = 0$:

$$\mathrm{ES}(S) = \max_{i \in \{1,\ldots,n\}} \left| \sum_{\substack{g_j \in S_{T'} \\ j \le i}} \frac{1}{|S_{T'}|} - \sum_{\substack{g_j \notin S_{T'} \\ j \le i}} \frac{1}{|S| - |S_{T'}|} \right|. \qquad (14)$$

The significance of $\mathrm{ES}(S_{T'})$ is then estimated by a permutation test.

PAGE is a GSEA method based on a parametric statistical analysis model [27]. For each gene set, $S_{T'}$ a Z-score is calculated, which is the fraction of the mean deviation to the standard deviation of the ranking score values:

$$Z(S_{T'}) = (\mu_{S_{T'}} - \mu)\frac{1}{\sigma}\sqrt{|S_{T'}|}, \qquad (15)$$

where σ is the standard deviation, and μ and $\mu_{S_{T'}}$ are the means of the score values for all genes and for the genes in set $S_{T'}$, respectively. The Z-score is high if the deviation of the score values is small or if the means largely differ between the gene set and all genes. As gene sets may vary in size, the fraction is scaled by the square root of the set size. However, because of this scaling the Z-score is also high if $S_{T'}$ is very large. Assuming a normal distribution, a p-value for each gene set is calculated. Finally, the p-values are corrected by a permutation test.

Kim and Volsky [27] studied different datasets for which PAGE generally detected a larger number of significant gene sets than GSEA. On the other hand, GSEA makes no assumptions about the variability and can be used if the distribution is not normal or is unknown.

Trajkovski *et al.* [7] used the sum of GSEA's and PAGE's p-values, weighted by percentages (e.g. one-third of GSEA's and two-thirds of PAGE's or half of both). Hence, gene sets with small p-values for GSEA and PAGE are output as enriched gene sets.

5.3. Finding enriched gene sets with searching for enriched gene sets

In our experiments, we use the searching for enriched gene sets (SEGS) method [7] to find interesting subgroups of objects (that is, enriched gene sets). There, a subgroup of objects is considered interesting, when the subgroup is large enough, and its p-value obtained by a statistical test is smaller than the given significance level α.

SEGS uses hierarchies of attribute values (here, terms) to construct subgroups by individual terms as well as by logical conjunctions of terms. Ontologies are extensively used in gene set enrichment [27, 30]. Commonly used ontologies include gene ontology[1] (GO) [31], KO[2] [Kyoto Encyclopedia of Genes and Genomes (KEGG) Orthology] [32] and GoMapMan[3], an extension of the MapMan [33] ontology, for plants.

SEGS combines terms from the same level as well as from different levels into term conjunctions as follows. Several ontologies can be modeled by a single ontology [34]. To construct all possible subgroups, one merged ontology is used, where the root has n children, one for each individual ontology. We start with the root term and recursively replace each term by each of its children.

We are not interested in constructing all possible subgroups, but only those representing at least a minimal number *min* of objects. This parameter *min* is specified by the user. We conjunctively extend a rule condition only if the subgroup defined by it contains more than a minimum number of objects. If a condition defines the same group of objects as a more general condition, the more general condition is deleted. Further, in each recursive step we add other terms to the rule condition to obtain intersections of two or more subgroups.

5.4. Finding contrasting gene sets

To find contrasting gene sets, that is, to find enriched gene sets (interesting subgroups) that are characteristic for their class, we can apply our proposed method described in Section 4.

[1] http://www.geneontology.org/.
[2] http://www.genome.jp/kegg/ko.html.
[3] http://www.gomapman.org/.

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

Note that there are a couple of issues to take into account in the case of gene set enrichment. In Step 1, the classical subgroup discovery, the subgroups can be analyzed by a statistical test, like Fisher's exact test followed by a permutation test or alternatively by GSEA and PAGE in the case of a gene set enrichment application. In Step 2, the user can then choose to contrast different time points or different phenotypes. In Step 3, the contrasting subgroup discovery, we need to analyze the constructed gene sets by a statistical test, like the Fisher's exact test. There, GSEA and PAGE cannot be used for analyzing the constructed gene sets since we analyze the subgroups with respect to two classes P_c and $\overline{P_c}$ (and not with respect to the differential expression which would provide a ranking for GSEA and PAGE).

6. EXPERIMENTS AND RESULTS

For our experiments, we used a *S. tuberosum* (potato) time labeled gene expression dataset for virus-infected and non-infected plants. *S. tuberosum* is severely damaged by the *potato virus Y*. When infected, the plant shows severe symptoms within 1 week and dies after several weeks. Biologists aim to understand the plants disease response by utilizing gene set enrichment.

The dataset consists of three time points: 1, 3 and 6 days after virus infection when the viral-infected leaves as well as leaves from non-infected plants were collected. The aim is to find enriched gene sets that are common to virus-infected plants compared with non-infected plants and at the same time specific for one or all time points. Hence, we transform the expression values of our four samples (four virus-infected and four non-infected plants) into a class attribute, called differential expression, for each time point separately (see Section 5 for details). Afterward we have three class attributes, one for each time point, and can apply our proposed contrasting subgroup discovery method to contrast the different time points.

Recently, *S. tuberosum*'s genome has been completely sequenced [35], but only a few GO or KEGG annotations of *S. tuberosum* genes exist. However, plenty of GO and KEGG annotations exist for the well-studied model plant *Arabidopsis thaliana*. Therefore, we perform two approaches: First, we use homologs between *S. tuberosum* and *A. thaliana* and ontologies for *A. thaliana*. Second, we build *S. tuberosum* ontologies using homolog sequences in the National Center for Biotechnology Information (NCBI) and their GO annotations. For both approaches, we carried out gene set enrichment experiments in an Orange4WS[4] workflow [36].

Our interest is in assisting biologists to generate new research hypotheses. Therefore, we evaluate our results by counting the quantities of gene sets which are unexpected as well as those that are useful to a plant biologist (as in [37]). In this context, *unexpected* means that the knowledge was contained in GO, KEGG or GoMapMan, but it was not shown previously to be related to *S. tuberosum*'s response to viral infection. A gene set is *useful* if it is of interest for the plant biologist, that is, the gene set description tells him something about the virus response, and/or he might want to have a closer look at the genes of that gene set. We compare the results obtained by our proposed method (Steps 1–3) to those results obtained with a classical subgroup discovery method (Step 1).

[4] http://orange4ws.ijs.si/.

6.1. *A. thaliana* homologs approach

Experimental setting. We use homologs between *S. tuberosum* and *A. thaliana* to make GSEA for *S. tuberosum* possible. There are more than 26 000 homologs for more than 42 000 *S. tuberosum* genes. GSEA is performed based on expression values in the dataset, the gene IDs of the *A. thaliana* homologs, and GO and KEGG annotations for *A. thaliana*.

We restricted gene sets to contain a minimum of three genes ($min = 3$) as only these are biologically relevant, the gene set description to contain a maximum of four terms, and the p-value to be 0.05 or smaller. For analyzing the constructed gene sets obtained by classical subgroup discovery (Step 1), we used Fisher's exact test, GSEA, PAGE and the combined GSEA and PAGE with equal percentages. Fisher's exact test was used to analyze gene set enrichment obtained by contrasting subgroup discovery (Step 3).

We considered two types of contrast classes for gene set enrichment (Step 2). First, the intersection: genes that are common to all classes compared with the genes occurring in some gene sets, but not in all (obtained by Equation (8)). Second, the set differences: genes that are specific for one class compared with the genes of the gene sets of the other classes (obtained by Equation (9)). The choice was made by the plant biologists, who are interested in understanding which biological processes, pathways, etc. are active at all time points, and which are active only at a specific time point.

Results. The quantities of enriched gene sets found with the *A. thaliana* homolog approach are shown in Table 4. The first subgroup discovery instance (Step 1), that is, the classical subgroup discovery method, found only a few, if any, gene sets. All gene sets that were found for the classical subgroup discovery method (Step 1) are described by either

```
protein.synthesis.ribosomal protein.
prokaryotic (GoMapMan:29.2.1.1)
```

more general terms of this gene set description (that is, for example, by `GoMapMan:29.2.1`), or by

```
Plant-pathogen interaction (KEGG:04626)
```

As we construct the set differences and intersection from the gene sets found in Step 1, it is no surprise that also the second subgroup discovery instance (Step 3), the contrasting subgroup discovery method, found only a few, if any, gene sets at all. Some of the gene sets that were found by the contrasting subgroup

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

TABLE 4. Quantities of enriched gene sets found with the classical subgroup discovery (Step 1) and with the contrasting subgroup discovery method (Step 3) for the *A. thaliana* homologs approach with Fisher (F), GSEA (G), PAGE (P) and GSEA and PAGE combined (C).

	F	G	P	C
Classical SD (Step 1)				
Day 1	1	0	2	0
Day 3	0	0	1	0
Day 6	1	0	0	0
Contrasting SD (Step 3)				
Day 1 set difference	6	0	0	0
Day 3 set difference	0	0	0	0
Day 6 set difference	3	0	0	0
Intersection	0	0	0	0

TABLE 5. Quantities of *useful* enriched gene sets found with the *A. thaliana* homologs approach.

	F	G	P	C
Classical SD (Step 1)				
Day 1	0	0	0	0
Day 3	0	0	0	0
Day 6	1	0	0	0
Contrasting SD (Step 3)				
Day 1 set difference	2	0	0	0
Day 3 set difference	0	0	0	0
Day 6 set difference	2	0	0	0
Intersection	0	0	0	0

For the contrasting subgroup discovery (Step 3), only enriched gene sets are counted that are useful as well as new or more specific in comparison to the classical subgroup discovery (Step 1).

discovery method (Step 3) are more specific than those found by the classical subgroup discovery method (Step 1). For instance,

```
protein.synthesis.ribosomal protein.
prokaryotic.chloroplast.50S subunit
(GoMapMan:29.2.1.1.1.2)
```

is more specific than `GoMapMan:29.2.1.1`, which is a term located higher in the term hierarchy. Another example is

```
calmodulin-dependent protein kinase activity
(GO:0004683)
∧ signalling.calcium (GoMapMan:30.3)
∧ Plant-pathogen interaction (KEGG:04626)
```

where `KEGG:04626` became combined with terms from other hierarchies. This combination was not statistically significant for the classical subgroup discovery method (Step 1), but is for the contrasting subgroup discovery method (Step 3), when comparing the contrast sets constructed in Step 2.

No gene sets at all were unexpected using the *A. thaliana* homologs approach. A gene set is *useful* if it is of interest for the plant biologist, that is, the gene set description tells him something about the virus response, and/or he might want to have a closer look at the genes of that gene set. The quantities of unexpected enriched gene sets found using the *A. thaliana* homologs approach are shown in Table 5. The only gene set that is useful for the classical subgroup discovery method (Step 1) is

```
Plant-pathogen interaction (KEGG:04626)
```

which covers 51 genes with a p-value $\leq 10^{-6}$. This gene set description is expected as it describes the plant's defense pathway to disease infections.

Two enriched gene sets were found to be useful for the contrasting subgroup discovery method (Step 3) on the first day:

```
protein.synthesis.ribosomal protein.
prokaryotic.chloroplast
(GoMapMan:29.2.1.1.1)
```

which covers 28 genes with a p-value $\leq 10^{-6}$ and its more specific variant

```
protein.synthesis.ribosomal protein.
prokaryotic.chloroplast.50S subunit
(GoMapMan:29.2.1.1.1.2)
```

which covers 22 genes with a p-value $\leq 10^{-6}$. The more general as well as the more specific gene set description was output as they define different gene sets. More precisely, the gene set of the more specific description is a subset of the gene set of the more general description.

The two enriched and useful gene sets found for the contrasting subgroup discovery method (Step 3) on Day 6 are

```
Plant-pathogen interaction (KEGG:04626)
∧ signalling.calcium (GoMapMan:30.3)
```

which covers 26 genes with a p-value $\leq 10^{-6}$, and

```
Plant-pathogen interaction (KEGG:04626)
∧ signalling.calcium (GoMapMan:30.3)
∧ Calmodulin-dependent protein kinase
activity (GO:0004683)
```

which is more specific than the previous one, and covers only 14 genes with a p-value of 0.0001. All these gene sets are described by more specific concepts than those found with the classical subgroup discovery method (Step 1) and hence give the plant biologists more detailed information.

For the intersection in Step 3, we obtained no enriched gene sets at all. This reflects the characteristics of a defense response: The gene expression of the first days (when activating the defense response) differs from the gene expression on Day 6 (when the defense response is active) and therefore the intersection reveals no enriched gene sets that are active at all time points.

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

TABLE 6. Quantities of enriched gene sets found with the classical (Step 1) and with the contrasting subgroup discovery method (Step 3) for the *S. tuberosum* GO approach with Fisher (F), GSEA (G), PAGE (P), and GSEA and PAGE combined (C).

	F	G	P	C
Classical SD (Step 1)				
Day 1	7	2	5	2
Day 3	2	0	5	0
Day 6	3	1	12	1
Contrasting SD (Step 3)				
Day 1 set difference	16	3	15	3
Day 3 set difference	3	0	15	0
Day 6 set difference	29	2	29	2
Intersection	0	0	0	0

6.2. *S. tuberosum* GO approach

Experimental setting. We built *S. tuberosum* ontologies independently using Blast2GO[5] to obtain homolog sequences in NCBI and their GO annotations. Enrichment analysis is then performed using *S. tuberosum*'s gene IDs and expression values, and GO and KEGG annotations obtained with Blast2GO.

Again, we restricted gene sets to contain a minimum of three genes ($min = 3$), the gene set description to contain a maximum of four terms and the p-value to be 0.05 or smaller. For analyzing the constructed gene sets, we used the Fisher's exact test, GSEA, PAGE and the combined GSEA and PAGE with equal percentages, in Step 1, and Fisher's exact test in Step 3. We considered the same two types of contrast classes for gene set enrichment (Step 2) as in the *A. thaliana* approach: the intersection (Equation (8)) and the set differences (obtained by Equation (9)).

Results. The quantities of enriched gene sets found with the *S. tuberosum* GO approach are shown in Table 6. In comparison with the *A. thaliana* approach, we found more enriched gene sets. This is probably due to the following reason: Many potato genes have no homologs in *A. thaliana* or the homologs are not known yet, but with the *S. tuberosum* GO approach we obtain extensive GO annotation of the genes.

However, when using GSEA (either alone or in combination with PAGE) to analyze the constructed gene sets of the first subgroup discovery instance (Step 1), that is, the classical subgroup discovery method, only a few more enriched gene sets are found. When Fisher's exact test or PAGE are used instead, more enriched gene sets are found. This suggests that especially in the *S. tuberosum* gene ontology approach one of these methods should be preferred.

When PAGE is used, several enriched gene sets are found on Day 6 by the classical subgroup discovery method (Step 1). Even more enriched gene sets are found by the contrasting subgroup discovery method (Step 3) on Day 6 when PAGE or Fisher's exact test are used in Step 1. (As stated before, in Step 3 always Fisher's exact test is used to analyze the constructed gene sets.) The fact that more enriched gene sets are found on Day 6 reflects that *S. tuberosum* activates the defense response in the first days, and the full effect can be witnessed only on Day 6.

Several gene sets that are known to relate to *S. tuberosum*'s response to virus infection were found, including molecular functions, biological processes and pathways with a central role in it, such as

```
auxin mediated signalling pathway
(GO:0009734)
```

which covers 42 genes with a p-value $\leq 10^{-6}$,

```
fatty acid catabolic process (GO:0009062)
∧ lipid metabolism.lipid degradation.
beta-oxidation (GoMapMan:11.9.4)
```

which covers 17 genes with a p-value of 0.0001, and

```
protein.postranslational modification
(GoMapMan:29.4)
∧ protein serine/threonine phosphatase
complex (GO:0008287)
```

which covers 16 genes with a p-value of 0.0001.

As before, we counted the quantities of enriched gene sets that are unexpected to a plant biologist when using the *S. tuberosum* GO approach (see Table 7). In contrast to the *A. thaliana* approach, we found some enriched genes set that are unexpected. For the classical subgroup discovery method (Step 1), we found unexpected gene sets only on the first day, which all relate to the Golgi complex, such as

```
protein.targeting.secretory pathway.golgi
(GoMapMan:29.3.4.2)
```

which covers 19 genes with a p-value $\leq 10^{-6}$.

TABLE 7. Quantities of *unexpected* enriched gene sets found with the *S. tuberosum* GO approach.

	F	G	P	C
Classical SD (Step 1)				
Day 1	1	2	1	2
Day 3	0	0	0	0
Day 6	0	0	0	0
Contrasting SD (Step 3)				
Day 1 set difference	0	1	4	1
Day 3 set difference	0	0	0	0
Day 6 set difference	2	1	0	0
Intersection	0	0	0	0

For the contrasting subgroup discovery (Step 3), only enriched gene sets are counted that are unexpected as well as new or more specific in comparison to the classical subgroup discovery (Step 1).

[5] http://www.blast2go.org/.

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

For the contrasting subgroup discovery method (Step 3), we found unexpected gene sets for the first and sixth day. Some of those relate also to the Golgi complex, but were not found with the classical subgroup discovery method (Step 1), such as

```
ER to Golgi vesicle-mediated transport
(GO:0006888)
∧ vesicle coat (GO:0030120)
```

which covers 14 genes with a p-value of 0.0001. Other examples of unexpected gene sets are novel when compared with the enriched gene sets found by the classical subgroup discovery method (Step 1). Hence, they might reveal new research hypotheses for the plant biologists. Examples of such gene sets are

```
RNA.regulation of transcription.Chromatin
Remodeling Factors (GoMapMan:27.3.44)
```

which covers 15 genes with a p-value $\leq 10^{-6}$,

```
unidimensional cell growth (GO:0009826)
```

which covers 7 genes with a p-value of 0.0001 or

```
root development (GO:0048364)
∧ hormone metabolism.auxin (GoMapMan:17.2)
```

which covers 5 genes with a p-value of 0.001.

As before, we also counted the quantities of gene sets that are useful to a plant biologist when using the *S. tuberosum* GO approach (see Table 8).

An enriched gene set that was found by the classical subgroup discovery method (Step 1) and is considered useful is

```
protein.targeting.secretory pathway.golgi
(GoMapMan:29.3.4.2)
```

which covers 19 genes with a p-value $\leq 10^{-6}$. Another example is

TABLE 8. Quantities of *useful* enriched gene sets found with the *S. tuberosum* GO approach.

