Combining RapidMiner operators with bioinformatics services – a powerful combination

Simon Jupp[1], James Eales[1], Simon Fischer[2], Sebastian Land[2]
Rishi Ramgolam[1], Alan Williams[1], Robert Stevens[1]

[1]School of Computer Science, University of Manchester, UK
[2]Rapid-I GmbH, Stockumer Str. 475, 44227 Dortmund, Germany

Abstract

Knowledge discovery through pattern finding in data is central to modern molecular biology, which now has thousands of databases and similar numbers of tools for processing those data. Any data analysis in molecular biology involves gathering and processing data from many sources, even before the analysis for the central biological question takes place. Taverna is a workflow workbench that allows bioinformaticians to create data pipelines involving distributed Web services and other forms of tool; these workflows gather and manage data in order to perform analyses that answer biological questions. RapidMiner brings a large suite of data processing, visualisation and data mining tools to bear upon tables of data, but there is a disconnect between these operators and the services available to users of Taverna. Through a RapidMiner extension to Taverna we have combined the ability to gather and process data from many molecular biological sources with RapidMiner's data mining capabilities to provide a powerful tool for scientific analysis. In this article we describe this RapidMiner extension to Taverna and some preliminary analyses we have performed using RapidMiner on biological data.

1 Introduction

Data in molecular biology are characterised by their complexity, volatility and, more recently, by their large volume. Bioinformatics specialists gather and process these data to find patterns that may form hypotheses for new laboratory experiments [1]. This places data mining activity at the centre of modern biology. Yet bringing classic data mining operators to bear upon biology's data is not as straight-forward as it might appear. As a discipline bioinformatics has over one thousand data resources [3]. It also has a similar number of tools that process these data in some way to find clauses within; the clauses represent some biological property or relationship with another entity. These tools and data resources are highly distributed and heterogeneous with respect to API, schema, vocabulary [4]. This means that a bioinformatician has to gather and organise these data before any analysis to address a particular biological question can take place. Such analyses can be performed in a variety of ways, one of which is through data mining [7].

Knowledge Discovery in Databases (KDD) is primarily about extracting interesting, nontrivial, implicit previously unknown and potentially useful information out of these data. Biology's data has various interesting challenges including high-dimensionality and low numbers of samples, especially in clinical situations. The data span many levels of granularity from the molecular analysis of genes, proteins, their regulation and other interactions such as in biochemical pathways, to larger scale experiments like clinical trials and drug discovery. Data mining examples in biology include finding sequence motifs, protein family classification, prediction of protein folding patterns, molecular structures, drug targets, whole genome arrays, building phylogenetic trees etc. [7]. Advances in data mining means we can begin to tackle these bio data.

As new biological data are generated the progression from raw data, through analysis, to hypothesis generation forms an important part of the modern scientific method [2]. It is becoming increasingly important to capture this process so results can be disseminated in public and scrutinised by peer review. The computational analysis involve steps that form workflows, where data is manipulated and transformed through a series of software tools, before the outputs are interpreted by the scientists. The software used can take many forms from small applications that can be run locally on users machines, to more complex services that require enormous amounts of resources from large distributed clusters of computers. Often service providers will provide programmatic access via web services that can then be used as components of the workflow. The distributed nature of these services has led to the development of applications that allow scientists to build these workflows, connecting a range of services to form an analysis pipeline; one such system is the Taverna workflow management system [6]. Taverna allows bioinformaticians to develop data gathering and analysis pipelines using distributed resources. Taverna has a pluggable architecture and can handle a range of service types including, WSDL operations, REST services, Beanshells and also contains access to several domain specific resources such as BioMart databases, BioMoby services and the ability to execute R scripts.

