Panoply of Utilities in Taverna

K. Wolstencroft[1], T. Oinn[2], C. Goble[1], J. Ferris[3], C. Wroe[1],
P. Lord[1], K. Glover[4], R. Stevens[1]

[1]School of Computer Science, Kilburn Building, University of Manchester, Manchester
[2]EBI, Wellcome Trust Genome Campus, Hinxton, Cambridge
[3]IT Innovation Centre, University of Southampton, Southampton
[4]School of Computer Science and Information Technology, University of Nottingham, Nottingham
Contact: kwolstencroft@cs.man.ac.uk

Abstract

The Taverna e-Science Workbench is a central component of myGrid, a loosely coupled suite of middleware services designed to support in silico experiments in biology. Taverna enables the construction and enactment of complex workflows over resources on local and remote machines, allowing the automation of otherwise labour-intensive multi-step bioinformatics tasks. As the Taverna user community has grown, so has the demand for new features and additions. This paper outlines the functional requirements that have become apparent over the last year of working with domain scientists, along with the solutions implemented in both the Taverna workbench and the Freefluo enactment engine to address concerns relating to workflow construction and enactment, respectively.

1. Introduction

In silico experiments in the life sciences domain often involve chaining together disparate analysis tools and database resources. This has been achieved either by manual cutting and pasting between web pages, or by writing bespoke programmes, which chain resources together. These methods have been necessary because of the way in which the bioinformatics community has developed. Autonomous research groups produce and maintain different databases and develop associated algorithms and analysis tools. The number of these resources and the volume of publicly available data have dramatically increased in the post-genome era. These factors make integration a major concern in the community. However, there are few common data standards and the legacy of heterogeneity and geographical separation remains a problem.

The *ad hoc* nature of development additionally means that the scientific process of recording methods and results, can be overlooked, reducing the possibility of reproducibility and collaboration. To address these problems, myGrid provides a toolkit for designing, executing and managing *in silico* biological workflow experiments, from the users desktop, over distributed computational resources.

myGrid supports the e-science life cycle. Workflow experiments can be designed and executed; monitored and recorded; and shared and re-used. Some of the myGrid components responsible for the different stages in the life cycle are: Taverna and Freefluo for designing and enacting workflows, KAVE for recording experiment provenance, BioNanny for monitoring service use and performance, and Feta for discovering services.

myGrid is accessed by most users through the Taverna workbench. In this paper, we describe the developments in myGrid from the users' perspective, focusing on the Taverna component and the design and enactment of myGrid workflows.

1.1 myGrid Development and Case Studies

From the outset, myGrid has taken a user-driven approach to development, creating close ties with the community. Several projects were adopted as case studies. For example, the analysis of genetic data in the mapping of gapped regions associated with Williams-Beuren syndrome (WBS) [1] and the identification of genes involved in Graves Disease (GD) [2].

WBS is a sporadic microdeletion disorder in human caused by the deletion of approximately 1.5 Mb (encompassing 24 genes) from chromosome 7q11.23. Flanking the deletion are highly repetitive regions of DNA;

this makes sequencing difficult. Consequently, there are gaps in this region of the human genome map, which may contain further deleted genes contributing to the complex WBS phenotype. We used myGrid workflows to find new sequences that would close the genomic gaps and characterise any genes or regulatory elements discovered therein.

GD is a multigenic autoimmune disease of the thyroid. The immune system attacks cells in the thyroid gland resulting in hyperthyroidism. The aim of the project was to identify and characterise genes in chromosomal regions showing linkage to GD. After the initial identification of 50 candidate genes from microarray analyses, we used myGrid workflows to query public resources for SNPs associated with each gene, and information about each gene product, its function and functional domains.

These case studies quickly revealed that e-Science workflows could be complex artefacts. Not only did each study use varied types and numbers of services in their workflows, but the process flows required complex invocation patterns (for example, iteration over sets of data). These factors, together with the increase in the number of available web services, have presented challenges during the development of Taverna. This paper focuses on the myGrid workflow component, Taverna and describes recent developments and innovations that have evolved to meet these user requirements. We discuss how Taverna supports service discovery, complex invocation patterns, workflow re-use and interaction with diverse resource types, from standard SOAP web services to customised JDBC connections.

Finally, we will discuss the challenges presented as the Taverna user base extends to other scientific communities, which will present their own challenges and drive development into the future.