	F	G	P	C
Classical SD (Step 1)				
Day 1	3	2	3	2
Day 3	0	0	3	0
Day 6	3	1	12	1
Contrasting SD (Step 3)				
Day 1 set difference	6	1	6	1
Day 3 set difference	1	0	6	0
Day 6 set difference	22	1	18	1
Intersection	0	0	0	0

For the contrasting subgroup discovery (Step 3), only enriched gene sets are counted that are useful as well as new or more specific in comparison with the classical subgroup discovery (Step 1).

```
RNA.regulation of transcription.
WRKY domain transcription factor family
(GoMapMan:27.3.32)
```

which covers 30 genes with a p-value of 0.0001.

Useful gene sets found by the contrasting subgroup discovery method (Step 3), which are novel or more specific when compared with the classical subgroup discovery method (Step 1) include

```
protein.degradation.ubiquitin.E3.SCF.FBOX
(GoMapMan:29.5.11.4.3.2)
```

which covers 40 genes with a p-value $\leq 10^{-6}$,

```
enoyl-CoA hydratase activity (GO:0004300)
```

which covers 7 genes with a p-value $\leq 10^{-6}$,

```
post-embryonic development (GO:0009791)
∧ reproductive structure development
(GO:0048608)
∧ RNA (GoMapMan:27)
```

which covers 21 genes with a p-value $\leq 10^{-6}$, and

```
ER to Golgi vesicle-mediated transport
(GO:0006888)
∧ vesicle coat (GO:0030120)
```

which covers 14 genes with a p-value of 0.0001. From these four gene sets, the last one is more specific, while the first three gene sets are novel when compared with the classical subgroup discovery method (Step 1).

Note that a gene set can be either expected and not useful, unexpected but not useful, expected, but useful, or both, unexpected as well as useful. Gene sets that are expected as well as not useful might be simply described by too general terms, such as

```
protein.postranslational modification
(GoMapMan:29.4)
```

which covers 217 genes with a p-value $\leq 10^{-6}$. A gene set can be expected and not useful also because it is not informative for some other reason, such as

```
coated vesicle membrane (GO:0030662)
```

which covers 27 genes with a p-value of 0.0001, but is not informative to plant biologists as it describes a cellular component only. An example of an enriched gene set that is unexpected, but not useful is

```
organ development (GO:0048513)
∧ RNA (GoMapMan:27)
```

which covers 21 genes with a p-value $\leq 10^{-6}$. It is not useful because the biological term it is too general.

Gene sets that are unexpected, useful or both may contain genes that are interesting for further (tough, time-consuming) wet-lab experiments. From the gene sets mentioned before, an example of an enriched gene set that is expected but useful is

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

```
RNA.regulation of transcription.
WRKY domain transcription factor family
(GoMapMan:27.3.32)
```

which is expected as it is known that these proteins have an important role in defense against virus, but still useful as it tells the plant biologist that these proteins are differentially expressed on Day 6. Examples of enriched gene sets that are unexpected and useful are

```
post-embryonic development (GO:0009791)
∧ reproductive structure development
(GO:0048608)
∧ RNA (GoMapMan:27)
```

and

```
ER to Golgi vesicle-mediated transport
(GO:0006888)
∧ vesicle coat (GO:0030120)
```

These rules combine two or more ontology terms that have not been associated with the viral infection response of plants (to the knowledge of the plant biologists). Therefore, the genes covered by these gene set descriptions are potentially interesting to the plant biologists and might help them to generate new hypotheses.

As in the *A. thaliana* approach, we did not obtain any enriched gene sets for the intersection in Step 3. Again, this reflects the characteristics of a defense response: The gene expression of the first days differs from the gene expression on Day 6.

7. CONCLUSIONS

We defined the problem of contrasting subgroup discovery; that is, the aim is to find subgroups of objects characteristic for their class, even if the classes are not identical. Further, we allow the user to specify contrast classes in which they are interested, for example, to contrast several time points. We proposed to find such subgroups by combining well-known algorithms. We showed that our approach finds subgroups of objects that are characteristic for their class, even if the classes are not identical. Our results on a time series dataset for virus-infected *S. tuberosum* (potato) plants indicate that such subgroups can be unexpected and useful for biologists. Studying the genes of such subgroups may reveal new research hypotheses for biologists.

Further experimental evaluation is planned, including an extensive evaluation of the quality of gene set descriptions which possibly relate to *S. tuberosum*'s virus response, but are unexpected for a plant biologist. Further, we will address the redundancy of gene set descriptions, and we will investigate how redundancy can be avoided, or at least decreased, for example, by rule clustering or filtering. In addition, we will evaluate the results at the gene level, including a selection of genes for wet-lab experiments, which will affect the understanding of the biological mechanisms of virus response, particularly that of *S. tuberosum*. Finally, we will perform further experiments on other, non-biological datasets and use simple as well as more complex set-theoretic functions.

ACKNOWLEDGEMENTS

We would like to thank Kamil Witek, Ana Rotter and Špela Baebler for the test data and the help with interpreting the results.

FUNDING

This work has been supported by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898, by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland and by the Slovenian Research Agency grants P2-0103, J4-2228 and P4-0165.

REFERENCES

[1] Klösgen, W. (1996) Explora: A Multipattern and Multistrategy Discovery Assistant. In Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. (eds), *Advances in Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park, CA, USA.

[2] Wrobel, S. (1997) An Algorithm for Multi-relational Discovery of Subgroups. In Komorowski, J. and Zytkow, J. (eds), *Principles of Data Mining and Knowledge Discovery*. Springer, Berlin.

[3] Bruner, J., Goodnow, J. and Austin, G. (1956) *A Study of Thinking*. Wiley, Hoboken, NJ, USA.

[4] Michalski, R. (1983) A theory and methodology of inductive learning. *Artif. Intell.*, **20**, 111–161.

[5] Gruber, T. (1995) Toward principles for the design of ontologies used for knowledge sharing. *Int. J. Hum.-Comput. Stud.*, **43**, 907–928.

[6] Weber, I. (2000) Levelwise search and pruning strategies for first-order hypothesis spaces. *J. Intell. Inf. Syst.*, **14**, 217–239.

[7] Trajkovski, I., Lavrač, N. and Tolar, J. (2008) SEGS: search for enriched gene sets in microarray data. *J. Biomed. Inform.*, **41**, 588–601.

[8] van Belle, G., Fisher, L., Heagerty, P. and Lumley, T. (1993) *Biostatistics: A Methodology for the Health Sciences*. Wiley, Hoboken, NJ, USA.

[9] Li, J., Liu, G. and Wong, L. (2007) Mining statistically important equivalence classes and delta-discriminative emerging patterns. *Proc. KDD '07*, San Jose, CA, USA, August 12–15, pp. 430–439. ACM Press, New York City, NY, USA.

[10] Agrawal, R., Imieliński, T. and Swami, A. (1993) Mining Association Rules Between Sets of Items in Large Databases. *Proc. SIGMOD '93*, Washington, DC, USA, May 26–28, pp. 207–216. ACM Press, New York City, NY, USA.

[11] Mielikäinen, T. (2003) Intersecting Data to Closed Sets with Constraints. *Proc. FIMI '03*, Melbourne, FL, USA, Novermber 19, CEUR-WS.org, http://ceur-ws.org/Vol-90/mielikainen.pdf.

[12] Pan, F., Cong, G., Tung, A., Yang, J. and Zaki, M. (2003) Carpenter: Finding Closed Patterns in Long Biological Datasets.

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

Proc. KDD '03, Washington, DC, USA, August 24–27, pp. 637–642. ACM Press, New York City, NY, USA.

[13] Borgelt, C., Yang, X., Nogales-Cadenas, R., Carmona-Saez, P. and Pascual-Montano, A. (2011) Finding Closed Frequent Item Sets by Intersecting Transactions. *Proc. EDBT/ICDT '11*, Uppsala, Sweden, March 21–25, pp. 367–376. ACM Press, New York City, NY, USA.

[14] Dong, G. and Li, J. (1999) Efficient Mining of Emerging Patterns: Discovering Trends and Differences. *Proc. KDD '99*, pp. 43–52. ACM Press, New York City, NY, USA.

[15] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. and Verkamo, A. (1996) Fast Discovery of Association Rules. In Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. and Uthurusamy, R. (eds), *Advances in Knowledge Discovery and Data Mining*. AAAI Press, Menlo Park, CA, USA.

[16] Bay, S. and Pazzani, M. (2001) Detecting group differences: mining contrast sets. *Data Min. Knowl. Discov.*, **5**, 213–246.

[17] Böttcher, M. (2011) Contrast and change mining. *Data Min. Knowl. Discov.*, **1**, 215–230.

[18] Suzuki, E. (1997) Autonomous Discovery of Reliable Exception Rules. *Proc. KDD '97*, Newport Beach, CA, USA, August 14–17, pp. 259–262. AAAI Press, Menlo Park, CA, USA.

[19] Webb, G., Butler, S. and Newlands, D. (2003) On Detecting Differences between Groups. *Proc. KDD '03*, Washington, DC, USA, August 24–27, pp. 256–265. ACM Press, New York City, NY, USA.

[20] Hilderman, R. and Peckham, T. (2007) Statistical Methodologies for Mining Potentially Interesting Contrast Sets. In Guillet, F. and Hamilton, H. (eds), *Quality Measures in Data Mining*. Springer, Berlin.

[21] Kralj Novak, P., Lavrač, N. and Webb, G. (2009) Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. *J. Mach. Learn. Res.*, **10**, 377–403.

[22] Azevedo, P. (2010) Rules for contrast sets. *Intell. Data Anal.*, **14**, 623–640.

[23] del Jesus, M., Gonzalez, P., Herrera, F. and Mesonero, M. (2007) Evolutionary fuzzy rule induction process for subgroup discovery: a case study in marketing. *Trans. Fuzzy Syst.*, **15**, 578–592.

[24] May, M. and Ragia, L. (2002) Spatial Subgroup Discovery Applied to the Analysis of Vegetation Data. In Karagiannis, D. and Reimer, U. (eds), *Practical Aspects of Knowledge Management*. Springer, Berlin.

[25] Allison, D., Cui, X., Page, G. and Sabripour, M. (2006) Microarray data analysis: from disarray to consolidation and consensus. *Nat. Revi. Genet.*, **5**, 55–65.

[26] Subramanian, A. *et al.* (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl Acad. Sci.*, **102**, 15545–15550.

[27] Kim, S.-Y. and Volsky, D. (2005) PAGE: parametric analysis of gene set enrichment. *BMC Bioinform.*, **6**, 144.

[28] Cui, X. and Churchill, G. (2003) Statistical tests for differential expression in cDNA microarray experiments. *Gen. Biol.*, **4**, 210.1–210.10.

[29] Baldi, P. and Long, A. (2001) A Bayesian framework for the analysis of microarray expression data: regularized *t*-test and statistical inferences of gene changes. *Bioinformatics*, **17**, 509–519.

[30] Mootha, V. *et al.* (2003) PGC-1α-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.*, **34**, 267–273.

[31] Khatri, P. and Drăghici, S. (2005) Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, **21**, 3587–3595.

[32] Aoki-Kinoshita, K. and Kanehisa, M. (2007) Gene Annotation and Pathway Mapping in KEGG. In Walker, J. and Bergman, N.H. (eds), *Comparative Genomics*. Humana Press, New York City, NY, USA.

[33] Thimm, O., Bläsing, O., Gibon, Y., Nagel, A., Meyer, S., Krüger, P., Selbig, J., Müller, L., Rhee, S. and Stitt, M. (2004) MapMan: a user-driven tool to display genomics datasets onto diagrams of metabolic pathways and other biological processes. *Plant J.*, **37**, 914–939.

[34] Srikant, R. and Agrawal, R. (1995) Mining Generalized Association Rules. *Proc. VLDB '95*, Zurich, Switzerland, September 11–15, pp. 407–419. Morgan Kaufmann Publishers, San Francisco, CA, USA.

[35] The Potato Genome Sequencing Consortium (2011) Genome sequence and analysis of the tuber crop potato. *Nature*, **475**, 189–195.

[36] Podpečan, V. *et al.* (2011) SegMine workflows for semantic microarray data analysis in Orange4WS. *BMC Bioinform.*, **12**, 416.

[37] Suzuki, E. and Tsumoto, S. (2000) Evaluating Hypothesis-Driven Exception-Rule Discovery with Medical Data Sets. *Proc. PADKK '00*, Kyoto, Japan, April 18–20, pp. 208–211. Springer, Berlin.

[38] Westfall, P. and Young, S. (1993) *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*. Wiley, Hoboken, NJ, USA.

[39] Bender, R. and Lange, S. (2001) Adjusting for multiple testing—when and how? *J. Clin. Epidemiol.*, **54**, 343–349.

[40] Ge, Y., Dudoit, S. and Speed, T. (2003) Resampling-based multiple testing for microarray data analysis. *Test*, **12**, 1–77.

[41] Holm, S. (1979) A simple sequentially rejective multiple test procedure. *Scand. J. Stat.*, **6**, 65–70.

[42] Benjamini, Y. and Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. B (Methodol.)*, **57**, 289–300.

[43] Vavpetič, A. and Lavrač, N. (2012) Semantic subgroup discovery systems and workflows in the SDM-toolkit. *Comput. J.*, advance access published online 4 June 2012.

APPENDIX. PERMUTATION TEST

Subgroup discovery methods typically evaluate a large number of potentially interesting subgroups. It is possible that some of them are apparently statistically significant just by chance. To address the multiple testing problem, that is, to control the type I error (false positive) rates, we perform a permutation test to obtain adjusted *p*-values (see, e.g. [38–40]). We randomly permute the classes (class attribute values) and calculate the *p*-value for each subgroup. We repeat this first step for 10 000

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

permutations, create a histogram by the p-values of each permutation's best subgroup and estimate the (corrected) p-value of the original subgroups using the histogram: The corrected p-value is the relative number of permutations, including the original one, in which the best p-value is smaller or equal to the original p-value. This approach returns only an approximation of the exact p-values, which is sufficient enough for our application, where we primarily use the resulting corrected p-values to rank the subgroups. For stronger statistical tests, one can use a method such as Holm's simple sequentially rejective multiple test procedure [41] or the false discovery rate [42] instead.

Downloaded from http://comjnl.oxfordjournals.org/ by guest on September 28, 2012

5 Summary and Further Work

In this thesis, we have conceived and developed a modern software platform, based on principles of service-oriented architecture and design, scientific workflow and visual programming. The developed platform enabled us to design and implement novel knowledge discovery scenarios from different scientific domains. Most notably, we have developed the SegMine methodology which allows for semantic analysis of microarray experimental data, and the contrasting subgroup discovery (CSD) methodology, which allows for finding contrasting patterns in data that can not be found using classical subgroup discovery.

In the rest of this chapter the achievements and main scientific contributions of the thesis are briefly summarised. We conclude by discussing directions of further work and improvements of the presented software platform, developed methodologies, their constituents and implementations.

5.1 Summary

Modern data mining algorithms and platforms are faced with the ever increasing amount and complexity of heterogeneous data and information sources, semantically annotated content, ontologies, distributed software components and novel computer system configurations and environments. Some of these challenges are addressed in this thesis and some are discussed as potential directions for further work.

The goal of the presented software platform Orange4WS is to utilise, combine and interconnect the principles of service-orientation with web services, interactive scientific workflows, visual programming and interactive visualisations, open-source machine learning software, knowledge discovery ontology and automated construction of data mining workflows in order to create a modern and easy-to-use scientific environment where novel knowledge discovery scenarios can be designed, implemented and executed.

The platform, described in Chapter 2, is based on Orange (Demšar et al., 2004), a mature open-source data mining toolkit, and thus benefits from all available Orange's components and features. At the lowest implementation level, Orange4WS makes use of a large number of machine learning, optimisation and data preprocessing algorithms, efficiently implemented in C++. Second, at the scripting layer (the Python programming language), Orange offers numerous procedures ranging from data management (loading, saving, preprocessing, and transformation) to statistical evaluation procedures and numerical methods, all of which are inherited by Orange4WS. Finally, the most important feature of Orange, also inherited and extended in Orange4WS, is the interactive workflow editor which allows for user-friendly construction of scientific workflows. The workflow editor enables the drag-drop-connect visual programming paradigm in a component called *Orange Canvas* where more complex procedures can be constructed by connecting GUI elements called widgets which implement various functions, procedures, visualisations, and user interaction.

Orange4WS extends and upgrades Orange with the following features. First, it enables seamless integration of web services into the Orange Canvas by implementing a code generator

which can produce a valid Orange Canvas widget from any SOAP 1.1 compatible web service. As a result, web services can be easily composed into complex workflows by the user while Orange4WS takes care of service invocation, message passing, error handling, and results retrieval. This feature has enabled us to effectively construct novel knowledge discovery scenarios which utilise web services which are already available on the internet, as well as our newly developed web services. Second, Orange4WS implements tools for web service development, which adhere to the contract-first design principle. Using these tools we have successfully designed and developed different web services, most notably, the SEGS algorithm web service and its variants for plants, the relational subgroup discovery (RSD) web services, Weka web services which expose some parts of the Weka machine learning software, text mining services, and services which implement natural language processing methods for extracting triplets from biological literature. Third, Orange4WS is enriched with the knowledge discovery ontology (KD ontology), implemented in OWL-DL, which defines relationships among the components of knowledge discovery scenarios, both declarative (various knowledge representations) and algorithmic. The KD ontology enables the workflow planner to reason about which algorithms can be used to produce the results required by a specified knowledge discovery task and to query the results of knowledge discovery tasks. Finally, Orange4WS allows for automated construction of abstract data mining workflows using a planning algorithm which queries the KD ontology. We have successfully tested and evaluated this feature of Orange4WS by generating workflows for a text mining scenario and discovery of association rules.

Orange4WS has enabled the construction and execution of various knowledge discovery scenarios. The two most important scenarios presented in this thesis are the SegMine methodology and contrasting subgroup discovery. Both are focused on the discovery of knowledge from experimental data in the systems biology domain. In the rest we briefly summarise both methodologies, their goals and scientific contributions.

The objective of the SegMine methodology is to help domain experts (i.e., biologists) in the analysis of gene expression data from microarray experiments and in the process of formulating research hypotheses. SegMine achieves this by integrating a semantic subgroup discovery algorithm, public ontologies, an interactive clustering component, and probabilistic link discovery search engine in an easy-to-use interactive workflow environment.

