Scalable Biomedical and Bioinformatics Applications

(Invited Paper)

Mario Cannataro*
Bioinformatics Laboratory,
University Magna Græcia,
88100 Catanzaro, Italy
cannataro@unicz.it

Pietro H. Guzzi
Bioinformatics Laboratory,
University Magna Græcia,
88100 Catanzaro, Italy
hguzzi@unicz.it

Giuseppe Tradigo
Bioinformatics Laboratory,
University Magna Græcia,
88100 Catanzaro, Italy
gtradigo@libero.it

Pierangelo Veltri
Bioinformatics Laboratory,
University Magna Græcia,
88100 Catanzaro, Italy
veltri@unicz.it

ABSTRACT

Biomedical and bioinformatics applications manage and process huge datasets and are characterized by complex workflows involving the application of different algorithms and tools. In the last years the application of high-throughput technological platforms, such mass spectrometry or medical imaging, and the combined use of different databases (e.g. disease-specific databases), is producing an ever increasing volume of data that need to be processed in an efficient way. Thus, the need for scalable solutions at the different layers of biomedical/bioinformatics applications arises. The paper discusses some emerging scalable solutions for relevant biomedical and bioinformatics applications.

Keywords

Biomedical Applications, Bioinformatics, Systems Biology

1. INTRODUCTION

Biomedical and bioinformatics applications manage and process huge datasets and are characterized by complex workflows involving the application of different algorithms and tools. In the last years the application of high-throughput technological platforms, such mass spectrometry or Computed tomography medical imaging, and the combined use of different databases (e.g. disease-specific databases containing images like mammography, or biological databases), is producing an ever increasing volume of data that need to be processed in an efficient way. Thus, the need for scalable solutions at the different layers of biomedical/bioinformatics applications arises.

*Corresponding author.

Infoscale2008, June 4-6, 2008, Vico Equense, Napoli, Italy.

Bioinformatics regards the application of advanced algorithms and computational platforms to solve problems in biology and medicine. Important tasks are the methods used for acquiring, storing, retrieving and analyzing biological data produced by technological platforms such as mass spectrometry, micro-array, computational tomography, magnetic resonance imaging, positron emission technology, etc.

Another important issue is the access and eventually the integration of different biological databases relevant for the specific problem. For instance, applications studying the sequence and structure of proteins usually access to protein sequence and structure databases (e.g. UniProt, PDB), while many biomedical applications access images databases.

Another common issues in the bioinformatics and biomedical fields is the distribution of data and users. For instance, many bioinformatics laboratories or health centers are more and more interested in sharing their data to improve quality of research and allow cross-validation, so analysis requires the remote and distributed collection of data and the sharing of results.

Moreover, emerging applications such as the large scale screening of population may be based on the distributed collection and analysis of biological samples.

Thus, the need for providing scalable solutions at the different layers of bioinformatics and biomedical applications arises. At the data layer, biological and biomedical databases need to support both an increasing number of queries, as well as complex data integration functions. Following a trend observed in commercial applications, also biological databases and query services starts to be provided in a parallel fashion (e.g. Mascot Server [18]). Moreover, many bioinformatics applications need to find different information about proteins that are stored in different databases. Other than using static links and naming conventions among databases, ontologies (e.g. GeneOntology [21]) and controlled taxonomies/vocabularies are more and more used for broad access to information.

At the computing layer, different applications can be modeled through workflows of basic tools (e.g. data filtering, data preprocessing, data classification and results interpretation) that are often implemented as web-based applications and more recently as web services. Different comput-

ing platforms facing the requirements of scientific workflows or e-science applications have emerged. They are often available on the Internet but high-performance solutions are more and more implemented on the Grid. In fact, issues such as transparent data replication and high-performance data transfer are not faced by current web applications but are solved in Grid middleware.

