

# Automation of in-silico data analysis processes through workflow management systems

Paolo Romano

Submitted: 31st August 2007; Received (in revised form): 17th October 2007

## Abstract

Data integration is needed in order to cope with the huge amounts of biological information now available and to perform data mining effectively. Current data integration systems have strict limitations, mainly due to the number of resources, their size and frequency of updates, their heterogeneity and distribution on the Internet. Integration must therefore be achieved by accessing network services through flexible and extensible data integration and analysis network tools. EXtensible Markup Language (XML), Web Services and Workflow Management Systems (WMS) can support the creation and deployment of such systems. Many XML languages and Web Services for bioinformatics have already been designed and implemented and some WMS have been proposed. In this article, we review a methodology for data integration in biomedical research that is based on these technologies. We also briefly describe some of the available WMS and discuss the current limitations of this methodology and the ways in which they can be overcome.

**Keywords:** *biological data integration; automation of retrieval and analysis processes; XML; web services; workflow management systems; ontologies*

## INTRODUCTION

### Some relevant characteristics of biological information

In the post-genomic era, a huge amount of biological and medical information is publicly available. Genome projects contributed only a fraction of all available data. Emerging research domains, like the analysis of mutations and of metabolic pathways, and high-throughput technologies are contributing with even huger amounts of data. The scientific literature remains one of the most important sources of biological information.

This information is increasing at an impressive rate. The size of the European Molecular Biology Laboratory Data Library reached 97 361 640 sequence entries in its 91 release. It grew by 6.78% since the previous release and by 31.50% in 1 year. ArrayExpress [1], a microarray experiments database maintained by the European Bioinformatics Institute

(EBI), included 2256 experiments and occupied 1 471 587 Mb in June 2007 with an increase of 19.02% from June 2006. As regards the literature, Medline includes >15 000 000 references.

Only a few databases are managed under a coordination effort. This is the case for nucleotide sequences databanks available at EBI, National Center for Biotechnology Information (NCBI) and the Japanese National Institute of Genetics that exchange data on a peer-to-peer basis under the framework of the International Nucleotide Sequence Database Collaboration [2]. Instead, databases on similar biological objects can be managed without a common information and data structures. For example, the International Agency for Research on Cancer Tumour Protein 53 (TP53) somatic mutation database [3], the Universal Mutation Database-TP53 [4] and the Catalogue Of Somatic Mutations In Cancer [5] all refer to mutations of the

Corresponding author. Paolo Romano, Bioinformatics, Istituto Nazionale per la Ricerca sul Cancro, Largo Rosanna Benzi 10, I-16132, (IST), Genova, Italy. Tel: +39-010-5737288; Fax: +39-010-5737295; E-mail: paolo.romano@istge.it

**Paolo Romano** obtained a Bioengineering PhD degree at the Polytechnic of Milan. Since 1993 he has been a researcher at the National Cancer Research Institute of Genoa. His interests include biological databases, data modelling and integration, automation of retrieval and analysis processes through semantic tools and programming interfaces.

TP53 human gene, but each of them has its data structure.

Information in secondary databases is of the highest quality. Data are derived from primary databases and undergo a careful procedure for removal of errors and of duplications and an extended annotation. They often represent an essential resource for researchers since it is aimed at special research interests. Many databanks are created and maintained by small groups or even by a single researcher. This leads to a high number of heterogeneous databases, the majority of which are of a great interest to researchers. The Nucleic Acids Research Supplement devoted to molecular biology databases gives a precise idea of this situation. In its 14th edition, in 2007 [6], it listed 968 databases, 110 more than in the previous one. Also, the list of databases that are available in public Sequence Retrieval System (SRS) [7] sites includes >1000 names.

As a result of this diffused and uncoordinated development, data are spread over hundreds of Internet sites where it is stored, using heterogeneous database management systems and data structures. There are no common information sets and the semantics of data, i.e. the actual meaning associated to each piece of data, is left to the developers and can be different, even when using same or similar names, thus leading to potential confusion. User interfaces and query methods are also different and searching, retrieving and integrating information become very difficult. Data are often manually retrieved by researchers, making access to several servers through their web browsers with the 'cut and paste' technique being widely used to transfer data from one web resource to another for further analysis.

### **Biological data integration**

The main goals of biological data integration are the achievement of a wider view of information, the automatic carrying out of analysis involving more databases and software and the execution of large-scale analysis. Only a tight integration of all information can support an effective and real data mining.

Data integration has some pre-requisites: it can best be achieved when the information and desired analysis are stable in time. Additionally, it is made easier by a sound and thorough knowledge of the domain and when information and data are well defined. These conditions lead to a standardization of data models and formats. Also essential is a clear definition of the desired outputs of the integration.

Instead, integration is often impeded by heterogeneous data and systems, uncertain domain knowledge, highly specialized and quickly evolving information, lack of predefined clear goals and originality of procedures and processes. In biology, a pre-analysis of data is very difficult, because the domain's knowledge changes very quickly. Moreover, the complexity of information makes it difficult to design data models that are valid for different application domains and over time. Finally, the goals and the needs of researchers evolve very quickly, according to new theories and discoveries that lead to new data, goals and processes.

The majority of current integration methods are based on syntactical tools, like explicit cross references, implicit links (e.g. shared names) and common contents (terms from shared vocabularies and lexicons). These methods rely on the manual annotation of data, which is a long and costly task, prone to errors and very demanding. They are also unable to convey the semantics of the link. Integration methods based on semantics, such as those that can be implemented by using reference ontologies for associating metadata descriptions to data sources, seem more adequate and are increasingly used.