Essentially, the SegMine methodology, described in Chapter 3, consists of four steps each of which gives the domain expert valuable information extracted from the experimental data. First, the data is preprocessed (e.g. handling repeated measures and missing data) and the genes are ranked using one of the well-known approaches. This indicates which genes seem important according to the experimental data and allows for filtering of genes with little variability across samples. Second, SegMine integrates the SEGS algorithm (as a web service) which is used to identify differentially expressed gene sets. While there are numerous algorithms and systems available to accomplish this task the SEGS algorithm offers unique ability to discover gene groups, described by rules formulated as conjuncts of ontology terms from public ontologies (GO, KEGG, and Entrez). Such rules are of special value for domain experts as they semantically explain differentially expressed gene groups in terms of gene functions, components, processes, and pathways. Third, hierarchical rule clustering is performed interactively by the user. This aids greatly to comprehend the list of rules (which can be very long) by grouping them into a small number of groups according to the genes they cover. Finally, the last step of the SegMine methodology is link discovery and graph visualisation. This step complements the results, mined from experimental data using public ontologies, with probabilistic graph search, performed on data from several public databases which are integrated into a large graph. Probabilistic graph search is performed by the Biomine system which is available in Orange4W as a set of web

services. The Biomine system links the results of the SEGS algorithm to the data from public databases and thus enables a unique data interpretation environment which enables the domain expert to formulate new research hypotheses. For example, we have successfully applied SegMine to the analysis of human mesenchymal stem cell (MSC) data and were able to formulate three new scientific hypotheses. Moreover, one of the hypotheses (all of them were derived from an older (2008) dataset), may even substantiate a recent independently derived proposition based on newer (2010) senescence gene expression data (Schallmoser et al., 2010). To summarise, the SegMine methodology is a powerful tool in the hands of a domain expert as it offers improved data interpretation and hypothesis generation in an interactive scientific workflow environment.

Contrasting subgroup discovery, as described in Chapter 4, can be thought of as a generalisation of subgroup discovery where the goal is to search for population subgroups which are statistically interesting and which exhibit unusual distributional characteristics with respect to the property of interest (Klösgen, 1996). However, subgroup discovery methods can only reveal sets of objects characteristic for some class. On the other hand, using contrasting subgroup discovery we aim to find sets of objects where each object is characteristic for its class, but the classes are not necessarily identical. This is achieved using a three step approach. In the first step, classical subgroup discovery is applied. In the second step, the user defines two new classes of objects which are to be contrasted. In the third step, classical subgroup discovery is again applied to discover interesting subgroups of these two classes. Clearly, the resulting subgroups of the two contrasting classes contain objects which are characteristic for their (original) classes which are not necessarily identical.

The key of contrasting subgroup discovery is the formulation of two contrast classes, P_c and its complement $\overline{P_c}$. Their formulation depends of the objective and has to be specified by the user. Some of the typical choices are as follows. For example, if we are interested in finding interesting subgroups that are common to all class attributes, the contrast class is defined as the set of objects occurring in interesting subgroups of all class attributes. Or, for example, if we are interested in finding interesting subgroups that are specific for one class attribute in contrast to all the other class attributes, the contrast class is the set difference between the interesting objects of the selected class attribute and the rest. The complement contrast class $\overline{P_c}$ can be a true complement of P_c or can be more specific according to the user's contrasting objective.

The proposed contrasting subgroup methodology can be applied in various application areas. In our work we have employed the methodology, implemented as a collection of workflow components in Orange4WS, to the problem of finding enriched gene sets that are specific for virus-infected samples for a specific time point or a specific phenotype. As in the case of SegMine, our implementation also employs the SEGS algorithm, implemented as a web service and specially adapted for experimental gene expression data for plants. Our experiments, performed on a time series dataset for virus-infected Solanum tuberosum plants, indicate that the methodology can lead to the discovery of subgroups which are unexpected and useful for biologists, and may lead to new scientific research hypotheses. For example, we have obtained gene sets that – to the best domain experts' best knowledge – have not (yet) been associated with the viral infection response of plants.

5.2 Further work

In this thesis we have developed a modern knowledge discovery platform which served as a basis for the implementation of novel methodologies which allow for advanced analyses of experimental data that have the potential to lead to new scientific discoveries. In the rest of this section we first discuss the directions for further work related to the software platform followed by a

discussion on the potential directions for further work on the developed methodologies, SegMine and contrasting subgroup discovery.

5.2.1 Orange4WS

Orange4WS is a platform independent software tool which benefits from service-oriented technologies, especially SOAP-based web services. While it offers a fairly complete set of features, there are several lines of further development.

First, the existing support for automated integration of web services in Orange4WS is focused on SOAP web services with WSDL descriptions. However, currently well-supported WSDL 1.1 standard supported by Orange4WS only provides means of describing SOAP and HTTP GET/POST services. REST web services, for example, which are becoming increasingly popular due to their simplicity, are only supported by WSDL 2.0 and WADL[1] which are still poorly supported in libraries for web service development. Although Orange4WS can utilise REST web services, service widgets are not generated automatically. For example, the Biomine web services, which are an essential part of the SegMine methodology, are REST style services and were integrated manually by implementing the corresponding widgets. The integration of programming libraries, such as *wadllib*[2] and *suds*[3] in Orange4WS could thus greatly improve the support for the latest web service technologies and give access to new data and processing resources.

Second, the annotations of web services and widgets in the KD ontology used in Orange4WS currently needs to be performed manually which requires a certain level of expertise related to the KD ontology and components' internals. Semi-automated annotation, coupled with an interactive visual annotation component, such as Visual OntoBridge (Grčar et al., 2012), could significantly simplify and speed up the annotation of new services and components. The workflow planner can also be improved by taking into account user-defined constraints and preferences. Moreover, by incorporating an optimisation ontology, such as DMOP[4] (Hilario et al., 2009), and databases of past data mining experiments, parameters and results, Orange4WS could gain the ability of optimising the data mining process represented by a given workflow. Integration of deep/heavy-weight ontologies for data mining ontologies, such as OntoDM (Panov, 2012), and ontologies of general purpose data types, such as OntoDT (Panov et al., 2009) would bring more generality into Orange4WS although this would require nontrivial modifications and adaptations.

Finally, in recent years the focus of software development has noticeably shifted towards mobile, system independent solutions which are primarily based on web technologies, such as dynamic, interactive web pages, client-side scripting, HTML5 Canvas, WebGL[5], Ajax[6], and WebSocket[7]. Therefore, it is reasonable to expect that the development of data analysis and mining software will also gradually shift towards mobile computing. While we cannot expect that mobile platforms will ever provide sufficient computing power for large data analysis they can provide mobile, independent interfaces to data analysis software which runs on powerful server

[1] Web Application Description Language

[2] `https://launchpad.net/wadllib`

[3] `https://fedorahosted.org/suds/`

[4] `http://www.e-lico.eu/DMOP.html`

[5] Web Graphics Library (WebGL) is a JavaScript API for rendering interactive 3D graphics and 2D graphics without the use of technology-specific browser plugins.

[6] Asynchronous JavaScript and XML (Ajax) is a web development technique used on the client-side to create asynchronous web applications.

[7] WebSocket is a web technology for bi-directional communications most typically implemented in web browsers and web servers.

equipment or even in a cloud. Our recent efforts on further Orange4WS development have thus focused on an entirely new workflow environment that is platform, system and browser independent, and can run on modern mobile devices, including smartphones and tablets. The ClowdFlows (Kranjc et al., 2012a) platform is an attempt to implement a state of the art browser-based scientific workflow environment with the emphasis on data and stream mining.

5.2.2 SegMine

The SegMine methodology enables the analysis of gene expression data by integrating several components of which the SEGS algorithm and the Biomine system are the most important. The SEGS algorithm, which discovers gene sets described as conjunctions of ontology terms from GO, KEGG and Entrez, employs Fisher's exact test, GSEA, and PAGE to assess gene set enrichment. SEGS requires a cutoff parameter which is set by the user and employed in search space pruning, as well as in the Fisher's exact test. The later can be replaced or complemented by a threshold-free method to give a better aggregated scoring of gene sets. For example, in the minimum hyper-geometric (mHG) score (Eden, 2007; Eden et al., 2007), which is implemented in the GOrilla (Eden et al., 2009) gene set enrichment analysis tool, the cutoff between the top of the list and the rest is chosen in a data driven manner which maximises the enrichment. Moreover, as the computation of the mHG score introduces multiple hypothesis testing, the exact p-values are always computed to assess the statistical significance of the enrichment. Therefore, instead of a threshold parameter the user only needs to specify the desired p-value limit. mHG score computation can be easily incorporated into the SEGS procedure to either complement current gene set evaluation methods or to replace Fisher's exact test.

The second main component, the independently developed Biomine system, also offers several lines for improvement. Biomine is currently focused on human genetics and uses somewhat limited number of curated sources (9 databases as of October 2012). A prototype version of Biomine for plant data using the GoMapMan ontology (Baebler et al., 2010) and other relevant knowledge sources has already been developed, but requires further development and evaluation. Several other data sources including publicly available full text publications from PubMed would provide relevant knowledge to be included into Biomine's probabilistic graph, for example, by using natural language processing techniques, such as the extraction of triplets. A more detailed proposal for further Biomine development is discussed by Biomine's authors Eronen and Toivonen (Eronen and Toivonen, 2012).

5.2.3 Contrasting subgroup discovery

Contrasting subgroup discovery methodology upgrades subgroup discovery and allows for finding subgroups of objects that could not be found with classical subgroup discovery. Therefore, its application in several domains where subgroup discovery has been traditionally applied could lead to the discovery of novel patterns. Our current experiments with contrasting subgroup discovery were focused on the domain of biology where we applied the methodology to analyse time-labelled gene expression data. A natural further development is to use the methodology to analyze non-biological data and to develop set-theoretic functions defining contrast classes suitable for a given data analysis task. With respect to the current biological experiments, further work could also address the redundancy in gene set descriptions (rules obtained by the SEGS algorithm) and how to reduce or eliminate it by filtering or clustering, for example. Contrasting subgroup discovery could also be combined with subgroup explanation approaches (Vavpetič et al., 2012) to provide semantic explanations of discovered contrasting subgroups.

6 Acknowledgements

I am sincerely and heartily grateful to my supervisor Nada Lavrač for her guidance, support, and ideas which made this thesis possible.

I would like to thank Monika Zemenova for her great work on automated workflow construction in Orange4WS, and Laura Langohr for the development of the contrasting subgroup discovery methodology and fruitful collaboration during her six-month visit at the Jožef Stefan Institute.

Many thanks to the colleagues from the National Institute of Biology, especially Kristina Gruden, Helena Motaln and Marko Petek for expert interpretation of results, discussions, ideas, comments, and continuous collaboration.

I am thankful to Joost Kok and Jeroen de Bruin from Leiden University, who contributed numerous ideas to this thesis and co-organised a series of successful ECML PKDD workshops on service-oriented knowledge discovery.

The SegMine methodology would not have been possible without the Biomine search engine developed at the University of Helsinki. I wish to thank Hannu Toivonen and his research group for their work on the Biomine system and our successful and friendly collaboration.

At the Jožef Stefan Institute, Igor Mozetič supervised the development and testing of SegMine in the productive and successful collaboration with the National Institute of Biology. I am truly grateful for his efforts.

I wish to acknowledge the significant contribution of Igor Trajkovski to the SegMine methodology by his continuous development of the semantic subgroup discovery algorithm.

I would also like to thank the developers of Orange from the Faculty of Computer and Information Science, Ljubljana, for their hard work on this great open source data mining platform.

It is a great pleasure to thank my colleagues from the Department of Knowledge Technologies at the Jožef Stefan Institute. In long (and sometimes stormy) discussions they contributed ideas and solutions many of which have been incorporated into this thesis.

I wish to thank several funding bodies for the financial support of my study: the Slovenian Research Agency, the Jožef Stefan Institute and the Department of Knowledge Technologies, and the European Commission (through the research projects BISON – Bisociation networks for creative information discovery and ENVISION – Environmental services infrastructure with ontologies).

Finally, I owe sincere and earnest thankfulness to my family and friends for their support, understanding, and patience throughout my study.

7 References

Agrawal, R.; Faloutsos, C.; Han, J.; Kargupta, H.; Kumar, V.; Motwani, R.; Yu, P. S. (eds.). *National Science Foundation Symposium on Next Generation of Data Mining and Cyber-Enabled Discovery for Innovation (NGDM'07): Final Report.* (Unpublished proceedings, 2007). `http://www.cs.umbc.edu/~hillol/NGDM07/abstracts/NGDM07-Report.pdf` (Accessed: January, 2013).

Agresti, A. A survey of exact inference for contingency tables. *Statistical Science* **7**, 131–153 (1992).

Ali, A. S.; Rana, O. F.; Taylor, I. J. Web services composition for distributed data mining. In: *Proceedings of the 34th International Conference on Parallel Processing Workshops (ICPP 2005 Workshops)*. 11–18 (IEEE Computer Society, Washington, DC, 2005).

Altintas, I.; Berkley, C.; Jaeger, E.; Jones, M. B.; Ludäscher, B.; Mock, S. Kepler: An extensible system for design and execution of scientific workflows. In: *Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM 2004)*. 423–424 (IEEE Computer Society, Washington, DC, 2004).

Alves, A.; Arkin, A.; Askary, S.; Barreto, C.; Bloch, B.; Curbera, F.; Ford, M.; Goland, Y.; Guízar, A.; Kartha, N.; Liu, C. K.; Khalaf, R.; König, D.; Marin, M.; Mehta, V.; Thatte, S.; van der Rijn, D.; Yendluri, P.; Yiu, A. OASIS Web Services Business Process Execution Language (WSBPEL) v2.0. *OASIS specification*, OASIS (2007). `http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf` (Accessed: January, 2013).

Anand, S.; Büchner, A.; Financial Times Management. *Decision Support Using Data Mining. Managment Briefings - Information Technology Series* (Financial Times/Prentice Hall, London, 1998).

Ashburner, M.; Ball, C. A.; Blake, J. A.; Botstein, D.; Butler, H.; Cherry, J. M.; Davis, A. P.; Dolinski, K.; Dwight, S. S.; Eppig, J. T.; Harris, M. A.; Hill, D. P.; Issel-Tarver, L.; Kasarskis, A.; Lewis, S.; Matese, J. C.; Richardson, J. E.; Ringwald, M.; Rubin, G. M.; Sherlock, G. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics* **25**, 25–29 (2000).

Baebler, Š.; Korbar, M.; Mozetič, I.; Hren, M.; Petek, M.; Stare, T.; Gruden, K. GoMapMan: helping plant scientist with the omics data. In: Koce, J. D.; Vodnik, D.; Pongrac, P. (eds.) *Proceedings of the 5th Slovenian Symposium on Plant Biology with International Participation*. 85 (Slovenian Society of Plant Biology, Ljubljana, 2010).

Bahree, A.; Cicoria, S.; Mulder, D.; Pathak, N.; Peiris, C. *Pro WCF: Practical Microsoft SOA Implementation* (Apress, New York, 2007).

Banks, T. Web Services Resource Framework (WSRF) - Primer v1.2. *OASIS committee draft*, OASIS (2006). `http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf` (Accessed: January, 2013).

Berthold, M. R.; Cebron, N.; Dill, F.; Gabriel, T. R.; Kötter, T.; Meinl, T.; Ohl, P.; Thiel, K.; Wiswedel, B. KNIME - the Konstanz information miner: Version 2.0 and beyond. *SIGKDD Explorations* **11**, 26–31 (2009).

Bhagat, J.; Tanoh, F.; Nzuobontane, E.; Laurent, T.; Orlowski, J.; Roos, M.; Wolstencroft, K.; Aleksejevs, S.; Stevens, R.; Pettifer, S.; Lopez, R.; Goble, C. A. Biocatalogue: A universal catalogue of web services for the life sciences. *Nucleic Acids Research* **38**, 689–694 (2010).

Bouckaert, R. R.; Frank, E.; Hall, M. A.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. H. WEKA - Experiences with a Java Open-Source Project. *Journal of Machine Learning Research* **11**, 2533–2541 (2010).

Burnett, M. M. *Visual Programming*, 275–283 (John Wiley & Sons, Inc., New York, 2001).

Cabena, P. *Discovering Data Mining: From Concept to Implementation. An IBM Press Book Series* (Prentice Hall, Upper Saddle River, New Jersey, 1998).

Charif, Y.; Sabouret, N. An overview of semantic web services composition approaches. *Electron. Notes Theor. Comput. Sci.* **146**, 33–41 (2006).

Chiaretti, S.; Li, X.; Gentleman, R.; Vitale, A.; Vignetti, M.; Mandelli, F.; Ritz, J.; Foa, R. Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood* **103**, 2771–2778 (2004).

Chinnici, R.; Hadley, M.; Mordani, R. The Java API for XML-based web services (JAX-WS) 2.0. *Specification JSR-000224*, Sun Microsystems, Inc. (2006). `http://download.oracle.com/otn-pub/jcp/jaxws-2_0-fr-eval-oth-JSpec/jaxws-2_0-fr-spec.pdf` (Accessed: January, 2013).

Cios, K.; Pedrycz, W.; Swiniarski, R.; Kurgan, L. *Data Mining: A Knowledge Discovery Approach* (Springer Science+Business Media, New York, 2010).

de Bruin, J. S. *Service-oriented discovery of knowledge: foundations, implementations and applications.* Ph.D. thesis (Leiden Institute for Advanced Computer Sciences (LIACS), Faculty of Science, Leiden University, Leiden, Belgium, 2010).

De Roure, D.; Goble, C.; Stevens, R. The design and realisation of the myExperiment Virtual Research Environment for social sharing of workflows. *Future Generation Computer Systems* **25**, 561–567 (2009).

Demšar, J.; Zupan, B.; Leban, G.; Curk, T. Orange: From experimental machine learning to interactive data mining. In: Boulicaut, J.-F.; Esposito, F.; Giannotti, F.; Pedreschi, D. (eds.) *Knowledge Discovery in Databases: PKDD 2004, Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases.* **3202**, 537–539 (Springer, Berlin, 2004).

Džeroski, S. Towards a general framework for data mining. In: Džeroski, S.; Struyf, J. (eds.) *Knowledge Discovery in Inductive Databases, Proceedings of the 5th International Workshop, KDID 2006, Revised Selected and Invited Papers.* **4747**, 259–300 (Springer, Berlin, 2006).

Eden, E. *Discovering Motifs in Ranked Lists of DNA Sequences*. Master's thesis (Israel Institute of Technology, Haifa, Israel, 2007).

Eden, E.; Lipson, D.; Yogev, S.; Yakhini, Z. Discovering motifs in ranked lists of dna sequences. *PLoS Computational Biology* **3** (2007).

Eden, E.; Navon, R.; Steinfeld, I.; Lipson, D.; Yakhini, Z. GOrilla: A tool for discovery and visualization of enriched GO terms in ranked gene lists. *BMC Bioinformatics* **10**, 48 (2009).

Erl, T. *Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall service-oriented computing series from Thomas Erl* (Pearson Education, Upper Saddle River, New Jersey, 2005).

Eronen, L.; Toivonen, H. Biomine: predicting links between biological entities using network models of heterogeneous databases. *BMC Bioinformatics* **13**, 119 (2012).

Fayyad, U. M.; Piatetsky-Shapiro, G.; Smyth, P. The KDD Process for Extracting Useful Knowledge from Volumes of Data. *Commun. ACM* **39**, 27–34 (1996).