2. Supporting Service Description and Discovery

Many of the requirements for more traditional Grid applications have stemmed from the physics community, and, consequently, technologies have focused on harnessing high performance computing and networking power to enable the processing of complex equations and algorithms over large numeric data sets. In contrast, bioinformatics data are relatively small, but highly distributed and heterogeneous at many levels: differing syntax; differing storage paradigms; and semantic heterogeneities at schema and content levels.

myGrid provides a service-oriented architecture to access and integrate these heterogeneous biological data resources. There are now over 1000 services available to Taverna and this number is continuing to grow. While this provides a rich, integrated resource for the user, discovering appropriate services has become increasingly difficult. In addition, the majority of services are owned and developed by third-parties. There are no standards enforced regarding formatting of data or input parameters, and the user documentation available for each service is variable. To address this issue, myGrid developed the Feta semantic discovery component. Feta is a service, called the Feta Engine, with a client plugin to Taverna that supports the discovery of web services based on descriptions of the functions they perform [3].

The Feta data model uses bioinformatics domain information captured in the myGrid service ontology [4], which provides descriptions of bioinformatics tasks, data types and data sources. Describing services with these properties enables users to search, for example, for: all services that perform a multiple sequence alignment; all services that require a protein sequence as input; all services that search the UniProt database [5]. Knowing not only the function of the service, but the data it accesses and the format and type of input it is expecting, or output it produces aides the user in both service discovery and designing service interaction in workflow design.

Services are described using the service ontology and, optionally stored in a UDDI service registry [6]. The Feta engine enables the querying of this registry for services matching functional descriptions. Suitable services can then be easily added to a workflow with the standard drag and drop metaphor used elsewhere in the workbench. While, there is redundancy in the types of services available, the number of any particular type (and therefore any particular result) is presently manageable by manual selection.