The release of the RapidAnalytics enterprise server from Rapid-I provides a platform for interacting with data mining operators from RapidMiner via Web services. In this paper we present a new plugin for Taverna so users can gain access to the wealth of operators available in RapidMiner. We begin by describing the Taverna workbench along with the architecture and functionality of the RapidMiner plugin. We demonstrate its use in some bioinformatics workflows that capture the data-gathering, pre-processing, data mining and evaluation stages of analysis.

1.1 Taverna

Taverna[1] is one of a family of tools that support e-Science:

- the Taverna Workflow Workbench in which users create and edit workflows and interface to the enactment of the workflows by

- the Taverna Engine (bundled with the Workbench or a Command Line Tool) that enacts the workflows and collects the provenance of workflow runs, and

- the Taverna Server that allows workflows to be enacted on remote machines.

Figure 1 shows the Taverna architecture illustrating the separation of the workbench and engine. As described, Taverna supports a broad variety of types of implementation of services; All of these Taverna *service types* are implemented as plugins into the basic architecture. There are also a set of useful “local services” built into Taverna for common operations such as reading a file or flattening a list of values. There have been many extensions made to Taverna, for example to include Chemistry Development Kit tools, to call services running on a grid and to interact with myExperiment.

The Workbench is the user interface where users can open, create and edit workflows. It is also used to run workflows and view the results of a run. Taverna does not have a fixed set of services; instead users are able to inform it of new services that they want to include in their workflows. For example, a user can import a WSDL definition into the Workbench, thus making the operations described in the file available as services within the service panel. The workbench allows users to form services into pipelines, connect ports, and specify data inputs and locations in which output data are gathered. It also allows the control of how the workflow is enacted and the storage of the data. Taverna also has a mechanism for recording the provenance of the workflow enactments on particular datasets.

[1] `http://www.taverna.org.uk`

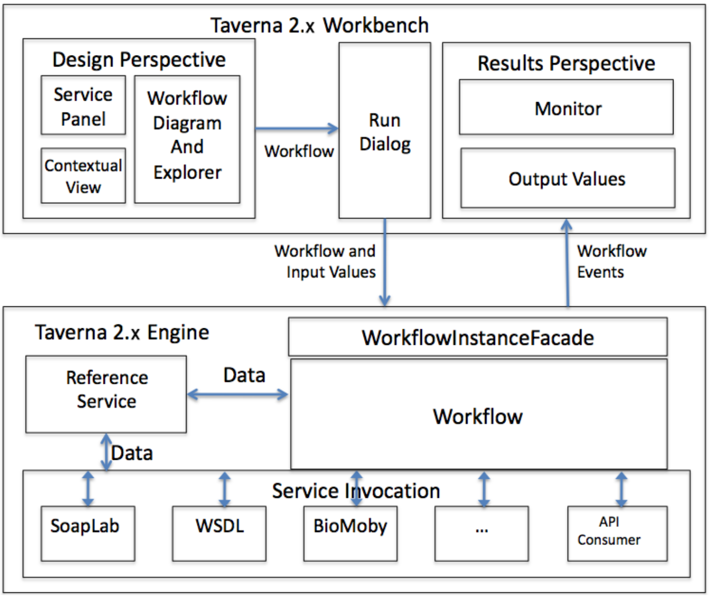


Figure 1: Taverna Architecture.

1.2 RapidMiner Web services

RapidAnalytics is a server environment where rapidMiner (RM) processes can be stored, shared and executed to generate dynamic reports and visualisations. RapidAnalytics exposes the operators within RapidMiner as web services, along with additional services for data management and meta-data generation. External applications can now utilise the services exposed by RapidAnalytics to bring RapidMiner functionality into new application domains.

One of the drawbacks when working with web services and large volumes of data is that much of the execution time is spent passing the data around between the services. RapidAnalytics addresses this issues by decoupling the upload and download of data from the actual execution of the operators, passing data to the operators only by reference. Thus, transmission of data is not necessary in cases where

- an operator execution fails and has to be restarted,
- an operator is executed repeatedly with different parameter settings, or
- a second operator is executed on the result of the first, but the intermediate result is of no interest to the client.