Fielding, R. T. *Architectural styles and the design of network-based software architectures*. Ph.D. thesis (University of California, Irvine, Irvine, California, USA, 2000).

Goble, C. A.; Bhagat, J.; Aleksejevs, S.; Cruickshank, D.; Michaelides, D.; Newman, D.; Borkum, M.; Bechhofer, S.; Roos, M.; Li, P.; De Roure, D. myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Research* **38**, W677–W682 (2010).

Grčar, M.; Podpečan, V.; Sluban, B.; Mozetič, I. Ontology querying support in semantic annotation process. In: Anthony, P.; Ishizuka, M.; Lukose, D. (eds.) *Trends in Artificial Intelligence - 12th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2012)* **7458**, 76–87 (Springer, Berlin, 2012).

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. H. The WEKA data mining software: An update. *SIGKDD Explorations* **11**, 10–18 (2009).

Hilario, M.; Kalousis, A.; Nguyen, P.; Woznica, A. A data mining ontology for algorithm selection and meta-mining. In: Podpečan, V.; Lavrač, N.; Kok, J. N.; de Bruin, J. (eds.) *Proceedings of the 2nd workshop on service-oriented knowledge discovery (SoKD'09)*. 76–88 (Unpublished proceedings, 2009). `http://www.ecmlpkdd2009.net/wp-content/uploads/2008/09/service-oriented-knowledge-discovery_2.pdf` (Accessed: January, 2013).

Hobbs, C. Using ZSI. *Technical report*, NORTEL Advanced Technology Research (2007). `http://pywebsvcs.sourceforge.net/cookbook.pdf` (Accessed: January, 2013).

Huang, D. W.; Sherman, B. T.; Lempicki, R. A. Bioinformatics enrichment tools: Paths toward the comprehensive functional analysis of large gene lists. *Nucleic acids research* **37**, 1–13 (2009a).

Huang, D. W.; Sherman, B. T.; Lempicki, R. A. Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. *Nature protocols* **4**, 44–57 (2009b).

Hull, D.; Wolstencroft, K.; Stevens, R.; Goble, C. A.; Pocock, M. R.; Li, P.; Oinn, T. Taverna: A tool for building and running workflows of services. *Nucleic Acids Research* **34**, 729–732 (2006).

Jahn, D. Service-Oriented Architectures - Going from Buzz to Business. In: LeBouton, K. J. (ed.) *Proceedings of the Thirty-first Annual SAS Users Group International Conference* (SAS Institute Inc., Cary, North Carolina, 2006).

Johnston, W. M.; Hanna, J. R. P.; Millar, R. J. Advances in dataflow programming languages. *ACM Comput. Surv.* **36**, 1–34 (2004).

Kanehisa, M.; Goto, S. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research* **28**, 27–30 (2000).

Kargupta, H.; Han, J.; Yu, P. S.; Motwani, R.; Kumar, V. (eds.) *Next Generation of Data Mining (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series)* (Chapman and Hall/CRC, St Helier, Jersey, 2008).

Kim, S.-Y.; Volsky, D. J. PAGE: Parametric Analysis of Gene Set Enrichment. *BMC Bioinformatics* **6**, 144 (2005).

Klösgen, W. Explora: A Multipattern and Multistrategy Discovery Assistant. In: *Advances in Knowledge Discovery and Data Mining*. 249–271 (American Association for Artificial Intelligence, Menlo Park, CA, 1996).

Kranjc, J.; Podpečan, V.; Lavrač, N. ClowdFlows: A cloud based scientific workflow platform. In: Flach, P. A.; Bie, T. D.; Cristianini, N. (eds.) *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), Part II*. **7524**, 816–819 (Springer, Berlin, 2012a).

Kranjc, J.; Podpečan, V.; Lavrač, N. Knowledge Discovery Using a Service Oriented Web Application. In: Mauri, J. L.; Lorenz, P. (eds.) *Proceedings of the Fourth International Conference on Information, Process, and Knowledge Management 2012 (eKNOW 2012)*. 82–87 (Curran Associates, Inc., New York, 2012b).

Langohr, L.; Podpečan, V.; Petek, M.; Mozetič, I.; Gruden, K.; Lavrač, N.; Toivonen, H. Contrasting subgroup discovery. *The Computer Journal* (2012). In press.

Liu, C. K.; Booth, D. Web services description language (WSDL) version 2.0 part 0: Primer. *W3C recommendation*, W3C (2007). `http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626` (Accessed: January, 2013).

Maglott, D. R.; Ostell, J.; Pruitt, K. D.; Tatusova, T. A. Entrez Gene: Gene-centered information at NCBI. *Nucleic Acids Research* **33**, 54–58 (2005).

Majithia, S.; Shields, M. S.; Taylor, I. J.; Wang, I. Triana: A Graphical Web Service Composition and Execution Toolkit. In: *Proceedings of the IEEE International Conference on Web Services (ICWS'04)*. 514–524 (IEEE Computer Society, Washington, DC, 2004).

McCabe, F.; Booth, D.; Ferris, C.; Orchard, D.; Champion, M.; Newcomer, E.; Haas, H. Web services architecture. *W3C note*, W3C (2004). `http://www.w3.org/TR/ws-arch/` (Accessed: January, 2013).

Microsoft Corporation. Predictive analysis with SQL Server 2008. *White paper*, Microsoft, Inc. (2008). `http://download.microsoft.com/download/1/D/0/1D0AA2A5-E2FB-4F72-B41D-04548D25A9D5/SQL%20Server%202008%20R2%20Data%20Mining%20Whitepaper%20Overview.docx` (Accessed: January, 2013).

Mierswa, I.; Wurst, M.; Klinkenberg, R.; Scholz, M.; Euler, T. Yale: Rapid prototyping for complex data mining tasks. In: Ungar, L.; Craven, M.; Gunopulos, D.; Eliassi-Rad, T. (eds.) *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*. 935–940 (ACM, New York, NY, USA, 2006).

Mozes, A. Oracle Data Mining 11g, Release 2: Mining star schemas, a Telco Churn case study. *Oracle white paper*, Oracle, Inc. (2011). `http://www.oracle.com/technetwork/database/options/advanced-analytics/odm/odmtelcowhitepaper-326595.pdf` (Accessed: January, 2013).

Newcomer, E.; Lomow, G. *Understanding SOA with Web services. Independent technology guides* (Addison-Wesley, Boston, Massachusetts, 2005).

Panov, P. *An Ontology of Data Mining*. Ph.D. thesis (Jožef Stefan International Postgraduate School, Ljubljana, Slovenia, 2012).

Panov, P.; Soldatova, L.; Džeroski, S. Towards an ontology of data mining investigations. In: Gama, J.; Costa, V.; Jorge, A.; Brazdil, P. (eds.) *Discovery Science*. 257–271 (Springer, Berlin, Heidelberg, 2009).

Podpečan, V.; Lavrač, N.; Mozetič, I.; Kralj Novak, P.; Trajkovski, I.; Langohr, L.; Kulovesi, K.; Toivonen, H.; Petek, M.; Motaln, H.; Gruden, K. SegMine workflows for semantic microarray data analysis in Orange4WS. *BMC Bioinformatics* **12**, 416 (2011).

Podpečan, V.; Zemenova, M.; Lavrač, N. Orange4WS environment for service-oriented data mining. *The Computer Journal* **55**, 82–98 (2012).

Quinlan, J. R. *C4.5: Programs for Machine Learning* (Morgan Kaufmann, Burlington, Massachusetts, 1993).

Rao, J.; Su, X. A survey of automated web service composition methods. In: Cardoso, J.; Sheth, A. (eds.) *Semantic Web Services and Web Process Composition*. 43–54 (Springer, Berlin, Heidelberg, 2005).

Robnik Šikonja, M.; Kononenko, I. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning* **53**, 23–69 (2003).

Salz, R.; Blunck, C. ZSI: The Zolera Soap Infrastructure. *Technical report*, Zolera Systems, Inc. (2005). `http://pywebsvcs.sourceforge.net/zsi.html` (Accessed: January, 2013).

Schallmoser, K.; Bartmann, C.; Rohde, E.; Bork, S.; Guelly, C.; Obenauf, A. C.; Reinisch, A.; Horn, P.; Ho, A. D.; Strunk, D.; Wagner, W. Replicative senescence-associated gene expression changes in mesenchymal stromal cells are similar under different culture conditions. *Haematologica* **95**, 867–874 (2010).

Sevon, P.; Eronen, L.; Hintsanen, P.; Kulovesi, K.; Toivonen, H. Link discovery in graphs derived from biological databases. In: Leser, U.; Naumann, F.; Eckman, B. A. (eds.) *Proceedings of the Third International Workshop on Data Integration in the Life Sciences (DILS 2006)* **4075**, 35–49 (Springer, Berlin, 2006).

Shearer, C. The CRISP-DM Model: The new blueprint for data mining. *Journal of Data Warehousing* **5** (2000).

Subramanian, A.; Kuehn, H.; Gould, J.; Tamayo, P.; Mesirov, J. P. GSEA-P: a desktop application for Gene Set Enrichment Analysis. *Bioinformatics* **23**, 3251–3253 (2007).

Talia, D.; Trunfio, P.; Verta, O. The Weka4WS framework for distributed data mining in service-oriented Grids. *Concurrency and Computation: Practice and Experience* **20**, 1933–1951 (2008).

The LCG TDR Editorial Board. LHC Computing Grid. *Technical design report*, CERN-LHCC (2005). `http://cdsweb.cern.ch/record/840543/files/lhcc-2005-024.pdf` (Accessed: January, 2013).

Thimm, O.; Bläsing, O.; Gibon, Y.; Nagel, A.; Meyer, S.; Krüger, P.; Selbig, J.; Müller, L. A.; Rhee, S. Y.; Stitt, M. MAPMAN: A user-driven tool to display genomics data sets onto diagrams of metabolic pathways and other biological processes. *Plant Journal* **37**, 914–939 (2004).

Trajkovski, I. *Functional Interpretation of Gene Expression Data*. Ph.D. thesis (Jožef Stefan International Postgraduate School, Ljubljana, Slovenia, 2007).

Trajkovski, I.; Lavrač, N.; Tolar, J. SEGS: Search for enriched gene sets in microarray data. *Journal of Biomedical Informatics* **41**, 588–601 (2008).

Vanschoren, J.; Blockeel, H. A community-based platform for machine learning experimentation. In: Buntine, W. L.; Grobelnik, M.; Mladenić, D.; Shawe-Taylor, J. (eds.) *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD), Part II*. **5782**, 750–754 (Springer, Berlin, 2009).

Vanschoren, J.; Blockeel, H.; Pfahringer, B.; Holmes, G. Experiment databases - a new way to share, organize and learn from experiments. *Machine Learning* **87**, 127–158 (2012).

Vavpetič, A.; Podpečan, V.; Meganck, S.; Lavrač, N. Explaining subgroups through ontologies. In: Anthony, P.; Ishizuka, M.; Lukose, D. (eds.) *Trends in Artificial Intelligence - 12th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2012)* **7458**, 625–636 (Springer, Berlin, 2012).

Žáková, M.; Kremen, P.; Železný, F.; Lavrač, N. Automating knowledge discovery workflow composition through ontology-based planning. *IEEE T. Automation Science and Engineering* **8**, 253–264 (2011).

Wagner, W.; Horn, P.; Castoldi, M.; Diehlmann, A.; Bork, S.; Saffrich, R.; Benes, V.; Blake, J.; Pfister, S.; Eckstein, V.; Ho, A. D. Replicative senescence of mesenchymal stem cells: A continuous and organized process. *PLoS ONE* **3** (2008).

Wrobel, S. An algorithm for multi-relational discovery of subgroups. In: Komorowski, H. J.; Zytkow, J. M. (eds.) *Proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD'97)* **1263**, 78–87 (Springer, Berlin, 1997).

Zeeberg, B.; Feng, W.; Wang, G.; Wang, M.; Fojo, A.; Sunshine, M.; Narasimhan, S.; Kane, D.; Reinhold, W.; Lababidi, S.; Bussey, K.; Riss, J.; Barrett, J.; Weinstein, J. GoMiner: A resource for biological interpretation of genomic and proteomic data. *Genome Biology* **4**, R28+ (2003).

Zeshan, F.; Mohamad, R. Semantic web service composition approaches: Overview and limitations. *International Journal on New Computer Architectures and Their Applications (IJNCAA)* **3**, 640–651 (2011).

Publications related to the dissertation

This section lists all publications related to the thesis. It consists of publications included in the thesis, publications which are directly related to the topic of the thesis, as well as publications which are indirectly related to the thesis, such as Orange4WS use cases, applications and further developments of the platform.

Original scientific article (1.01)

Kranjc, J.; Podpečan, V.; Lavrač, N. ClowdFlows: A cloud based scientific workflow platform. In: Flach, P. A.; Bie, T. D.; Cristianini, N. (eds.) *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), Part II.* **7524**, 816–819 (Springer, Berlin, 2012).

Langohr, L.; Podpečan, V.; Petek, M.; Mozetič, I.; Gruden, K.; Lavrač, N.; Toivonen, H. Contrasting subgroup discovery. *The Computer Journal* (2012). In press.

Miljkovic, D.; Stare, T.; Mozetič, I.; Podpečan, V.; Petek, M.; Witek, K.; Dermastia, M.; Lavrač, N.; Gruden, K. Signalling network construction for modelling plant defence response. *PLoS ONE* **7**, e51822 (2012).

Podpečan, V.; Lavrač, N.; Mozetič, I.; Kralj Novak, P.; Trajkovski, I.; Langohr, L.; Kulovesi, K.; Toivonen, H.; Petek, M.; Motaln, H.; Gruden, K. SegMine workflows for semantic microarray data analysis in Orange4WS. *BMC Bioinformatics* **12**, 416 (2011).

Podpečan, V.; Žáková, M.; Lavrač, N. Workflow Construction for Service-Oriented Knowledge Discovery. In: Margaria, T.; Steffen, B. (eds.) *Proceedings of the 4th International Symposium on Leveraging Applications of Formal Methods, Verification, and Validation (ISoLA 2010), Part I.* **6415**, 313–327 (Springer, Berlin, 2010).

Podpečan, V.; Zemenova, M.; Lavrač, N. Orange4WS environment for service-oriented data mining. *The Computer Journal* **55**, 82–98 (2012).

Vavpetič, A.; Podpečan, V.; Meganck, S.; Lavrač, N. Explaining subgroups through ontologies. In: Anthony, P.; Ishizuka, M.; Lukose, D. (eds.) *Trends in Artificial Intelligence - 12th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2012).* **7458**, 625–636 (Springer, Berlin, 2012).

Published scientific conference contribution (1.08)

Krajnc, J.; Podpečan, V.; Lavrač, N. A browser-based platform for service-oriented knowledge discovery. In: Jörg-Uwe, K.; Fisher, S.; Lavrač, N.; Podpečan, V. (eds.) *Proceedings of the 4th workshop on planning to learn and service-oriented knowledge discovery (PlanSoKD'11).*

30–35 (Unpublished proceedings, 2011). https://files.ifi.uzh.ch/ddis/oldweb/ddis/fileadmin/pdf/kietz/PlanSoKD-proceedings.pdf (Accessed: January, 2013).

Kranjc, J.; Podpečan, V.; Lavrač, N. Knowledge Discovery Using a Service Oriented Web Application. In: Mauri, J. L.; Lorenz, P. (eds.) *Proceedings of the Fourth International Conference on Information, Process, and Knowledge Management 2012 (eKNOW 2012)*. 82–87 (Curran Associates, Inc., New York, 2012).

Lavrač, N.; Kralj Novak, P.; Mozetič, I.; Podpečan, V.; Motaln, H.; Petek, M.; Gruden, K. Semantic subgroup discovery: using ontologies in microarray data analysis. In: He, B.; Pan, X.; Kim, Y.; Worrell, G. (eds.) *Engineering the future of biomedicine: Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 5613–5616 (Institute of Electrical and Electronics Engineers, New York City, New York, 2009a).

Lavrač, N.; Mozetič, I.; Podpečan, V.; Kralj Novak, P.; Motaln, H.; Petek, M. Gene analytics: discovery and contextualization of enriched gene sets. In: Nürnberger, A.; Berthold, M. R.; Kötter, T.; Thiel, K. (eds.) *Proceedings of the Workshop on Explorative Analytics of Information Networks*. 39–49 (Unpublished proceedings, 2009b). http://www.ecmlpkdd2009.net/wp-content/uploads/2009/08/explorative-analytics-of-information-networks.pdf (Accessed: January, 2013).

Miljković, D.; Mihăilă, C.; Podpečan, V.; Grčar, M.; Gruden, K.; Stare, T.; Lavrač, N. Workflow-based information retrieval to model plant defence response to pathogen attacks. In: Hilario, M.; Lavrač, N.; Podpečan, V.; Kok, J. N. (eds.) *Proceedings of the 3rd workshop on service-oriented knowledge discovery (SoKD'10)*. 51–60 (Unpublished proceedings, 2010). http://cui.unige.ch/~hilario/sokd10/sokd10-proceedings.pdf (Accessed: January, 2013).

Mozetič, I.; Lavrač, N.; Podpečan, V.; Kralj Novak, P.; Motain, H.; Petek, M.; Gruden, K.; Toivonen, H.; Kulovesi, K. Bisociative Knowledge Discovery for Microarray Data Analysis. In: Ventura, D.; Pease, A.; Pérez, R.; Ritchie, G.; Veale, T. (eds.) *Proceedings of the International Conference on Computational Creativity (ICCC-X)*. 190–199 (Department of Informatics Engineering, University of Coimbra, Lisbon, Portugal, 2010).

Podpečan, V.; Juršič, M.; Žáková, M.; Lavrač, N. Towards a service-oriented knowledge discovery platform. In: Podpečan, V.; Lavrač, N.; Kok, J. N.; de Bruin, J. (eds.) *Proceedings of the 2nd workshop on service-oriented knowledge discovery (SoKD'09)*. 25–38 (Unpublished proceedings, 2009). http://www.ecmlpkdd2009.net/wp-content/uploads/2008/09/service-oriented-knowledge-discovery_2.pdf (Accessed: January, 2013).

Žáková, M.; Železný, F.; Podpečan, V.; Lavrač, N. Advancing data mining workflow construction: A framework and cases using the Orange toolkit. In: Podpečan, V.; Lavrač, N.; Kok, J. N.; de Bruin, J. (eds.) *Proceedings of the 2nd workshop on service-oriented knowledge discovery (SoKD'09)*. 39–51 (Unpublished proceedings, 2009). http://www.ecmlpkdd2009.net/wp-content/uploads/2008/09/service-oriented-knowledge-discovery_2.pdf (Accessed: January, 2013).