In summary, many bioinformatics/biomedical applications (i) are naturally distributed, due to the high number of involved data sets; (ii) require high computing power, due to the large size of data sets and the complexity of basic computations; (iii) may access heterogeneous data, where heterogeneity is in data format, access policy, distribution, etc.; (iv) and require a secure software infrastructure, because they could access private data owned by different organizations.

The rest of the paper describes some emerging scalable solutions for relevant biomedical and bioinformatics applications. Section 2 describes a software platform for the distributed collection and analysis of voice samples for screening of diseases. Section 3 introduces interactomics and describes a distributed protein complexes meta-prediction tool based on the integration of different predictors. Section 4 discusses a software platform for the design and distributed execution of computational proteomics applications modeled as workflows of services. Finally, Section 5 concludes the paper and outlines future trends.

2. DISTRIBUTED BIOMEDICAL APPLICATIONS

This section describes an example of a biomedical application that is a possible template for different medical fields. It involves the remote acquisition of biomedical data, their preprocessing, transmission, storage and analysis. Moreover, to accomplish comparative analysis, the extraction of key parameters from the samples and their coordinated analysis, e.g. by using data mining techniques, is often performed.

The application considered here is the distributed collection and analysis of voice samples. Voice is the result of the coordination of the whole pneumophonoarticulatory apparatus. Dysphonia is one of the major symptoms of benign laryngeal diseases, such as polyps or nodules, but it is often the first symptom of neoplastic diseases such as laryngeal cancer.

The analysis of the voice can allow the identification of some diseases of the vocal apparatus and usually is carried out from an expert doctor in a non automatic way. To overcome this a distributed web-based system for the remote collection and automatic analysis of vocal signals is presented. Vocal signals are submitted by the users through a simple web-interface or may be collected by a portable device and analyzed in real-time providing first-level information on possible voice alterations.

REVA (Remote Voice Analysis) is a web based system for the distributed collection and automatic analysis of vocal signals [3]. The system consists of a client module (a Java Applet loaded into the user's browser) where a user can register his/her voice by using some rules (e.g. the user is asked to register a vowel and provide information about sex and age, that are relevant to perform voice analysis), after a verification of some minimum hardware requirements on microphone quality.

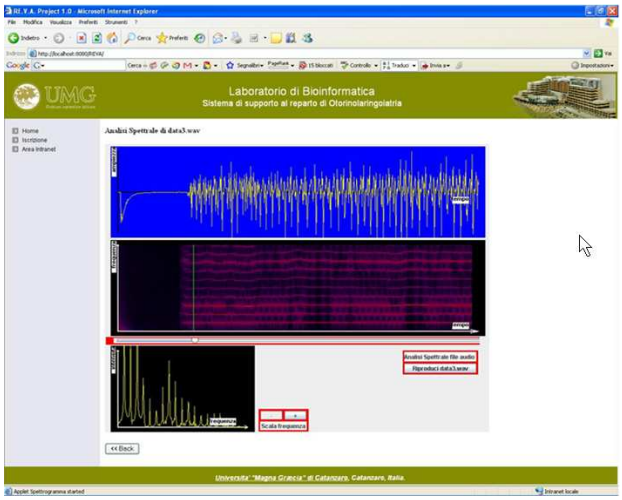

Figure 1: The REVA Interface

The voice signal, cleaned from noises, is sent through the Internet to a remote server which is in charge of storing and analyzing it. The server performs on the voice sample some signal processing activities to extract relevant signal parameters and then attempt to classify the sample as healthy or diseased. Then, it returns to the client the signal analysis results and the possible voice anomalies are related to potential diseases. Moreover, data are available to the doctors for further analysis.

The interaction between the system and the users happens through different phases.

In a first phase the subject/patient has to register to the system; such registration has to be extremely simplified, but at the same time has to allow the unambiguous identification of the user, through his or her personal data, and the acquisition of previous data (if existing) about his or her clinical status.

