CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2007; **19**:2047–2066
Published online 19 December 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1144

MS-Analyzer: preprocessing and data mining services for proteomics applications on the Grid

Mario Cannataro*,† and Pierangelo Veltri

Department of Experimental Medicine and Clinic,
University Magna Græcia of Catanzaro, Italy

SUMMARY

Mass spectrometry proteomics data contain much information about cell functions and disease conditions. The discovery of such information is enabled by the combined use of novel bioinformatics tools and data mining techniques requiring the integration of huge data sources and the composition of different software tools. The main phases of such emerging applications comprise the loading, management, preprocessing, mining, and visualization of spectra, as well as the analysis of discovered knowledge models. The collection, storage, and analysis of spectra produced in different laboratories can make use of the services of computational Grids, which offer efficient data transfer primitives, effective management of large data stores, and large computing power. In this paper we present MS-Analyzer, a Grid-based software platform for the integrated management and analysis of spectra data. MS-Analyzer provides efficient spectra management through a specialized spectra database, and supports the semantic composition of spectra preprocessing services and data mining services to analyze spectra on the Grid. Copyright © 2006 John Wiley & Sons, Ltd.

Received 26 August 2006; Accepted 27 August 2006

KEY WORDS: mass spectrometry data analysis; Grid problem solving environment; ontology; workflow

1. INTRODUCTION

Mass spectrometry (MS) is an analytical tool used for measuring the molecular mass of a sample. MS-based proteomics is a powerful technique for identifying molecular targets in different pathological conditions [1]. Classical bioinformatics tasks, such as protein sequence alignment, protein structure prediction, peptide identification, etc., are increasingly combined with data mining and machine

*Correspondence to: Mario Cannataro, University Magna Græcia of Catanzaro, Campus of Germaneto, Viale Europa, 88100 Catanzaro, Italy.
†E-mail: cannataro@unicz.it

Contract/grant sponsor: MIUR (GRID.IT Project); contract/grant number: RBNE01KNFP

learning algorithms to obtain specialized computational platforms. MS is increasingly used in clinical studies leading to huge collections of data. Moreover, MS data, i.e. spectra, may be affected by errors and noise as a result of sample preparation and instrument approximation. Thus, the main requirements for the analysis of spectra data are: (i) efficient spectra representation and management to enable the high throughput and large-scale analysis required in clinical studies; (ii) effective and efficient preprocessing algorithms for noise cleaning and data size reduction; (iii) flexible and semantic-based composition of software tools, to face heterogeneous instruments and data formats, and to enable different analysis techniques.

The collection, storage, and analysis of mass spectra can make use of the computational power of Grids [2], which offer efficient data transfer primitives (e.g. Globus GridFTP [3]), effective management of large data stores (e.g. replica management), and high computing power. However, the heterogeneity of technological platforms and the need for scientists to execute the same experiment by changing preprocessing and data mining algorithms as parameters, may necessitate the semantic modeling of bioinformatics resources as well as the use of workflows to represent *in silico* experiments. Modeling the different available algorithms, data sources, and analysis strategies can be carried out using domain ontologies [4]. High-throughput application execution can be supported by workflows of Grid services [5]. Thus, the use of ontologies and workflows provides biomedical researchers with easy problem formulation and application coding, hiding the software configuration details.

In this paper we present the design and implementation of MS-Analyzer, a software tool that allows the modeling and design of distributed applications for the preprocessing and data mining analysis of MS data on the Grid. The scenario we envision consists of a set of distributed proteomics facilities that produce spectra and make use of the specialized services of MS-Analyzer through the Grid; the produced raw spectra are collected by using the data transfer capabilities of the Grid and stored in a central spectra database managed by MS-Analyzer.

The basic services regarding storing, preprocessing, and data mining of MS data are modeled through ontologies, whereas the applications are designed by combining such services through workflows. This task is accomplished by an ontology-based workflow designer and scheduler, a central component of the system that guides users in the choice of algorithms, enforces constraints on services composition, and executes a workflow of services on the Grid. By using MS-Analyzer a user can produce different workflows of the same application, in a short time, by considering different combinations of preprocessing and data mining techniques, and can thus evaluate the best strategies to analyze mass spectra data. The main contribution of this paper is the MS-Analyzer software platform, which comprises some other relevant contributions: a specialized spectra database, an Ontology-based Service Discovery system, and an ontology-based workflow designer.

The rest of the paper is organized as follows. In Section 2 we report on related works. In Section 3 we introduce MS data and the main preprocessing techniques. In Section 4 we describe the architecture and functions of the proposed MS-Analyzer system. In Section 5 we present the Ontology-based Service Discovery system used in MS-Analyzer. In Section 6 we describe the implemented prototype. Some initial performance results are discussed in Section 7 while Section 8 concludes the paper.

2. RELATED WORKS

In the last few years many systems dealing with spectra management have been developed, but to the best of our knowledge none of these systems implements the complete knowledge discovery process for the analysis of mass spectra and biological information extraction. To compare MS-Analyzer to existing systems and approaches, the discussion is moved from domain-specific tools to general-purpose platforms, considering also the availability on the Grid. The relevant systems considered are mass spectra management tools, laboratory information management systems (LIMSs), bioinformatics and scientific workflow systems, and distributed data mining systems. A comprehensive taxonomy of workflow management systems for Grids can be found in [5], whereas a survey of distributed data mining systems is reported in [6].

Systems such as SpecAlign [7], MSAnalyzer (please note the similarity of this name with our MS-Analyzer) [8], and those developed in [9], are all specialized in preprocessing, visualization, and analysis of multiple mass spectra, but they neither support data mining of spectra and workflow composition, nor do they include domain ontologies. LabBase [10] and similar LIMSs are useful for managing experiments conducted in a laboratory and the related data, but are inadequate to support sophisticated analysis. Bioinformatics platforms, such as the genomics Research Network Architecture (gRNA) [11] and the Pegasys [12] bioinformatics system, offer some sort of configurable engine to pipeline a set of tasks and data, but neither support mass spectra data mining, nor the various knowledge discovery steps for mining other data types. myGrid [13] is a powerful toolkit to build bioinformatics workflows that offers a large set of bioinformatics tools wrapped as Web services, makes use of ontologies, and uses the powerful Taverna workflow editor [14]. Although MS-Analyzer shares many architectural choices and solutions with myGrid, it is specialized for mass spectra data mining and to this end it offers spectra preprocessing and preparation tools not present in myGrid.

General-purpose workflow editors (see [5]), such as Kepler, Pegasus, Triana, and Taverna, are suitable to support the composition of knowledge discovery workflows, but all lack spectra management functions and few of them use ontologies. Finally, distributed data mining environments, such as the Discovery Net [15] or the emerging Knowledge Grids [6,16], although providing support for the entire knowledge discovery process, do not offer efficient spectra management functions and storage or specialized spectra preprocessing and preparation services.

In summary, although many bioinformatics workflow systems for application do exist, few of these are related to MS proteomics and none of them support the entire set of operations needed to preprocess and analyze spectra.

3. MS DATA

MS is a technique that allows the identification of macromolecules in a compound. The mass spectrometer separates gas phase ions according to their m/z (mass-to-charge ratio) values [1]. The sample can be inserted directly into the ionization source, or can undergo some type of separation, such as liquid chromatography (LC), gas chromatography (GC), or capillary electrophoresis (CE), where the sample is separated into different components which enter the spectrometer sequentially for individual analysis. Commonly used ionization techniques are electrospray ionization (ESI), surface enhanced laser desorption/ionization (SELDI), and matrix-assisted laser desorption/ionization (MALDI), coupled with different types of mass analyzer such as time of flight (TOF) or quadrupole ion traps. Tandem (MS–MS) mass spectrometers are instruments that have more than one analyzer and can be used for structural and sequencing studies. Some popular tandem mass spectrometers use the quadrupole-TOF (Q-TOF) geometry.