Independent scientific component part or a chapter in a monograph (1.16)

Langohr, L.; Podpečan, V.; Mozetič, I.; Petek, M.; Gruden, K. Contrast mining from interesting subgroups. In: Berthold, M. R. (ed.) *Bisociative Knowledge Discovery: An Introduction to Concept, Algorithms, Tools, and Applications*. 390–406 (Springer, Berlin, 2012).

Mozetič, I.; Lavrač, N.; Podpečan, V.; Kralj Novak, P.; Motaln, H.; Petek, M.; Toivonen, H.; Kulovesi, K. Semantic subgroup discovery and cross-context linking for microarray data analysis. In: Berthold, M. R. (ed.) *Bisociative Knowledge Discovery: An Introduction to Concept, Algorithms, Tools, and Applications*. 379–389 (Springer, 2012).

Index of Figures

Index of Tables

Appendix

A Orange4WS: availability and user's manual

A.1 Software availability

The Orange4WS platform is available as an extension of the Orange data mining toolkit. It upgrades Orange with (a) transparent integration of web services as Orange's workflow components with one click import, (b) software tools for web service development, and (c) knowledge discovery ontology and planner that enables automated composition of data mining workflows. Note that the last feature is not yet provided in the current stable version (0.7) of the Orange4WS software.

Orange4WS is written in Python, and thus requires the Python interpreter version 2.6.x or 2.7.x which is also required by Orange. Additionally, Orange4WS requires ZSI (Hobbs, 2007; Salz and Blunck, 2005) version 2.0 or newer, which provide libraries for using and developing web services and protocols, such as SOAP and WSDL. Orange4WS is platform independent and runs on all platforms which are supported by Orange (currently Linux/Unix, Windows and MacOS). The home web page of Orange4WS at `http://orange4ws.ijs.si/` provides binary installers for Microsoft Windows systems and sources for Linux, Unix, and MacOS systems. The installation instructions are also provided.

A.2 User's manual

This section documents the most important features of Orange4WS and provides outlines of the most common procedures supported by Orange4WS.

Importing a web service into Orange4WS

Orange4WS enables transparent and automated integration of SOAP 1.1 web services with service descriptions in WSDL. This feature is implemented by providing a widget code generator and a user interface. The process of importing a web service is as follows.

- The user invokes the command *Import web service* from the "Options" menu. This opens a user interface dialogue window.
- The user enters the address or addresses of web services and selects a widget category (or creates a new one) where the automatically created widgets will be placed. A screenshot of the interface and its options is shown in Figure A.1.
- The user closes the user interface by confirming the entered values. Orange4WS validates the service addresses and invokes the code generator to create new web service widgets in the specified directories (each widget category resides in a different directory).
- Orange4WS automatically reloads all Orange and Orange4WS widgets. Newly imported web services are now available under the specified widget category.

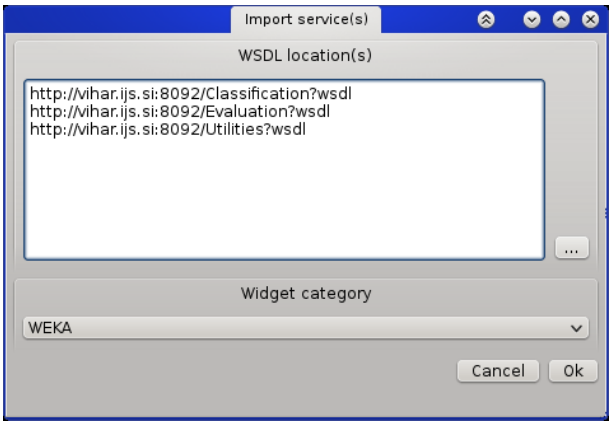

Figure A.1: A graphical user interface dialogue window for importing new web services into Orange4WS. Note that unlimited number of services can be imported at the same time.

Development of new web services in Orange4WS

The steps required for the development of new web services using the tools provided by Orange4WS and ZSI are as follows.

1. *Web service contract design.* As Orange4WS encourages contract-first (WSDL-first) web service design, the first step is to design the WSDL document. While this document can be written manually, it is recommended to use a visual design tool, such as Eclipse Web Tools Platform[1], Altova XML Spy[2] or WSDL Designer[3] which allows automated WSDL document construction according to specified functions and messages. Messages are typically described in separate documents using the XML Schema (XSD) language. Note that the results of this step do not depend on the target platform or employed software tools. For example, the top-level visual representation of the WSDL document for the SEGS algorithm web service is shown in Figure A.2.

2. *Code stub generation.* In this step a ZSI code generator tool is employed to automatically generate the server code skeleton. The code generator tool is available as a command line utility *wsdl2py*, but can also be called programmatically. For each given WSDL document, it produces three Python sources: client module, server module and types module. For example, the code skeleton for the SEGS web service is shown in Figure A.3.

3. *Web service functionality implementation.* In this step the server code skeleton obtained in the previous step is extended with the code performing the main functionalities of the web service. Due to versatile connectivity of the Python language framework the Orange4WS web services can employ algorithm implementations and libraries written in languages, such as C, C++, Fortran, Java, C#, Delphi, R and others. For example, the SEGS web service integrates the SEGS algorithm which is implemented in C for performance reasons. Orange4WS also supports the development of batch (job) processing web services by providing classes which allow job creation, progress polling, and results retrieval. SEGS web service is an example of such a service.

[1] http://www.eclipse.org/webtools/
[2] http://www.altova.com/
[3] http://www.stylusstudio.com/wsdl-designer.html

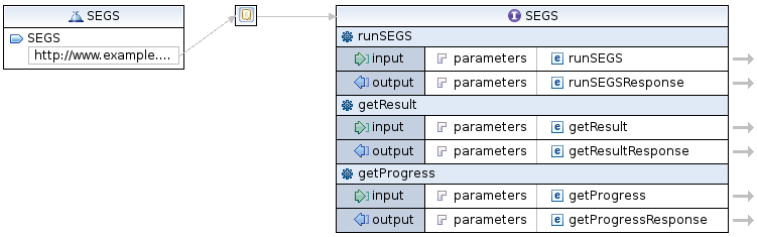

Figure A.2: *WSDL document of the SEGS web service.* A top-level visual representation of the WSDL document for the SEGS algorithm web service in the Eclipse WSDL editor. Note that this level does not show the structure of the messages, and that the service address is only provisional (`http://www.example.org/SEGS`) as the actual address is determined and inserted on the fly when the server hosting the service is started.

```
from ZSI.ServiceContainer import ServiceSOAPBinding
import SEGS_client as client

class SEGS(ServiceSOAPBinding):
    def soap_runSEGS(self, ps, **kw):
        request = ps.Parse(client.runSEGSRequest.typecode)
        response = client.runSEGSResponse()
        ### main function code goes here
        return request,response

    def soap_getResult(self, ps, **kw):
        request = ps.Parse(client.getResultRequest.typecode)
        response = client.getResultResponse()
        ### main function code goes here
        return request,response

    def soap_getProgress(self, ps, **kw):
        request = ps.Parse(client.getProgressRequest.typecode)
        response = client.getProgressResponse()
        ### main function code goes here
        return request,response
```

Figure A.3: A code skeleton for the SEGS web service.

4. *Web server deployment.* The last step is the deployment of the implemented web service using the simple, minimalistic web server provided by Orange4WS. Using the specified port number and log file the server exposes the given service on the current system and inserts the correct address into the service's WSDL document.

Development of new Orange4WS widgets

As Orange4WS upgrades Orange while retaining all of its features, the widget development process for Orange4WS is the same as for Orange. The official documentation for widget development in Orange, which is available at `http://orange.biolab.si/docs/latest/extend-widgets/rst/`, is thus also valid for Orange4WS.

However, Orange4WS provides additional software modules which can be employed in custom widgets. The following modules are provided.

- *Data serializer* module provides functions for the serialisation and deserialisation of several data types and manipulation of temporary objects (files and directories).
- *Server base* module provides functions and classes for the development of custom, lightweight web servers.
- *Stub importer* module enables importing web service stub code which is created on the fly.
- *Extractor* module implements functions for the manipulation of web service message objects.
- *Base WS widget* implements a generic web service class widgets which is a parent class of all web service widgets.
- *Widget generator* and *Generator* modules implement automated code generation and provide the user interface for the *Import web service* command.
- *Process pool* module implements custom process management. This module enables the creation of computationally complex job processing web services.
- *GUI tools* module provides helper function for the graphical user interfaces of Orange4WS widgets.
- *orngOWL* module provides classes and functions for the integration of the KD ontology into Orange4WS.

Annotating a new algorithm in Orange4WS

Existing and newly implemented algorithms in Orange4WS and Orange can be annotated in the ontology by performing the following steps.

1. Create instances of the `AlgorithmParameter` for all inputs and outputs of the algorithm.
2. Create an instance of the `NamedAlgorithm`.
3. For each instance of the `AlgorithmParameter` create a class defining its range and add the necessary subclasses of `Knowledge` (this is only required when a new type of algorithm is added).
4. Create an XML file defining the mapping between the algorithm representation in Orange4WS and in the KD ontology.
5. Generate a mapping using the Orange-Map (OM) ontology by means of the provided tools. This is accomplished by creating an instance of the *Orange2OntoMapping* class from the *orngOWL* module and executing its *addMappingsFromFile(·,·,·)* method.

Manually constructed Orange4WS and Orange workflows can also be annotated using the KD ontology and later used for querying and reasoning. For this task, the *Orange2Owl* class from the *orngOWL* module is used. Its two methods *convert2Owl(·,·,·)* and *convert2Ows(·,·)* perform the conversion between Orange's workflow representation (OWS) and OWL.

B Implementation of the SegMine methodology

Our implementation of the SegMine methodology consists of a set of Orange4WS interactive workflow components (widgets) and three instances of the SEGS algorithm web service which are also available as workflow components. The first instance allows for the analysis of Homo sapiens, Mus musculus, and Rattus rattus experimental data, the second supports Arabidopsis thaliana while the last accepts Solanum tuberosum experimental data.

SegMine workflow components are implemented in Python and require a working Orange4WS environment. Additionally, the SegMine's graph visualisation component requires Java Runtime Environment (JRE) while SEGS web service widgets also require internet connection for job submission and results retrieval. The home page of SegMine at `http://segmine.ijs.si/` provides binary installers for Microsoft Windows systems and sources for Linux, Unix, and MacOS systems along with the installation instructions.

The presented SegMine implementation offers the following Orange4WS workflow components.

1. *Microarray parser.* The Microarray data parser widget performs three functions: (1) parses the input data, recognises values in rows and columns, treats missing values as specified and performs syntax check, (2) creates a native Orange data structure `Orange.core.ExampleTable` from the data, and (3) computes logFC values (logarithms of fold change). A screenshot of the widget is shown in Figure B.1.

   Inputs:

   - microarrayAsString [type: string]

   Outputs:

   - microarrayTable [type: Orange.core.ExampleTable]
   - logFCvalues [type: list of tuples (string, float)]

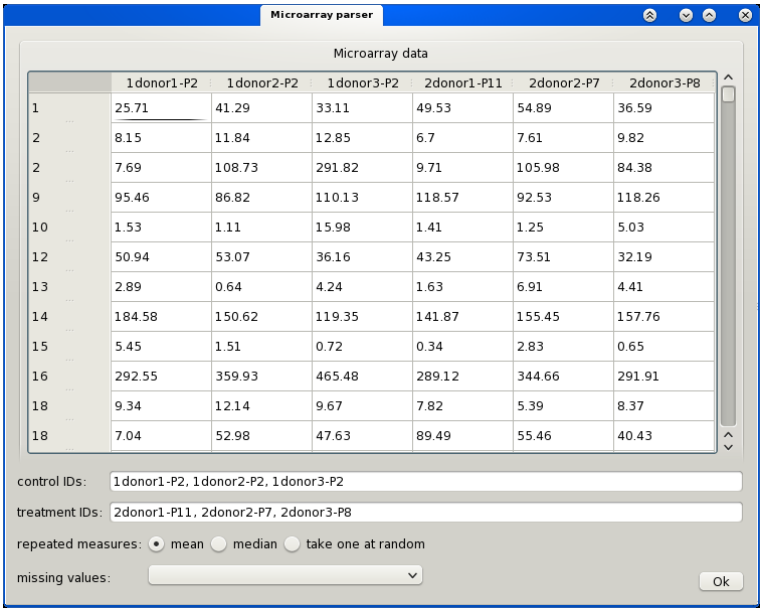

Figure B.1: A screenshot of the Microarray data parser widget depicting loaded experimental data and available options.

2. *Gene ranker*. The Gene ranker widget ranks input genes using the ReliefF (Robnik Šikonja and Kononenko, 2003) algorithm. It also computes t-statistic for independent two-sample t-test assuming unequal sample sizes and unequal variances. The widget returns a list of tuples (geneName, rank), sorted in descending order according to the rank. A screenshot of the widget is shown in Figure B.2.

   Inputs:

   · microarrayTable [type: Orange.core.ExampleTable]

   Outputs:

   · geneRanks [type: list of tuples (string, float), sorted in descending order]
   · tScores [type: list of tuples (string, float), sorted in descending order]

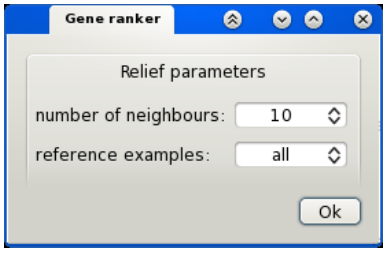

Figure B.2: A screenshot of the Gene ranker widget depicting available settings for the gene ranking algorithm.

3. *Rank plotter*. The Rank plotter widget draws a dotted line graph of the sorted input list of ranked genes. It is typically used to inspect the rank curve and enables easy estimation of the positions in the ranked list where the ranks change significantly. X-axis also supports infinite zoom. A screenshot of the widget is shown in Figure B.3.

   Inputs:

   · geneRanks [type: list of tuples (string, float)]

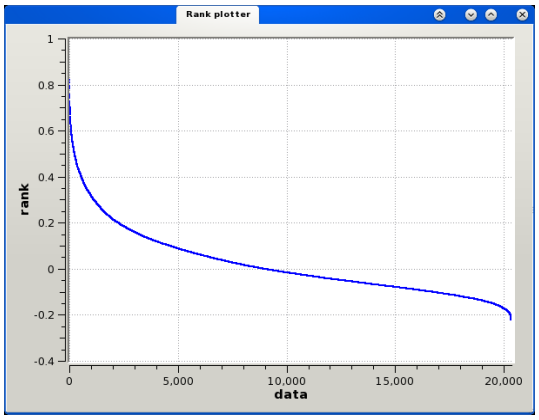

Figure B.3: A screenshot of the Gene ranker widget depicting a typical plot of a sorted list of gene ranks obtained with the ReliefF algorithm.

4. *Cutoff.* The Cutoff widget receives a list of ranked genes and a list of their logFC values[1] and removes the genes whose logFC values do not fit in the specified range. It allows for upper and lower bound filtering which can also be absolute. Most typically, genes with absolute logFC value around 0 are filtered out. A screenshot of the widget is shown in Figure B.4.

   Inputs:

   - geneRanks [type: list of tuples (string, float)]
   - logFCvalues [type: list of tuples (string, float)]

   Outputs:

   - reduced geneRanks [type: list of tuples (string, float)]
   - reduced logFCvalues [type: list of tuples (string, float)]

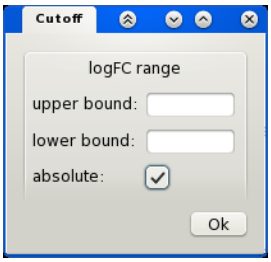

Figure B.4: A screenshot of the Cutoff widget depicting available settings for filtering a ranked list of genes according to their logFC values.

5. *Resolve gene names.* The Resolve gene names widget tries to resolve symbolic names and synonyms of input genes from human, mouse and rat experimental data, and map them into their Entrez identifiers. It should be applied to the data just before they are put to the SEGS algorithm web service widget. The widget is based on the mammalian gene info data available from NCBI[2] and Affymetrix gene identifiers. Finally, if an input gene is not recognised, it is ignored and removed from the list.

   Inputs:

   - geneRanks [type: list of tuples (string, float)]

   Outputs:

   - geneRanks [type: list of tuples (string, float)]

6. *Hierarchical clustering.* The Hierarchical clustering widget implements interactive hierarchical clustering. It provides the following linkage criteria: (1) single linkage, (2) average linkage, (3) Ward linkage, and (4) complete linkage. The widget also offers many options for controlling the cluster visualization and annotation. Essentially, the Hierarchical clustering widget is provided by Orange, but it is enhanced by Orange4WS, so that it also outputs the actual clustering which can be sent to the Rule browser to display cluster information. A screenshot of the widget is shown in Figure B.5.

   Inputs:

[1] Fold change (FC) is used in the analysis of gene expression data for measuring change in the expression level of a gene.

[2] `http://mirrors.vbi.vt.edu/mirrors/ftp.ncbi.nih.gov/gene/DATA/GENE_INFO/Mammalia/`

· DistanceMatrix [type: Orange.core.SymMatrix]

Outputs:

· SelectedExamples [type: Orange.core.ExampleTable]

· UnselectedExamples [type: Orange.core.ExampleTable]

· Centroids [type: Orange.core.ExampleTable]

· Structured Data Files [type: DataFiles]

· HierarchicalClustering [type: Orange.core.HierarchicalClustering]

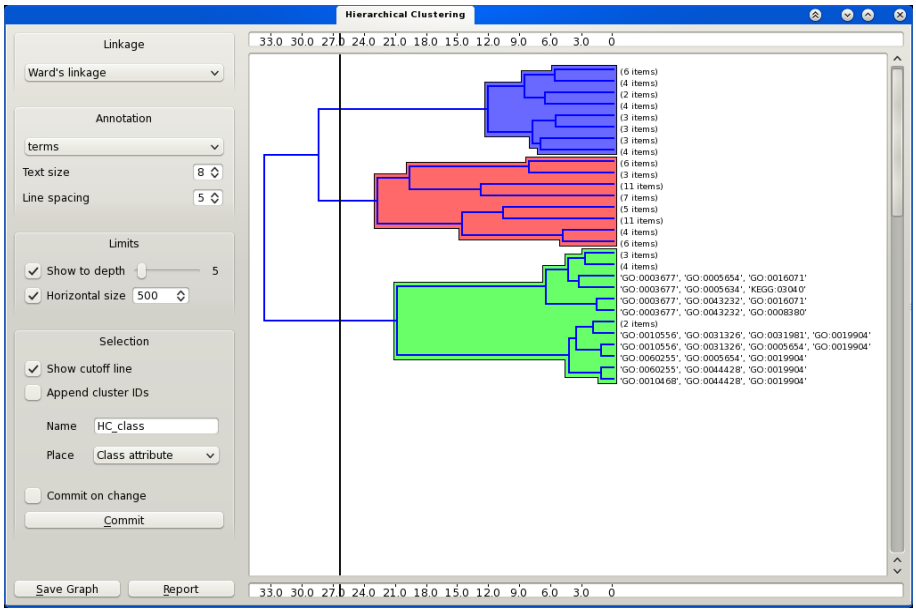

Figure B.5: A screenshot of the Hierarchical clustering widget depicting a sample clustering with three clusters obtained using Ward's linkage criteria and a manually set threshold (shown as a vertical line).