The second phase will consist of assisting the user during the registration of the voice signal, by using instruments for the audio acquisition available on the local computer (i.e. audio card and microphone of average quality). The voice signal, conveniently cleaned from possible external disturbances, will be sent to the remote server which will analyze it and will return to the client system the outcome of the elaboration. The rules that will allow to distinguish in an automatic way the healthy signal from a pathological one, are based on the analysis of voice samples through signal processing techniques [19]. The system is currently tested on data coming from adult patients provided by the University of Catanzaro Hospital.

From the web server side, the system will make use of the know-how of the medical components for a deeper analysis of the signal. In those cases for which the automatic analysis of the parameters returns an uncertain result, the software will be set so to commit the result evaluation to a human operator. As an example, Figure 1 shows the interface of REVA accessible to the doctors which can visualize and perform deep analysis of the voice samples that are classified in an uncertain way by the automatic procedure.

The system described so far adopt a quite classical centralized architecture, but different extensions are possible.

First of all, the acquired data are collected into a local database to allow a further analysis via classification and clustering procedures, by using data mining methodologies. Such analysis will allow to verify the correctness of the screening rules of the vocal signals, and to identify possible further indicators of pathologies to be submitted to the attention of the medical components. In fact, performing measures on a large control group, will allow the definition of objective "normality" parameters. Any measure outside such values, will allow defining pathological vocal folds vibration and level of dysphony. As a consequence, precise indexes could be defined, to classify different nosological elements, and evaluate any modification due to logopaedic and/or surgical treatment within different pathologies.

Second, the availability of a database will allow the controlled sharing of voice samples, to perform multi-centric studies, or to compare, by analysing the voice quality, the follow-up of different interventions. The storage and sharing of voice samples requires scalable solutions at the data layer.

Third, by embedding the analysis software on hand-held portable devices, the collection and first analysis of the voice sample could be performed directly near the body of the patients, to monitor in a continuous way the voice of patients that had surgical interventions, or to allow monitoring also when the patients is disconnected.

3. PROTEIN-COMPLEXES PREDICTION

Proteins within a cell interact composing a very broad network of interactions, also known as Protein-Protein Interaction (PPI) [12]. If two or more proteins interact for a long time forming a stable association, their interaction is known as *protein complex*. *Interactomics* focuses on the determination of all possible interactions and on the identification of a meaningful subset of interactions. Protein complexes play a biological relevant role and many of them, such as proteasomes, are central components of vital cellular tasks. Recently, different studies have shown relevance of protein complexes in the development of diseases [15].

Due to the high number of proteins within a cell, manual investigation and analysis of protein interactions is unfeasible, so computational methods to model and investigate interactions are needed [24].

The most natural way to model PPIs network is by using graphs [11], where proteins are represented as nodes and interactions as edges linking them. The simplest representation is an undirected graph in which the nodes are labeled with the protein identifiers, while the edges are simple connections (i.e. no labels or directions).

The biological investigation of a PPI network consists in studying the structural properties of the graph [16], for instance highly connected subgraphs may represent biological relevant and meaningful proteins interactions [4].

The possibility to find protein complexes in a PPI network searching for highly interconnected regions has been demonstrated in the early work of Bader [4]. Predicted complexes can be already known, i.e. their composition is known, or can denote a new protein complex. In this case, if the experiments confirm this relation, the algorithms can be used as predictors. These algorithms, also called complex prediction tools (or simply predictors), belong to the general class of graph clustering algorithms, where each cluster is defined as a set of nodes of the graph with their connections. Thus, clustering algorithms aim to identify subgraphs. The quality

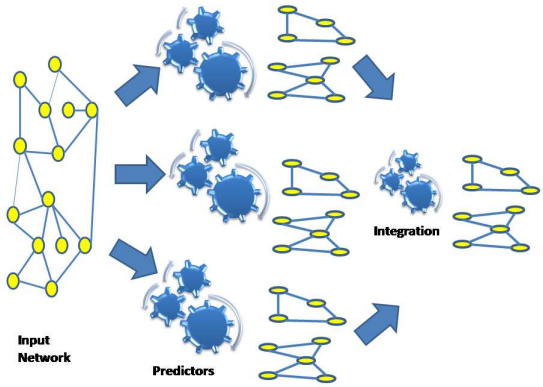

Figure 2: Structure of the Metaprediction

of predictors is measured in terms of percentage of complexes biologically meaningful with respect to the meaningless ones.