The MS output, i.e. the spectrum, can be represented as a (large) sequence of value pairs. Each pair contains a measured intensity, which depends on the quantity of the detected biomolecule, and a

DOI: 10.1002/cpe

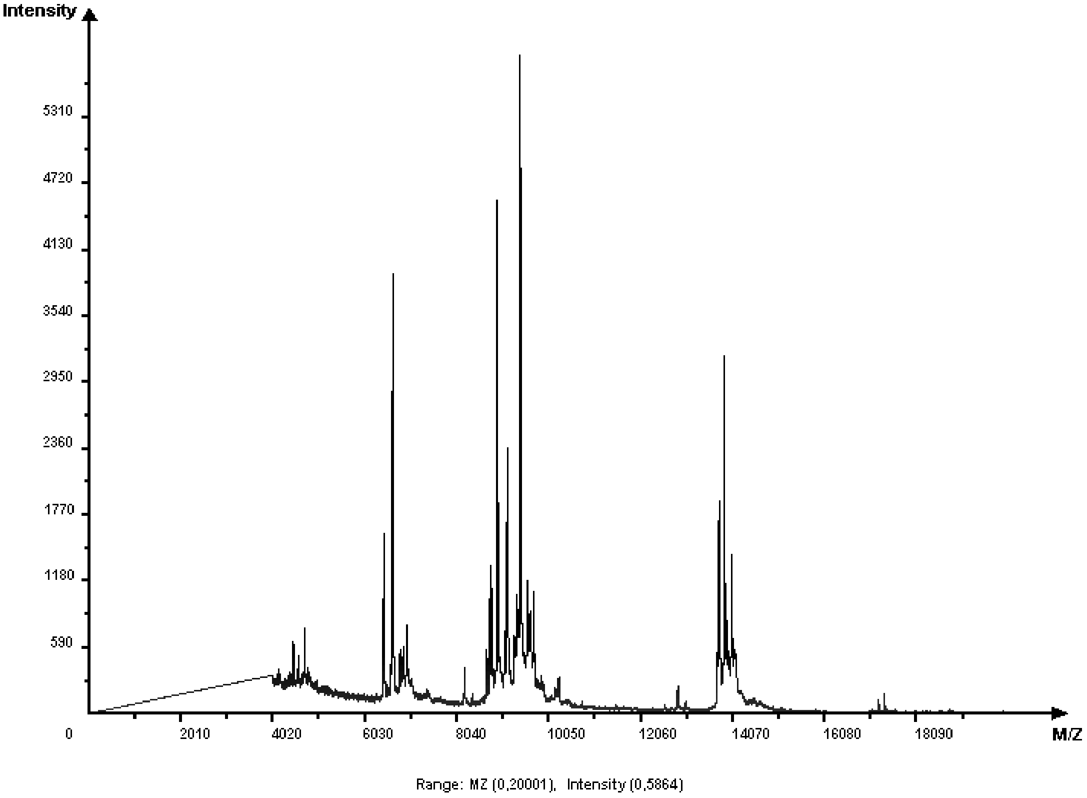


Figure 1. Example of a raw MALDI–TOF spectrum.

mass-to-charge ratio (m/z), which depends on the molecular mass of the detected biomolecule. As an example, Figure 1 shows a MALDI–TOF spectrum generated from a real biological sample.

Files dimensions range from a few kilobytes per spectrum to a few gigabytes. This variability depends on the type of spectrometer and the bin dimension, i.e. the total number of measurements. As an example, in an experiment performed in our proteomics laboratory the spectrometer executed 4000 scans acquiring approximately 2000 mass spectra and another 2000 spectra of selected and fragmented peptides (MS/MS spectra) per biological sample. The tandem mass spectrometer comprises an on-line strong cation exchange (SCX) and reversed-phased (RP) chromatography coupled to a QSTAR XL Q-TOF hybrid mass spectrometer [17]. The dimensions of each elementary spectrum, stored in an ASCII file, vary from 50 to 800 kB. Thus, the dimension of a single datum ranges from 200 MB to 3.2 GB in uncompressed format, while a complete dataset of 20 samples is at least 2 GB. Increasing either the resolution of the spectrometer or the number of analyzed biological samples may lead to very huge datasets that require large storage systems and high computing power.

DOI: 10.1002/cpe

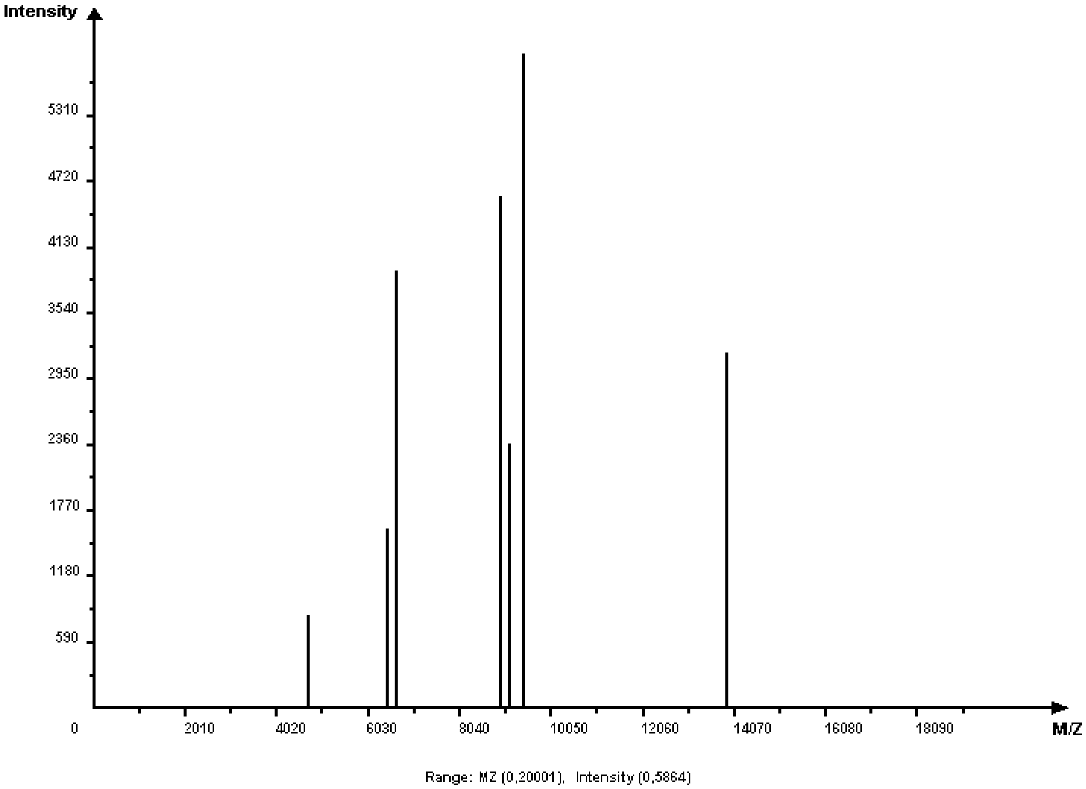


Figure 2. Example of a preprocessed MALDI–TOF spectrum.

The measurements contained in a spectrum may be affected by noise, so spectra preprocessing aims to correct intensity and m/z values in order to reduce noise, reduce the amount of data, and make spectra comparable [18]. Noise reduction comprises baseline subtraction and smoothing. The former flattens the base profile of a spectrum subtracting the so-called baseline, i.e. a base intensity level which varies from fraction to fraction of the spectrum. The latter reduces the noise level in the whole spectrum increasing the signal-to-noise ratio. Normalization enables the comparison of different samples by normalizing intensities. Binning performs data dimensionality reduction by aggregating measured data into bins: a set of peaks from a spectrum is substituted with a unique peak $(I, m/z)$, whose intensity I is an aggregate function of the original intensities (e.g. their sum), and the mass m/z is usually chosen from the original mass values (e.g. the median value). Peak extraction consists of separating real peaks (e.g. corresponding to peptides) from those representing noise. Peak alignment corrects errors on m/z measurements, finding a common set of peak locations in a set of spectra, in such a way that all aligned spectra will have common m/z values for the same biological entities. As an example, Figure 2 shows the spectrum of Figure 1 after performing noise reduction and peak extraction preprocessing.