7. *Example distances.* The Example distances widget computes distances between examples (rows) in the input table. In SegMine scenarios, rows in the table are the rules as computed by the SEGS algorithm while the columns can be individual genes or individual terms. Different distance metrics can be used: Euclidean, Pearson correlation, Spearman rank correlation, Manhattan, Hamming, and Relief. Note that the Example distances widget is provided by Orange. A screenshot of the widget is shown in Figure B.6.

   Inputs:

   · Examples [type: Orange.core.ExampleTable]

   Outputs:

   · DistanceMatrix [type: Orange.core.SymMatrix]

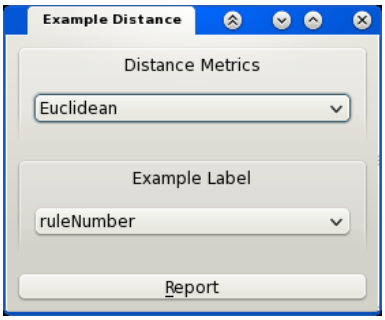

Figure B.6: A screenshot of the Example distances widget depicting available settings (distance metric and example labelling).

8. *Fetch SEGS results.* The Fetch SEGS results widget fetches existing results of the SEGS algorithm from the server where the SEGS web service is running. A screenshot of the widget is shown in Figure B.7.

   Inputs:

   - jobID [type: string]

   Outputs:

   - combinedRules [type: list of web service response objects]
   - fisherRules [type: list of web service response objects]
   - GSEArules [type: list of web service response objects]
   - PAGErules [type: list of web service response objects]
   - resultsAsXML [type: string]

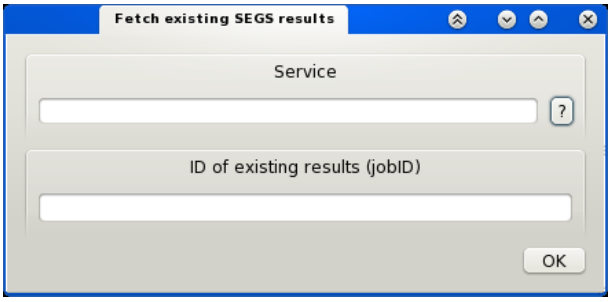

Figure B.7: A screenshot of the Fetch SEGS results widget depicting its interface where the user can input the SEGS web service URL and the job ID.

9. *SEGS HMR.* The SEGS HMR widget provides an interface to the web service implementing the SEGS algorithm (Trajkovski et al., 2008) for Homo sapiens, Mus musculus, and Rattus rattus. The widget offers numerous settings which control the SEGS algorithm. A screenshot of the widget is shown in Figure B.8.

   Inputs:

   - geneRanks [type: list of tuples (string, float), sorted in descending order]

   Outputs:

   - combinedRules [type: list of web service response objects]

- fisherRules [type: list of web service response objects]
- GSEArules [type: list of web service response objects]
- PAGErules [type: list of web service response objects]
- jobID [type: string]
- resultsAsXML [type: string]

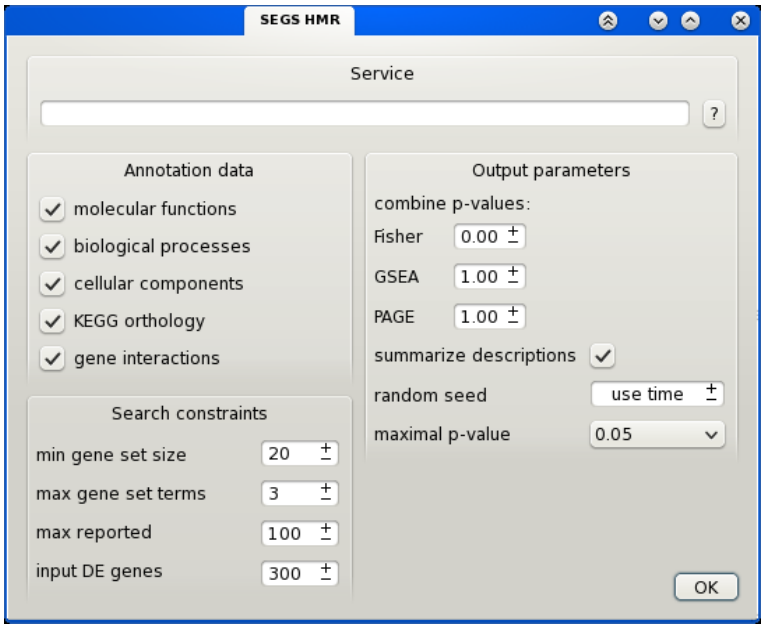

Figure B.8: A screenshot of the SEGS HMR widget depicting available parameters controlling the SEGS algorithm.

10. *Input loader for SEGS.* The Input loader for SEGS widget loads ranking of genes from a file. The widget can be used to run the SEGS algorithm using input which was prepared with a different tool. A screenshot of the widget is shown in Figure B.9.

    Outputs:

    - GeneRanks [type: list of tuples (string, float), sorted in descending order]

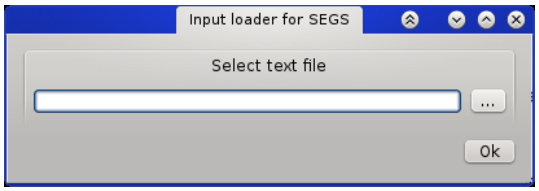

Figure B.9: A screenshot of the Input loader for SEGS widget. The input is a text file provided by the user.

11. *Load SEGS results.* The Load SEGS results loads the existing results of the SEGS algorithm from an XML file provided by the user. The purpose of this widget is to allow loading stored results of the SEGS algorithm for further analyses in order to avoid re-running SEGS. A screenshot of the widget is shown in Figure B.10.

    Outputs:

- combinedRules [type: list of web service response objects]
- fisherRules [type: list of web service response objects]
- GSEArules [type: list of web service response objects]
- PAGErules [type: list of web service response objects]

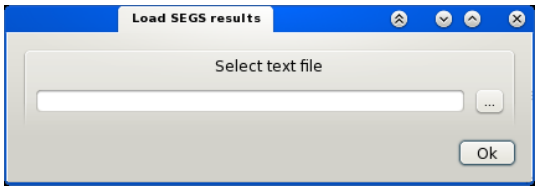

Figure B.10: A screenshot of the Load SEGS results widget. The input is an XML file provided by the user.

12. *Rule browser.* The Rule browser widget displays ontology term conjuncts (i.e., rules) describing gene sets obtained by the SEGS algorithm in an interactive HTML table. Ontology terms as well as corresponding sets of genes can be sent to the widget connected to the output by clicking on the provided links. Rule browser can also display rule cluster information if the Hierarchical clustering component is connected to the appropriate input. A screenshot of the widget is shown in Figure B.11.

    Inputs:

    - rules [type: list of web service response objects]
    - HierarchicalClustering [type: Orange.core.HierarchicalClustering]

    Outputs:

    - termList [type: list of strings]

13. *ExampleTable from rules.* The ExampleTable from rules widget takes a list of ontology term conjuncts (rules) describing gene sets found by the SEGS algorithm and constructs an Orange.core.ExampleTable data structure. The widget constructs two tables: the first one has genes as columns and the second has terms as columns.

    Inputs:

    - rules [type: list of web service response objects]

    Outputs:

    - RuleGeneTable[type: orange.ExampleTable]
    - RuleTermTable[type: orange.ExampleTable]

14. *Biomine neighbourhood search.* The Biomine neighbourhood search widget provides an interface to the Biomine search engine (Eronen and Toivonen, 2012; Sevon et al., 2006). It performs neighbourhood search, which searches for neighbouring entities of given query terms. The search results are returned as a directed, weighted, and annotated graph in Biomine's .bmg format. A screenshot of the widget is shown in Figure B.12.

    Inputs:

    - startNodes [type: list of strings]

Outputs:

- graphAsString [type: string]
- bestPathsGraphAsString [type: string]

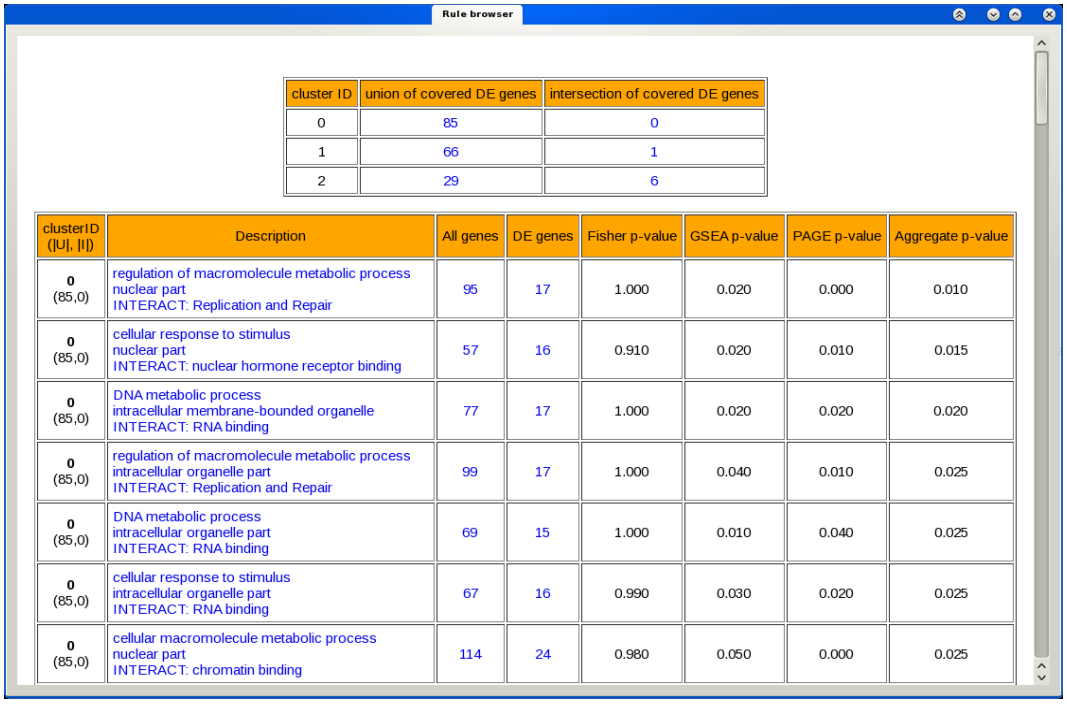

Figure B.11: A screenshot of the Rule browser widget depicting an ordered table of gene sets discovered by the SEGS algorithm. Cluster information is also shown. A more detailed explanation of the output of the SEGS algorithm is given in Appendix D.

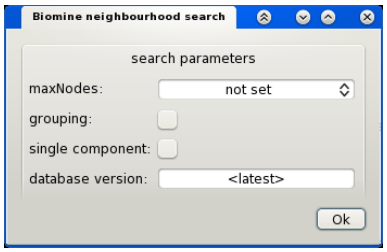

Figure B.12: A screenshot of the Biomine neighbourhood search widget depicting various search parameters. Note that the search is repeatable by specifying the same database version.

15. *Biomine connections search.* The Biomine connections search widget provides an interface to the Biomine search engine. It performs connections search, which searches for connections between two sets of given query terms. The search results are returned as a directed, weighted, and annotated graph in Biomine's .bmg format. A screenshot of the widget is shown in Figure B.13.

    Inputs:

    - startNodes [type: list of strings]

- endNodes [type: list of strings]

Outputs:

- graphAsString [type: string]
- bestPathsGraphAsString [type: string]

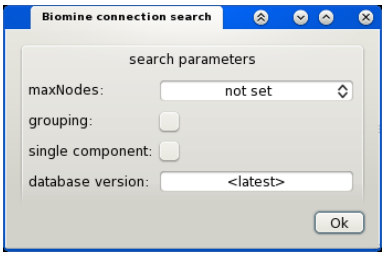

Figure B.13: A screenshot of the Biomine connections search widget depicting various search parameters. Note that the search is repeatable by specifying the same database version.

16. *Biomine medoids search.* The Biomine medoids search widget provides an interface to the Biomine search engine. It performs medoids search, which searches for medoid entities of given query terms, and returns a list of medoids as well as a list of all query terms with their medoid scores. A screenshot of the widget is shown in Figure B.14.

    Inputs:

    - startNodes [type: list of strings]

    Outputs:

    - medoids [type: list of strings]
    - rankedTerms [type: list of tuples (string, float)]

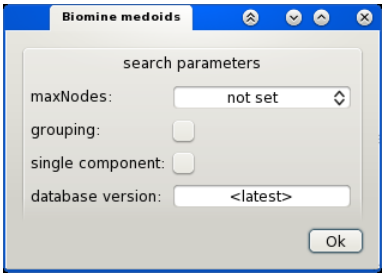

Figure B.14: A screenshot of the Biomine medoids search widget depicting various search parameters. Note that the search is repeatable by specifying the same database version.

17. *Add logFC to Biomine graph.* The Add logFC to Biomine graph widget adds logFC values to the vertices representing input genes in a given Biomine graph.

    Inputs:

    - logFCs [type: list of tuples (string, float)]
    - graphAsString [type: string]

    Outputs:

- graphAsString [type: string]

18. *Biomine graph painter.* The Biomine graph painter widget applies the chosen colour to given terms in the input Biomine graph. Using this widget interesting terms can be easily recognised in a large graph.

    Inputs:

    - termsList [type: list of strings]
    - graphAsString [type: string]

    Outputs:

    - graphAsString [type: string]

19. *Biomine graph visualiser.* The Biomine graph visualiser widget provides a powerful interactive graph visualiser which is a part of the Biomine system. The visualiser is distributed along with SegMine and runs locally as a Java Applet. Note that this component requires Java Runtime Environment[1] (JRE) to run. A screenshot of the widget is shown in Figure B.15.

    Inputs:

    - graphAsString [type: string]

    Outputs:

    - graphAsString [type: string]

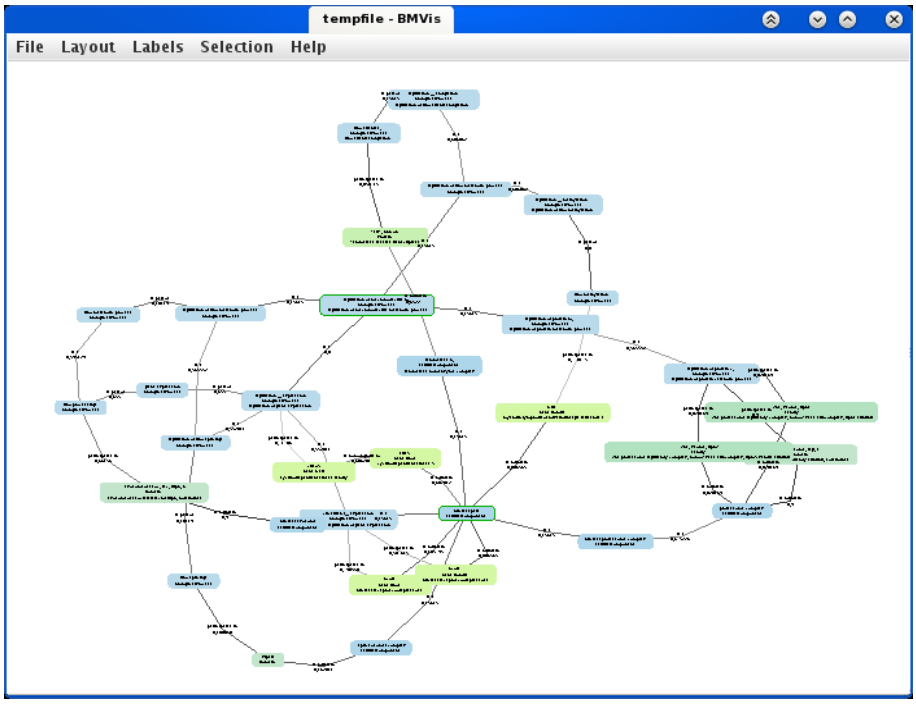

Figure B.15: A screenshot of the Biomine graph visualiser widget depicting a directed, annotated graph of biological entities, obtained by querying Biomine.

[1] http://www.java.com/en/download/manual.jsp

C Implementation of contrasting subgroup discovery

Our implementation of contrasting subgroup discovery (CSD) consists of a set of interactive workflow components for Orange4WS. The core component of the implementation is the SEGS algorithm web service (available in our SegMine implementation) which currently supports experimental gene expression data from the following organisms: Homo sapiens, Mus musculus, Rattus rattus, Arabidopsis thaliana, and Solanum tuberosum.

Contrasting subgroup discovery workflow components are implemented in Python and require a working Orange4WS environment. Additionally, a working SegMine installation is also required as our implementations share some of the components.

The presented implementation of the CSD methodology consists of the following Orange4WS widgets. Note that only a listing is provided for the widgets which are also a part of SegMine as they are described in Appendix B.

1. *Microarray parser*
2. *Gene ranker*
3. *Rank plotter*
4. *Cutoff*
5. *Hierarchical clustering*
6. *Example distances*
7. *ExampleTable from rules*
8. *Fetch plant SEGS results.* The Fetch plant SEGS results widget fetches existing results of the SEGS algorithm on Arabidopsis thaliana and Solanum tuberosum data from the server where the SEGS web service is running.

   Inputs:

   · jobID [type: string]

   Outputs:

   · combinedRules [type: list of web service response objects]
   · fisherRules [type: list of web service response objects]
   · GSEArules [type: list of web service response objects]
   · PAGErules [type: list of web service response objects]
   · resultsAsXML [type: string]

9. *SEGS STU.* The SEGS STU widget provides an interface to the web service implementing the SEGS algorithm (Trajkovski et al., 2008) for Solanum tuberosum (using the GoMapMan (Baebler et al., 2010) ontology). The widget offers numerous settings which control the SEGS algorithm. A screenshot of the widget is shown in Figure C.1.