Integration tools that manage local copies of the information sources (like SRS) pose many problems, mainly deriving from the size of the sources, the need for continuous updates and for coping with frequently modified data structures and new databases. Data warehouses have similar problems and considerable effort is required to define them and set them up. Flexibility of systems, including the ability to support frequent changes of data models, software and objectives of the analysis, is needed. Integrating biological information in a distributed, heterogeneous environment requires flexible, expandable and adaptable technologies and tools that allow to move toward an automation of data analysis through systems that automatically access remote sites, retrieve information from the databases of interest and/or use the appropriate software to achieve the desired analysis, while at the same time are able to cope with the heterogeneity of data sources and to select and manage properly the right information (semantics).

### **ICT TECHNOLOGIES SUPPORTING DATA INTEGRATION**

Among current Information and Communication Technologies (ICT), Workflow Management

Systems (WMS), in connection with eXtensible Markup Language (XML), Web Services (WS) and ontologies, seem to be the most promising one. In the following paragraphs, some characteristics of these technologies and their increasing utilization in the biomedical domain are presented.

### **EXtensible Markup Language**

Structured information contains both data and its contextual meaning. When semantics is implicit, only a person can understand it from the context; when it is explicit, software can also manage the data. A markup language is a mechanism for identifying parts of a document by inserting start and end tags. The XML specification defines a way to add markup to documents and assign meanings to data explicitly. A set of tags and their relationships defines an XML and constitutes a namespace (context in which these definitions are valid). XML languages are defined by using Document Type Definitions or XML schemas. An XML file may include tags from more namespaces thus combining definitions of various XML-based languages. In conclusion, XML allows for a machine-readable description of the data through languages that are valid in a knowledge domain.

Reasons for adopting XML languages in bioinformatics have been presented in [8–10]. Many languages have been created for bioinformatics applications. Among them, some support the storage of information in formats that can improve traditional flat-file management. These include the Bioinformatics Sequence Markup Language (BSML) and the UniProt XML. InterPro, a databank incorporating many information from primary protein-related databases, uses XML for storing information. Languages have also been defined to describe output of analysis tools: NCBI Blast output is formatted according to the BlastXML.

XML languages have been used in specialized knowledge domains. For example, the Polymorphism Markup Language [11] has been developed to overcome the heterogeneity of Single Nucleotide Polymorphism databases as a common data exchange format. Similar goals lead to the development of the Genomic Sequence Variation Markup Language, by the Health Level 7 community, and of the Biological Variation Markup Language [12, 13], a model for polymorphism data-describing clinical genotypes.

XML languages have also been defined for information of complex systems. This is the case of

the Systems Biology Markup Language [14], which is defined as ‘a computer-readable format for representing models of biochemical reaction networks’. The same objectives are driving the development of the Cell System Markup Language (CSML). An XML has also been defined for microarray experiments and gene expression data. This represents a special case, as it is the result of the effort of a community. It will be briefly described in a separate paragraph later.

XML languages can support data interchange. In order to simplify interoperability between bioinformatics tools, the Helmholtz Open Bioinformatics Technology (HOBIT) XML schemas and the BioDOM library were developed [15]. HOBIT XML schemas refer to some bioinformatics data types (sequence, RNA structure and alignment) and BioDOM is a java library for their management. The proteomics standards initiative of the Human Proteomics Organization is now developing a set of interchange formats for proteomics applications. New XML languages for bioinformatics are being continuously developed.

### **Web services**

Web services are machine-oriented network services based on XML, usually communicating by using the Simple Object Architecture Protocol (SOAP). WS offer a standardized programming interface so that software tools can effectively make access to information and services. Standards have been defined for their description [Web Services Description Language (WSDL)], retrieval and identification [Universal Description, Discovery, Identification (UDDI)] and composition [Web Services Choreography Description Language (WSCDL)]. WS can have ontological metadata added to them, thus allowing software applications to access data in a semantic aware manner.

Reasons for the setting up of WS in bioinformatics have been presented [16–18]. These include the need for overcoming the scaling problem arising from the use of high-throughput experimental protocols: these provide such huge amounts of data that their analysis in an adequate time scale needs a ‘high-throughput’ sequence analysis process that could not be achieved through the traditional approach of manual access to web sites. Also, WS would offer bioinformatics the possibility of implementing a real distributed analysis environment, while protecting intellectual property rights for data,

algorithms and source code, that would not be copied and would remain on the owners' information system.

WS have already been implemented by many institutes and service providers in the biomedical field. EBI WS [19] are a suite of tools allowing access to homology searches, multiple sequence alignment and text mining. NCBI developed Entrez Utilities Web Service for querying databases. DNA Data Bank of Japan developed Kyoto Encyclopedia of Gene and Genomes WS for accessing the Kyoto Encyclopedia of Genes and Genomes.

Many other WS have been implemented. The Distributed Annotation System defines a communication protocol to exchange biological sequence annotations [20]; GeneCruiser [21] and Biosphere [22] support microarray experiments analysis and Common Access to Biological Resources and Information WS give access to biological resources information [23, 24].

SoapLab [25] is an effective tool for the deployment of WS by non-experts. It supports implementation of WS interfacing both command line software and web forms. The SoapLab implementation at EBI implements access to each software of the European Molecular Biology Open Source Software suite.

BioMOBY [26, 27] is an open source software implementing an architecture for the discovery and distribution of biological data through WS; data and services are decentralized, but the availability of these resources, and the description of interaction methods, are registered in a central location called MOBY Central. Since BioMOBY offers a unique interface, its clients can interact with multiple sources of biological data, regardless of the underlying format or schema, in a homogeneous way. BioMOBY includes a classification of bioinformatics data, elaborations and contexts (see following paragraph for details). MobyServlet [28] is a framework that allows existing Java applications to be easily converted into WS conforming to BioMOBY semantics. Although it requires a programming effort, this tool has the advantage of improving interoperability of services in the BioMOBY environment.