DOI: 10.1002/cpe

4. MS-ANALYZER

MS-Analyzer is a Grid-based problem solving environment for the design and execution of proteomics applications. It uses domain ontologies to model software tools and spectra data, and workflow techniques to design data analysis applications (*in silico* experiments). In particular, ontologies model bioinformatics knowledge about the following: (i) biological databases (e.g. Protein Data Bank [19], and SwissProt [20]); (ii) experimental data sets (e.g. a set of spectra); (iii) bioinformatics software tools (e.g. preprocessing tools, peptide identification tools, etc.); (iv) bioinformatics processes (e.g. a workflow of a classification experiment).

MS-Analyzer glues distributed proteomic facilities and data analysis tools through a specialized spectra database and a set of preprocessing and data mining services. In particular it supports: (i) interfacing remote and heterogeneous proteomics facilities; (ii) storing and managing MS proteomics data abstracting experimental datasets; (iii) integrating off-the-shelf data mining and visualization software tools. Services may be discovered using an ontology-based discovery system supporting ontology browsing and querying. It is based on the cooperation and integration between ontologies and the basic discovery services of the Semantic Web (i.e. UDDI [21]) and of the Grid (i.e. Globus MDS [3]).

4.1. Architecture

MS-Analyzer adopts the service-oriented architecture: it provides a collection of specialized spectra management services and integrates publicly available off-the-shelf data mining and visualization software tools. Composition and execution of such services is carried out through an ontology-based workflow designer and scheduler, whereas services are discovered with the help of the ontologies. Finally, spectra are managed by a specialized database. MS-Analyzer comprises the following components (see Figure 3).

4.1.1. Ontology-based workflow designer and scheduler

This allows the design of a proteomics application as a workflow of services selected by searching the MS-Analyzer ontologies. It produces an abstract graphical workflow schema that is translated to a workflow language and then scheduled using a proper workflow scheduler. The ontology-based workflow designer and scheduler is used by the final user on a Grid node to compose applications and to submit the related workflow to the Grid middleware. It comprises the following components.

- The ontology-based assistant suggests to the user available tools for a given bioinformatics problem through a concept-based search of software and data components. It uses the services offered by the Ontology-based Service Discovery system described in the following, for example the browsing of ontologies.
- The workflow editor, used with the ontology-based assistant, allows the user to specify and design applications as workflows. It also provides visualization facilities for loading and browsing existing workflow schemas. Currently, workflows are designed by using a notation based on Unified Modelling Language (UML) and are translated into a subset of Business Process Execution Language for Web Services (BPEL4WS) [22].

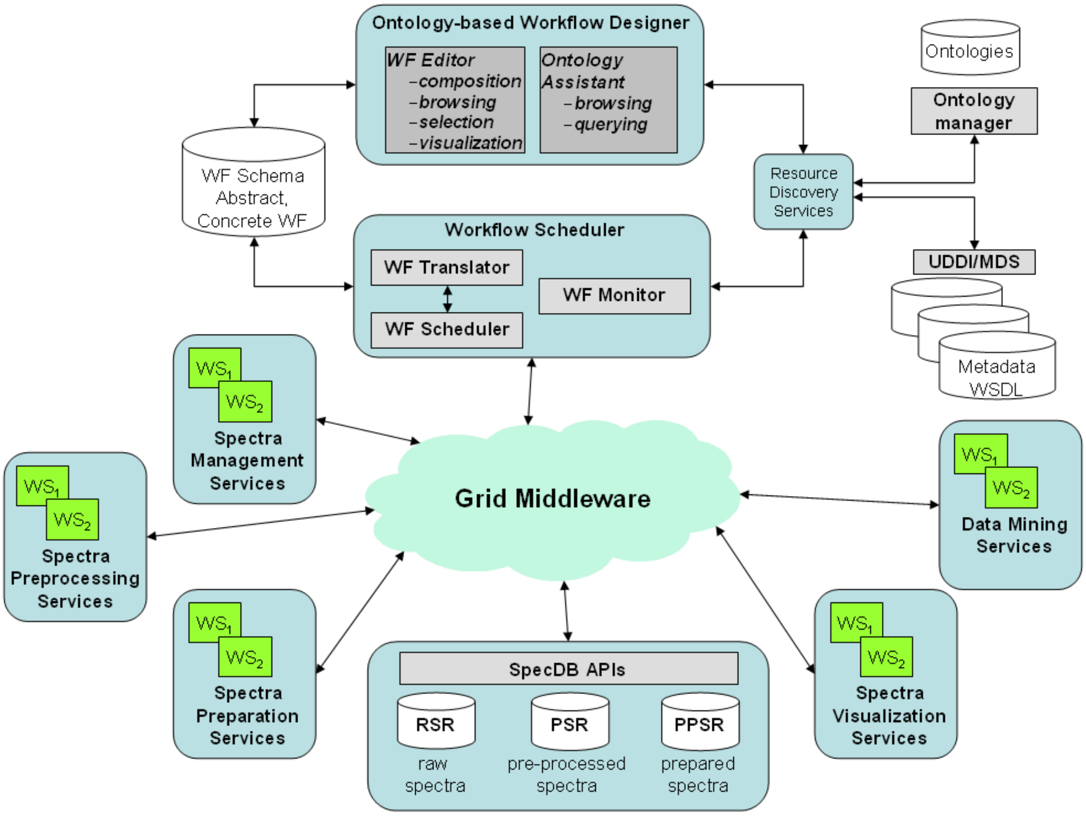


Figure 3. Architecture of MS-Analyzer.

- The workflow scheduler schedules and controls the execution of activities on the Grid. The scheduling depends on available Grid nodes and takes into account any execution constraints (e.g. a data source cannot be moved). To optimize application execution or to satisfy constraints, the workflow scheduler may move data calling Grid functions.
- The workflow metadata repository contains all of the information on the workflow schema. Information is organized such that static aspects (workflow schema metadata), such as data about control flow, are separated from application details (workflow application metadata), such as data used to perform a task.

4.1.2. *Metadata and ontology management*

In MS-Analyzer, the semantics of software and data is organized in a hierarchical schema. At the top layer (ontology repository), ontologies model bioinformatics services and applications, whereas

DOI: 10.1002/cpe

at the bottom layer (metadata repository), metadata provide specific information about available (i.e. installed) bioinformatics services and data sources. Users are guided by the top layer to choose available services or applications (ontology-based application design). The bottom layer allows the scheduler to access services and databases, providing information such as installed version, format of input and output data, parameters, etc. Metadata about services are provided by Universal Description, Discovery, and Integration (UDDI) and Web Service Description Language (WSDL) data that contain references to ontology data. Layers are updated whenever new software tools or data sources are added to the system, or new applications are developed.

4.1.3. Services and application library

This is a virtual repository containing the software tools, databases, and user-defined applications useful to manage and analyze MS data. Preprocessing [18] and Weka [23] tools are currently implemented as Web services and will be migrated towards Globus Toolkit 4 Grid services [3]. The services offered by MS-Analyzer and the SpecDB spectra database are described in the following.

4.2. Services

MS-Analyzer provides the following services that can be available wherever in the network.