   Inputs:

   · geneRanks [type: list of tuples (string, float), sorted in descending order]

Outputs:

- combinedRules [type: list of web service response objects]
- fisherRules [type: list of web service response objects]
- GSEArules [type: list of web service response objects]
- PAGErules [type: list of web service response objects]
- jobID [type: string]
- resultsAsXML [type: string]

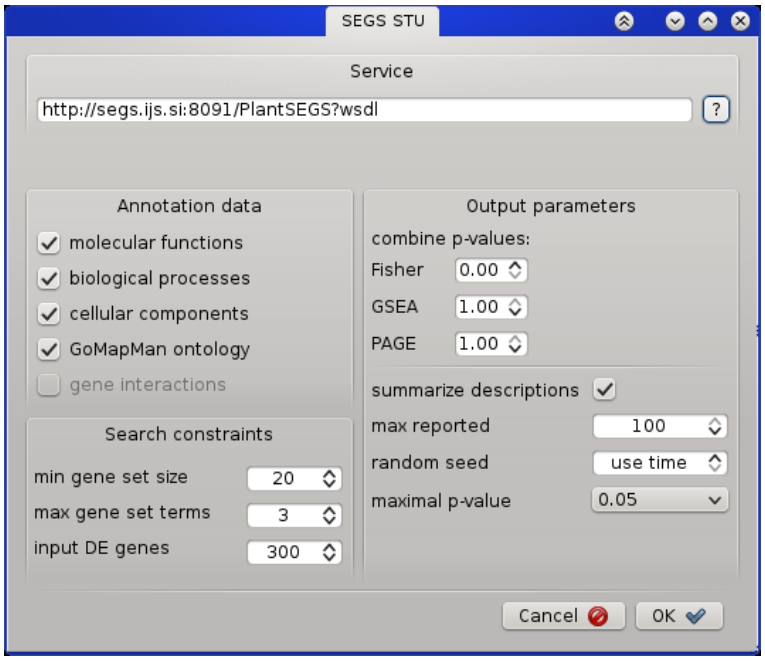

Figure C.1: A screenshot of the SEGS STU widget depicting available parameters controlling the SEGS algorithm.

10. *SEGS ATH.* The SEGS ATH widget provides an interface to the web service implementing the SEGS algorithm (Trajkovski et al., 2008) for Arabidopsis thaliana (using the GoMapMan (Baebler et al., 2010) ontology). The widget offers numerous settings which control the SEGS algorithm. The parameters of the widget are the same as those shown in Figure C.1.

11. *Load plant SEGS results.* The Load plant SEGS results loads the existing results of the SEGS algorithm on plant data from an XML file provided by the user. The purpose of this widget is to allow loading stored results of the SEGS algorithm for further analyses in order to avoid re-running SEGS.

12. *Plant rule browser.* The Plant rule browser widget displays ontology term conjuncts (i.e., rules) describing gene sets obtained by the SEGS algorithm adapted for plant gene expression data in an interactive HTML table. Ontology terms as well as corresponding sets of genes can be sent to the widget connected to the output by clicking on the provided links. Rule browser can also display rule cluster information if the Hierarchical clustering component is connected to the appropriate input.

13. *Set operations.* The Set operations widgets provides the following functions on sets: union, intersection, difference, and symmetric difference. The widget can be used to construct contrasting sets.

    Inputs:

    - ElementListA [type: list objects]
    - ElementListB [type: list objects]

    Outputs:

    - Union [type: list objects]
    - Intersection [type: list objects]
    - Difference [type: list objects]
    - SymmetricDifference [type: list objects]

14. *List concatenator.* The List concatenator widget concatenates two input lists. The widget can be used to construct an input to the SEGS algorithm web service.

    Inputs:

    - FirstList [type: list objects]
    - SecondList [type: list objects]

    Outputs:

    - ConcatenatedList [type: list objects]

15. *Plant rule browser.* The Plant rule browser widget displays ontology term conjuncts (i.e., rules) describing gene sets obtained by the SEGS algorithm adapted for plant gene expression data in an interactive HTML table. Ontology terms as well as corresponding sets of genes can be sent to the widget connected to the output by clicking on the provided links. Rule browser can also display rule cluster information if the Hierarchical clustering component is connected to the appropriate input.

16. *Get size.* The Get size widgets reports the size of the input object. The widget can be used to construct an input for the SEGS algorithm web service.

    Inputs:

    - InputObject [type: object]

    Outputs:

    - Size [type: integer]

17. *Number generator.* The Number generator widget constructs a list of numbers of specified length according to the selected rule. The widget can be used to construct an input for the SEGS algorithm web service. A screenshot of the widget is shown in Figure C.2.

    Inputs:

    - Length [type: integer]

    Outputs:

· NumberList [type: list of floats]

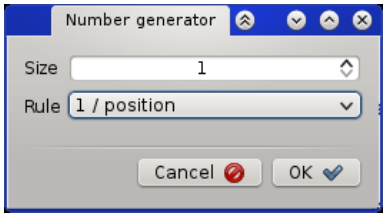

Figure C.2: A screenshot of the Number generator widget depicting available parameters.

18. *Input constructor for SEGS*. The Input constructor for SEGS widgets constructs a list of pairs (gene, rank) from two separate input lists of genes and ranks. The widget can be used to construct an input to the SEGS algorithm web service.

    Inputs:

    · GenesList [type: list of strings]
    · RanksList [type: list of floats]

    Outputs:

    · GeneRanks [type: list of tuples (string, float), sorted in descending order]

D Experimental results of the SEGS algorithm on human MSC data

This appendix presents the results of the experiments conducted using the implementation of the SEGS algorithm available in SegMine on the dataset where gene expression profiles from late senescent passages of MSC (mesenchymal stem cells) from three independent donors were compared to the MSC of early passages (Wagner et al., 2008). The presented experimental results served as a basis for expert data analysis with SegMine, which resulted in the construction of three novel scientific research hypotheses (discussed in depth in Chapter 3).

The dataset used in the experiments is available in two forms. First, the original dataset as deposited by Wagner et al. (2008) at GEO[1] is available through the series accession number GSE9664 at `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE9664`. Second, the preprocessed and reformatted version which was used in the presented experiments is available at `http://segmine.ijs.si/media/supplemental/datasets/hMSC_dataset.csv`.

The experimental results consist of four tables of rules (conjuncts of ontology terms), as obtained using the SEGS algorithm in the SegMine methodology data analysis workflow (the Rule browser widget was used to produce the tables). SEGS discovers relevant gene groups, described by logical rules formulated as conjunctions of ontology terms from GO, KEGG and Entrez. The rules semantically explain differentially expressed gene groups in terms of gene functions, components, processes, and pathways as annotated in biological ontologies.

Tables D.1, D.2, and D.3 contain rules constructed using SEGS operators and evaluated using Fisher's exact test, the GSEA method and the PAGE method, respectively. Table D.4 aggregates the first three using a weighted aggregation of p-values according to the following formula (Trajkovski, 2007):

$$p_{agg} = \frac{(W_{Fisher} * p_{Fisher} + W_{GSEA} * p_{GSEA} + W_{PAGE} * p_{PAGE})}{(W_{Fisher} + W_{GSEA} + W_{PAGE})}$$

Up to 100 most significant[2] rules are reported ($p \leq 0.05$) for each test and the aggregation. Note that the rules can contain optional *interacting* terms (shown enclosed in square brackets) which is a distinguished capability of the SEGS algorithm, explained in more detail in Chapter 3 and in the thesis of Trajkovski (2007). The results also include the total number of genes and the number of differentially expressed genes covered by each rule. The parameters of the SEGS algorithm are shown in Figure D.1

[1]Gene Expression Omnibus (GEO) is a public functional genomics data repository supporting MIAME-compliant data submissions.

[2]The p-values returned by SEGS are estimated using the permutation testing procedure. The enrichment scores are computed on the permuted input a specified number of times and the p-values can then be estimated according to the ratio of better scores from the random repetitions and the total number of permutations. Given a sufficient number of permutations this procedure gives a relatively reliable estimation of the p-value.

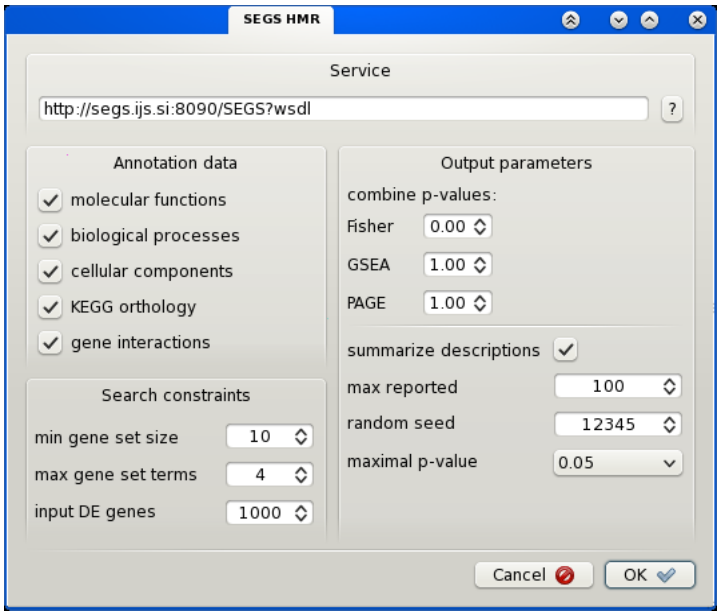

Figure D.1: The parameters of the SEGS algorithm in the presented experiments with the MSC dataset were set as shown in the figure.

Table D.1: List of rules obtained using the SEGS algorithm and Fisher's exact test on the MSC dataset in the SegMine data analysis workflow. Interacting ontology terms are shown enclosed in square brackets. The statistical significance of the rules is estimated using permutation testing.

#	ontology terms	all genes	DE genes	adjusted p-value
1	nucleus [mitochondrion organization]	229	53	0.000
2	intracellular organelle [regulation of neurogenesis]	367	73	0.010
3	[envelope]	726	123	0.010
4	intracellular membrane-bounded organelle [mitochondrion organization]	277	59	0.010
5	[organelle envelope]	714	121	0.030
6	[endomembrane system]	1258	191	0.040
7	RNA splicing intracellular non-membrane-bounded organelle	42	17	0.040
8	[mitochondrion organization]	371	72	0.040
9	cell surface receptor linked signaling pathway nuclear lumen	38	16	0.040
10	intracellular organelle [mitochondrion organization]	292	60	0.050

Table D.2: List of rules obtained using the SEGS algorithm and the GSEA method on the MSC dataset in the SegMine data analysis workflow. Interacting ontology terms are shown enclosed in square brackets. The statistical significance of the rules is estimated using permutation testing.

#	ontology terms	all genes	DE genes	adjusted p-value
1	DNA binding intracellular non-membrane-bounded organelle [mRNA metabolic process]	50	15	0.000
2	DNA binding intracellular non-membrane-bounded organelle [RNA splicing]	48	15	0.000
3	cellular macromolecule metabolic process intracellular non-membrane-bounded organelle [chromatin binding]	77	16	0.000
4	DNA metabolic process nucleus [chromatin binding]	73	17	0.000
5	chromosome [chromatin binding]	82	15	0.000
6	intracellular non-membrane-bounded organelle [protein-DNA complex assembly]	87	15	0.010
7	organelle organization nuclear part [chromatin binding]	75	16	0.010
8	organelle organization nuclear part [transcription corepressor activity]	60	16	0.010
9	RNA binding [microtubule cytoskeleton]	48	16	0.010
10	organelle organization intracellular non-membrane-bounded organelle [chromatin binding]	70	15	0.010
11	[microtubule motor activity]	70	15	0.010
12	DNA metabolic process intracellular organelle part [RNA binding]	73	15	0.010
13	nuclear part [protein-DNA complex assembly]	100	16	0.010
14	DNA binding nucleoplasm [mRNA processing]	61	16	0.010
15	regulation of macromolecule biosynthetic process regulation of cellular biosynthetic process intracellular non-membrane-bounded organelle [chromatin binding]	74	17	0.010
16	organelle organization nuclear lumen [transcription corepressor activity]	51	16	0.010
17	organelle organization nuclear part [Cell cycle]	97	15	0.010

Continued on next page

Table D.2 – *Continued from previous page*

#	ontology terms	all genes	DE genes	adjusted p-value
18	cellular response to stimulus intracellular organelle part [RNA binding]	71	16	0.010
19	nuclear part [p53 binding]	72	15	0.010
20	DNA metabolic process intracellular organelle part [receptor binding]	65	15	0.010
21	cellular response to stimulus nuclear part [nuclear hormone receptor binding]	59	16	0.010
22	organelle organization intracellular non-membrane-bounded organelle [transcription coactivator activity]	79	18	0.010
23	intracellular signal transduction [chromosome, centromeric region]	61	17	0.010
24	DNA metabolic process nuclear lumen [transcription factor binding]	83	16	0.010
25	regulation of macromolecule metabolic process intracellular non-membrane-bounded organelle [helicase activity]	70	16	0.010
26	organelle organization [heterochromatin]	73	16	0.010
27	regulation of macromolecule metabolic process intracellular non-membrane-bounded organelle [chromatin binding]	84	18	0.010
28	DNA metabolic process nuclear part [transcription cofactor activity]	72	15	0.010
29	DNA binding nucleus [Spliceosome]	64	18	0.010
30	cellular macromolecule metabolic process nuclear part [chromatin binding]	116	23	0.010
31	cellular macromolecule metabolic process intracellular non-membrane-bounded organelle [protein heterodimerization activity]	73	15	0.010
32	DNA metabolic process nucleus [RNA binding]	81	17	0.010
33	nucleus [protein complex localization]	52	15	0.010
34	DNA metabolic process nuclear part [zinc ion binding]	84	15	0.010

Continued on next page

Table D.2 – *Continued from previous page*

#	ontology terms	all genes	DE genes	adjusted p-value
35	cellular macromolecule metabolic process nuclear part [Glioma]	82	16	0.010
36	DNA binding nuclear lumen [mRNA metabolic process]	82	22	0.010
37	intracellular organelle part [protein-DNA complex assembly]	122	20	0.010
38	nuclear part [chromatin assembly or disassembly]	109	21	0.010
39	intracellular organelle part [nucleosome assembly]	91	15	0.010
40	intracellular organelle part [DNA packaging]	108	16	0.010
41	nucleoplasm [regulation of S phase]	87	19	0.010
42	organelle organization nucleus [steroid hormone receptor binding]	78	17	0.010
43	DNA metabolic process nuclear part [transcription factor binding]	98	18	0.010
44	nucleoplasm [G1/S transition of mitotic cell cycle]	87	18	0.010
45	enzyme binding nucleus [cell cycle arrest]	49	15	0.010
46	regulation of transcription intracellular non-membrane-bounded organelle [chromatin binding]	66	15	0.010
47	chromosome organization nucleus [chromatin binding]	78	17	0.010
48	DNA metabolic process intracellular organelle [RNA binding]	82	17	0.020
49	DNA metabolic process intracellular organelle lumen [transcription factor binding]	85	16	0.020
50	macromolecule metabolic process intracellular non-membrane-bounded organelle [protein heterodimerization activity]	74	15	0.020
51	enzyme binding intracellular organelle [Colorectal cancer]	44	15	0.020
52	regulation of transcription nuclear part [protein domain specific binding]	91	22	0.020

Continued on next page

Table D.2 – *Continued from previous page*

#	ontology terms	all genes	DE genes	adjusted p-value
53	organelle organization nucleus [chromatin binding]	99	21	0.020
54	nucleoplasm [Melanoma]	83	15	0.020
55	regulation of macromolecule biosynthetic process regulation of cellular biosynthetic process nuclear part [Oocyte meiosis]	66	16	0.020
56	DNA binding nuclear lumen [mRNA processing]	74	21	0.020
57	nucleoplasm [DNA conformation change]	97	15	0.020
58	chromosome organization [chromatin]	97	19	0.020
59	regulation of macromolecule metabolic process nuclear lumen [Oocyte meiosis]	72	15	0.020
60	nucleoside-triphosphatase activity intracellular organelle part [gene expression]	68	15	0.020
61	nuclear lumen [regulation of S phase]	98	22	0.020
62	regulation of macromolecule metabolic process intracellular non-membrane-bounded organelle [Oocyte meiosis]	67	15	0.020
63	organelle organization [chromosome, centromeric region]	130	18	0.020
64	regulation of gene expression intracellular non-membrane-bounded organelle [chromatin binding]	68	16	0.020
65	organelle organization intracellular organelle part [chromatin binding]	89	18	0.020
66	nuclear part [regulation of fatty acid metabolic process]	67	15	0.020
67	DNA metabolic process intracellular organelle part [zinc ion binding]	87	16	0.020
68	regulation of macromolecule metabolic process nuclear part [Oocyte meiosis]	87	18	0.020
69	DNA binding nuclear part [Spliceosome]	52	17	0.020

Continued on next page

Table D.2 – *Continued from previous page*

#	ontology terms	all genes	DE genes	adjusted p-value
70	regulation of macromolecule biosynthetic process regulation of cellular biosynthetic process nucleoplasm [protein domain specific binding]	68	17	0.020
71	DNA binding intracellular organelle lumen [mRNA metabolic process]	76	21	0.020
72	organelle organization intracellular organelle part [protein N-terminus binding]	65	15	0.020
73	organelle organization intracellular non-membrane-bounded organelle [Small cell lung cancer]	65	16	0.020
74	chromosome organization nuclear part [transcription cofactor activity]	74	18	0.020
75	chromosome organization intracellular organelle part [transcription coactivator activity]	72	15	0.020
76	DNA metabolic process nucleus [transcription coactivator activity]	77	18	0.020
77	nucleus [motor activity]	74	15	0.020
78	nucleoside-triphosphatase activity intracellular organelle [positive regulation of apoptosis]	67	15	0.020
79	enzyme binding nucleus [Pancreatic cancer]	53	15	0.020
80	intracellular non-membrane-bounded organelle [chromatin assembly or disassembly]	93	17	0.020
81	nucleoplasm [enzyme activator activity]	61	16	0.020
82	nucleus [protein-DNA complex assembly]	146	24	0.020
83	organelle organization intracellular organelle part [steroid hormone receptor binding]	75	16	0.020
84	cellular response to stimulus intracellular organelle part [nuclear hormone receptor binding]	62	16	0.020
85	nucleoplasm [Oocyte meiosis]	98	16	0.020
86	nuclear lumen [chromatin assembly or disassembly]	89	16	0.020

Continued on next page

Table D.2 – *Continued from previous page*

#	ontology terms	all genes	DE genes	adjusted p-value
87	DNA metabolic process intracellular organelle part [transcription cofactor activity]	76	16	0.020
88	regulation of gene expression nuclear part [protein domain specific binding]	94	25	0.020
89	cellular macromolecule metabolic process intracellular non-membrane-bounded organelle [helicase activity]	95	16	0.030
90	nucleoplasm [mitotic cell cycle G1/S transition checkpoint]	66	16	0.040
91	nucleoplasm [chromatin binding]	122	28	0.040
92	nucleoplasm [mitotic cell cycle checkpoint]	83	17	0.040
93	regulation of macromolecule biosynthetic process regulation of cellular biosynthetic process nuclear part [chromatin binding]	133	31	0.040
94	nucleoplasm [Bladder cancer]	83	15	0.040
95	organelle organization [kinetochore]	114	15	0.040
96	cellular macromolecule metabolic process nuclear lumen [chromatin binding]	97	20	0.040
97	nuclear part [regulation of S phase]	116	24	0.040
98	cellular response to stimulus nuclear lumen [receptor binding]	61	15	0.040
99	intracellular organelle part [nucleosome organization]	100	16	0.040
100	DNA binding organelle lumen [mRNA metabolic process]	77	21	0.040

Table D.3: List of rules obtained using the SEGS algorithm and the PAGE method on the MSC dataset in the SegMine data analysis workflow. Interacting ontology terms are shown enclosed in square brackets. The statistical significance of the rules is estimated using permutation testing.