## Ontologies

An ontology is the specification of conceptualization in a given domain of interest. It consists of a set of concepts expressed by using a controlled vocabulary

and the relationships among these concepts. Ontologies can add semantic metadata to the resources, improve data accessibility and support integrated searches. Many biomedical ontologies have been or are being developed, mainly in the context of the Open Biomedical Ontologies initiatives. Unfortunately, available ontologies are still only partially applied and the vast majority of data sources do not use them.

In the context of the automation of biological data analysis processes, ontologies referring to bioinformatics data and tasks are of the highest relevance. They can be used to characterize WS by annotating their inputs and outputs, data sources and computation type (e.g. alignment, retrieval and gene prediction). Such characterization can support both search and discovery of services and interoperation between them.

The BioMOBY ontology [26] consists of three interdependent hierarchies related to data types, services and namespaces. The data types hierarchy specifies possible MOBY objects, i.e. data that can be transferred between a client and a service. The services hierarchy specifies the possible analyses, like alignments, data retrieval and computation of phylogenetics distance. The namespaces hierarchy includes contexts where services and data types can be applied.

The MyGrid ontology [29] has been designed to support semantic discovery of bioinformatics services. It includes two components: the service ontology and the domain ontology. The latter includes descriptions of data types relevant to bioinformatics and their relationships, while the former describes characteristics of WS. By combining the two ontologies, WS can be characterized on the basis of their computation, data sources and I/O data types. A similar ontology was developed to support search and selection of workflows in the biowep workflow enactment portal [30].

The System for the Integration of Bioinformatics Services (SIBIOS) ontology [31] is used to support the discovery of WS within the SIBIOS workflow system. This ontology is structured as three connected components referring to biological and bioinformatics concepts and software tools.

Finally, a proposal has recently been published for the setting up of a registry of all bioinformatics resources, the Resourceome [32], where the resources are annotated on the basis of a domain

ontology including definitions of bioinformatics data types and tasks.

It is important to point out that WS constitute the only interface to the systems they expose. In an automated analysis process, all data exchange is carried out through WS. So, it is essential that a shared reference ontology of bioinformatics data types and tasks be used by WS. At the same time, association of semantic metadata to databases' components (such as tables and attributes) becomes useless since these are not directly accessed by users.

### Workflow management systems

Workflows are defined as 'computerized facilitations or automations of a business process, in whole or part' (Workflow Management Coalition, WfMC). Their goal is the implementation of data analysis processes in standardized environments. All data-processing steps in a complex process are ordered in the proper way and interlinked so that the overall process can be carried out by executing each task when all needed requisites are fulfilled and by transferring data from one step to the following one. The main advantages of automated workflows relate to effectiveness, reproducibility, reusability of procedures and of intermediate results and traceability. Effectiveness is achieved through the automation of repetitive procedures: being an automatic procedure, a workflow can free bio-scientists from repetitive interactions with the web, at the same time supporting good practice. Reproducibility is also granted by the implementation of repetitive procedures, although it is limited by the frequent update of information sources; anyway, analyses can be replicated over time. Reusability is implemented by storing intermediate results and by allowing their use in subsequent workflows executions and by making workflows widely available in the scientific community. Finally, traceability is achieved by storing intermediate results and allowing their analysis: the workflow is then carried out in a transparent analysis environment where data provenance can be checked and/or controlled. This is especially important when unexpected data are obtained.

A WMS is a system that defines, creates and manages the execution of workflows. Its main components are:

- a graphical interface for composing workflows, entering data, watching execution, displaying results,
- an archive to store workflow descriptions, results of executions and related traces,
- a registry of available services, either local or remote,
- a scheduler able to invoke services included in the workflow at the appropriate time,
- a set of programming interfaces able to dialogue with remote services,
- a monitor tool for controlling the execution of the workflow,
- a set of visualization capabilities for displaying different types of results.

Many WMS have already been proposed, both by industries and by academic and research institutes, and are being increasingly applied in the biomedical domain. A classification of WMS is introduced in the following Section.

### A methodology for the automation of retrieval and analysis processes

WMS are the most promising technology for supporting the creation and deployment of flexible, network based, integration systems. For this to happen, a methodology must be defined and adopted by the highest possible number of developers and service providers. This should implement ways for sharing data models and, when possible, data definitions and should be based on common data interchange formats. The following methodology could then be devised:

- XML schemas can be used for the creation of common models of biological information,
- XML-based languages can be adopted for data storage, representation and exchange,
- WS can be made available for the interoperability of software,
- ontologies can semantically support WS discovery, selection and interoperation,
- workflows can be created and maintained for the execution of analysis processes,
- workflow enactment portals can provide widespread utilization of automated processes.

This methodology can partially cope with the problem of format changes in data sources since it limits interoperability issues to the interface level. This implies that changes, even important ones, can occur at the data management level without

**Table 1:** List of some WMS and portals for bioinformatics

Tool	Class	Language	Distrib	URL	Bib
Biopipe	Library	Pipeline XML	Open source	<a href="http://www.gmod.org/biopipe/">http://www.gmod.org/biopipe/</a>	[33]
BioVBI	Web-based, local services	Proprietary	Commercial	<a href="http://www.alphaworks.ibm.com/tech/biowbi">http://www.alphaworks.ibm.com/tech/biowbi</a>	[44]
BioWMS	Web-based, remote services	XPDL	Public access	<a href="http://litbio.unicam.it:8080/biowms/">http://litbio.unicam.it:8080/biowms/</a>	[50]
Taverna Workbench	Stand alone	XScufl	Open source	<a href="http://taverna.sourceforge.net/">http://taverna.sourceforge.net/</a>	[36–40]
Kepler	Stand alone	MoML	Open source	<a href="http://kepler-project.org/">http://kepler-project.org/</a>	[41]
SIBIOS	Client-server, remote services		Open source		[47]
Pegasys	Stand alone	Pegasys DAG	Open source	<a href="http://bioinformatics.ubc.ca/pegasys/">http://bioinformatics.ubc.ca/pegasys/</a>	[49]
Pegasus	Client-server, Grid services				[48]
Wildfire	Stand-alone	GEL	Open source	<a href="http://wildfire.bii.a-star.edu.sg/wildfire/">http://wildfire.bii.a-star.edu.sg/wildfire/</a>	[42]
Triana	Client-server system	TaskGraph	Open source	<a href="http://www.trianacode.org/">http://www.trianacode.org/</a>	[45, 46]
BioWEP	Portal	Xscufl XPDL	Open source	<a href="http://bioinformatics.istge.it/biowep/">http://bioinformatics.istge.it/biowep/</a>	[30]
MOWServ	Portal	XScufl	Public access	<a href="http://www.inab.org/MOWServ/">http://www.inab.org/MOWServ/</a>	[51]