(1) *Spectra management services* implement different spectra management functions and support the different stages of spectra. They allow loading of raw spectra produced by different types of mass spectrometer (e.g. MALDI–TOF, LC–MS/MS) and the storing of these in a raw spectra repository (RSR). Moreover, they support access to the spectra in their different stages (raw, preprocessed, prepared) when spectra need to be analyzed: data movement is based on the Grid data transfer functions. An important feature regards the ingestion of spectra datasets produced in remote proteomics facilities, as well as the spectra format conversion to a common format as mzData [24].

(2) *Spectra preprocessing services* load raw spectra (directly from the mass spectrometer or from the RSR), apply the preprocessing techniques described before, and store data into a preprocessed spectra repository (PSR). Preprocessing can be applied to one spectrum or contemporarily to many spectra.

(3) *Spectra preparation services* load preprocessed spectra and prepare them to be given as input to different types of data mining tools. For example, Weka [23] requires spectra to be organized in a unique file called an attribute-relation file format (ARFF). Data ready to be mined are stored in the prepared preprocessed spectra repository (PPSR), for further analysis.

(4) *Data mining services* implement common data mining tasks (e.g. classification, clustering, pattern analysis). Following a trend common to some recent projects [25,26], the data mining tools provided by Weka [23] are wrapped as Web/Grid services.

(5) *Data visualization services* allow the user to visualize raw and preprocessed spectra, as well as the knowledge models produced by data mining analysis. Visualization can be useful to compare different spectra belonging either to different samples (e.g. healthy and diseased patients), or to the same sample, but taken at different times, as happens in LC-MS/MS.

(6) *Data Access services* allow the user to wrap relevant biological data sources such as PDB [19] and SwissProt [20], as Web services. We currently have not yet implemented any data

source service. We plan to use publicly available databases whenever they are released as Web services. For instance, SRS [27] is a search engine for biological data sources that will be made available as a Web service.

4.3. Spectra database

The SpecDB spectra database is the data layer of MS-Analyzer. It implements basic spectra management functions and stores spectra in their different stages (raw, preprocessed, prepared) keeping trace of the different phases of proteomics experiments. To deal with the huge volumes of mass spectra data, which could not be analyzed just in the main memory, and to allow easy and efficient access to a single spectrum, to multiple spectra, and to relevant portions of spectra (e.g. to select all peaks in the range 0–500 Dalton (Da) for further data mining analysis), a hybrid XML-relational database for spectra data has been developed. The SpecDB database implements the spectra repositories described so far by using a relational data model to store spectra values (couples), and the mzData XML-based data model [24], to store metadata information about proteomics experiments. Moreover, mzData comprises also a compressed representation of spectra data.

The main requirements in designing SpecDB were the following: (i) supporting efficient storing and retrieval of data (single spectrum, set of spectra and portions of spectra); (ii) supporting import/export functions (e.g. loading of raw spectra available in different text files, exporting of spectra in mzData format); (iii) offering query/update functions able to enhance performance of data preprocessing and analysis (e.g. avoiding full main memory processing). Such functions are offered through a set of specialized APIs, for instance range queries can be used to implement basic preprocessing steps such as aggregation of peaks.

SpecDB is organized in such a way that each spectrum is stored in a set of relations, each containing portions of the spectrum according to a predefined window of mass/charge values, while a main relation maintains the organization of the single spectrum, improving data management. Spectra in the database are loaded from the raw format and are divided according to m/z windows. Such subdivision allows the optimization of data organization especially for spectra treatment. The idea is to map the multivalued attribute couple into a set of relational tables which maintain a trace of the original spectrum and divide data peaks depending on predefined m/z window ranges. Thus, a spectrum with values of m/z contained in the range 0–20 000 Da is divided into a set of relations such that intensities of peaks that belong in the range 0–500 Da are stored in a first relation, and peaks in the range 500–1000 Da are stored in a different relation, and so on. Figure 4 represents the logical data organization in a relational model. The main entity MassSpectrum represents the set of spectra, each represented by a unique identifier, information on date, and experimental details (laboratory, operator, etc.).

4.4. The role of the ontologies

Ontologies model the key domains of interest: data mining and MS-based proteomics.

4.4.1. Data mining ontology

WekaOntology models concepts and relations of the data mining domain. In particular, its instances represent the features of the data mining tools of the Weka suite [23]. The categorization of the data mining software has been made on the basis of the following classification parameters: (i) the data mining task performed by the software; (ii) the methodology used by the software in the data

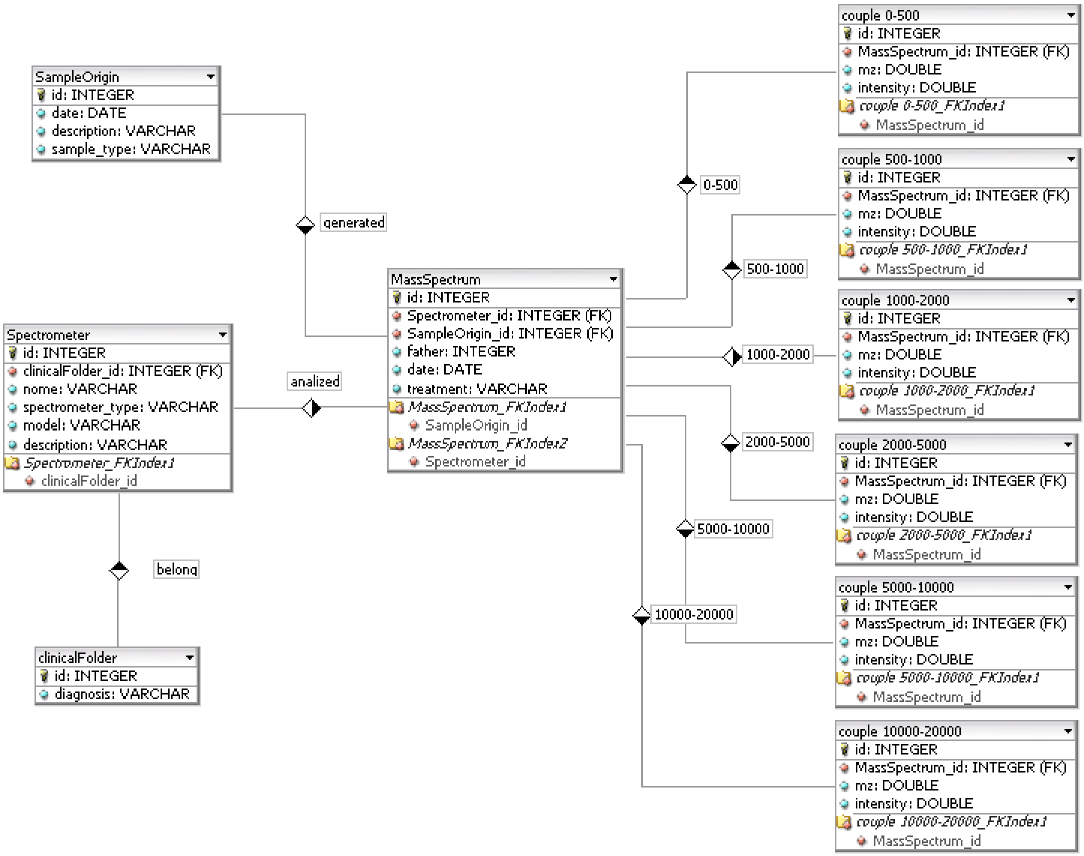


Figure 4. SpecDB logical data organization.

mining process; (iii) the type of data source the software works on; (iv) the degree of required interaction with the user. By browsing WekaOntology a researcher can choose the most suitable data mining technique for the current problem. Moreover, having specified a type of data to analyze, a user can browse the ontology taxonomy and find the preprocessing phase, the algorithms, and their software implementations. In summary, such ontology allows the semantic search of data mining software and other data mining resources, and suggests to the user the methods and software to use on the basis of stored knowledge and problem requirements.

