

Association of variations in I kappa B-epsilon with Graves' disease using classical and ^{my}Grid methodologies

Peter Li¹, Keith Hayward¹, Claire Jennings², Kate Owen², Tom Oinn³, Robert Stevens⁴, Simon Pearce² and Anil Wipat¹

¹School of Computing Science, University of Newcastle upon Tyne, NE1 7RU, ²Institute of Human Genetics, University of Newcastle upon Tyne, International Centre for Life, NE1 3BZ, ³European Bioinformatics Institute, Hinxton, Cambridge, CB10 1SD, ⁴Department of Computer Science, University of Manchester, M13 9PL.

Abstract

Bioinformatics experiments can be modelled as workflows whereby the order of each computational resource used has been pre-defined. Workflows in the ^{my}Grid project are composed and enacted using the Taverna workflow system. We have compared the use of Taverna with classical approaches for performing bioinformatics experiments in the genetic analysis of Graves' disease. Both classical and ^{my}Grid methodologies identified I kappa B-epsilon as a candidate gene involved in Graves' disease, demonstrating that ^{my}Grid is capable of producing the same results as the classical bioinformatics approach.

Introduction

Bioinformatics analyses are *in silico* experiments involving the use of local and remote resources to test a hypothesis, derive a summary or search for patterns (Stevens *et al.*, 2003a). These resources may be information repositories such as the EMBL (Kulikova *et al.*, 2004) and Swiss-Prot (Boeckmann *et al.*, 2003) databases, or computational analysis tools like BLAST (Altschul *et al.*, 1990) and ClustalW (Higgins *et al.*, 1994). The analysis performed in an *in silico* experiment frequently involves a combination of these resources that each perform a task. Each of these tasks are linked in a specific order to form a workflow process. For example, a workflow to investigate the evolutionary relationships between proteins might begin with acquiring amino acid sequences belonging to a protein family from Swiss-Prot and then applying the ClustalW algorithm to align and identify patterns between sequences.

Organisations have begun to provide programmatic access to bioinformatics information repositories and analysis tools based on Web Services (Stein, 2002), a new distributed computing architecture which uses existing Internet communication and data exchange standards (Booth *et al.*, 2003). Resources with Web Service access provide a web-based, published, application programming interface for interaction with other applications. Examples of bioinformatics Web Services include the XEMBL (Wang *et al.*, 2002), openBQS (Senger, 2002) and Soaplab analysis

services (Senger *et al.*, 2003) hosted by the European Bioinformatics Institute (EBI), the services provided by XML Central of DDBJ (Miyazaki and Sugawara, 2000), the KEGG API (Kawashima *et al.*, 2003) and a range of analysis services offered by the PathPort project (Eckart and Sobral, 2003).

The ^{my}Grid e-Science project aims to provide high-level, service-based middleware to support data-intensive *in silico* bioinformatics experiments using distributed resources (Goble *et al.*, 2003; Stevens *et al.*, 2003b). These bioinformatics analyses depend on a workflow system which can converse with the interfaces of Web Services and mediate how data flows between resources. This led to the inception of the Taverna project within ^{my}Grid which has developed an open source workflow tool enabling scientists to orchestrate bioinformatics Web Services and existing bioinformatics applications in workflows. We have used the Taverna workflow system to build and enact workflows which model the *in silico* analyses undertaken for the genetic analysis of Graves' disease (GD) (Imrie *et al.*, 2001), and compared the performance of this new methodology with the classical bioinformatics approach.

Taverna workflow system

The emphasis taken in the Taverna project has been to provide working tools for e-Scientists to perform their *in silico* experiments. The Taverna software is available as open source and can be downloaded at <http://taverna.sourceforge.net/>.

In Taverna, a workflow is considered to be a graph of processors, each of which transforms a set of data inputs into a set of data outputs. These workflows are represented in the Simple conceptual unified flow language (Scufl). Current languages were deemed unsuitable for composing scientific workflows since the existing standards are in flux, and high quality, free tools were not available to support standards (Oinn *et al.*, 2004). In addition, Web Service standards do not have the levels of user abstraction necessary for most bioinformaticians and do not offer support for the specification of data, processes or resources at a semantic level. These requirements led to the specification of Scufl which is a high-level, XML-based, conceptual language where each processing step of the workflow represents one atomic task. A workflow in the Scufl language consists of three main entities:

1. Processors

A processor is a transformation that accepts a set of input data and produces a set of output data (Fig. 1B). Processors have a name within the Scufl model and a set of both input and output ports. During the execution of a workflow, each processor has a current execution status which is one of initialising, waiting, running, complete, failed or aborted. The main processor types currently available are:

Arbitrary WSDL type: This type of processor allows a single call on a Web Service operation.