Clustering algorithms take as input a graph representing an interaction network among a set of proteins and an initial configuration (i.e. algorithms parameters such as the number of clusters). While initial configurations mostly depends on clustering algorithms, the initial interaction graph mostly depends on known protein interactions.

Recently, a number of algorithms that predict protein complexes (predictor) starting from graphs have been developed [24],[14],[4], [17], [2]. The available predictors are usually implemented in a centralized and sequential way.

Recently, a system based on the integration of different prediction tools to improve quality has been developed. Such a system, named IMPRECO (IMproving PREdiction of Complexes), combines different predictor results using an integration algorithm able to gather (partial) results from different predictors, to improve the biological relevance of the protein complexes associated to the output identified clusters [8].

An advantage of the meta-predictor approach is the possibility to increase the scalabilty of the system: whenever partial results coming from different predictors are available, integration can be carried out producing results in a pipeline fashion. The rest of the section presents an internally parallel service oriented architecture IMPRECO.

3.1 Algorithm and Architecture

The IMPRECO meta-predictor combines different predictor results using an integration algorithm able to gather (partial) results from different predictors invoked in parallel, as depicted in Figure 2.

The integration algorithm starts by integrating results (i.e. clusters) obtained by running different available predictors. Three different cases are considered by evaluating the topological relations among clusters coming from the considered predictors:

1. *equality*: the same clusters are returned by all (or by a significant number of) predictors,

2. *containment*: it is possible to identify a containment relation among (a set of) clusters returned by all (or by a significant number of) predictors;

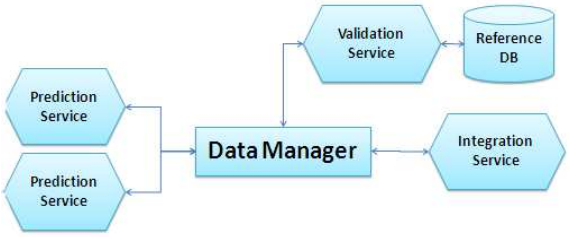

Figure 3: Architecture of IMPRECO

3. *overlap*: it is possible to identify an overlap relation among (a set of) clusters returned by all (or by a significant number of) predictors;

The proposed algorithm works in three phases: i) it firstly parses results coming from different predictors, then (ii) tries to associate them in one of the three possible considered configurations and finally (iii), it performs the integration phase among clusters. The latter phase is performed by selecting clusters from the set obtained during the second phase. All phases are integrated into an on line available tool[1].

Main modules of the system are:

Data manager module. It collects the outputs of the different predictors and translates them into a a single formalism known to IMPRECO. Currently, three existing predictors are used, which are MCODE [4], RNSC [14] and MCL [10].

Integration Module. It implements the integration strategy. The first version of IMPRECO verifies the three relations in a sequential way.

Evaluation Module. It evaluates the predictions with respect to a **reference database**, i.e. a catalog of verified complexes .

The service oriented architecture of IMPRECO is depicted in Figure 3. It allows: (i) to wrap each predictor as a web/grid service; (ii) to realize an internally parallel integration service wrapping it as web/grid service; (iii) to wrap the evaluation module as a web/grid service.

The core of the architecture is represented by the data manager module. It receives as input a network, a list of predictors (and relative parameters) that have to be used, and a set of parameters of the integration. Then it builds an execution plan and executes it. Then it invokes each prediction service using a replica of input network for input. Finally it has to merge together the results and to invoke the integration service with the resulting set of predictions.