For every system, the above data are specified, when known or applicable: type of tool (see classification in the text), workflow definition language, distribution policy, reference URL and bibliography (see list of references in the text)

influencing how external applications can interact with the system. Of course, care must be taken by database curators to ensure that the interface is still working according to advertised application programming interfaces (API) and these are based on consensus data models.

This methodology can be seen as a speculative idea, very difficult to implement. The Microarray Gene Expression Data Group (MGED) [33] initiative, lead along the above lines, can instead be seen as a success story. MGED is an international society of biologists, computer scientists and data analysts that aim to facilitate the sharing of microarray data. This initiative was devoted to the creation of a common data structure for communicating microarray-based gene expression (MAGE) data. This activity started by defining the Minimum Information About a Microarray Experiment (MIAME) data set. MIAME describes the data that is needed to interpret unambiguously results of any experiment and potentially reproduce it [34]. MIAME includes raw and normalised data for each hybridisation in the study, annotations of the sample and of the array, and other related information. In order to improve specification of MIAME information, and therefore its accessibility, a data exchange model (MAGE-OM) and related data formats were then defined. Formats are specified as spreadsheets and as an XML language (MAGE-ML). In addition, the MGED Ontology was developed for the description of key concepts. A software toolkit was finally developed to facilitate the adoption of MAGE-OM and MAGE-ML.

## WORKFLOWS MANAGEMENT SYSTEMS IN BIOINFORMATICS

Many integration systems have been proposed for biological information. Many of them use workflows for achieving the needed flexibility to cope with researchers' needs. However, some of them are limited to local data sources, either internally generated or downloaded, and analysis tools. In this article, distinction is made between workflow systems and local integration systems with workflow composition capabilities and the analysis is limited to the former. Some information on the workflows systems that are described in following paragraphs is presented in Table 1.

### A classification of WMS

Many classifications can be proposed for WMS. Here, we propose a classification of systems that is based both on complexity of their use for researchers and actual possibility of accessing and making use of either local or remote services. This classification includes software libraries, standalone systems, client/server systems and enactment portals.

#### Software libraries

Bioinformatics tools are often the result of efforts of single researchers who make use of software libraries for facilitating further software development. Examples are the bio\* initiatives (biojava, bioperl and biopython) associated to the Open Bioinformatics Foundation. Biopipe [35] is a perl module designed to be used with bioperl for high-throughput sequence analysis. It includes job

management routines that constitute an interface to load-sharing tools. These ensure proper ordering of the analysis and their monitoring. Biopipe considers a bioinformatics experiment just as a sequential pipeline, hence it does not support synchronization operators. Software libraries are only useful for skilled programmers developing new software.

### ***Standalone systems***

Standalone applications implement both workflow building and execution in a single tool running on desktops. They allow the user to integrate remote services so that information can be downloaded as part of the workflow execution, while computational tasks can be assigned to remote services for execution. The computational power of the desktop is not relevant, since the most demanding tasks are executed remotely.

These autonomous systems can be downloaded and installed on personal computers and workstations and they do not rely on any devoted server application. They may include support for access to WS and for the deployment of workflows in Grid environments. Standalone applications can be an effective tool for the development and enactment of small-to-medium size tasks, not demanding high performances (memory, storage and execution time), by researchers that have a good knowledge of existing network services and some programming skills.

Taverna Workbench [36, 37] from EBI is the best-known standalone system. It is an open source designed to build workflows for bioinformatics and was developed in the frame of the myGrid project [38]. It is a Java application that is able to build complex workflows, to access both remote and local processors of various kinds, to launch execution of workflows and to display different types of results, including text, web pages and various kinds of images and diagrams. Its main strength resides in its nice interface, which can be used without any special skill, and in its flexibility, i.e. its ability to support access to services offered through many kind of interfaces. In fact, Taverna can interact with WS having a standard WSDL interface definition, Soaplab servers, BioMOBY services [39] and BioMart implementations. It currently offers access to >3000 resources. It is also able to find both definitions of WS and workflows by navigating Internet sites. Internal Java functions and the ability

to include original scripts offer further processing capabilities. Taverna also includes a plug-in mechanism for the addition of interfaces to network services, not already accessible. Taverna explicitly recognizes the importance of data provenance and semantic issues: it includes Feta [40], a semantic-based tool for searching WS, a user interaction tool that is based on email messaging, and an additional component to record provenance information.

Kepler [41], developed at the SDSC, is an open source system based on the University of California Berkeley Ptolemy II tool. Ptolemy's focus is on the assembly of concurrent components, which are independent and autonomous, through a 'well-defined' model of computation governing all needed interactions and thus ensuring a proper working of the system. In this model, independent components are called 'actors'; 'actors' have input and output 'ports' that are connected through 'channels'. The model of computation, including the definition of actors, ports and channels, is expressed in a programming language whose instructions are executed by one special component, called the 'director'. The actors' behaviour is therefore determined by the director, instead of its connections and data flow only. This allows actors to be reused for different goals in workflows having different models. Various general directors and actors are available for the most useful functions, including those needed to harvest and access WS, query databases, execute R and java scripts, and distribute computation in a Grid environment. Kepler includes many mathematical and statistical components and graph visualization tools and therefore it is particularly useful when simulation and modelling tasks are required.