Soaplab type: This processor type calls a complete invocation of a Soaplab service as one unit. Soaplab services are Web Services which have been created using Soaplab, a tool which can wrap command-line programs with a Web Service interface based on a description of the analysis tool to be deployed (Senger *et al.*, 2003). Tools which have been wrapped by Soaplab include European Molecular Biology Open Software Suite (EMBOSS) programs (Rice *et al.*, 2000) and algorithms such as BLAST (Altschul *et al.*, 1990).

Nested workflow type: A processor of this type can invoke another child workflow. Currently, only child workflows in Scufl are supported.

String constant type: This type of processor has a single output port on which it returns a constant string value. This processor is of particular use when another processor in the same workflow requires a default value which acts as a parameter. Another use of this

processor is the replacement of an input entity in test workflows.

Local processor type: This processor can be used to add new local functions which are coded as classes to comply with a simple Java interface. The local functions currently available for use in workflows are shown in Figure 1C.

A workflow can also possess input and output data entities. A workflow input can be considered to be a source processor which executes instantaneously and makes the input value available on its virtual output port. A workflow output can be considered as a sink processor which receives a value from its virtual input port but never actually executes. Both workflow sources and sinks can be annotated with metadata. Three types of metadata can be associated with workflow inputs and outputs: a MIME type, a semantic type based on the myGrid bioinformatics ontology (Wroe *et al.*, 2003) and a free textual description.

2. Data links

Data links mediate the flow of data between a data source and a data sink. The data source can be a processor output or a workflow input. The data sink can be a processor input port or a workflow output. Each data sink will receive the same value if there are multiple links from a data source.

3. Coordination constraints

A coordination constraint links two processors and controls their execution. This level of control is required when there is a process where the stages must execute in a certain order and yet there is no direct data dependency between them. For example, coordination constraints can be used to allow one processor to go from scheduled to running if another processor has status completed. In most cases, no concurrency constraints are required since data links will ensure that some processors stay in their waiting state until the data they require is available.

Scufl workbench

The Taverna tool contains an application called the Scufl workbench which enables bioinformaticians to write workflows without having to learn the Scufl language (Fig. 1). This application acts as a container for a number of user interface components which provide