4.4.2. Proteomics ontology

ProtOntology (proteomics ontology) models concepts, methods, algorithms, tools, and databases relevant to the proteomics domain. It comprises some main classes for the general proteomics concepts and various specializations. A main aspect is the distinction between biological concepts (such as proteins) and non-biological concepts (such as software tools).

ProtOntology biological concepts comprise three primary entities: aminoacid, protein, and structure. The structure concept is specialized into primary, secondary, tertiary, and quaternary, respecting the biological classification.

ProtOntology non-biological concepts comprise the following main classes. Analysis models the theoretic study of proteins. Introducing this concept is important because in medical research the literature citations of a method are the criteria of evaluation. This class comprises the following subclasses: mass-spectra analysis, interaction, primary structure analysis, secondary structure analysis, tertiary structure analysis, and quaternary structure analysis. Task describes concrete proteomics problems that a researcher needs to solve. Specializations of this concept are interpretation of MS data (e.g. methods used to identify a protein, or to recognize a disease), alignment (e.g. sequence alignment and structural alignment), and prediction (e.g. prediction of the secondary or tertiary structure of a protein starting with its primary sequence). Method is a way to perform a task. Software is an implementation of a method.

The main relations between the concepts of the previous classes are the following: is_Chain_of explains that a protein is a sequence of aminoacids, has_A links a protein to its own structures, studies links a particular analysis and a protein structure, and implements links a software with a method.

4.4.3. Ontology-based workflow design

Ontologies can help with the cooperation between the biological and the bioinformatics groups linking knowledge about experimental research (e.g. wet lab) and bioinformatics applications. Using the ontology-based workflow editor of MS-Analyzer, the design and execution of an application is conducted using the following steps.

(1) *Ontology-based component selection.* Browsing the ontologies, a user can first visualize the main tasks of a proteomics analysis, and then can select the dataset to be analyzed, the proper preprocessing techniques, the type of data mining task, and related software tools.
(2) *Workflow design.* Selected components are combined producing a graphic workflow schema that can be translated into a standard language such as BPEL4WS [22].
(3) *Application execution on the Grid.* The workflow is scheduled by a workflow scheduler on the Grid. In particular, the MS-Analyzer scheduler takes care of data movement and communication between services. In turn, such functions make use of Grid middleware services.
(4) *Results visualization and storing*. After application execution, the user can visualize results and eventually enrich and extend the ontologies with knowledge about application execution.

5. DISCOVERY OF SERVICES IN MS-ANALYZER

Discovering a service that fits user requirements can be a challenge in a distributed environment. Currently existing directory services, such as UDDI [21] or MDS [3], support a simple key-based discovery based on a matchmaking of a set of attribute types describing the service itself. This model implies that services that have the same functionalities will not be simply reachable together if their

DOI: 10.1002/cpe

```
<tmodel tmodelKey="string" operator="string"
  authorizedName="string">
    <name> tModel name </name>
    <description> ... </description> *
    <overviewDoc> ... </overviewDoc>?
    <identifierBag> ... </identifierBag> ?
    <categoryBag> ... <categoryBag> ?
</tModel>
```

Figure 5. A general tModel schema.

keywords are different. In a distributed scenario where resources are dynamic and descriptions can be different for the same type of service, there is a need for a standard service description and a semantic-based service discovery [28]. The proposed Ontology-based Service Discovery approach aims to use ontologies to find semantic properties of services and to use basic middleware information systems such as UDDI or MDS to find more specific details about services and resources.

5.1. UDDI: service discovery on the Semantic Web

UDDI [21] is an industrial initiative whose purpose is to create an Internet-wide network of registries containing Web services. Information contained in UDDI is stored in four data structures: (i) businessEntity containing information about the business owner of the published service; (ii) businessServices containing descriptive and technical information; (iii) bindingTemplate that supplies both technical information about the entry point of a service and references to a tModel; (iv) tModel (see Figure 5) related to specifications of service taxonomies. These four data structures carry all information expressible by UDDI.

BindingTemplate carries technical details (entry point, parameters, and specification) about stored services in a tModelInstanceInfo structure with reference to a tModel and other descriptive parameters. tModelInstanceInfo constitutes a type of digital fingerprint of the service because it contains a number of references to specific descriptions (tModelKey). Then a search agent can utilize these keys to select services containing references to a certain specific tModel. Once a specification has been established and registered as a tModel, developers can determine how a service reflects that specification including simply the relative key in its bindingTemplate field.

Although UDDI allows a wide range of searches (services can be searched by name, location, business, bindings, or tModels), the search mechanism is limited to keyword matches and UDDI does not support any inference based on the taxonomies referred to by the tModels. Moreover, a search by category is the only way to search for services; however, the search may produce many results which may be of no interest.

5.2. MDS: service discovery on the Grid

In the Globus Grid middleware the resource discovery is provided by the Metacomputing Directory Service (MDS) [3]. This is based on a central repository storing all of the information and on an homogeneous mechanism for discovering different types of information. MDS uses the Lightweight Directory Access Protocol (LDAP), a protocol that implements a standardized way to access directory data. MDS comprises two components: Grid Resource Information Service (GRIS) and Grid Index Information Service (GIIS). The former provides the resource description, using the Web Service Resource Framework (WSRF) resource property model, and has a modular structure. Each set of information has a simple structure: a set of entries that have a set of attribute–value pairs. The latter provides an aggregate directory by hierarchically grouping of resources by using LDAP. Moreover, MDS implements two protocols: the Grid Resource Information Protocol (GRIP) used to access information about resources, and the Grid Resource Registration Protocol (GRRP) used to notify aggregate directory services on how to use this information. In such a way an aggregate directory service uses GRIP and GRRP to obtain information from a set of information providers and to reply to requests related to same resources.

Also, MDS does not yet provide semantic capabilities for the description and searching of services.

5.3. Ontology-based Service Discovery

The approach proposed here is based on a combination and cooperation between ontologies and current standard discovery services, such as UDDI and MDS (referred to as middleware in the rest of this section). An ontology layer stores the semantic properties of resources whereas the UDDI/MDS layers store information to reach and use a particular service. Links between ontology and UDDI/MDS are provided by: arranging a mapping between the name of a UDDI class and the name of an ontology resource (e.g. the J48 class name must have an equivalent entry in UDDI); defining a tModel inside UDDI (or a new LDAP entry inside GRIS) containing the relevant semantic properties of ontology instances, for example serialized in a string.

For example, let us suppose that the ontology fragment for J48 indicates that it implements a classification task, employs the decision tree method, receives_in_input an ARFF file, and produces_as_output a decision tree knowledge model. Now, at the middleware layer, each installed instance of the J48 service will be characterized by its WSDL file (stored locally where the services run) that is referred to by an entry in UDDI/MDS. The description of J48 inside UDDI/MDS will contain, in addition to the links to the various WSDL, a tModel containing a set of concept–value pairs extracted by the ontology instance of J48. In the example, such a set is (task = classification, method = decision tree, receives_in_input = ARFF, produces_as_output = decision tree). In general, all taxonomical and non-taxonomical relations that involve a class of the ontology are derived and stored in the tModel of a certain class instance.

From an architectural point of view, the glue between the ontology and the UDDI/MDS layers is implemented by a service called Ontology-based Service Discoverer (see Figure 6). The ontology layer is inspected by using an ontology manager that supports ontology browsing (through visual user interfaces) and querying (through ontology query languages such as RDQL [29], or through reasoning systems).