#	ontology terms	all genes	DE genes	adjusted p-value
1	nuclear part [DNA binding]	503	80	0.000
2	intracellular organelle part [DNA binding]	709	105	0.000
3	nuclear part [programmed cell death]	324	58	0.000
4	nuclear part [regulation of transcription]	559	81	0.000
5	nuclear part [transcription factor binding]	392	66	0.000
6	nuclear part [gene expression] [RNA metabolic process]	419	72	0.000
7	nuclear part [apoptosis]	283	54	0.000
8	nuclear lumen [DNA binding]	422	70	0.000
9	nuclear part	948	129	0.000
10	intracellular organelle part [RNA metabolic process]	568	92	0.000
11	nuclear part [regulation of transcription from RNA polymerase II promoter]	388	68	0.000
12	nuclear lumen [programmed cell death]	274	53	0.000
13	nucleus [gene expression] [RNA metabolic process]	741	118	0.000
14	intracellular organelle part [gene expression]	619	99	0.000
15	intracellular organelle part [regulation of transcription]	859	118	0.000
16	nuclear part [transcription cofactor activity]	309	57	0.000
17	nuclear lumen [apoptosis]	239	49	0.000
18	intracellular organelle part [purine nucleoside binding] [purine ribonucleotide binding]	732	118	0.000
19	intracellular organelle part [RNA processing]	417	72	0.000
20	nuclear part [regulation of transcription, DNA-dependent]	475	74	0.000
21	nuclear part [cellular macromolecule metabolic process]	625	88	0.000

Continued on next page

Table D.3 – *Continued from previous page*

#	ontology terms	all genes	DE genes	adjusted p-value
22	intracellular organelle part [cellular macromolecule metabolic process]	1080	149	0.000
23	nuclear part [organelle organization]	438	66	0.000
24	nuclear lumen [gene expression] [RNA metabolic process]	391	68	0.000
25	intracellular organelle part [regulation of transcription, DNA-dependent]	685	104	0.000
26	nucleoplasm [DNA binding]	321	54	0.000
27	nuclear part [DNA metabolic process]	312	53	0.000
28	intracellular organelle lumen [cellular macromolecule metabolic process]	569	85	0.000
29	nuclear part [RNA processing]	316	57	0.000
30	nuclear lumen	741	106	0.000
31	nuclear part [transcription activator activity]	347	59	0.000
32	intracellular organelle part [post-translational protein modification]	793	119	0.000
33	cellular macromolecule metabolic process nuclear part [DNA binding]	336	51	0.000
34	intracellular organelle part [regulation of macromolecule metabolic process]	1020	137	0.000
35	nuclear part [enzyme binding]	381	64	0.000
36	nuclear part [positive regulation of transcription, DNA-dependent]	381	64	0.000
37	nuclear part [post-translational protein modification]	473	69	0.000
38	intracellular organelle part [ATP binding]	677	104	0.000
39	nuclear lumen [cellular macromolecule metabolic process]	510	78	0.000
40	nuclear lumen [post-translational protein modification]	398	65	0.000
41	nuclear part [response to chemical stimulus]	407	68	0.000
42	nuclear part [chromatin binding]	180	41	0.000
43	nucleus [RNA processing]	527	85	0.000
44	intracellular organelle lumen	927	126	0.000

Continued on next page

Table D.3 – *Continued from previous page*

#	ontology terms	all genes	DE genes	adjusted p-value
45	cellular macromolecule metabolic process intracellular organelle part [DNA binding]	385	55	0.000
46	[ribonucleoprotein complex]	594	101	0.000
47	intracellular membrane-bounded organelle [RNA metabolic process]	872	130	0.000
48	intracellular membrane-bounded organelle [RNA processing]	621	96	0.000
49	nucleus [DNA metabolic process]	589	91	0.000
50	nuclear lumen [regulation of transcription]	501	75	0.000
51	organelle lumen [programmed cell death]	310	56	0.000
52	cellular macromolecule metabolic process nucleus [DNA binding]	580	79	0.000
53	nuclear lumen [organelle organization]	386	61	0.000
54	intracellular organelle lumen [ATP binding]	375	63	0.000
55	nuclear part [Pathways in cancer]	307	55	0.000
56	nucleoplasm	449	66	0.000
57	intracellular organelle part [cellular macromolecule biosynthetic process]	523	81	0.000
58	cellular macromolecule metabolic process nucleus [ATP binding]	445	64	0.000
59	nuclear lumen [transcription factor binding]	333	58	0.000
60	nuclear part [negative regulation of transcription]	371	59	0.000
61	nuclear part [cellular macromolecule biosynthetic process]	381	61	0.000
62	organelle lumen [apoptosis]	272	52	0.000
63	nuclear lumen [RNA processing]	287	54	0.000
64	nuclear lumen [regulation of cell growth] [regulation of cell size]	191	41	0.000
65	nuclear part [ATP binding]	405	61	0.000
66	nuclear lumen [regulation of transcription from RNA polymerase II promoter]	333	60	0.000
67	cellular macromolecule metabolic process nuclear part	516	71	0.000

Continued on next page

Table D.3 – *Continued from previous page*

#	ontology terms	all genes	DE genes	adjusted p-value
68	nuclear part [Cell cycle]	272	44	0.000
69	nuclear part [cellular response to stress]	357	58	0.000
70	nucleoplasm [gene expression] [RNA metabolic process]	272	51	0.000
71	nuclear part [positive regulation of macromolecule metabolic process] [positive regulation of cellular metabolic process]	433	66	0.000
72	intracellular organelle part [cellular biosynthetic process]	612	94	0.000
73	nuclear lumen [DNA metabolic process]	275	50	0.000
74	intracellular organelle part [regulation of transcription from RNA polymerase II promoter]	546	86	0.000
75	nucleoplasm [cellular macromolecule metabolic process] [regulation of macromolecule metabolic process]	364	55	0.000
76	intracellular organelle part [protein metabolic process]	874	124	0.000
77	intracellular organelle part [RNA binding]	383	63	0.000
78	nuclear part [regulation of apoptosis]	366	60	0.000
79	nuclear part [sequence-specific DNA binding transcription factor activity]	330	54	0.000
80	nuclear part [cell cycle process]	335	48	0.000
81	nuclear lumen [post-translational protein modification]	379	59	0.000
82	cellular macromolecule metabolic process nuclear lumen [DNA binding]	286	46	0.000
83	nucleoplasm [regulation of transcription from RNA polymerase II promoter]	266	49	0.000
84	regulation of macromolecule metabolic process nuclear part [DNA binding]	311	55	0.000
85	cellular macromolecule metabolic process [nucleoplasm]	707	95	0.000
86	nucleus [cellular macromolecule metabolic process]	1305	169	0.000
87	intracellular organelle part [apoptosis]	503	78	0.000
88	cellular macromolecule metabolic process nuclear part [enzyme binding]	250	43	0.000

Continued on next page

Table D.3 – *Continued from previous page*

#	ontology terms	all genes	DE genes	adjusted p-value
89	cellular macromolecule metabolic process intracellular membrane-bounded organelle [ATP binding]	519	73	0.000
90	nuclear part [nervous system development]	324	57	0.000
91	nucleoplasm [metal ion binding]	274	46	0.000
92	intracellular organelle part [organelle organization]	770	111	0.000
93	regulation of macromolecule metabolic process nuclear part [transcription cofactor activity]	210	44	0.000
94	nuclear part [sequence-specific DNA binding]	311	54	0.000
95	intracellular membrane-bounded organelle [cellular macromolecule metabolic process]	1729	224	0.000
96	nuclear part [negative regulation of macromolecule metabolic process]	438	64	0.000
97	nuclear part [positive regulation of transcription from RNA polymerase II promoter]	303	54	0.000
98	[nucleoplasm part]	1262	183	0.000
99	nuclear part [macromolecular complex assembly]	378	57	0.000
100	intracellular organelle [cellular macromolecule metabolic process]	1907	249	0.000

Table D.4: List of rules obtained using the SEGS algorithm on the MSC dataset in the SegMine data analysis workflow. Interacting ontology terms are shown enclosed in square brackets. The statistical significance of the rules is computed using the aggregation function on the adjusted p-values of Fisher's exact test, GSEA, and PAGE. The weight vector used in the aggregation was set to $\langle 0.0, 1.0, 1.0 \rangle$.

#	ontology terms	all genes	DE genes	aggregate p-value
1	intracellular organelle part [protein-DNA complex assembly]	122	20	0.005
2	nucleoplasm [regulation of S phase]	87	19	0.005
3	DNA binding intracellular non-membrane-bounded organelle [mRNA metabolic process]	50	15	0.005
4	DNA binding intracellular organelle part [mRNA metabolic process]	85	22	0.005
5	DNA binding intracellular non-membrane-bounded organelle [RNA splicing]	48	15	0.005
6	DNA metabolic process nucleus [chromatin binding]	73	17	0.005
7	cellular macromolecule metabolic process nuclear part [chromatin binding]	116	23	0.005
8	organelle organization nuclear part [chromatin binding]	75	16	0.005
9	cellular macromolecule metabolic process nuclear part [Glioma]	82	16	0.010
10	organelle organization nuclear part [Cell cycle]	97	15	0.010
11	organelle organization [chromosome, centromeric region]	130	18	0.010
12	organelle organization [heterochromatin]	73	16	0.010
13	chromosome organization [chromatin]	97	19	0.010
14	intracellular signal transduction [chromosome, centromeric region]	61	17	0.010
15	nuclear part [chromatin assembly or disassembly]	109	21	0.010
16	nucleoplasm [G1/S transition of mitotic cell cycle]	87	18	0.010
17	nucleus [protein-DNA complex assembly]	146	24	0.010
18	intracellular non-membrane-bounded organelle [protein-DNA complex assembly]	87	15	0.010

Continued on next page

Table D.4 – *Continued from previous page*

#	ontology terms	all genes	DE genes	aggregate p-value
19	nuclear lumen [regulation of S phase]	98	22	0.010
20	DNA binding nuclear lumen [mRNA metabolic process]	82	22	0.010
21	DNA metabolic process intracellular membrane-bounded organelle [RNA binding]	81	17	0.010
22	cellular response to stimulus intracellular organelle part [RNA binding]	71	16	0.010
23	organelle organization nucleus [chromatin binding]	99	21	0.010
24	chromosome organization nucleus [chromatin binding]	78	17	0.010
25	DNA metabolic process intracellular organelle part [receptor binding]	65	15	0.010
26	cellular response to stimulus nuclear part [nuclear hormone receptor binding]	59	16	0.010
27	organelle organization nucleus [steroid hormone receptor binding]	78	17	0.010
28	DNA metabolic process nuclear part [transcription factor binding]	98	18	0.010
29	DNA metabolic process nuclear lumen [transcription factor binding]	83	16	0.010
30	DNA metabolic process nuclear part [transcription cofactor activity]	72	15	0.010
31	chromosome organization nuclear part [transcription cofactor activity]	74	18	0.010
32	organelle organization intracellular non-membrane-bounded organelle [transcription coactivator activity]	79	18	0.010
33	organelle organization nuclear lumen [transcription corepressor activity]	51	16	0.010
34	organelle organization nuclear part [transcription corepressor activity]	60	16	0.010

Continued on next page

Table D.4 – *Continued from previous page*

#	ontology terms	all genes	DE genes	aggregate p-value
35	regulation of transcription nuclear part [protein domain specific binding]	91	22	0.010
36	regulation of gene expression nuclear part [protein domain specific binding]	94	25	0.010
37	nucleoplasm [Oocyte meiosis]	98	16	0.015
38	regulation of macromolecule metabolic process nuclear part [Oocyte meiosis]	87	18	0.015
39	nuclear part [regulation of fatty acid metabolic process]	67	15	0.015
40	DNA binding intracellular organelle lumen [mRNA metabolic process]	76	21	0.015
41	DNA binding nuclear lumen [mRNA processing]	74	21	0.015
42	DNA metabolic process nucleus [RNA binding]	80	17	0.015
43	DNA metabolic process intracellular organelle part [RNA binding]	73	15	0.015
44	chromosome [chromatin binding]	82	15	0.015
45	cellular macromolecule metabolic process intracellular non-membrane-bounded organelle [chromatin binding]	77	16	0.015
46	organelle organization intracellular organelle part [chromatin binding]	89	18	0.015
47	cellular response to stimulus intracellular organelle part [nuclear hormone receptor binding]	62	16	0.015
48	organelle organization intracellular organelle part [steroid hormone receptor binding]	75	16	0.015
49	DNA metabolic process intracellular organelle part [transcription cofactor activity]	76	16	0.015
50	DNA metabolic process nucleus [transcription coactivator activity]	77	18	0.015
51	chromosome organization intracellular organelle part [transcription coactivator activity]	72	15	0.015

Continued on next page

Table D.4 – *Continued from previous page*

#	ontology terms	all genes	DE genes	aggregate p-value
52	regulation of macromolecule biosynthetic process regulation of cellular biosynthetic process nucleoplasm [protein domain specific binding]	68	17	0.015
53	intracellular membrane-bounded organelle [protein-DNA complex assembly]	155	26	0.020
54	nuclear part [protein-DNA complex assembly]	100	16	0.020
55	intracellular organelle part [DNA packaging]	108	16	0.020
56	intracellular organelle lumen [regulation of S phase]	100	22	0.020
57	nuclear part [regulation of S phase]	116	24	0.020
58	DNA binding nucleus [Spliceosome]	64	18	0.020
59	DNA binding nucleoplasm [mRNA metabolic process]	61	16	0.020
60	RNA binding [microtubule cytoskeleton]	48	16	0.020
61	regulation of macromolecule metabolic process intracellular non-membrane-bounded organelle [helicase activity]	70	16	0.020
62	nuclear part [chromatin binding]	180	41	0.020
63	nucleoplasm [chromatin binding]	122	28	0.020
64	macromolecule metabolic process nuclear lumen [chromatin binding]	97	20	0.020
65	regulation of macromolecule biosynthetic process regulation of cellular biosynthetic process nuclear part [chromatin binding]	120	28	0.020
66	DNA metabolic process intracellular organelle lumen [transcription factor binding]	85	16	0.020
67	organelle organization nuclear part [transcription cofactor activity]	98	22	0.020
68	organelle organization nuclear lumen [transcription cofactor activity]	78	20	0.020
69	regulation of macromolecule metabolic process nucleoplasm [protein domain specific binding]	78	18	0.020

Continued on next page

Table D.4 – *Continued from previous page*

#	ontology terms	all genes	DE genes	aggregate p-value
70	DNA metabolic process intracellular organelle part [transition metal ion binding]	93	17	0.020
71	nucleoplasm [Glioma]	85	16	0.025
72	organelle organization intracellular non-membrane-bounded organelle [Small cell lung cancer]	65	16	0.025
73	organelle organization [kinetochore]	114	15	0.025
74	organelle organization [chromatin]	139	25	0.025
75	chromosome organization [nuclear chromatin]	72	15	0.025
76	regulation of macromolecule metabolic process intracellular organelle lumen [Oocyte meiosis]	74	15	0.025
77	nucleoplasm [negative regulation of organelle organization]	85	16	0.025
78	nuclear part [mitotic cell cycle checkpoint]	112	19	0.025
79	nucleoplasm [mitotic cell cycle checkpoint]	83	17	0.025
80	nuclear part [mitotic cell cycle G1/S transition checkpoint]	88	18	0.025
81	DNA binding organelle lumen [mRNA metabolic process]	77	21	0.025
82	DNA binding intracellular organelle lumen [mRNA processing]	68	20	0.025
83	regulation of macromolecule metabolic process nuclear part [hydrolase activity, acting on ester bonds]	75	15	0.025
84	regulation of macromolecule metabolic process [DNA helicase activity]	96	18	0.025
85	intracellular signal transduction nuclear part [DNA binding]	70	15	0.025
86	chromosome organization nuclear part [sequence-specific DNA binding]	78	16	0.025
87	cellular response to stimulus intracellular organelle [RNA binding]	94	20	0.025
88	cellular macromolecule metabolic process nuclear lumen [chromatin binding]	96	19	0.025

Continued on next page

Table D.4 – *Continued from previous page*

#	ontology terms	all genes	DE genes	aggregate p-value
89	organelle organization intracellular membrane-bounded organelle [chromatin binding]	103	21	0.025
90	organelle organization intracellular membrane-bounded organelle [steroid hormone receptor binding]	80	17	0.025
91	DNA metabolic process organelle lumen [transcription factor binding]	86	16	0.025
92	DNA metabolic process intracellular organelle part [transcription factor binding]	102	19	0.025
93	organelle organization intracellular organelle lumen [transcription cofactor activity]	80	20	0.025
94	organelle organization intracellular non-membrane-bounded organelle [transcription corepressor activity]	72	20	0.025
95	regulation of macromolecule metabolic process nuclear part [protein domain specific binding]	109	26	0.025
96	DNA metabolic process nuclear part [zinc ion binding]	87	15	0.025
97	regulation of macromolecule metabolic process nuclear lumen [Oocyte meiosis]	72	15	0.030
98	nucleoside-triphosphatase activity intracellular organelle [positive regulation of apoptosis]	67	15	0.030
99	macromolecule metabolic process intracellular non-membrane-bounded organelle [chromatin binding]	78	16	0.030
100	organelle organization intracellular organelle part [protein N-terminus binding]	65	15	0.030

E Biography

Vid Podpečan was born on 19 June 1982 in Slovenj Gradec, Slovenia. After completing the secondary education at Gimnazija Velenje in 2001 he enrolled in the undergraduate programme at the Faculty of Computer and Information Science, University of Ljubljana. He completed the undergraduate study with a diploma thesis under the supervision of prof. dr. Igor Kononenko and co-supervision of prof. dr. Nada Lavrač.

In 2007 he was accepted to the Young researchers programme of the Slovenian Research Agency under the supervision of prof. dr. Nada Lavrač, and he enrolled in the PhD programme New Media and E-science at the Jožef Stefan International Postgraduate School. He has collaborated on European projects BISON, Envision, and MUSE. The results of his research work were published in several journal and conference publications.