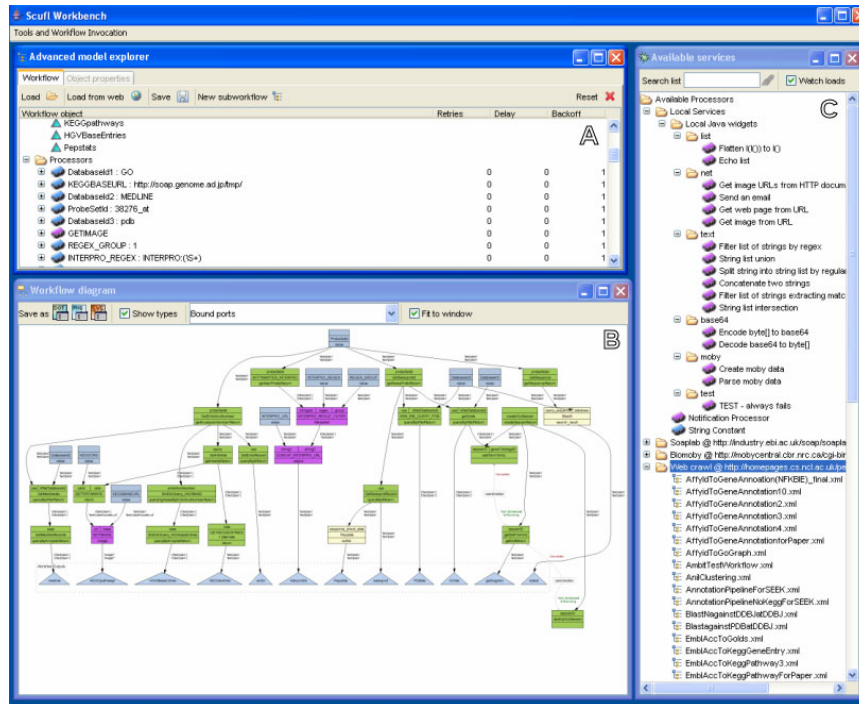


Figure 1. The ScufI workbench application - version beta10. The advanced model explorer is used to manipulate the workflow (A). At the top level are the different types of entities within a ScufI model; overall workflow inputs and outputs, processors, data links and coordination controls (A). The workflow is visualised using the workflow diagram window (B). Processors are displayed in different colours depending on their type. Web Service processors are shown as green boxes, Soaplab processors are yellow and local processors are purple. There is a range of display options supported by the graphical view. Users can view processors with all ports displayed, no ports or only those ports which are bound to data links (B).

read-only views and read-write controllers/views involved in the composition and enactment of ScufI workflows. The ScufI model explorer is a controller view that shows the state of the current model as a tree structure, and is also used for defining the flow of data between processors (Fig. 1A). The ScufI diagram view provides a graphical display of the current workflow (Fig. 1B). The graphical display of the workflow is read-only since only the ScufI model explorer is used to edit workflows.

The ScufI Workbench contains a service browser which provides a palette of processors (Fig. 1C). Context menus in the service panel allow new processors to be added to the current ScufI workflow model. There are two methods of populating the palette with services. Processors in the current ScufI model can be ‘scavenged’ which involves extracting the set of processors contained within the model and adding them to the service palette. The palette can also be populated from the Web

using scavengers for each processor type. Each scavenger requires a URL which, when pointed to a directory, will perform a naïve search to find files which it can process. The service browser can also be populated with workflows when directed with a URL to a directory of ScufI workflows (Fig. 1C).

Workflows can be executed in the ScufI Workbench using the enactor launch panel. This panel allows inputs to be specified for the workflow and launches a local instance of the Freefluo enactment engine which has also been developed by the ^{my}Grid project. Freefluo is a Java workflow orchestration tool for Web services which supports a subset of the Web Services Flow Language as well as ScufI (Addis *et al.*, 2003). This flexibility of Freefluo is provided at its core by a reusable orchestration framework that is not tied to any workflow language or execution architecture. The enactor core supports an object model of a workflow in the form of a directed graph where each node has a state machine that defines its lifecycle.