In the proposed approach, all information useful to design an application can be reached starting from the ontology; information used to schedule applications is provided by the UDDI/MDS middleware. The ontology describes the semantic properties and relations of resources whether they are available or not. Ontology classes classify resources; instances of classes describe the resources itself. In this way domain ontologies can completely describe the semantic properties of resources of

DOI: 10.1002/cpe

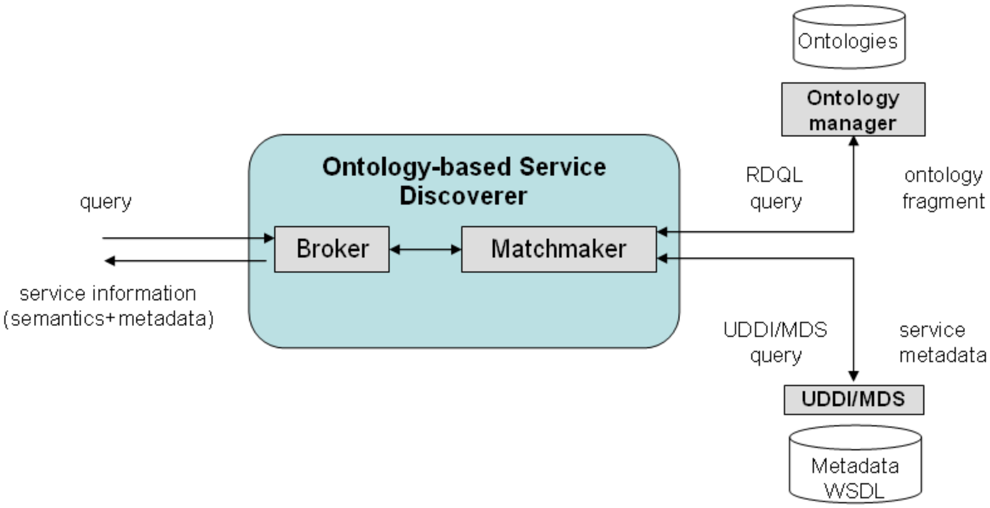


Figure 6. Architecture of the Ontology-based Service Discoverer.

a domain (e.g. software tools). Finding a resource that satisfies some properties (e.g. a data mining software that implements the classification task and uses the decision tree method) can be done at the ontology layer. Using a resource may require access to middleware that provides metadata.

The Ontology-based Service Discoverer service is composed of the following two subsystems.

(1) The *broker*, which receives queries expressed through an ontology query language or by a reasoner, and allows the discovery of services (their semantic properties) by forwarding the query to the matchmaker.
(2) The *matchmaker*, which retrieves in the ontology the description of a resource by redirecting (and eventually reformulating) the query to the ontology layer. Depending on the request, the matchmaker can only return the semantic description of returned instances or may, in turn, use such information to query the UDDI/MDS. In this case both semantics information (taxonomy and properties) and invocation information (e.g. reference to WDSL) are returned to the user.

For example, a user could query the ProtOntology for retrieving all binning software, by sending a query or by browsing taxonomies. The matchmaker retrieves ontology classes representing software, then queries the UDDI registry and retrieves the links to WSDL files. Then it returns a result composed of (i) information about invocation of services, provided by the middleware, and (ii) semantic information provided by the ontology.

Compared to the UDDI and MDS models, the proposed approach has the main advantage of allowing the concept-based discovery of services useful in many cases, such as workflow composition. Moreover, knowledge coded in the ontology is useful to guide the workflow composition by imposing the pre/post-condition for each tool. Another approach to add semantic information to UDDI is

DOI: 10.1002/cpe

described in [30]. It provides a semantic algorithm to match the inputs and outputs of Web service requests using service ontology. Compared to this, the approach proposed here avoids overloading UDDI with a large amount of ontological information. While in [30] all fields of each OWL file are mapped to a specialized field of UDDI (tModel), our approach stores in the UDDI only a small amount of ontology information, avoiding the overload of updating ontology when a modification of UDDI occurs. Naumenko *et al.* [31] presented an evaluation of UDDI capabilities to store semantic descriptions enabling semantic discovery. Moreover, they also presented an approach to mapping RDFS upper concepts to a UDDI data model using a tModel structure.

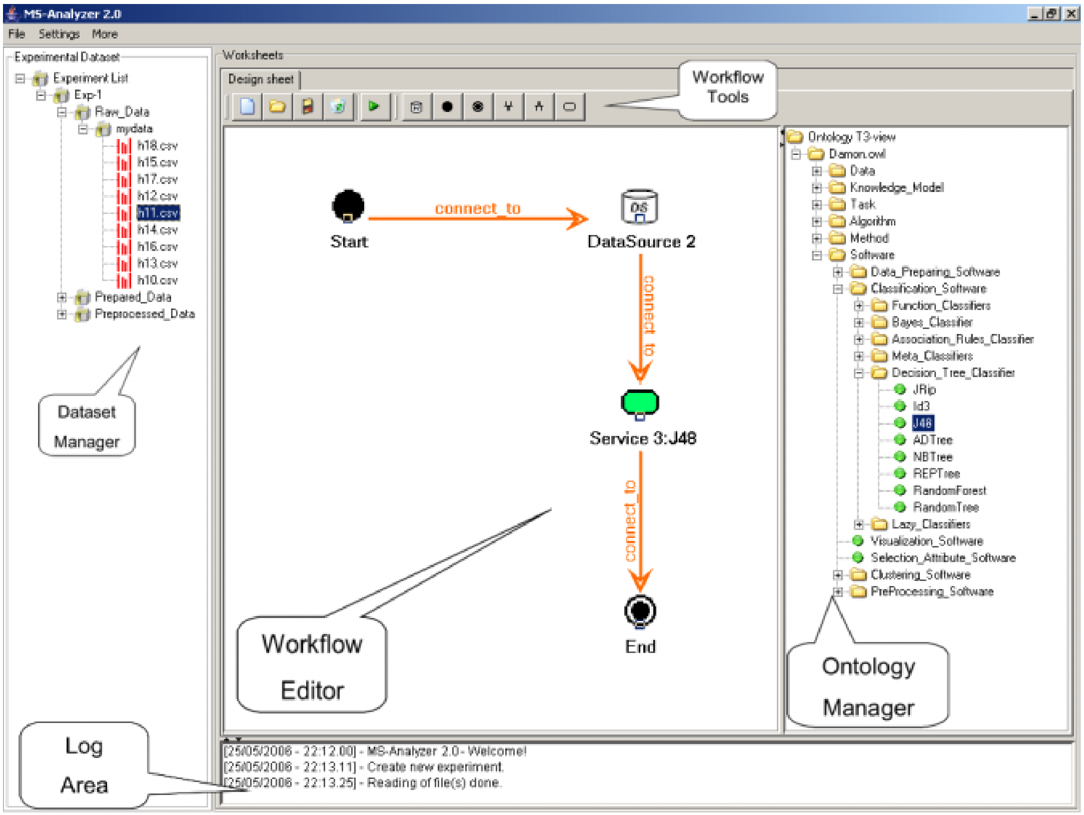
6. SYSTEM PROTOTYPE

A working prototype of MS-Analyzer has recently been implemented. All of the preprocessing algorithms described in Section 3 have been implemented as Web services using Java technology and they are deployed through the Apache Axis Web services implementation. Moreover, both WekaOntoloy and ProtOntology have been implemented using the OWL Web ontology language [32]. The SpecDB spectra database has been fully implemented on top of an open-source database management system. The data mining services are provided by the Weka data mining suite [23]. While the ontology-based workflow editor has been fully implemented, scheduling activities are currently managed by a rough internal scheduler that analyzes the workflow schema and activates the services. We plan to use the Karajan [33] scheduler provided by the CoG Toolkit available on Globus [34].

The graphical user interface of MS-Analyzer is shown in Figure 7. It comprises the dataset manager and the ontology-based workflow editor. The former manages the experimental data modeled through the dataset concept, i.e. a view over the SpecDB database containing the raw, preprocessed, and prepared spectra of an experiment. The latter, after loading the described ontologies, allows the user to browse and search the bioinformatics tools, and to select and compose workflows of services through drag&drop. The workflow is designed by using a UML-based notation, providing basic control blocks such as fork/join, etc. Elements used to compose the workflow can be either datasets selected through the dataset manager or services selected through the ontologies. Constraints expressed by the ontologies (e.g. the type of data to be given as input to a service) are enforced at composition time. For instance, all data mining services require an ARFF file as input, which can be obtained from a (preprocessed) dataset through a proper transformation. The produced abstract workflow schema is translated into a schedulable concrete workflow using the BPEL4WF workflow language [35], which in turn is used by a Grid workflow engine.