3.2 Prediction Services

The first step is to obtain the partial prediction results by using the existing prediction tools. To parallelize the computation of each independent prediction we wrapped the existing predictors as web/grid services. The main issue in the development of these service is to face with the different syntaxes used by predictors to format both input and output. In such a way the resulting service has to translate the input network into a format readable by predictors. The resulting predictions are finally translated in a common format readable by subsequent modules of IMPRECO.

[1]http://bioingegneria.unicz.it/~guzzi

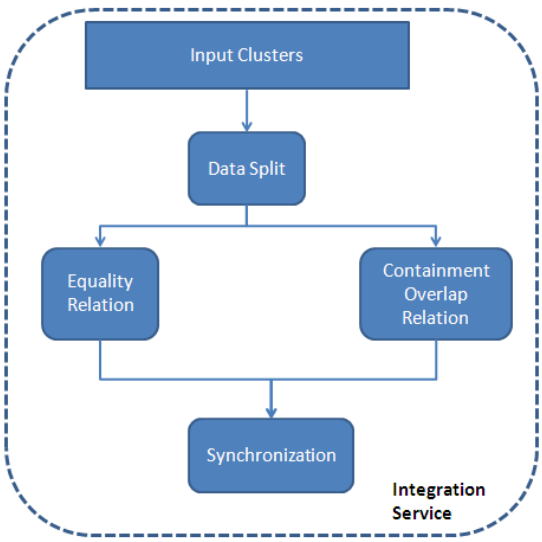

Figure 4: Parallel Integration

3.3 Integration Service

The integration module of IMPRECO implements the integration strategy. The first prototype of IMPRECO verified the three relations in a sequential way. Initially, it builds the set of all clustering outputs starting from data parsed from the data manager. Then it tries to verify the equality relation. After this first phase, IMPRECO considers clusters bigger than a threshold TD. Initially it finds those that verify the Containment relation. Finally, the clusters that do not satisfy this relation are considered. IMPRECO searches for those that satisfy the Overlap relation. At the end of each phase, found clusters are inserted into the integrated set.

The service implementation of IMPRECO wraps this module as a service that internally executes three phases in a parallel way as depicted in Figure 4. Data coming from Prediction services can be partitioned in two groups. The first group includes clusters whose dimension are lower than TD and is processed by a module verifying the equality relation. The second group constitutes the input for a module that verifies the containment and overlap relation in a sequential way.

3.4 Validation Service

To estimate the integration quality, IMPRECO uses an evaluation module based on a **reference database**, i.e. a catalog of verified complexes. IMPRECO actually uses the MIPS catalog [20], but a user can build IMPRECO's database autonomously. The evaluation module calculates the measurements of sensitivity, positive predictive values (PPV) and accuracy for each cluster. The first measure is an average representing the fraction of proteins of a complex that are found in a common cluster. When only a big cluster is found, the sensitivity tends to one. The second measure

represents the fraction of members of a cluster that belong to a given complex. When each protein belongs to one cluster, PPV is 1, conversely to the previous measure. Thus, the third measure, being the geometric average of sensitivity and PPV represents a trade-off.

As the reference for each clustering, we considered a weighted average of both measures for each cluster as defined in [5], and we calculated the accuracy over these. These measurements are calculated with respect to a reference database storing the verified protein complexes. Currently, only a few such databases exist, including the MIPS catalog [20], the Mammalian Protein Complex Database (MPCDB) [20] and the CORUM Complexes Database [23]. We used the first one, a manually annotated catalog of complexes determined in yeast.

4. SERVICES FOR COMPUTATIONAL PROTEOMICS

Computational Proteomics is about the computational methods, algorithms, databases, and methodologies used to process, manage, analyze and interpret the data produced in proteomics experiments [6].