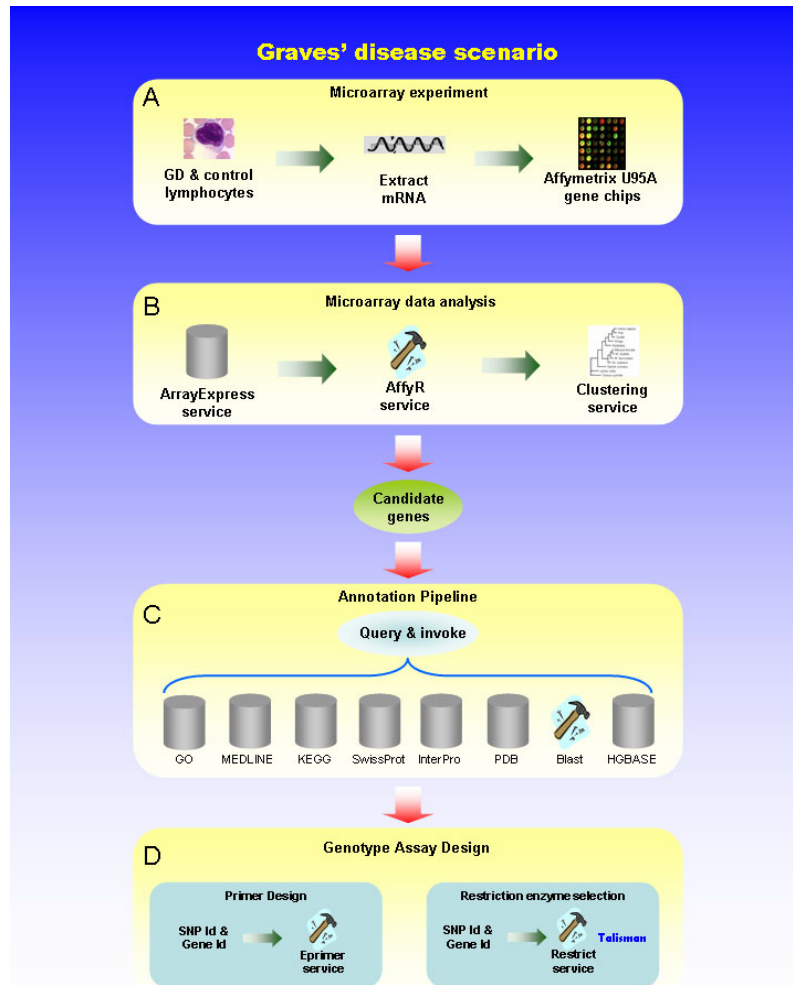


Figure 2. The Graves' disease scenario. The scenario begins with a microarray analysis experiment (A) which is followed by three workflows involving the analysis of microarray data (B), the retrieval of annotations (C) and the design of genotyping experiments (D).

Scheduling and state transitions are driven by message passing between nodes as the workflow progresses. The core of the enactor is decoupled from both the textual form of a workflow specification and the details of service invocation and data model allowing it to orchestrate workflows in a generic way. This flexibility is exploited when Taverna is extended to cope with a new processor type. The core of the enactor is unchanged but Freefluo is extended with a parser for the new XScufl processor and a plug-in for the required service invocation.

Bioinformatics experiments for Graves' disease

The genetic analysis of GD (Imrie *et al.*, 2001) is a real-life biological scenario which was used

by the Taverna and ^{my}Grid projects to test the extent of whether its tools could meet the requirements of the scientists. GD is an autoimmune disease primarily affecting the thyroid gland. In this condition, lymphocytes secrete auto-antibodies which bind to receptors on cells in the thyroid, resulting in hyperthyroidism. Symptoms of the disease include weight loss, trembling, muscle weakness, increased pulse rate, heat intolerance and exophthalmos.

The analysis of GD genetics involves the discovery of genes involved in the diseased state and the genotyping of single nucleotide polymorphisms (SNPs) which are nucleotide variations that occur in those genes. The analysis begins with a laboratory microarray experiment where the mRNA expression levels of over 10 000 genes in lymphocytes from GD

patients and healthy controls were measured using Affymetrix HG-U95A gene chips (Fig. 2A). Three workflows were designed, composed and enacted as bioinformatics experiments for the analysis of the microarray data. Each workflow corresponds to a distinct phase in the classical *in silico* process normally carried out using a number of web based resources, and has a specific function:

1. The first workflow is concerned with the analysis of the microarray data to generate a list of candidate genes which are differentially-expressed in GD and in healthy individuals (Fig. 2B).
2. An annotation pipeline workflow allows the retrieval of annotated information for each gene in the list including its location in the genome, function, other similar genes, information about the gene from the scientific literature and the SNPs identified in the gene. In addition, it analyses the composition and structure of the protein encoded by the gene and also the pathways that the protein participates in (Fig. 2C).
3. A final workflow is required to help design the wet-lab experiments to test the hypothesis generated by the preceding workflows. This workflow aids in primer design and identifies restriction sites for use in Restriction Fragment Length Polymorphism (RFLP) experiments to genotype control and GD patient lymphocyte DNA samples for a given SNP (Fig. 2D).

Classical *in silico* approach

The classical approach to analysing microarray data involved using the Affymetrix data mining software to identify genes differentially-expressed between controls and GD patients. The information required to make a judgement about its candidacy in GD was then collected by visiting the web sites of a number of bioinformatics databases and tools. This was a time consuming and laborious procedure, requiring copying and pasting queries or results of one tool as input on web forms and then repeating the same procedure for each gene.

Once a gene with a potential role in GD had been selected based on the collected information and the experience of the biologist, the SNPs within the gene are found by querying SNP databases such as dbSNP, HGVBASE and HAPMAP, again by using their forms on their web-based interfaces. Confirmation that these SNPs are associated with GD was obtained

from RFLP experiments to determine the genotypes of the SNPs in the patient and control groups. The design of the RFLP experiment involved visiting websites to find the enzyme restriction sites that cut around the SNP whilst primers to amplify the region of DNA flanking the SNP were designed manually. A gene called I kappa B-epsilon was found to be differentially-expressed between control and GD patients. The I kappa B-epsilon gene also maps to an area with GD linkage (6p21.1) suggesting that it is a good functional candidate involved in GD. Preliminary evidence for allelic association at a SNP marker in the 3' untranslated region of the gene was also found when SNPs in I kappa B-epsilon were genotyped.