7. PERFORMANCE EVALUATION

The goal of the following study is mainly to show how different choices in preprocessing and data mining strategies can affect the results of a data analysis task. MS-Analyzer allows the user to easily design, execute, and then evaluate the effectiveness and performance of a data mining experiment varying preprocessing and mining strategies. The following performance evaluation has been conducted measuring the performance of the steps of a MS-based data mining experiment



```
[25/05/2006 - 22:12.00] - MS-Analyzer 2.0 - Welcome!
[25/05/2006 - 22:13.11] - Create new experiment.
[25/05/2006 - 22:13.25] - Reading of file(s) done.
```

Figure 7. MS-Analyzer graphical user interface.

that comprises the following phases: (i) loading of the raw spectra; (ii) preprocessing of the raw spectra; (iii) preparation of the data mining input file (e.g. the Weka ARFF file); and (iv) data mining analysis (e.g. classification) of prepared spectra. The impact of using different preprocessing algorithms on execution times, memory occupancy, and, most importantly, quality of classification has been evaluated.

Three different mass spectra datasets publicly available on the Internet have been considered. The pancreatic cancer dataset contains 142 spectra, each with 6772 (intensity, m/z) measurements, partitioned into two classes: healthy and diseased patients [36]. The prostate cancer dataset contains 322 spectra, each with 15 154 measurements, partitioned into four classes: no disease, benign cancer, pc410, and pcg10 patients [36]. The ovarian cancer dataset contains 49 spectra, each with 59 386 measurements, partitioned into two classes: control and disease [37]. Each dataset has been classified by using Weka data mining tools and the following performance data have been measured: (i) execution times of, respectively, loading, preprocessing, preparation, and classification activities; (ii) memory

Table I. Execution times versus preprocessing strategy (times in milliseconds).

Dataset		No preprocessing	Normalization	Binning	Binning + Align
Pancreatic	Loading	10 758	10 758	10 758	10 758
	Preprocessing	0	67	47	520
	Preparation	3385	3385	418	421
	Mining	300	297	7	7
	Total execution time	14 443	14 510	11 229	11 706
Prostate	Loading	26 122	26 122	26 122	26 122
	Preprocessing	0	384	221	2480
	Preparation	7438	7438	763	759
	Mining	561	551	14	13
	Total execution time	33 860	34 244	27 119	29 374
Ovarian	Loading	21 831	21 831	21 831	21 831
	Preprocessing	0	201	120	1456
	Preparation	5432	5432	599	607
	Mining	720	722	120	122
	Total execution time	27 983	28 184	22 670	24 106

occupancy of, respectively, raw spectra, preprocessed spectra, and ARFF files; (iii) quality indexes of classification, under different preprocessing conditions.

7.1. Execution times

For each experiment, Table I reports the following execution times, expressed in milliseconds (ms): (i) spectra loading time, i.e. the time needed to load the entire dataset into the main memory; (ii) preprocessing execution time considering different preprocessing strategies, i.e. no preprocessing, normalization, binning, or binning plus alignment; (iii) preparation execution time, i.e. the time needed to generate and write the Weka ARFF file (it should be noted that only one ARFF file is generated for a dataset comprising a set of spectra); (iv) classification execution times, i.e. the execution times of the C4.5 classification method offered by Weka (i.e. J48), comprising the loading of the ARFF file and the generation of the classification decision tree. The reference time is the experiment without applying any preprocessing. It is possible to note that normalization alone increases the overall execution time, whereas binning and binning plus alignment, which sensibly reduce the size of ARFF data, also decrease the classification time and the overall experiment time, even if a preprocessing time has to be included.

7.2. Secondary memory occupancy

For each experiment (see Table II), the size of the initial raw spectra, the spectra after preprocessing, if any, and the corresponding ARFF files have been measured. Spectra sizes are expressed in kilobytes.

DOI: 10.1002/cpe

Table II. Spectra and ARFF sizes versus preprocessing strategy (sizes in kilobytes).

Dataset		No preprocessing	Normalization	Binning	Binning + Align
Pancreatic	Spectra sizes	15 393	15 393	151	244
	ARFF sizes	7393	7393	77	142
Prostate	Spectra sizes	104 311	104 311	1002	2200
	ARFF sizes	52 311	52 311	502	1100
Ovarian	Spectra sizes	48 505	48 505	524	998
	ARFF sizes	24 505	24 505	255	492

Table III. Classification indices versus preprocessing strategy.

	No preprocessing			Normalization			Binning			Binning + Align		
	TP	Prec	Recall	TP	Prec	Recall	TP	Prec	Recall	TP	Prec	Recall
Pancreatic Cl1	0.513	0.5	0.513	0.513	0.5	0.513	0.536	0.578	0.536	0.613	0.563	0.613
Pancreatic Cl2	0.488	0.5	0.488	0.488	0.5	0.488	0.614	0.573	0.614	0.525	0.575	0.525
Prostate Cl1	0.774	0.786	0.774	0.774	0.786	0.774	0.847	0.885	0.847	0.937	0.922	0.937
Prostate Cl2	0.794	0.704	0.794	0.794	0.704	0.794	0.825	0.813	0.825	0.937	0.967	0.937
Prostate Cl3	0.269	0.269	0.269	0.269	0.269	0.269	0.577	0.405	0.577	0.577	0.556	0.577
Prostate Cl4	0.395	0.447	0.395	0.395	0.447	0.395	0.558	0.615	0.558	0.698	0.732	0.698
Ovarian Cl1	0.84	0.913	0.84	0.84	0.913	0.84	0.849	0.913	0.845	0.96	0.96	0.96
Ovarian Cl2	0.917	0.846	0.917	0.917	0.846	0.917	0.934	0.944	0.917	0.958	0.958	0.958

It is possible to note that binning and binning plus alignment sensibly reduce spectra and ARFF sizes by two orders of magnitude. This has a great impact on preparation and classification executions that have to deal with fewer data, as shown in Table I.

7.3. Quality of classification

For each classification experiment (see Table III), and for each class found, the following performance indices obtained by using tenfold cross validation have been measured: (i) true positive (TP) rate; (ii) precision (often called positive predictive value or specificity); (iii) recall (often called sensitivity). Precision is the ratio of correctly classified positives with respect to all predicted positives, whereas recall is the ratio of correctly classified positives with respect to all real positives. In Table III it is possible to note that classification indices are improved when binning or binning plus alignment preprocessing are used; this means that preprocessing really eliminates noise and allows a more effective classification.

DOI: 10.1002/cpe

In summary, the capability of MS-Analyzer to easily design, execute, and evaluate different workflows of the same experiment, gives information to the scientist about the best strategy to use. In previous experiments it has been shown that the choice of the right preprocessing techniques may improve execution times, memory occupancy, and quality of data mining.

8. CONCLUSION

Considering the complexity and the heterogeneity of the various technological platforms involved in MS proteomics, we have proposed MS-Analyzer, a system for the management, preprocessing, and analysis of MS proteomics data on the Grid. A specialized spectra database is used to store, manipulate, and efficiently access spectra datasets, while a novel approach combining ontologies and Semantic Web/Grid information services has been proposed to discover services. By using MS-Analyzer a user can produce, in a short time, different workflows of the same application by considering different combinations of preprocessing, preparation, and data mining techniques, with the help and the constraint checks provided by WekaOntology and ProtOntology, and without worrying about preparation of the proper input files for data mining tools. The performance of the enacted workflows can be easily visualized, allowing a comparison of the effect of different preprocessing and data mining techniques and an evaluation of the best strategies to analyze mass spectra data.