Mass spectrometry, a main experimental technique used in proteomics, is an analytical tool used for measuring the molecular mass of a sample. Mass Spectrometry-based proteomics is a powerful technique for identifying molecular targets in different pathological conditions [1]. Mass Spectrometry output can be represented, at a first stage, as a (large) sequence of value pairs, said spectrum. Each pair contains a measured *intensity*, which depends on the quantity of the detected biomolecule, and a mass to charge ratio (m/z), which depends on the molecular mass of the detected biomolecule. 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, that is the total number of measurements. 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.

Finally, 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 [7].

In summary, 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.

4.1 MS-Analyzer

MS-Analyzer is a Grid-based Problem Solving Environment for the design and execution of proteomics applications [9]. 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: (i) biological databases; (ii) experimental data sets (e.g. a set of spectra); (iii) bioinformatics software tools (e.g. preprocessing tools, peptide identification tools, etc.); and (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 pre-processing and data mining services. In particular it supports: (i) interfacing remote and heterogeneous proteomics facilities; (ii) storing and managing MS proteomics data; (iii) integrating data mining and visualization software tools.

MS-Analyzer adopts the Service Oriented Architecture: it provides a collection of specialized spectra management services and integrates public 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. MS-Analyzer provides the following services.

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 kind of mass spectrometers (e.g. MALDI-TOF, LC-MS/MS) and their storing in a specialized spectra database named SpecDB [25]. To face the huge volumes of mass spectra data, that could not be analyzed only in main memory, and to allow an easy and efficient access to single spectrum, to multiple spectra, and to relevant portions of spectra, 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 a XML-based data model [22], to store metadata information about proteomics experiments. The relational model allows fast access to portions of spectra, while the XML model allows easy querying of information and remote sharing of datasets.

2. **Spectra pre-processing services** load raw spectra, apply pre-processing algorithms, and store data back in the spectra database. Pre-processing can be applied to one spectrum or contemporarily to many spectra.

3. **Spectra preparation services** load pre-processed spectra and prepare them to be given in input to different kind of data mining tools. E.g. Weka [27] requires spectra being organized in a unique file named ARFF (Attribute-Relation File Format).

4. **Data Mining services** implement common data mining tasks (e.g. classification, clustering, pattern analysis).

5. **Data Visualization services** allow to visualize raw and preprocessed spectra, as well as the knowledge models produced by data mining analysis.

4.2 Ontology-Based Workflow Designer

Ontologies are used in MS-Analyzer to link knowledge about experimental research (e.g. wet lab) and bioinformatics applications. They model the key domains of interest: data mining and mass spectrometry-based proteomics. A first ontology, named *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 [27]. A second ontology, named *ProtOntology* (Proteomics Ontology), models concepts, methods, algorithms, tools and databases relevant to the proteomics domain (e.g. spectra, preprocessing, etc).

The ontology-based workflow editor of MS-Analyzer allows to improve the design of a computational application by using the previous ontologies. In particular it supports the following steps:

1. **Ontology-based component selection.** The main tasks of a proteomics analysis can be seen by browsing the ontologies, then the dataset to be analyzed, the proper preprocessing techniques, and the kind of data mining task and related software tools can be selected.

2. **Workflow design.** Selected components are combined producing a graphic workflow schema that is further translated into a workflow language.

3. **Application execution.** The workflow is scheduled (e.g. on the Internet or on the Grid) by a workflow scheduler. In particular, the MS-Analyzer scheduler takes care of data movement and communication between services.

4. **Results visualization and storing**. Both spectra data and data mining results can be visualized and eventually annotated.

A working prototype of MS-Analyzer has been currently implemented. Preprocessing algorithms have been implemented as Web Services in Java, while the data mining services are provided by the Weka data mining suite [27]. The spectra database has been fully implemented on the top of an open source DBMS. Ontologies have been implemented using the OWL Web Ontology Language [26].