^{my}Grid approach

The ^{my}Grid approach used the Scuffl workbench application in Taverna to build and enact workflows to perform bioinformatics experiments required by the GD scenario. Prior to the composition of workflows, any resources required by the GD scenario had to be 'wrapped' as a local or Web Service so that it can be consumed by the Taverna workflow system. Resources which were wrapped included the ArrayExpress microarray database (Brazma *et al.*, 2003) and the Affy R package from the Bioconductor project (Gautier *et al.*, 2004).

With a service palette populated with services (Fig. 1C), workflows were composed using the model explorer window of the Scuffl workbench. It was essential that the semantic details of the data inputs and outputs of the service operations were known since the Scuffl workbench does not support semantic typing during workflow composition at the present time. The graphical view of the workflow is used during workflow composition to determine how input data is transformed by each Taverna processor. The composition of workflows was found to be a collaborative process between the biologists and bioinformaticians and so being able to record and graphically visualise workflows were important features.

The first workflow involved analysis of genes based on their expression levels in the microarray data set (Fig. 2B). This involved retrieving microarray data from an ArrayExpress database Web service to be transformed into Treeview format using the AffyR service which was then followed by running a number of array analyses such as outlier analysis or clustering using a microarray

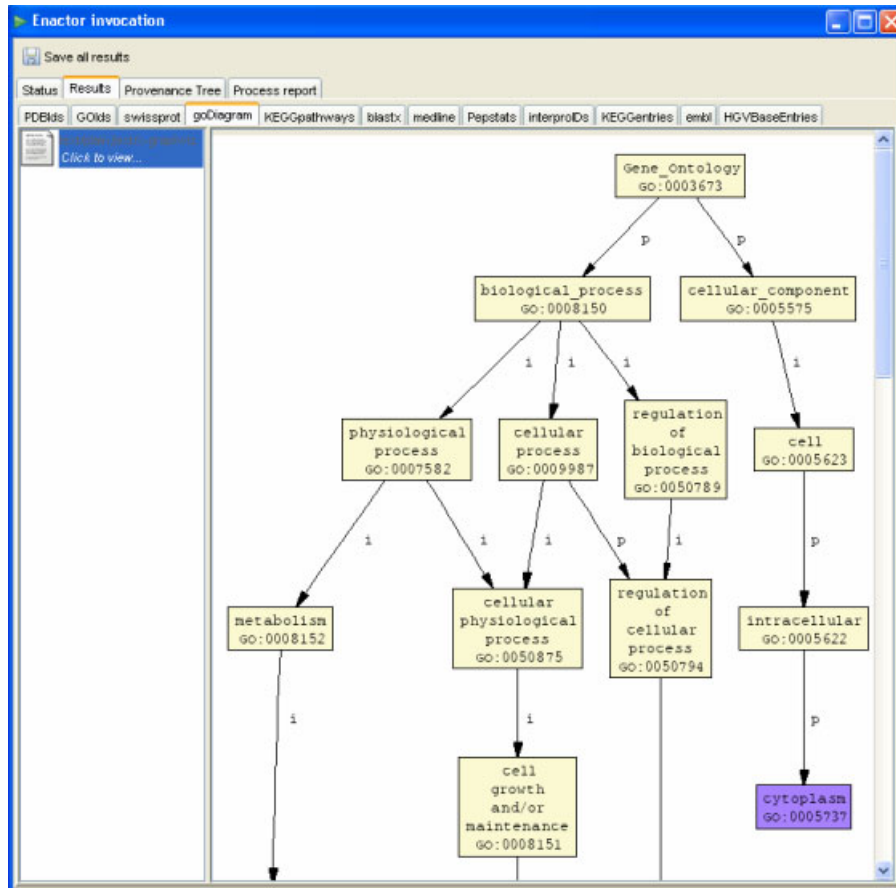


Figure 3. The enactor launch panel of the ScufI workbench which has completed the enactment of the annotation pipeline workflow of the Graves' disease scenario. The result item shown is the sub-graph of the Gene Ontology (GO) associated with the GO identifiers which have been assigned to the I kappa B-epsilon protein.

data analysis service hosted at Hong Kong University. Clustering analyses of microarray data can take several hours and notification of completion is received via email.