Wildfire [42], from the Singapore Bioinformatics Institute, is a graphical user interface for constructing workflows. It borrows user interface features from Jemboss [43] and adds a drag-and-drop interface allowing the user to compose programs into workflows. For execution, Wildfire uses Grid Execution Language (GEL), a parallel scripting language including explicit parallel constructs that allows to define tasks to be executed in parallel. It is worth noting that GEL can also work autonomously and be run on the command line.

### ***Client/server systems***

Workflow tools can also be based on a client/server approach. In this case, workflows are always

executed at the server side, while their design and creation can occur at both sides. On the server, workflows can be enacted locally, on the server itself, or remotely, on network nodes. In the former case, all data and analysis tools are also maintained locally, while in the latter case they reside on remote machines. These can be available either on the Internet or in a GRID network. Such workflow systems can run large-scale e-science applications. They have to support concurrent users, must take care of security issues and can run workflows taking a long time to complete. As a consequence, these WMS are especially useful for those workflows requiring high performances (with reference to data size and execution time). User interfaces normally are simple and intuitive and do not require neither programming skills nor previous knowledge of existing services. These systems must be managed by an administrator and therefore they are not easily adaptable to the needs of single users.

The Bioinformatic Workflow Builder Interface, BioWBI, from IBM [44], is a web-based graphical user interface for building and executing workflows. The description of data sources is stored in a database, together with workflows and analyses. The workflow execution engine is a back-end application able to process and execute the requests from BioWBI and to return results. The system supports pipes, conditions and iterations. Data sources can be flat files, queries to databases and analysis software.

Triana [45, 46] is an open source, platform-independent tool developed at Cardiff University. It offers users both workflow editing and execution capabilities through a graphical interface accessing the Triana Central Service (TCS). It supports access to WS properly described with a standard language and registered in a standard registry. It also gives access to peer-to-peer and Grid services that are made available through standard interfaces. Users can log in to the TCS, launch a workflow (which is executed by a Triana Service), log off, enter again later and control execution of previous workflows. Triana also allows workflows to be exposed as WS through a UDDI register. Workflows are defined by using a proprietary language, TaskGraph, but can also be exported to BPEL4WS and imported from XML Simple Conceptual Unified Flow Language (Xscufl) (see references further on).

SIBIOS [47] is also based on a client-server architecture that includes three independent

modules. The Workflow Builder and the Result Manager components reside on the client side. On the server side, the Task Engine supports the invocation of services and controls possible interoperability issues among services. SIBIOS is inherently based on WS: its modules are presented as WS and communicate among them by using related standards. Services integrated by SIBIOS include database searches and data analysis tools invocation. During workflow execution, the Workflow Builder queries the server to obtain the current status of running workflows. Intermediate results are fetched, thus allowing the user to inspect partial results before the entire workflow execution is completed. This also allows the user to intervene by pausing the execution, changing the workflow and restarting it. This approach is particularly useful for long-running workflows: by allowing an early evaluation of results, it helps in reducing waste of time possibly derived by planning errors.

Pegasus [48], developed at the University of Southern California, is a workflow manager that supports automatic conversion of high-level workflow descriptions to executable workflows and enacts them in a Condor-based Grid infrastructure. It includes a metadata catalogue service (MCS) and implements workflow reduction, resource selection and task clustering for optimizing execution performances. Pegasus approach is very original since it implements both data and computation abstraction and it is able to optimize performances, but it lacks standardization and openness.

Pegasys [49] is a modular software for data integration from heterogeneous sequence analysis tools. It includes tools for pair-wise and multiple sequence alignment, *ab initio* gene prediction, RNA gene detection and masking repetitive sequences in genomic DNA, as well as filters for database formatting and processing of raw output from various analyses. It includes a data structure for sequence analyses: users can create, save and run workflows through a graphical user interface. Workflows are executed through the Pegasys server and integrated results are presented to the user. Its focused, but limited, application scope makes it a useful tool only for a limited number of researchers.

BioWMS [50] supports workflows definition and execution and management of results through a web-based interface. It is implemented on the BioAgent/Hermes architecture that is based on the multi-agent system software technology.

This technology supports the development of software where basic elaboration tasks are performed by dedicated modules, the ‘agents’, that have autonomous (pro-active) behaviours: they are able to request needed data and return results of their elaborations by communicating with other agents, check if further requests are pending by communicating their availability to further processing, wait until their intervention is needed again, and eventually stop. ‘Mobile agents’ can also be created: they are transferred to a remote computer where they are executed. The BioAgent system dynamically generates agent-based workflow engines from the specification of the workflow to be executed. It both exploits pro-activeness and mobility of agents to embed the proper features inside agents behaviour. The resulting workflow engine is a multi-agent system where agents are distributed and run concurrently.

### ***Enactment portals***

WMS assume that researchers know all bioinformatics resources they need and that they are skilled in programming and composing workflows. They are therefore not viable for the majority of biologists and researchers. Web portals can be implemented for allowing all users to enact workflows in a user-friendly environment. The role of the user is limited to selection and execution of predefined workflows. Portals are essential for the exploitation of workflows in a real research environment, since they free researchers from the burden of the development of workflows and let them concentrate on the scientific problem. When using portals, authentication of users is needed. This allows access rights to workflows to be managed and to store executions metadata and related results in a personal area.