Altogether, in order to execute RapidMiner operators, the client will typically execute these steps:

- Upload data to the server.
- Select operator to apply and configure parameters and requirements.
- Invoke the operator on the uploaded data.
- Use the results:
  - download the results for inspection,
  - use these results as input to another operator invocation.

1.3 Taverna / RapidMiner plugin

RapidAnalytics offers several web services; for the Taverna plugin we currently focus on two of these:

- *RepositoryService*: The repository service can be used to upload and download data from a RapidAnalytics server as well as administration of the repository
- *ExecutionService*: The execution service is a single polymorphic web service that is used to execute a RapidMiner operator. As input it takes a reference to the input file locations along with a set of parameters including the named operator to execute

Taverna can easily import WSDL and REST based services, however, in the case of the execution service, its polymorphic functionality can make it difficult to work with in an environment like Taverna. To address this we developed a plugin extension to Taverna that presents the RapidMiner operators behind the web service in a similar way to how they are presented in RapidMiner 5. In addition we provide dialog based interactions for setting input file locations and operator invocation parameters. In addition we can expose the full operator tree in Taverna along with the appropriate input and output ports for each operator to the user (see Figure 2).

As previously stated, RapidAnalytics requires the data being processed to sit within the RapidAnalytics repository. This reduces the need to pass large amounts of data between services and improves the execution time on file transfer overheads associated with running distributed workflows. So before data can be analysed in Taverna, this data must be first uploaded onto the RapidAnalytics server. When any RapidMiner operator in Taverna is used the user is given a configuration dialog. From this dialog a user can access the RapidAnalytics server in order to browse or upload new data to the