The second workflow is a gene annotation pipeline in which a gene is selected from the output of the microarray data analysis workflow, and annotation from database services such as Swiss-Prot, KEGG, GO and MEDLINE with analyses such as BLAST are returned (Figs. 2C and 3). In addition, information on SNPs identified in the gene is retrieved from the HGVBASE database (Fig. 2C). Finally, a genotype assay design workflow takes one or more of these SNPs as input to design primer sequences and selects restriction enzymes for RFLP experiments using the Eprimer and Restrict programs from EMBOSS which have been wrapped as SoapLab services (Fig. 2D).

Based on the results of the microarray data analysis and the information retrieved by the annotation pipeline workflows, I kappa B-

epsilon (Whiteside *et al.*, 1997) was selected as a candidate gene involved in GD. The annotation pipeline also found a SNP associated with the 3' untranslated region of the I kappa B-epsilon gene. The third workflow detected the same restriction site as used in the classical approach and calculated primers which can be used to validate the presence of the SNP using RFLP.

Discussion

When classical and ^{my}Grid approaches were compared for performing the above bioinformatics experiments in the genetic analysis of GD, both methodologies identified I kappa B-epsilon as a candidate gene in GD. That the same result was achieved by both methodologies shows that ^{my}Grid is capable of producing the same results as the classical *in silico* approach.

The automation of the bioinformatics experiments through the use of ^{my}Grid workflow

technology enabled analyses to be performed faster than when they are performed using the classical approach. This saving in time was gained through a number of features which are present in the Taverna workflow system. Perhaps the most time consuming aspect of performing bioinformatics analyses when using the conventional approach is having to manually cut and paste data between the web based interfaces of bioinformatics resources. Taverna avoids this since the flow of data between bioinformatics resources is defined during the composition of the workflow using the ScufI workbench application. The Freefluo enactor then handles the transfer of data between resources through their service interfaces. In addition, Taverna has iteration functionality which allows workflows to iterate over a number of items within a data set. Time is also saved if workflows have to be repeated. Since workflows can be stored as XScufI scripts, *in silico* experiments in ^{my}Grid are repeatable with different parameters for services if required.

The workflows used in the genetic analysis of GD are typical examples of bioinformatics workflows involving the querying of information repositories and the analysis of data using computational tools (Fig. 2). The workflows which have been used in the analysis of GD genetics are generic in that they can be applied to other scenarios which require similar types of workflows. For example, the annotation pipeline workflow will allow a disease that is the focus of another group of researchers to be substituted in place of GD in order to retrieve information about a gene or the encoded protein.

Web Services has been suggested as a framework for providing seamless access to bioinformatics resources in the life sciences community (Stein, 2002). To this end, there are free, open source tools which can be used to deploy Web Service interfaces for any bioinformatics resource such as Apache Axis (<http://ws.apache.org/axis/>) and Soaplab. (Senger *et al.*, 2003). However, the generation of Web Service interfaces was a significant obstacle for biologists since it requires programming knowledge and technical knowledge of the resource to be Web Service-enabled.

Using the ^{my}Grid approach, it is possible to increase the efficiency of biologists in their work by reducing the time spent in performing their bioinformatics analyses. The application of the GD scenario has provided a real life problem to show that ^{my}Grid and the Taverna workflow system can produce similar results

compared to the conventional bioinformatics approach. Work has now begun on using ^{my}Grid as part of the process in confirming the preliminary results using more complex clustering algorithms with more microarray data from a different cohort of patients.

References

Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.* 215, 403-410.

Addis M, Ferris J, Greenwood M, Li P, Marvin D, Oinn T. and Wipat A. (2003) Experiences with e-Science workflow specification and enactment in bioinformatics. *Proc UK e-Science All Hands Meeting 2003*, pp. 459-466.

Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, Gasteiger E, Martin MJ, Michoud K, O'Donovan C, Phan I, Pilbout S and Schneider M. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.* 31:365-370.