Biowep, Web Enactment Portal for Bioinformatics [30], is a web application that allows the selection and execution of a set of predefined, annotated workflows. It is able to enact workflows that are described with the XScufl language through a server side implementation of the enacter of the Taverna Workbench. It can also execute workflows written in XML Process Definition Language (XPDL) by linking to BioAgent/Hermes server. Biowep’s workflow annotation consists of the registration of task and I/O data types for the main components of the workflow. This is done on the basis of an ontology of bioinformatics tasks and data types. Users can then select workflows on the basis of

their annotation. Users also have a personal space where they can store results.

MOWServ [51] is a web interface developed and implemented at INB (National Institute for Bioinformatics, Spain). It allows the access to BioMOBY compatible services using descriptions stored in BioMOBY central. It includes an automatic interface generator that produces uniform forms for entering data and running workflows. It also allows the discovery of services on the basis of their input data types by using the BioMOBY ontology of datatypes, services and namespaces. The system is able to monitor workflow execution. Workflows are stored using the XScufl language, and their execution is carried out by a custom-made engine.

### ***Workflow definition languages***

Workflows are defined by using a workflow definition language. Since a description and comparison of these languages is beyond the scope of this article, the following notes can be used for reference. The XPDL is a general standard defined by WfMC. WSCDL is a W3C standard for the composition of workflows based on interactions among WS. The XScufl is a proprietary workflow language developed for Taverna. The Business Process Execution Language for WS (BPEL4WS) is a standard proposed by IBM and is intended to specify the interactions between WS.

## **DISCUSSION**

Some limitations are currently hampering the wide deployment of workflow systems in biology. The most important probably are the limited abstraction, poor performances and limited availability of resources.

Abstraction is the key issue. Scientists’ activity is oriented toward scientific results and this is their most important objective. Building workflows and coping with the details of the invoked services is a burden. Nice graphical interfaces do not solve this issue, since they still require knowledge of services, data formats and programming skills. Instead, a rich semantic interface is needed. This should include features like metadata management, association of concepts to systems and to databases, format conversions, automatic iteration management and tools for the visualization of multiple formats. The best interface should allow researchers to build workflows by describing the required processing in natural language. This, of course, is a long-term

objective, but the methodology proposed here can effectively support this goal. Shared data definitions and ontologies of data types allow WS with homogeneous data types to be set up. This would allow workflow systems to automatically (and transparently) introduce in the workflow transformation processors between linked services having different data types. A semi-automatic procedure for identifying and placing customizable adapters into workflows has been presented in [52]. Only a few data sources currently have a semantic characterization of data: this, however, is not a strong limitation in this context, since semantics should be conveyed to WS, especially in terms of a shared reference ontology of bioinformatics data types and tasks. This would avoid the need for associating detailed semantic information to each and all database structural information, such as tables and attributes, that, anyway, would not be retrieved by software accessing the database remotely. In other words, semantics can be associated to the information that is actually exchanged, instead to every single data.

Scientists need the most viable results in the shortest possible time, regardless which database, site or supercomputer is used: these issues should be completely transparent to users. Performances should be improved by the reuse of intermediate results and by a proper policy for distribution of tasks (e.g. applications returning huge amounts of data should submit this for further processing to services in the same LAN or computer). This can be achieved by adding metadata to available services and allowing for an automatic selection of best services among those offering the same processing. Workflows must face many possible faults: high traffic networks can exhibit timeouts, network crashes can make sites unreachable, sites themselves can crash. Workflow systems must cope with these problems, otherwise the most complex workflows have a high failure risk. Transparency of sources, implying the possibility of selecting alternative services for the same processing, can support this need. Also the ability to identify a failure, retry failed processings, suspend workflow execution and restart stopped workflows are all useful features that could be simplified by using alternative services. Reuse of intermediate results, by a related workflow or even by a different workflow needing the same data, can also significantly improve performances. To this end, intermediate results should be saved and properly annotated.

The unavailability of WS for accessing some data sources makes automation of some procedures impossible. The majority of existing databases do not allow for a programmatic access yet. The difficulty of making access to free text annotation and literature is another important limitation. Some services allowing the access and retrieval of structured data from these information sources non-compliant to WS would therefore be extremely useful. Implementation of more databases in SRS sites and the development of a WS for querying any SRS libraries, as the SRS by WS (SWS) system [P.Romano and D.Marra, Submitted for publication], can be of help.

Another fundamental requirement for scientists is the repeatability of experiments. In the workflow scenario, this implies that enactment engines must generate and store a trace of an execution and reproduce it. The trace should include execution metadata describing all steps of the process including both trivial information, such as workflow description, inputs that were used, processing software and sites, and non-trivial ones, like software and databases versions, operating systems of computers that run software. This implies that such data should also be provided by WS, which is not normally the case. It must anyway be kept in mind that *in-silico* processing, unlike *in vivo* tests, are prone to updating and evolution of databases and therefore they do not generally give the same results even after a brief interval of time. Current WMS are increasingly providing facilities for data provenance.

Another important requirement is the ability to interact with external events. User interaction is now recognised as a necessity, especially for long-running workflows. The Taverna Workbench is already providing a user interaction module based on email messaging procedures. Grid-enabled systems are also providing monitoring and interaction tasks.

## CONCLUSIONS

In this article, a review of ICT technologies that can support the development of automated analysis processes in bioinformatics has been presented. Such technologies as XML languages, WS, ontologies and WMS have already been implemented in bioinformatics, but not in a consistent way. A methodology for the automation of data retrieval and analysis in bioinformatics, based on the extensive use of above technologies, has been described. Such automation will offer bioinformatics the

possibility of implementing a really machine-oriented, distributed analysis environment that will give researchers the possibility of improving efficiency of their procedures and that will allow for the implementation of data integration tools able to significantly improve biological data mining.