The graphical user interface of MS-Analyzer (see Figure 5) comprises the *Dataset Manager* and the *Ontology-based Workflow Editor*. The former manages the experiment spectra data available as raw, preprocessed and prepared spectra. The latter allows browsing, searching and selection of bioinformatics tools through the ontologies in order to compose services through a workflow. The workflow is designed by using a UML-based notation, providing basic control blocks such as fork/join, etc. Constraints expressed by the ontologies (e.g. the type of data to be given in input to a service) are enforced at composition time. For instance all data mining services require an ARFF file as input, that can be obtained from a (pre-processed) dataset through a proper transformation. The produced abstract workflow schema is translated into a schedulable concrete workflow using a subset of the BPEL4WF workflow language [13].

5. CONCLUSIONS

The huge production of biological data in the so called omics sciences (genomics, proteomics, interactomics, etc.) and the need to validate experiments and analysis of data, makes the sharing of data a key issue for future applications. Biological databases, containing both primary data such as protein sequences and structures, as well as derived information produced in local laboratories, will be more and more made available through the network.

A similar trend, but more slow, will certainly be observed in biomedical applications. Here the focus is mainly devoted

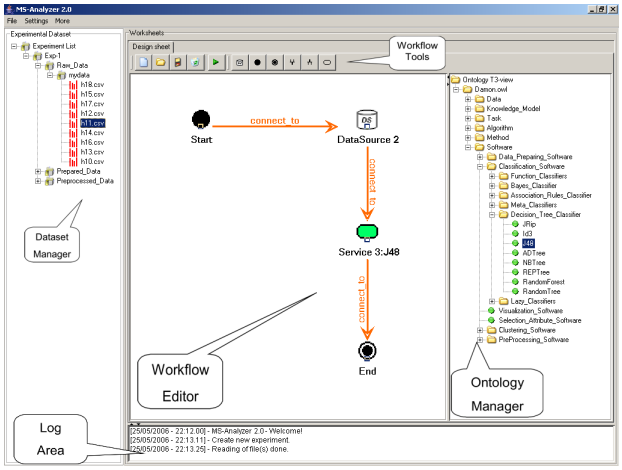

Figure 5: MS-Analyzer Graphical User Interface

to the high-performance analysis of huge volume of data, especially images, but there are also many projects that work on the sharing of biomedical data. In such field, data sharing and computer supported cooperative working can be used to support the discussion of medical cases (e.g. for second-opinion consultancy). But the sharing of biomedical data will also allow the comparative analysis of health procedures at different layers, e.g. the monitoring of medical protocols applied in different health centers and the analysis of follow-up.

To allow interoperability at the application level and to improve scalability, the service oriented architecture approach starts to be used in such field, while XML is more and more used as a neutral, portable data format. Grid is used both to support high-throughput applications as well as efficient data movement through specialized protocols (e.g. GridFTP) and efficient management of data files and their distributed replicas.

6. REFERENCES

[1] R. Aebersold and M. Mann. Mass spectrometry-based proteomics. *Nature*, 422:198–207, 13 March 2003.

[2] M. Altaf-Ul-Amin, Y. Shinbo, K. Mihara, K. Kurokawa, and S. Kanaya. Development and implementation of an algorithm for detection of protein complexes in large interaction networks. *BMC Bioinformatics*, 7(1):207, 2006.

[3] F. Amato, M. Cannataro, C. Cosentino, F. Montefusco, G. Tradigo, P. Veltri, A. Garozzo, N. Lombardo, S. Greco, and C. Manfredi. A web-based system for the collection and analysis of spectra signals for early detection of voice alterations. In R. L. Wainwright and H. Haddad, editors, *SAC*, pages 1405–1409. ACM, 2008.

[4] G. Bader and C. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4(1):2, 2003.

[5] S. Brohée and J. van Helden. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, 7:488, 2006.