Booth D, Haas H, McCabe F, Newcomer E, Champion M, Ferris C and Orchard D. (2003) Web Services Architecture. W3C <http://www.w3.org/TR/ws-arch/>

Brazma A, Parkinson H, Sarkans U, Shojatalab M, Vilo J, Abeygunawardena N, Holloway E, Kapushesky M, Kemmeren P, Lara GG, Oezcimen A, Rocca-Serra P, Sansone SA. (2003) ArrayExpress: a public repository for microarray gene expression data at the EBI. *Nucleic Acids Res.* 31:68-71.

Eckart JD and Sobral BW. (2003) A life scientist's gateway to distributed data management and computing: the PathPort/ToolBus framework. *OMICS*, 7, 79-88.

Gautier L, Cope L, Bolstad BM, Irizarry RA. (2004) affy: analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics.* 20:307-15.

Goble CA, Pettifer S, Stevens R. and Greenhalgh C. (2003) Knowledge Integration: In silico Experiments in Bioinformatics, in *The Grid 2: Blueprint for a New Computing Infrastructure Second Edition* eds. Ian Foster and Carl Kesselman, November 2003.

Higgins D, Thompson J, Gibson T, Thompson JD, Higgins DG, Gibson TJ. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* 22:4673-4680.

Imrie H, Vaidya B, Perros P, Kelly WF, Toft AD, Young ET, Kendall-Taylor P, Pearce SH. (2001) Evidence for a Graves' disease susceptibility locus at chromosome Xp11 in a United Kingdom population. *J Clin Endocrinol Metab.* 86:626-30.

Kawashima S, Katayama T, Sato Y and Kanehisa M. (2003) KEGG API. <http://www.genome.ad.jp/kegg/soap/>

Kulikova T, Aldebert P, Althorpe N, Baker W, Bates K, Browne P, van den Broek A, Cochrane G, Duggan K, Eberhardt R, Faruque N, Garcia-Pastor M, Harte N, Kanz C, Leinonen R, Lin Q, Lombard V, Lopez R, Mancuso R, McHale M, Nardone F, Silventoinen V, Stoehr P, Stoesser G, Tuli MA, Tzouvara K, Vaughan R, Wu D, Zhu W and Apweiler R. (2004) The EMBL Nucleotide Sequence Database. *Nucleic Acids Res.* 32: D27-D30.

Miyazaki S and Sugawara H. (2000) Development of DDBJ-XML and its application to a database of cDNA. *Genome Informatics.* Universal Academy Press, Inc (Tokyo), pp. 380-381.

Oinn T, Addis M, Ferris J, Marvin D, Greenwood M, Carver T, Pocock M, Wipat A and Li P. (2004) Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics.* Accepted for publication.

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

Senger M. (2002) Bibliographic query service. <http://industry.ebi.ac.uk/openBQS/>

Senger M, Rice P and Oinn T. (2003) SoapLab - a unified Sesame door to analysis tools. *Proc UK e-Science All Hands Meeting 2003.*

Stein L. (2002) Creating a bioinformatics nation. *Nature*, 417, 119-120.

Stevens R, Glover K, Greenhalgh C, Jennings C, Pearce S, Li P, Radenkovic M. and Wipat A. (2003a) Performing in silico experiments on the Grid: a users perspective. *Proc UK e-Science All Hands Meeting 2003*, 43-50.

Stevens R, Robinson A, and Goble CA. (2003b) myGrid: Personalised Bioinformatics on the Information Grid in proceedings of 11th International Conference on Intelligent Systems in Molecular Biology, 29th June–3rd July 2003, Brisbane, Australia, published *Bioinformatics* Vol. 19 Suppl. 1 2003, i302

Wang L, Riethoven JJ and Robinson A. (2002) XEMBL: distributing EMBL data in XML format. *Bioinformatics*, 18, 1147-8.

Wroe C, Stevens R, Goble C, Roberts A and Greenwood M. (2003) A suite of DAML+OIL Ontologies to Describe Bioinformatics Web Services and Data. In *International Journal of Cooperative Information Systems* special issue on Bioinformatics. 12 (2), 197-224.

Whiteside ST, Epinat JC, Rice NR and Israël A. (1997) I kappa B epsilon, a novel member of the I kappa B family, controls RelA and cRel NF-kappa B activity. *EMBO J* 16:1413-26.