It is now possible to devise a not-so-distant scenario where such automation can actually be achieved. For this to happen, a joint effort of developers and providers should be done for the definition of shared XML schemas and of a consensus ontology of data types and tasks. The development and implementation of WS compliant to these shared definitions and allowing access to the vast majority of molecular biology and biomedical databases, and the improvement of existing WMS along the line of data and computation abstraction and with improved performances will finally lead to the creation of effective and useful workflows by interested scientists, and hence significantly improving in-silico analyses.

### Key Points

- Biological data integration is made extremely difficult by the nature of the information: its heterogeneity, distribution, size, the need for frequent updating and poor explicit semantics.
- Current methods have strong limitations and integration must therefore be carried out by innovative ICT tools accessing network services.
- A possible methodology is based on XML for data modelling and storage, WS for data interchange, ontologies for semantics aware interoperation and WMS for automation of processes.
- Successful tools have been developed for each of the above steps, but their integration is still partial.
- Some issues are arising, mainly related to networks viability and the nature of the processing required, but there are clear indications that this is the way to go.
- An international effort is needed to support the adoption of the above methodology in the development of new biological information services and in the conversion of existing systems.

### Acknowledgements

The author is grateful to Andrew Emerson for his careful correction of the text and to the reviewers for their useful comments and suggestions. This work was partially funded by the Italian Ministry of Education, University and Scientific and Technology Research, project Laboratory for Interdisciplinary Technologies in Bioinformatics.

### References

1. Brazma A, Parkinson H, Sarkans U., *et al.* ArrayExpress—a public repository for microarray gene expression data at the EBI. *Nucleic Acids Res* 2003;**31**:68–71.
2. Brunak S, Danchin A, Hattori M., *et al.* Nucleotide sequence database policies. *Science* 2002;**298**:1333.
3. Bérout C, Soussi T. The UMD-p53 database: new mutations and analysis tools. *Hum Mutat* 2003;**21**:176–81.
4. Olivier M, Eeles R, Hollstein M., *et al.* The IARC TP53 Database: new online mutation analysis and recommendations to users. *Hum Mutat* 2002;**19**:607–14.
5. Bamford S, Dawson E, Forbes S., *et al.* The COSMIC (catalogue of somatic mutations in cancer) database and website. *Br J Cancer* 2004;**91**:355–8.
6. Galperin MY. The molecular biology database collection: 2007 update. *Nucleic Acids Res* 2007;**35**:D3–D4.
7. Etzold T, Ulyanov A, Argos P. SRS: information retrieval system for molecular biology data banks. *Meth Enzymol* 1996;**266**:114–28.
8. Guerrini VH, Jackson D. Bioinformatics and extended markup language (XML). *Online J Bioinform* 2000;**1**:1–13.
9. Achard F, Vaysseix G, Barillot E. XML, bioinformatics and data integration. *Bioinformatics* 2001;**17**:115–25.
10. Romano, P. (ed) Towards a bioinformatics integrated network environment. In: *Proceedings of NETTAB 2001 Workshop on XML and CORBA*, Genova, May 17–18, 2001 (available from the editor)
11. Sugawara H, Mizushima H, Kano T, *et al.* Polymorphism Markup Language (PML) for the interoperability of data on SNPs and other sequence variations. In: *15th International Conference on Genome Informatics*, December 16–18, 2004, Yokohama Pacifico, Japan.
12. Tyrelle GD, King GC. A genetic polymorphism object model and XML implementation: Biological Variation Markup Language. In: *11th International Conference on Intelligent Systems for Molecular Biology (ISMB 2003)*, June 29–July 3, 2003, Brisbane, Australia.
13. Tyrelle GD, King GC. A platform for the description, distribution and analysis of genetic polymorphism data. In: *Proceedings of the First Asia-Pacific Bioinformatics Conference 19. Australian Computer Science Communications*, February 4–7, Adelaide, Australia. 2003;**25**:173–80.
14. Hucka M, Finney A, Sauro HM, *et al.* The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics* 2003;**19**:524–31.
15. Seibel PN, Kruger J, Hartmeier S, *et al.* XML schemas for common bioinformatic data types and their application in workflow systems. *BMC Bioinformatics* 2006;**7**:490.
16. Stein L. Creating a bioinformatics nation. *Nature* 2002;**417**:119–20.
17. Jamison DC. Open bioinformatics (editorial). *Bioinformatics* 2003;**19**:679–80.
18. Neerincx PB, Leunissen JA. Evolution of web services in bioinformatics. *Brief Bioinform* 2005;**6**:178–88.
19. Labarga A, Valentin F, Lopez R. Web services at the European bioinformatics institute. In: *Proceedings 15th International Conference on Intelligent Systems for Molecular Biology (ISMB) and 6th European Conference on Computational Biology (ECCB)*, July 22–25, 2007, Vienna, Austria.
20. Dowell RD, Jokerst RM, Day A, *et al.* The distributed annotation system. *BMC Bioinformatics* 2001;**2**:7.
21. Liefeld T, Reich M, Gould J, *et al.* GeneCruiser: a web service for the annotation of microarray data. *Bioinformatics* 2005;**21**:3681–2.