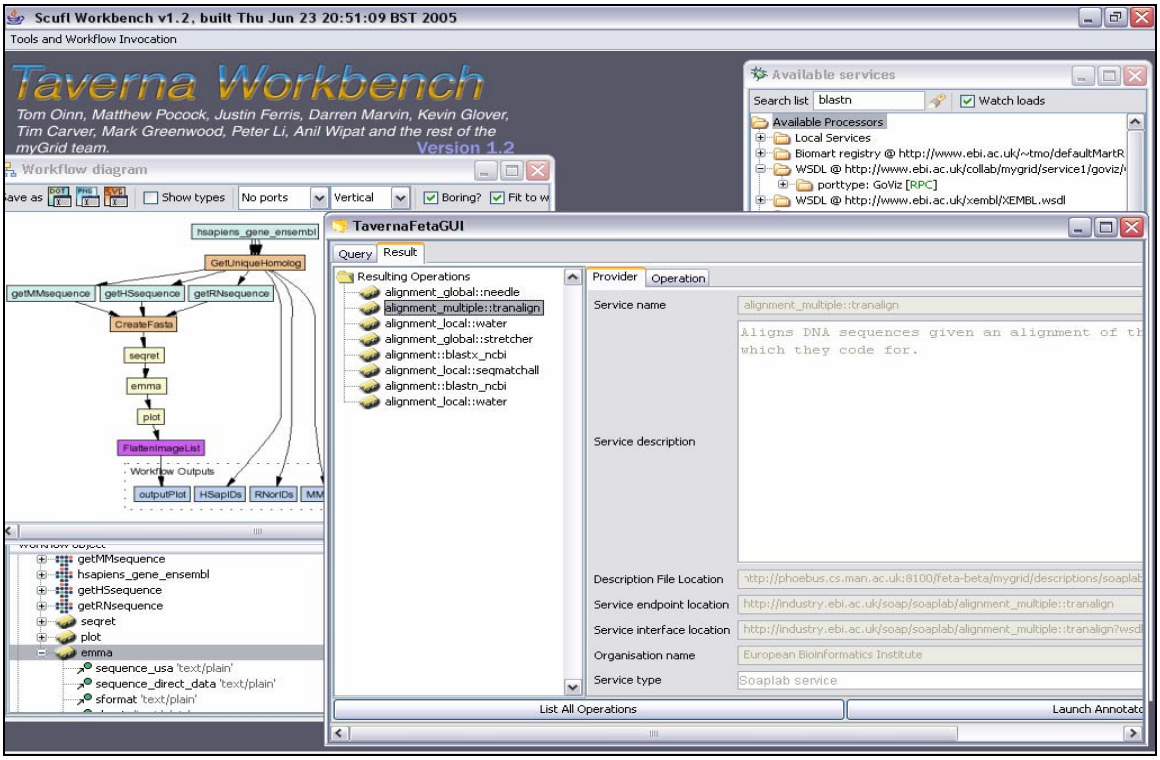

Figure 1. The Feta semantic discovery tool identifying alternative services for the 'emma' multiple alignment service. The query was for services that perform alignment tasks. Discovered services are presented with their operation definitions from the myGrid services ontology along with the location and provider of the service and an optional free-text description.

The Feta semantic discovery tool enables biologists with little bioinformatics experience to find services and use them in workflow design. It also provides a way for the more experienced bioinformatician to discover alternative, equivalent services from the ones they normally use. Figure 1 demonstrates the use of Feta to discover alternative sequence alignment services. While Feta can potentially be used to discover services from any source, certain libraries of services provide their own service metadata that is useful to the discovery process (i.e. Soaplab [7] and BioMoby [8]). This metadata provides a useful skeleton for Feta descriptions, enabling consistency and conformation to the Feta data model.

In order to perform semantic searches of service functionality, they have to be adequately and accurately described. This annotation is a relatively high cost, requiring human effort. Service providers who include service metadata help expedite this process, but as the majority of services are not owned or developed by myGrid, many provide little or no functional descriptions. We use the Pedro [9] annotation tool to encourage service providers or myGrid users to annotate such services. Pedro provides a service annotation GUI that presents annotation choices to the user from the underlying myGrid service ontology. Pedro generates service descriptions which conform to the Feta data model, which can be published in the registry and searched over using the Feta Engine.

Currently, results of Feta queries are not ranked. In the future, it is possible that Feta and another myGrid component, BioNanny, will interact to allow ranking by service performance. BioNanny can monitor the usage of web services, the number of times a service is invoked as well as the length of time it takes to return a response.

3. Supporting Complex Invocation Patterns

A major requirement for Taverna was to model, as a workflow, how users naturally deal with web-based resources. Workflows can be considered to be a series of inter-related components connected by data flow (the output of one component forming the input for the next) or control flow (setting the conditions for execution of the next component). When users

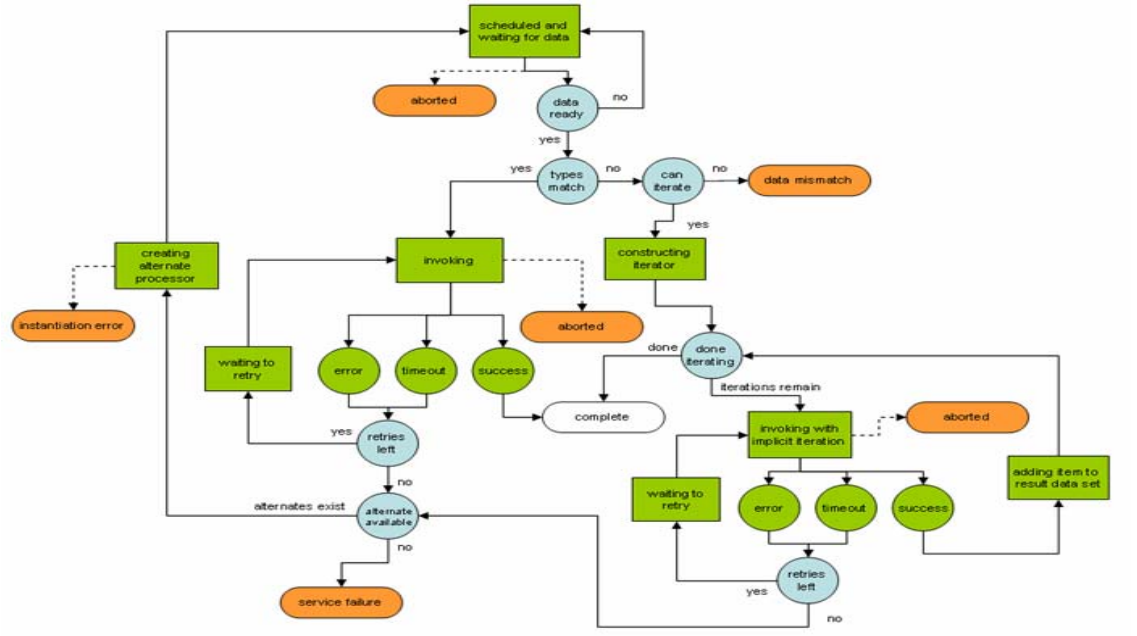

Figure 2. The Taverna state machine governing the processes of fault tolerance and iteration. The diagram details the processes of invoking services, retrying in the event of failure, and eventually aborting invocation or substituting services when a particular service is unavailable.