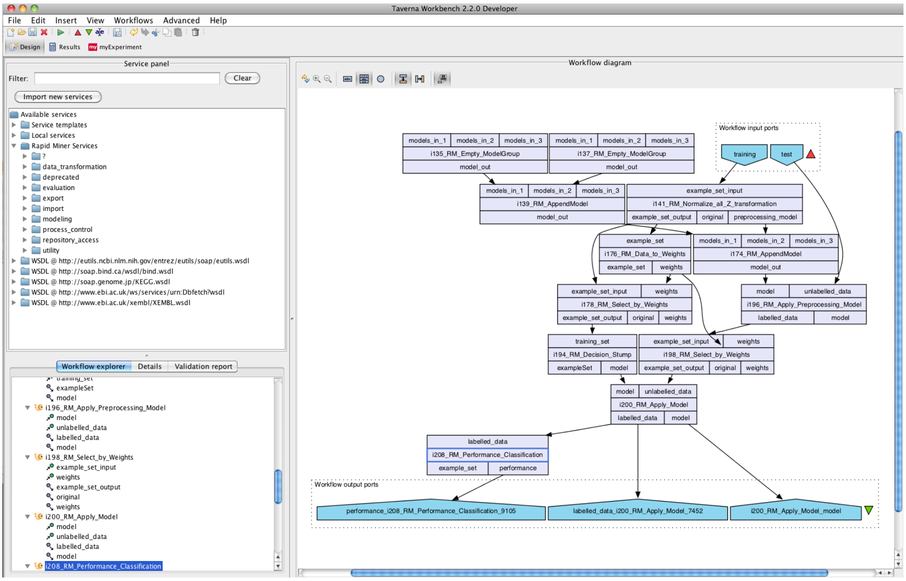

Taverna Workbench 2.2.0 Developer
File
Edit
Insert
View
Workflows
Advanced
Help
Design
Results
myExperiment
Service panel
Filter:
Clear
Import new services
Available services
Service templates
Local services
Rapid Miner Services
?
data_transformation
deprecated
evaluation
export
import
modeling
process_control
repository_access
utility
WSDL @ http://eutils.ncbi.nlm.nih.gov/entrez/eutils/soap/eutils.wsdl
WSDL @ http://soap.bind.ca/wsdl/bind.wsdl
WSDL @ http://soap.genome.jp/KEGG.wsdl
WSDL @ http://www.ebi.ac.uk/ws/services/urn:Dbfetch?wsdl
WSDL @ http://www.ebi.ac.uk/xembl/XEMBL.wsdl
Workflow explorer
Details
Validation report
training_set
exampleSet
model
i196_RM_Apply_Preprocessing_Model
model
unlabelled_data
labelled_data
model
i198_RM_Select_by_Weights
example_set_input
weights
example_set_output
original
weights
i200_RM_Apply_Model
model
unlabelled_data
labelled_data
model
i208_RM_Performance_Classification
Workflow diagram
Workflow input ports
training
test
models_in_1 | models_in_2 | models_in_3
i135_RM_Empty_ModelGroup
model_out
models_in_1 | models_in_2 | models_in_3
i137_RM_Empty_ModelGroup
model_out
models_in_1 | models_in_2 | models_in_3
i139_RM_AppendModel
model_out
example_set_input
i141_RM_Normalize_all_Z_transformation
example_set_output | original | preprocessing_model
example_set
i176_RM_Data_to_Weights
example_set | weights
models_in_1 | models_in_2 | models_in_3
i174_RM_AppendModel
model_out
example_set_input | weights
i178_RM_Select_by_Weights
example_set_output | original | weights
model | unlabelled_data
i196_RM_Apply_Preprocessing_Model
labelled_data | model
training_set
i194_RM_Decision_Stump
exampleSet | model
example_set_input | weights
i198_RM_Select_by_Weights
example_set_output | original | weights
model | unlabelled_data
i200_RM_Apply_Model
labelled_data | model
labelled_data
i208_RM_Performance_Classification
example_set | performance
Workflow output ports
performance_i208_RM_Performance_Classification_9105
labelled_data_i200_RM_Apply_Model_7452
i200_RM_Apply_Model_model

Figure 2: Taverna workbench showing RapidMiner operator panel on the left hand panel and a simple RapidMiner workflow on the main workflow canvas

repository. Once the input data is specified the user is offered the full set of available parameters for that operator in order to complete the configuration. RapidMiner operators are distinguished from other non-RapidMiner operators using a shade of purple on the main Taverna workflow canvas. Operators can be connected to create a workflow using a conventional drag and drop system between operator input and output ports, similar to the functionality of RapidMiner 5.

Executing workflows is handled using the standard Taverna mechanism. Users are displayed a graphical view to show the workflows progress and intermediate and output file locations can be previewed directly within Taverna. As the RapidMiner operators only pass data by reference, by default you only see the relative paths to files generated on the RapidAnalytics server. These files can be viewed via the RapidAnalytics web interface, alternatively, RapidAnalytics provides additional services for generating reports that can be included in the Taverna workflow. These reporting operators can generate outputs in various formats that can be viewed directly in Taverna.

1.4 Availability

In order to use the RapidMiner plugin in Taverna you will need access to a RapidAnalytics server. RapidAnalytics is a J2EE (Java 2 Enterprise Edition) application, running on any standard-compliant application server and is available from `rapid-i.com/`. The plugin works with Taverna workbench version 2.2 and above which is available from `http://www.taverna.org.uk`. The RapidMiner plugin can be obtained from the plugin menu directly in Taverna and more information and documentation is provided from `http://www.e-lico.eu/TavernaRM`.

2 Bioinformatics use case

RapidMiner provides a powerful workbench for building data mining workflows yet there is still much to be gained from exposing these operators in other applications. Workflow systems, such as Taverna, offer a greater degree of flexibility in terms of the services that can be consumed and used within a workflow. By exposing these RapidMiner operators in Taverna, a whole community of scientists can make use of them in their own domain specific workflows.