22. Cheung K-H, de Knikker R, Guo Y, *et al.* Biosphere: the interoperation of web services in microarray cluster analysis. *Appl Bioinformatics* 2004;**3**:253–6.
23. Romano P, Kracht M, Manniello MA, *et al.* The role of informatics in the coordinated management of biological resources collections. *Appl Bioinformatics* 2005;**4**:175–86.
24. Romano P, Marra D, Milanese L. Web services and workflow management for biological resources. *BMC Bioinformatics* 2005;**6**:S24([doi:10.1186/1471-2105-6-S4-S24](https://doi.org/10.1186/1471-2105-6-S4-S24)).
25. Senger M, Rice P, Oinn T. Soaplab. A unified sesame door to analysis tools. In: Simon J Cox (ed) *Proceedings, UK e-Science, All Hands Meeting 2003*, pp. 509–13. University of Southampton, Southampton, UK. ISBN: 1-904425-11-9 (meeting held on September 2–4, 2003, in Nottingham, UK).
26. Wilkinson MD, Links M. BioMOBY: an open-source biological web services proposal. *Brief Bioinform* 2002;**3**:331–41.
27. Wilkinson M, Schoof H, Ernst R, *et al.* BioMOBY successfully integrates distributed heterogeneous bioinformatics web services. The PlaNet exemplar case. *Plant Physiol* 2005;**138**:5–17.
28. Gordon PM, Trinh Q, Sensen CW. Semantic web service provision: a realistic framework for bioinformatics programmers. *Bioinformatics* 2007;**23**:1178–80.
29. Wroe C, Stevens R, Goble C, *et al.* A suite of DAML+OIL ontologies to describe bioinformatics web services and data. *International Journal of Cooperative Information Systems – Special issue on Bioinformatics*, 2003;**12**:197–224.
30. Romano P, Bartocci E, Bertolini G, *et al.* Biowep: a workflow enactment portal for bioinformatics applications. *BMC Bioinformatics* 2007;**8**:S19.
31. Mahoui M, Ben-Miled Z, Srinivasan S, *et al.* SIBIOS Ontology: a robust package for the integration and pipelining of bioinformatics services. In: *Proceedings of the Third International Workshop on Data Integration in the Life Sciences DILS 2006*, July 20–22, 2006, Hinxton, UK, Springer Lecture Notes in Bioinformatics LNBI 4075, 2006, pp. 104–13, Springer Verlag Berlin Heidelberg 2006.
32. Cannata N, Merelli E, Altman RB. Time to organize the bioinformatics resourceome. *PLoS Comput Biol* 2005;**1**:e76.
33. Spellman PT, Miller M, Stewart J, *et al.* Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol* 2002;**3**:0046.1–0046.9.
34. Brazma A, Hingamp P, Quackenbush J, *et al.* Minimum information about a microarray experiment (MIAME) – toward standards for microarray data. *Nat Genet* 2001;**29**:365–71.
35. Hoon S, Ratnapu KK, Chia JM, *et al.* Biopipe: a flexible framework for protocol-based bioinformatics analysis. *Genome Res* 2003;**13**:1904–15.
36. Oinn T, Addis M, Ferris J, *et al.* Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 2004;**20**:3045–54.
37. Hull D, Wolstencroft K, Stevens R, *et al.* Taverna: a tool for building and running workflows of services. *Nucleic Acids Res* 2006;**W729–W32**.
38. Stevens R, Robinson A, Goble C. MyGrid: personalised bioinformatics on the information grid. *Bioinformatics* 2003;**19**:i302–i304.
39. Kawas E, Senger M, Wilkinson MD. BioMoby extensions to the taverna workflow management and enactment software. *BMC Bioinformatics* 2006;**7**:523.
40. Lord, P. Alper P, Wroe C, *et al.* Feta: a light-weight architecture for user oriented semantic service discovery. In: *The Semantic Web: Research and Applications, Lecture Notes in Comp. Sci.* LNCS 2005, Vol. 3532, Springer, pp. 17–31.
41. Altintas I, Berkley C, Jaeger E, *et al.* Kepler: an extensible system for design and execution of scientific workflows. In: *Proceedings 16th International Conference on Scientific and Statistical Database Management, 2004*. June 21–23, 2004, pp. 423–4. Santorini Island, Greece.
42. Tang F, Chua CL, Ho LY, *et al.* Wildfire: distributed, grid-enabled workflow construction and execution. *BMC Bioinformatics* 2005;**6**:69.
43. Carver T, Bleasby A. The design of Jemboss: a graphical user interface to EMBOSS. *Bioinformatics* 2003;**19**:1837–43.
44. Leo P, Marinelli C, Pappadà G, *et al.* BioWBI: an integrated tool for building and executing bioinformatic analysis workflows. In: *Proceedings of the 1st Annual Meeting of the Bioinformatics Italian Society (BITS2004)*, March 26–27, 2004, Padua, Italy, abstract 25.
45. Taylor I, Wang I, Shields M, *et al.* Distributed computing with Triana on the grid. concurrency and computation. *Practice and Experience* 2005;**17**:1197–14.
46. Churches D, Gombas G, Harrison A, *et al.* Programming scientific and distributed workflow with triana services. *Concurrency and Computation: Practice and Experience* (Special Issue: Workflow in Grid Systems), 2006;**18**:1021–37.
47. Ben Miled Z, Gao N, Bukhres O, *et al.* SIBIOS: A system for the integration of bioinformatics services. In: *Second International Workshop on Challenges of Large Applications in Distributed Environments (CLADE) 2004*, June 7, 2004, Honolulu, Hawaii.
48. Deelman E, Singha G, Sua M-H, *et al.* Pegasus: a framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming* 2005;**13**:219–37.
49. Shah SP, He DY, Sawkins JN, *et al.* Pegasys: software for executing and integrating analyses of biological sequences. *BMC Bioinformatics* 2004;**5**:40.
50. Bartocci E, Corradini F, Merelli E, *et al.* BioWMS: a web based workflow management system for bioinformatics. *BMC Bioinformatics* 2007;**8**:S2.
51. Navas-Delgado I, del Mar Rojano-Munoz M, Ramirez S, *et al.* Intelligent client for integrating bioinformatics services. *Bioinformatics* 2006;**22**:106–11.
52. Radetzki U, Leser U, Schulze-Rauschenbach SC, *et al.* Adapters, shims, and glue – service interoperability for in silico experiments. *Bioinformatics* 2006;**22**:1137–43.