design workflows, they may wish to invoke a service, possibly repeatedly, or try an alternative service, or in some cases, ignore an unavailable service. Users may also wish to invoke the same workflow, or workflow fragment on each element in a list of input data.

In traditional procedural languages, these requirements would be fulfilled by the use of conditional or looping constructs. Our experiences suggest that, within the context of the higher level workflow design view, these constructs can produce counterintuitive behaviour. To avoid this necessity, we have therefore introduced additional semantics affecting the invocation of atomic processes within the workflow. In this section we illustrate some of these semantics.

3.1 Iterative Semantics

The desire to iterate over sets of data using the same service has been shown to be a common requirement when designing biological workflows. This should happen without direct user intervention. Therefore, the Freefluo enactment engine includes a mechanism to introspect over input data to a particular operation. When necessary, the workflow engine is able to split these data into a set of items that match those the operation is expecting. By default, this set is then iterated over, possibly in parallel. The user can, however, describe more complex configuration, such as generating the cross product of two sets of data.

Progress of iteration through a set is reported as the number of set elements that have been passed to the operation. Output from the iteration is then merged into a result set. This appears to be intuitive to our users, producing the expected behaviour in most cases.

3.2 Fault Tolerance

As Taverna operates over distributed services that are often not owned by myGrid, we do not have any control over the reliability of these services. Therefore, our users require mechanisms to deal with situations where services become unavailable, or perhaps temporarily overloaded.

Every operation in a Taverna workflow has fault tolerance settings. In the event of a service failure, the user can specify the number of times the service should be retried, the length of time between the retries, or an alternative service. If

all of these strategies fail, the user can specify the importance of the operation to the workflow as a whole. A failure in a *critical* operation will result in the entire workflow being aborted, otherwise the workflow will continue, but downstream processes will not be invoked. Figure 2 shows the Taverna state machine, which controls the iteration and fault tolerance processes.

3.3 Processor Invocation Co-ordination

When interacting, for example, with stateful web services, users may need to co-ordinate operations, to prevent the invocation of one until the completion of another.

There is already an implicit co-ordination between operations connected by data flow; however, Freefluo provides a mechanism to specify co-ordination between operations where there is no direct data transfer.

4. Supporting Interaction With Diverse Resource Types

Although much effort is being invested in standard SOAP based web services, we have found that our users' requirements do not always fit well with the web service model. There are several reasons for this

1. Technical limitations with the specification
2. The available toolkits
3. A lack of control over the required resources

There are several strategies that have been developed in Taverna and Freefluo to circumvent these problems. These include both the incorporation of non-standard web services; and the development of the API consumer, a tool to allow the invocation of arbitrary Java methods as part of a workflow.

The Styx Grid Service protocol [10] is an example of a non-standard web service; a plugin for its use is available in Taverna. Data can be streamed directly between these services without having to pass through the enactment engine, providing a clear optimisation in some circumstances, e.g. for very large data sets. This is particularly true where several linked services exist in close network proximity or a downstream service is capable of operating on partial data.

The Biomart [11] data warehouse system has also been incorporated into Taverna, bypassing the use of web service technology. Access is provided through direct JDBC connections to a central database. A Taverna plugin can extract the database metadata and present to the user a derived query generator interface. The Freefluo enactor has been extended to send the configured query and parse the returned results. This has given our bioinformatics users access to several critical genomic / proteomic data resources including Ensembl [12] and UniProt [13].

Finally, the API consumer tool has allowed the transparent integration of sizable third party Java libraries into Taverna, such as the JUMBO chemoinformatics library [14] and the caBIO cancer object model [15]. Table 1 describes the different types of service or service libraries accessible in Taverna. These include standard SOAP based web services as well as the non-standard additions our user community required.

