Workflow Scenarios
for a Semantic Web for Fungal Genomics

Greg Butler

Department of Computer Science and Software Engineering, Centre the Structural and Functional Genomics, Concordia University, Montréal, Quebec, Canada

Introduction

Over the last three years we have been developing a bioinformatics platform to support a comprehensive project [1] on fungal genomics for the discovery of novel enzymes with industrial or environmental applications. The project is comprehensive in that it is vertically integrated. In essence it is several major subprojects each carried out on several fungal species and many genes and enzymes, namely; a sequencing project which constructs cDNA libraries, sequences ESTs, assembles ESTs into unigenes, and analyses unigenes; a microarray project which constructs cDNA microarrays for each species and carries out transcription profiling experiments; a curation project where selected target genes and enzymes are manually curated and re-sequenced to obtain the full length genes; a gene expression project where full length genes are spliced into host organisms in order to secrete the selected enzyme; an enzomology project which biochemically assays the enzymes; and an applications testing project. The bioinformatics platform supports data collection and analysis across the entire project. Our aim was to use existing best practice, adopt widely used Software, and minimise our internal efforts at software development. Nevertheless, there were several significant systems developed as they had to be customised to our genomics project and its complexity.

In planning for the future of the fungal genomics project where we will have a considerable amount of data from which we need to create knowledge, we have been exploring the technologies of the semantic Web. This project is called FungalWeb [2]. The vision of the semantic Web is to extend the World Wide Web from a collection of data and documents that are often hard to find and use into a collection of knowledge that is very convenient to use. One pillar of the semantic Web is to associate an ontology with each web site: the ontology describes the information that is available on the site and it describes the information in a way that is precise and formal enough that the description can be manipulated by computer software. The second pillar of the semantic Web are software agents that utilise the Web to perform tasks. The tasks may range from the mundane such as straightforward data retrieval to the intelligent integration of heterogeneous knowledge sources. The third pillar is a concept of service broker or matchmaker, often using a service ontology, that connects the agents with available services on the web, both computational and data services.

In this paper we explore the range of tasks in genomics and the kinds of intelligent reasoning required for those tasks in order to gain a better understanding of the knowledge that must be captured in formally described ontologies. We do this by cataloguing a number of scenarios of scientific workflow.

Consider this paper an overview of the FungalWeb project. FungalWeb will be a prototype semantic Web for fungal genomics within the Montréal region that uses OWL description logic to represent ontologies, and the Racer reasoner [3] to perform the T-box and A-box reasoning. FungalWeb will complement the large-scale fungal genomic project, develop a variety of ontologies for genomic knowledge, and use the knowledge to determine the role of a gene from a variety of experimental evidence including microarray expression data.

FungalWeb

The scenarios envisaged in the FungalWeb project include the tasks of automated sequence analysis and annotation; the integration of several data sources to complement microarray data to determine regulatory information, and the use of text mining to extract information from the scientific literature.

The automated analysis of sequences is a fairly well-understood process. It involves many tasks, but widely-used software packages exist for these tasks. The process first does quality control on the EST sequences following base-calling, then assembles them into unigenes, and determines the translation into protein sequences. The automated analysis of a protein sequence may involve up to 30 different software tools, plus reasoning steps to integrate the results of the analysis. The general aim is to assign a set of Gene Ontology terms to each protein [4].

For the microarray data we are exploring the use of probablistic relational models introduced by Koller and Friedman [5]. We had to develop our own software for PRMs as none was publicly available. The first application of PRMs is to integrate TF binding site information in the upstream region of genes, the homology between genes and TFs, and microarray expression data to determine regulatory relationships between genes.

With text mining [6] we are working towards integration of statistical text mining with NLP techniques and ontological annotations of words and phrases within the text. One task is to determine numerical values for kinetic properties of enzymes from the scientific literature.

A broader range of scenarios will be discussed in the full paper.

Workflow and the Semantic Web

Workflow in a multi-agent system [7] makes use of the matchmaker or broker to dynamically locate services. The workflow may be represented by a traditional workflow process description that is interpreted by an agent acting as a workflow engine, however, is more common for a workflow in a multi-agent system to be represented as a hierarchical plan of subgoals with one or more agents available to achieve the subgoals through different strategies. Eventually these agents make use of data services or computational services to perform the atomic tasks. A service ontology [8] is required to describe services and to allow the matchmaker/broker to match service requests with service providers.

Workflow Scenarios

In this section we detail typical scenarios of each kind.

The base calling scenario is a linear pipeline that processes the chromatogram and generates the assemblies of high-quality sequence segments. The first step is base calling which is done by the commonly used software utility phred. The second step is to eliminate poor quality regions and contaminants. This is done by a program called Lucy that was developed at TIGR. The third step is assembly which is performed by the software utility phrap. At this stage we have a collection of consensus sequences of each assembly called unigenes which can be translated into open reading frames (ORFs) that are segments of protein sequences.

The protein annotation scenario runs a number of different software tools for each program using a collection of archival databases. The aim is to annotate each protein with one or more terms from the Gene Ontology. One major tool in this analysis is to perform a BlastP search of the protein against the Uniprot database. Many of the entries in the Uniprot database are annotated with GO terms and/or EC numbers. Thus the BlastP search is a convenient way to match to use sequence similarity to map to GO terms. Another major tool in this analysis is to run the classifiers that scan the protein for matches in the Interpro database of protein domain information. Again there are mappings from Interpro entries to corresponding GO terms. Another major set of tools in the analysis concerns cellular location prediction and the prediction of whether the protein is secreted or not. These tools include PSort, SignalP, TargetP, TMHMM, Phobius, GPI predictor, as well as motif finders. Other analysis tools such as the detection of low complexity regions and the use of the CDD database of conserved domains are also run. In terms of the analysis tools we are more complete than BioMAS but still lack many of the tools found in the Annie annotation system from Vienna. At this stage we do not do anything intelligent to combine the different GO terms into a summary annotation for a protein. This is done by human curators who are presented the above information.