ACKNOWLEDGEMENTS

This work has been partially supported by project GRID.IT funded by MIUR, grant no. RBNE01KNFP. We are grateful to P. H. Guzzi and T. Mazza for their contributions to the MS-Analyzer project, to Marco Gaspari for providing the spectra data, and to G. Tradigo for his contribution on SpecDB.

REFERENCES

1. Aebersold R, Mann M. Mass spectrometry-based proteomics. *Nature* 2003; **422**:198–207.
2. Berman F, Hey AJG, Fox G. *Grid Computing: Making The Global Infrastructure a Reality*. Wiley: New York, 2003.
3. Foster I, Kesselman C. Globus toolkit version 4: Software for service-oriented systems. *Proceedings of the IFIP International Conference on Network and Parallel Computing* (*Lecture Notes in Computer Science*, vol. 3779). Springer: Berlin, 2005; 2–13.
4. Gruber TR. A translation approach to portable ontologies. *Knowledge Acquisition* 1993; **5**(2):199–220.
5. Yu J, Buyya R. A taxonomy of scientific workflow systems for Grid computing. *SIGMOD Record* 2005; **34**(3):44–49.
6. Cannataro M, Congiusta A, Pugliese A, Talia D, Trunfio P. Distributed data mining on Grids: Services, tools, and applications. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics* 2004; **34**(6):2451–2465.
7. Wong JWH, Cagney G, Cartwright HM. Specalign—processing and alignment of mass spectra datasets. *Bioinformatics* 2005; **21**(9):2088–2090.
8. The Sashimi Project. http://sashimi.sourceforge.net/ [12 October 2006].
9. Jeffries N. Algorithms for alignment of mass spectrometry proteomic data. *Bioinformatics* 2005; **21**(14):3066–3073.
10. Goodman N, Rozen S, Stein LD, Smith A. The LabBase system for data management in large scale biology research laboratories. *Bioinformatics* 1998; **14**(7):562–574.
11. Laud A, Bhowmick S, Cruz P, Singh D, Rajesh G. The GRNA: A highly programmable infrastructure for prototyping, developing, and deploying genomics-centric applications. *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB 2002)*, Hong Kong, China, 20–23 August 2002. Morgan Kaufman: San Francisco, CA, 2002.

DOI: 10.1002/cpe

12. Shah SP, He DYM, Sawkins JN, Druce JC, Quon G, Lett D, Zheng GXY, Xu T, Ouellette BFF. Pegasys: Software for executing and integrating analyses of biological sequences. *BMC Bioinformatics* 2004; **5**(40). DOI: 10.1186/1471-2105-5-40.
13. Stevens RD, Robinson AJ, Goble CA. myGrid: Personalised bioinformatics on the information Grid. *Bioinformatics* 2004; **19**(1):302–302.
14. Oinn TM, Addis M, Ferris J, Marvin D, Greenwood RM, Carver T, Pocock MR, Wipat A, Li P. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 2004; **20**(17):3045–3054.
15. Rowe A, Kalaitzopoulos D, Osmond M, Ghanem M, Guo Y. The Discovery Net system for high throughput bioinformatics. *Bioinformatics* 2003; **19**(Suppl. 1):i225–i231.
16. Zhuge H. China's E-Science Knowledge Grid Environment. *IEEE Intelligent Systems* 2004; **19**(1):13–17.
17. AppliedBiosystems. http://www.appliedbiosystems.com [12 October 2006].
18. Cannataro M, Guzzi P, Mazza T, Tradigo G, Veltri P. Preprocessing of mass spectrometry proteomics data on the Grid. *Proceedings of the 18th IEEE International Symposium on Computer-Based Medical Systems (CBMS'05)*, Trinity College, Dublin, Ireland, 23–24 June 2005. IEEE Press: Piscataway, NJ, 2005; 549–554.
19. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The protein data bank. *Nucleic Acids Research* 2000; **1**(28):235–242.
20. Boechman B *et al.* The Swiss-Prot knowledgebase and its supplement TREMBL in 2003. *Nucleic Acid Research* 2003; **31**:365–370.
21. UDDI Organization. *Technical White Paper*, UDDI, 2000.
22. IBM. Business Process Execution Language for Web Services—BPEL4WS. http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/ [12 October 2006].
23. Witten IH, Frank E. *Data Mining: Practical Machine Learning Tools and Techniques* (2nd edn). Morgan Kaufmann: San Francisco, CA, 2005.
24. Orchard S, Hermjakob H, Apweiler R. The proteomics standards initiative. *Proteomics* 2003; **3**(7):1374–1376.
25. Perez MS, Sanchez A, Herrero P, Robles V, Pena JM. Adapting the Weka data mining toolkit to a Grid based environment. *Proceedings of the 3rd International Atlantic Web Intelligence Conference (AWIC 2005)*, Lodz, Poland, 6–9 June 2005. Springer: Berlin, 2005; 492–497.
26. Talia D, Trunfio P, Verta O. Weka4ws: A WSRF-enabled Weka toolkit for distributed data mining on Grids. *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2005)*, Porto, Portugal, October 2005 (*Lecture Notes in Artificial Intelligence*, vol. 3721). Springer: Berlin, 2005.
27. LION Bioscience AG. SRS search data bank system. http://srs.ebi.ac.uk/ and http://www.lionbioscience.com/ [12 October 2006].
28. Sivashanmugam K, Verma K, Sheth A, Miller J. Adding semantics to Web services standards. *Proceedings of the 1st International Conference on Web Services (ICWS'03)*, Las Vegas, NV, 23–26 June 2003 (*Lecture Notes in Computer Science*, vol. 2853). Springer: Berlin, 2003.
29. Miller L, Seaborne A, Reggiori A. Three implementations of SquishQL, a simple RDF query language. *Proceedings of the International Semantic Web Conference (ISWC2002)*, Sardinia, Italy, 9–12 June 2002 (*Lecture Notes in Computer Science*, vol. 2342). Springer: Berlin, 2002.
30. Paolucci M, Kawamura T, Payne TR, Sycara KP. Importing the Semantic Web in UDDI. *CAiSE'02/WES'02: Revised Papers from the International Workshop on Web Services, E-Business, and the Semantic Web*, London, U.K., 2002. Springer: Berlin, 2002; 225–236.
31. Naumenko A, Nikitin S, Terziyan V, Veijalainen J. Using UDDI for publishing metadata of the Semantic Web. *Proceedings of the 1st International IFIP/WG 12.5 Working Conference on Industrial Applications of the Semantic Web (IASW-2005)*, Jyväskylä, Finland, 25–27 August 2005 (*IFIP International Federation for Information Processing Series*, vol. 188). Springer: Berlin, 2005.
32. W3C. OWL Web Ontology Language Reference. http://www.w3.org/TR/owl-ref/ [12 October 2006].
33. Hategan M, von Laszewski G, Amin K. Karajan: A Grid orchestration framework. *Proceedings of Supercomputing 2004*, Pittsburgh, PA, 6–12 November 2004.
34. von Laszewski G, Hategan M. Workflow concepts of the Java CoG kit. *Journal of Grid Computing* 2006; **3**(3–4):239–258.
35. Juric MB, Sarang P, Mathew B. *Business Process Execution Language for Web Services*. Packt Publishing: Birmingham, U.K., 2004.
36. Clinical Proteomics Programs National Cancer Institute, Center for Cancer Research. http://home.ccr.cancer.gov/ncifdaproteomics/ppatterns.asp [12 October 2006].
37. Tibshirani R. PPC: Peak probability contrasts home page, Stanford University, Stanford, CA. http://www-stat.stanford.edu/∼tibs/PPC/index.html [12 October 2006].