Type	Description
BioMoby	Service based on BioMoby
Local Java	Local operation coded as a Java class, used for common or particularly generic functionality.
SeqHound	The SeqHound [16] library of web services.
Soaplab	Legacy command line applications, generally written in PERL, wrapped for use as web services. The EMBOSS [17] toolkit is an example
Workflow	A nested workflow exposed as a single operation.
Web Service	A standard SOAP service.
Beanshell	Scripting editor embedded in Taverna to enable users to script small transformation, or shim services
Biomart	Configurable query over the Biomart data warehouse.
Styx Grid Service	A service allowing data streaming directly between services.
API Consumer	Allows import of Java APIs as workflow components, processors correspond to constructors and static or instance methods.

Table 1. The different types of web services available in the Taverna workbench.

5. Supporting Collaboration and Reuse

Scientific workflows are often complex entities involving the integration of diverse resources and resource types with intricate invocation patterns. Consequently, the development of workflows is often a time consuming process involving many iterations as understanding grows of how best to address a task using the services available [18].

As such, individual workflows or components of workflows are potentially valuable artefacts that can be usefully shared amongst users. For example, different scientific workflows may be developed to address completely different biological hypotheses, but the underlying bioinformatics processes required could involve "standard" analyses, and therefore require similar services in similar orders.

For example, in the first workflow of the Williams-Beuren syndrome study, genomic sequence is masked for repetitive regions before being analysed by several gene prediction algorithms. This sequence of bioinformatics processes is not unique to WBS; it is a standard procedure for gene prediction analysis. Capturing this process in a workflow, however, means that other workflows can re-use this procedure. This occurred during an investigation into sleeping sickness in cattle.

This form of reuse is supported in Taverna. The workbench allows the import of workflows as service libraries. This allows sophisticated polymorphic services to be transparently shared amongst a collaborative group. Alternatively, imported workflows can be edited to allow repurposing for related experiments. Operations over Biomart services illustrate this as they require extensive configuration before use. Users can locate, describe and reuse pre-configured services from other workflows.

6. Discussion

Taverna is now a well established and widely used tool for e-science experiments. The early success of the WBS and GD projects clearly demonstrated its use in bioinformatics analyses, and the subsequent growth in the user community has produced many more success stories – new e-Science projects, such as, ISPIDER [19] and PsyGrid [20], are adopting Taverna and other myGrid components as a platform for development. Existing myGrid workflows are also being re-used.

The ease of use of Taverna is one of its greatest advantages, and the development of new features must always take account of this. Bioinformatics is an open access community. Data and the applications to manipulate it are open to all. The ability to efficiently integrate these disparate resources is consequently a powerful advantage and one which Taverna fully supports. The distributed computing nature of web service invocation also means that users can potentially access supercomputing resources from their desktops. The high level conceptual design process for constructing workflows in Taverna shields the user from the complexities of invoking services and dealing with data flow. Discovery and selection of services is aided by the use of Feta.

The experiences and feedback of the growing user community, in bioinformatics has accelerated the ongoing development of Taverna. New features and functionality are being continually added to increase the utility and user base of the biological e-science workbench.

The Taverna user community is now growing and diversifying to include researchers from outside bioinformatics, principally in the areas of chemoinformatics, health informatics, medical imaging and systems biology. Each variant of *in silico* science brings individual requirements, even though the core of workflow based experiments is very similar. For health informatics and medical imaging, for example, the necessity for patient confidentiality and auditing of access and analysis of data is vital, producing stringent requirements for provenance capture and increased security. Extending Taverna to support each of these research domains will need a clear understanding of these requirements, which we hope will be enabled by a continued user-focused approach to development.

Acknowledgements

The authors would like to acknowledge the assistance of the whole myGrid consortium. This work is supported by the UK e-Science programme EPSRC grant GR/R67743.