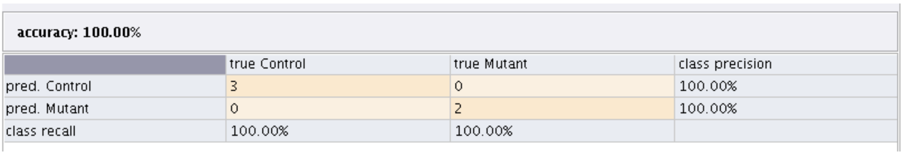
Taverna makes it possible for scientist to access many service providers and data centres from a single user interface. Whilst getting access to data is a pre-requisite for analysis, there is often a considerable amount of work required to transform this data so it can be processed by the different services. These transformations, also knows as shims [5], are commonplace in bioinformatics workflows. Taverna already contains sets of 'local' services that are intended to assist the user in common data manipulation tasks. With the release of the RapidMiner plugin, Taverna users have access to a richer set of operators for transforming and manipulating tabular data. Exposing RapidMiner in Taverna means users can build more complex workflows that bridge the gap between the data gathering steps and data pre-processing, both of which are pre-requisites to data mining.

In order to illustrate the benefits the RapidMiner plugin brings, we have developed several workflows to demonstrate its functionality in some typical bioinformatics tasks. These workflows are available for download from the myExperiment here `http://www.myexperiment.org/groups/402.html`.

2.1 Collecting, clustering and validating microarray data

Taverna is particularly useful for the collection of biological datasets from centralised data repositories, such as those provided by the EBI[2] and NCBI[3]. Of particular relevance to modern biology are high-throughput methods that produce many measurements (i.e. many thousands) simultaneously. In data mining terms this produces single examples each with many thousands of numerical attributes. These high-dimension data are subject to significant measurement error across experiment repeats, it is therefore vitally important to identify, normalise or possibly remove results that are clearly outliers in a wider dataset. Furthermore these datasets often contain multiple repeats of single biological states, the most common states being a diseased and a control state. These states and their experimental repeats are used to identify robust differences in biochemical composition between a diseased and control state. RM can therefore be very useful in the pre-processing and validation of biological datasets and by directly linking the collection and analysis of these data into a single workflow performed on a single platform (Taverna), we allow others to use the same process to analyse their data.

Our first example workflow collects a microarray dataset (GEO dataset GDS3685[4]) from the GEO database and forms this into a CSV formatted table. This is not a straightforward task and is made possible by the REST and XPath services provided as part of Taverna. The CSV data is then uploaded to RapidAnalytics. We set ‘label’ and ‘id’ roles on the appropriate attributes, and then we perform a split validation using an ‘SVM’ classifier (with default parameters) to test whether measurements from experimental states (mutant and control) are consistent across experimental repeats. Results from this analysis are provided as a performance vector which indicates that using a 50:50 stratified split, the test partition is classified with 100% accuracy (figure 3).



accuracy: 100.00%

	true Control	true Mutant	class precision
pred. Control	3	0	100.00%
pred. Mutant	0	2	100.00%
class recall	100.00%	100.00%	

Figure 3: Classification performance on 50:50 stratified split using default SVM classifier

Our second workflow performs a common task in microarray analysis, that

[2] http://www.ebi.ac.uk/Databases

[3] http://www.ncbi.nlm.nih.gov/guide/all

[4] http://www.ncbi.nlm.nih.gov/sites/GDSbrowser?acc=GDS3685

of assay clustering, this is another process used to assess data validity, which is hugely important when dealing with biological data that is often ‘noisy’. Using the same dataset stored in RapidAnalytics, we apply the RapidMiner ‘agglomerative clustering’ operator, again with default parameters. The output of the workflow, a cluster model, shows that one experimental repeat (GSM300685) from the mutant state, does not cluster with the other mutant assays and instead clusters with the control assays. This suggests that this result may be anomalous and therefore should be a candidate for exclusion from the dataset.