[6] M. Cannataro. Computational proteomics:

management and analysis of proteomics data. *Briefings in Bioinformatics*, 9(2):97–101, 2008.

[7] M. Cannataro, P. Guzzi, T. Mazza, G. Tradigo, and P. Veltri. Preprocessing of mass spectrometry proteomics data on the grid. In I. Press, editor, *18th IEEE International Symposium on Computer-Based Medical Systems (CBMS'05), Trinity College Dublin, 23-24 June 2005, Dublin, Ireland*, pages 549–554, 2005.

[8] M. Cannataro, P. H. Guzzi, and P. Veltri. A framework for the prediction of protein complexes. In *Abstract in Proceedings of the Bioinformatics Italian Society Conference (BITS 2007)*, 2007.

[9] M. Cannataro and P. Veltri. Ms-analyzer: preprocessing and data mining services for proteomics applications on the grid. *Concurrency and Computation: Practice and Experience*, 19(15):2047–2066, 2007.

[10] S. Enright, A.J. Van Dongen and C. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30(7):1575–1584, 2002.

[11] D. Fell and A. Wagner. The small world of metabolism. *Nat Biotechnol.*, 18(11):1121–2, 2000.

[12] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattor, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci USA*, 98:4569–4574, 2001.

[13] M. Juric, P. Sarang, and B. Mathew. *Business Process Execution Language for Web Services*. Packt Publishing, 2004.

[14] A. D. King, N. Przulj, and I. Jurisica. Protein complex prediction via cost-based clustering. *Bioinformatics*, 20(17):3013–3020, 2004.

[15] K. Lage, O. E. Karlberg, Z. M. Størling, Páll, A. G. Pedersen, O. Rigina, A. M. Hinsby, Z. Tümer, F. Pociot, N. Tommerup, Y. Moreau, and S. Brunak. A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nature Biotechnology*, 25(3):309–316, March 2007.

[16] A. Lesne. Complex networks: from graph theory to biology. *Letters in Mathematical Physics*, 78(3):235–262, December 2006.

[17] X. Li, S. Tan, C. Foo, and S. Ng. Interaction graph mining for protein complexes using local clique merging. *Genome Inform*, 16(2):260–9, 2005.

[18] M. S. Ltd. Mascot search engine. - http://www.matrixscience.com/.

[19] C. Manfredi and G. Peretti. A new insight into postsurgical objective voice quality evaluation: application to thyroplastic medialization. *IEEE Transactions on Biomedical Engineering*, 53(3):442–451, 2006.

[20] H. W. Mewes, D. Frishman, K. Mayer, M. Münsterkötter, O. Noubibou, P. Pagel, T. Rattei, M. A. R. Oesterheld, and V. Stümpflen. Mips: analysis and annotation of proteins from whole genomes in 2005. *Nucleic Acids Res.*, 34(DATABASE ISSUE):D169-D172, 2006.

[21] G. Ontology. Gene ontology web site. - http://www.geneontology.org.

[22] S. E. Orchard and H. Hermjakob. The hupo proteomics standards initiative - easing communication and minimizing data loss in a changing world. *Briefings in Bioinformatics*, 9(2):166–173, 2008.

[23] A. Ruepp, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, M. Stransky, B. Waegele, T. Schmidt, O. Doudieu, V. Stümpflen, and H. Mewes. Corum: the comprehensive resource of mammalian protein complexes. *Nucleic Acids Res*, page 36, 2008.

[24] R. Sharan, T. Ideker, B. Kelley, R. Shamir, and R. Karp. Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. *J Comput Biol.*, 12(6):835–46, 2005.

[25] P. Veltri, M. Cannataro, and G. Tradigo. Sharing mass spectrometry data in a grid-based distributed proteomics laboratory. *Inf. Process. Manage.*, 43(3):577–591, 2007.

[26] W3C. Owl web ontology language reference. - http://www.w3.org/TR/owl-ref/.

[27] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco (2nd Edition), 2005.