References

[1]. R. Stevens, H.J. Tipney, C. Wroe, T. Oinn, M. Senger, P. Lord, C.A. Goble, A. Brass and M. Tassabehji Exploring Williams-Beuren Syndrome Using myGrid *in Proceedings of 12th International Conference on Intelligent Systems in*

Molecular Biology, 31st Jul-4th Aug 2004, Glasgow, UK, published *Bioinformatics* Vol. 20 Suppl. 1 2004, i303-i310

[2]. Oinn T, Addis M, Ferris M, Marvin D, Senger M, Greenwood M, Carver T, Glover K, Pocock M, Wipat A and Li P. Taverna: A tool for the composition and enactment of bioinformatics workflows *Bioinformatics* Vol. 20(17) pp 3045-3054, 2004, doi:10.1093/bioinformatics/bth361

[3]. Lord, P. Alper, C. Wroe, and C. Goble. Feta: A light-weight architecture for user oriented semantic service discovery. In A. Gómez-Pérez and J. Euzenat, editors, *European Semantic Web Conference*, pages 17-31. Springer-Verlag, 2005

[4]. Chris Wroe, Robert Stevens, Carole Goble, Angus Roberts, and Mark Greenwood. Asuite of DAML+OIL Ontologies to Describe Bioinformatics Web Services and Data. *The International Journal of Cooperative Information Systems*, 12(2):597-624, 2003

[5]. Bairoch A., Boeckmann B., Ferro S., Gasteiger E.Swiss-Prot: Juggling between evolution and stability *Briefings in Bioinformatics*. 5:39-55(2004).

[6]. Tom Bellwood. UDDI Version 2.04 API Specification.UDDI Committee Specification, OASIS, July 2002

[7]. Senger M, Rice P, Oinn T (2003) Proceedings of UK e-science All Hands Meeting 2-4 September 2003

[8]. Wilkinson MD, Link M (2002) BioMoby: an open-source biological web services proposal. Briefings in Bioinformatics 3:4 331-341

[9]. Taylor CF, Paton NW, Garwood KL, Kirby PD, Stead DA, Yin Z, Deutsch EW, Selway L, Walker J, Riba-Garcia I, Mohamed S, Deery MJ, Howard JA, Dunkley T, Aebersold R, Kell DB, Liley KS, Roepstorff P, Yates JR, Brass A, Brown AJP, Cash P, Gaskell SJ, Hubbard SJ, Oliver SG (2003) A systematic approach to modelling, capturing and disseminating proteomics experimental data. *Nature Biotechnology* 21(3), 247-254

[10]. Rob Pike and Dennis M. Ritchie. The Styx- Architecture for distributed systems. Bell Labs Technical Journal, 4(2):146{152, 1999.

[11]. Pruess M, Kersey P, Apweiler R. The Intergr8 project - a resource for genomic and proteomic data 2004 In Silico Biol 22:5(2):0017

[12]. E Hubbard *et al* Ensembl 2005. *Nucleic Acids Res*. 2005 Jan 1;33(Database issue):D447-53

[13]. Bairoch A, Apweiler R, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N, Yeh LS. The Universal Protein Resource (UniProt). *Nucleic Acids Res.* 2005 Jan 1; 33(Database issue):D154-9.

[14]. Yong Zhang, Peter Murray-Rust, Martin T Dove, Robert C Glen, Henry S Rzepa, Joe A Townsend, Simon Tyrrell, Jon Wakelin, Egon L Willighagen. JUMBO – An XML Infrastructure for e-science, *All Hands Meeting*, Nottingham, 2004

[15]. Kraj P, McIndoe RA caBIONet--A .NET wrapper to access and process genomic data stored at the National Cancer Institute's Center for Bioinformatics databases *Bioinformatics*. 15;21(16):3456-8

[16]. E Hubbard *et al* Ensembl 2005. *Nucleic Acids Res*. 2005 Jan 1;33(Database issue):D447-53

[17]. Bairoch A, Apweiler R, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N, Yeh LS. The Universal Protein Resource (UniProt). *Nucleic Acids Res.* 2005 Jan 1; 33(Database issue):D154-9.

[18]. Michalickova K, Bader GD, Dumontier M, Lieu H, Betel D, Isserlin R, Hogue CW. SeqHound: biological sequence and structure database as a platform for bioinformatics research BMC *Bioinformatics* 2002 3:32

[19]. Rice P, Longden I, Bleasby A EMBOSS: the European Molecular Biology Open Software Suite *Trends Genet*. 2000 16(6):276-7

[20]. Goderis A, Goble G, Sattler U, Lord P, Seven Bottlenecks to Workflow Reuse and Repurposing 2005 I*nternational Semantic Web Conference,* accepted for publication

[21]. http://www.ispider.man.ac.uk/

[22]. http://www.psygrid.org