To further explore the differences between GSM300685 and the other mutant assays we used a third workflow that employs the ‘cross distances’ RapidMiner operator to generate the heatmap in figure 4. The heatmap clearly shows how GSM300685 is more distant from 3 of the 4 mutant assay repeats (GSM300681, GSM300682, GSM300684) than from the control repeats (GSM300676, GSM300677, GSM300678, GSM300679, GSM300680). The heatmap can be generated automatically by RA (figure 4) and can be downloaded for viewing in Taverna or in a web browser using a unique URL (`http://tinyurl.com/6b5qfnc`).

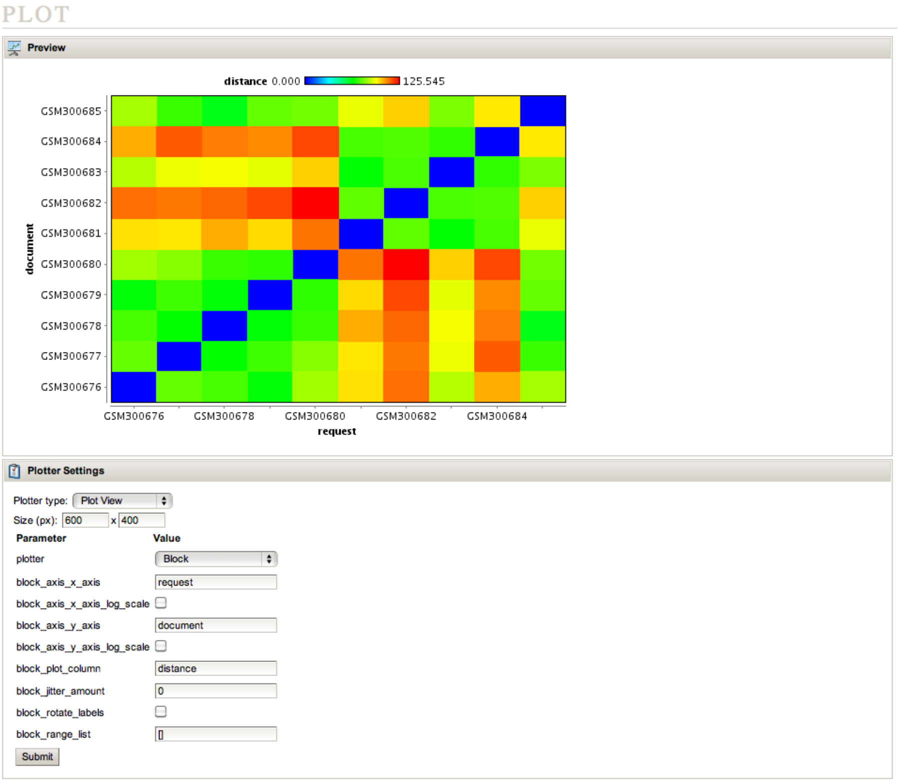
3 Discussion

Through the Taverna RapidMiner plugin we have brought together Taverna’s access to a wide variety of bioinformatics resources with RapidMiner’s array of tools for data processing, data mining and data visualisation. It follows a standard pattern of autonomous tools and database development and then post hoc combination.

3.1 Current limitations

The first release of this plugin is currently limited to a subset of operators available in RapidMiner. Simple operators that work on input files and generate a set of output files are easily handled in the Taverna plugin. However, RapidMiner also contains some specialised operators that we call *dominating operators*, that provide a special case. Most Taverna workflows assume a serialised flow of data between operators, where one operator receives input, generates some output that is then passed onto the next operator. Taverna can work with more complex workflows such as nested workflows and has functionality for conditional and looping operators, however, these are currently not sufficient to handle some of the RapidMiner operators from web services alone.