The scenario for analysis of microarray data consists of three sub-scenarios. The first sub-scenario determines the set of differentially expressed genes. The second sub-scenario users cluster analysis together with gene annotations to indicate statistically significant properties of clusters. The third sub-scenario uses PRMs to integrate microarray data, cluster data and transcription factor data obtained from the genome. In this discussion we are ignoring the multitude of quality control steps necessary for microarray experiments.

The determination of the set of differentially expressed genes involves a normalisation step followed by a step to set a threshold for differential expression and its statistical significance. We do not eliminate background in our analysis. The normalisation is generally performed using global Lowess normalization. This is complemented by pinhead group-specific normalisation if the quality control data indicates a skew between pinhead groups. We use an in-house algorithm that uses a sliding window to determine an intensity-dependent z-score for each gene.

Clustering around medoids as a variant of k-means clustering is often used for its robustness though the number of clusters k is still difficult to determine in practice. There are now many available programs to perform the statistical analysis of clusters.

The third scenario follows the approach of Eran Segal to determine from the microarrays data information about regulatory relationships between genes. This combines evidence from an analysis of the upstream and downstream regions of the gene for transcription factor binding sites with the microarray data and the cluster information. The transcription factor binding sites are determined using the HMM models from the SCPD database that catalogues these binding sites from yeast. We are applying this analysis to *Aspergillus niger*. We have information about expression levels of genes for growth in media with different sugar or carbon sources. These carbon sources include glucose, maltose, xylose and glycerol. As the genome for *Aspergillus niger* was not available to us, we mapped the genes to the genome of *Aspergillus nidulans* which was available. And we scanned that genome for transcription factor binding sites from yeast as catalogued in the SCPD. This workflow was necessary to collect the raw data. Following this was a manual step to construct the model for the PRM. Then the PRM software was run to learn the parameters of the model.

There are scenarios concerned with phylogenetic or phylogenomic analysis of protein families that we have carried out in order to better understand the relationship between protein sequences and their biochemical properties. This work is ongoing so we will not report it in detail here. An initial step in the study of each protein family is to scan the scientific literature and determine which proteins have already been biochemically characterised and what is known about numeric values of their kinetic parameters. Text mining can assist in this.

An interesting scenario for text mining is the MutationMiner system developed by Baker and Witte. The text mining component extracts from a PubMed article identifying information about an organism, a protein in that organism, and a site of a point substitution mutation in the protein. Many such mutation experiments are performed with proteins with industrial application. The experiments seek to enhance the biochemical properties of the protein. The article records the result of those experiments. In fact the text mining component treats a collection of articles related to will a family of proteins. A second component in this scenario seeks to determine the sequences under discussion in the articles and to carry out an analysis of the sequences. The analysis aligns the sequences, determines their conserved domains, and locates the mutation sites within those sequences as discussed in the articles. The analysis matches a consensus sequence for the protein family with an entry in the protein database (PDB) which has a three-dimensional structure. A third component is capable of visualising the locations of the mutations against the three-dimensional structure.

Conclusion

We have focused here on the workflow scenarios which involve several computational steps as well as data extraction and collection steps. Of course within any bioinformatics system as complex as FungalWeb there are numerous data sources that need to be integrated. This is a topic that we do not discuss here. Our related work details the FungalWeb ontology, our data warehouse of fungal genomics data, and the query mechanisms provided to access this data.

Acknowledgements

This work was partially have supported by Genome Québec and the Natural Sciences and Engineering Research Council of Canada. The author wishes to thank the following people for fruitful discussions and collaborations in this work: Adrian Tsang, Reg Storms, Justin Powlowski, Volker Haarslev, Chris Baker, and Bohdana Rattich.

[1] Fungal Genomics Project https://fungalgenomics.concordia.ca

[2] FungalWeb Project http://www.cs.concordia.ca/FungalWeb/

[3] Racer URL

[4] The Gene Ontology consortium, Gene Ontology: Tool for the unification of biology, Nature Genetics 25: 25-29, 2000.

[5] Lise Getoor, Nir Friedman, Ben Tasker, Daphne Koller, Learning probablistic models of relational structure, Journal of Machine Learning Research, 2003

[6] Lynette Hirschman, Jong C. Park, Junichi Tsujii, Limsoon Wong, Cathy Wu, Accomplishments and challenges in literature data mining for biology, Bioinformatics, 2002.

[7] Keith Decker, Salim Khan, Carl Schmidt, Gang Situ, Ravi Makkenna, Dennis Michaud, BioMAS: A a multiagent system for genomic annotation, International Journal of Cooperative Information Systems 11 (3-4): 265-292, 2002.

[8] Chris Wroe, Robert Stevens, Carol Goble, Angus Roberts, Mark Greenwood, A suite of DAML+Oil ontologies to describe bioinformatics Web services and data, International Journal of Cooperative Information Systems 12 (2): 197-224, 2003.

[9] Carol Goble, Chris Wroe, Robert Stevens and the myGrid consortium, The myGrid project: services, architecture and demonstrator.