One set of so called dominating operators is the set of validation operators in RapidMiner. These validation operators have a specialised functionality in that they control the execution of one or more sub data mining processes. An



Plotter type: Plot View

Size (px): 600 x 400

Parameter	Value
plotter	Block
block_axis_x_axis	request
block_axis_x_axis_log_scale	□
block_axis_y_axis	document
block_axis_y_axis_log_scale	□
block_plot_column	distance
block_jitter_amount	0
block_rotate_labels	□
block_range_list	[]

Figure 4: Usage of RapidAnalytics browser-based plotting interface

example is the X-validation operator in RapidMiner. X-validation is the process of partitioning data into subsets in order to perform analysis on a subset (i.e. a training set) and validate the analysis on the other subset (the test set). As these operators are represented as typical web services by RapidAnalytics, the X-validation operator itself must control the execution of the sub workflow. This control currently requires some logic that can not be expressed in Taverna via web services alone. Such is the importance of these operators in many data mining tasks, work is currently underway to extend the RapidMiner plugin to handle these dominating operators, and a solution will be provided later in 2011.

3.2 Future work

This work forms an early outcome of a the much broader objectives of the e-LICO project [5]. The e-LICO project are seeking to build a virtual laboratory for interdisciplinary collaborative research in data mining and data intensive sciences. The e-LICO platform, of which both Taverna and RapidMiner are a major component, will provide infrastructure to assist users in developing data mining workflows. This assistance will come in the form of an intelligent data mining assistant that will support users in the creation of complex data mining workflows.

At this early stage we can see that bringing together the different functionalities of Taverna and RapidMiner shows great promise. Taverna allows arbitrary workflows to be made from a wide variety of service types. This gives access to a wide range of data available in domains such as biology (but also many others). RapidMiner affords a rich set of specialist data mining operators, that include many pre-processing and visualisation tools. All of these have applications on the types of data that Taverna can gather and manage. We have brought these two tools together and begun to show the advantages this can bring.

4 Acknowledgements

This work is funded by the EU/FP7/ICT-2007.4.4 e-LICO project.

References

[1] *Here is the evidence, now what is the hypothesis? The complementary roles of inductive and hypothesis-driven science in the post-genomic era*, Bioessays **26** (2004), no. 1, 99–105.

[5] `http://www.e-lico.eu`

[2] Sean Bechhofer, David De Roure, Matthew Gamble, Carole Goble, and Iain Buchan, *Research objects: Towards exchange and reuse of digital knowledge*, The Future of the Web for Collaborative Science (FWCS 2010), February 2010, Co-located with WWW'10.

[3] Michael Y. Galperin and Guy R. Cochrane, *The 2011 nucleic acids research database issue and the online molecular biology database collection*, Nucleic Acids Research **39** (2011), no. suppl 1, D1–D6.

[4] Carole Goble and Robert Stevens, *State of the nation in data integration for bioinformatics*, Journal of Biomedical Informatics **41** (2008), no. 5, 687 – 693, Semantic Mashup of Biomedical Data.

[5] Duncan Hull, Robert Stevens, Phillip Lord, Chris Wroe, and Carole Goble, *Treating shimantic web syndrome with ontologies*, Advanced Knowledge Technologies Semantic Web Services (AKTSWS), 2004.

[6] Duncan Hull, Katy Wolstencroft, Robert Stevens, Carole A. Goble, Matthew R. Pocock, Peter Li, and Tom Oinn, *Taverna: a tool for building and running workflows of services*, Nucleic Acids Research **34** (2006), no. Web-Server-Issue, 729–732.

[7] Jason Tsong-Li Wang, Mohammed Javeed Zaki, Hannu Toivonen, and Dennis Shasha, *Introduction to data mining in bioinformatics*, Data Mining in Bioinformatics (Jason Tsong-Li Wang, Mohammed Javeed Zaki, Hannu Toivonen, and Dennis Shasha, eds.), Springer, 2005, pp. 3–8.