INTEGRATION STRATEGIES AND DATA ANALYSIS METHODS FOR PLANT SYSTEMS BIOLOGY

ARTEM LYSENKO

B. Sc. Biology

M. Res. Bioinformatics

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

July 2012

ABSTRACT

Understanding how function relates to multiple layers of inactions between biological entities is one of the key goals of bioinformatics research, in particular in such areas as systems biology. However, the realisation of this objective is hampered by the sheer volume and multi-level heterogeneity of potentially relevant information. This work addressed this issue by developing a set of integration pipelines and analysis methods as part of an Ondex data integration framework. The integration process incorporated both relevant data from a set of publically available databases and information derived from predicted approaches, which were also implemented as part of this work.

These methods were used to assemble integrated datasets that were of relevance to the study of the model plant species *Arabidopsis thaliana* and applicable for the network-driven analysis. A particular attention was paid to the evaluation and comparison of the different sources of these data. Approaches were implemented for the identification and characterisation of functional modules in integrated networks and used to study and compare networks constructed from different types of data. The benefits of data integration were also demonstrated in three different bioinformatics research scenarios. The analysis of the constructed datasets has also resulted in a better understanding of the functional role of genes identified in a study of a nitrogen uptake mutant and allowed to select candidate genes for further exploration.

ACKNOWLEDGEMENTS

I would like to take this opportunity to give my whole-hearted thanks to my supervisors: Chris Rawlings, Tony Miller, Charlie Hodgman and Jacob Koehler. I am particularly grateful for their patience, understanding and advice all the way throughout this project. I also would like to acknowledge Mansoor Saqi who has provided a lot of valuable input into my work and encouraged me to publish my research. Another, very welcome source of feedback and suggestions came from Michael Defoin-Platel, to whom I am also very grateful for all the support.

I also would like to acknowledge the invaluable contributions of all Ondex developers, past and present – in particular Matthew Hindle, Andrea Splendiani, Robert Pesch, Berend Hoekman, Catherine Canevet, Jan Taubert, Keywan Hassani-Pak, Matthew Pocock, and Rainer Winnenburg. This work was only made possible only because so many different aspects of the Ondex system were there to support it because of their hard work.

In addition to that, I would like to express my thanks to the Rothamsted Research, University of Nottingham for giving me this opportunity to undertake this research and to Biotechnology and Biological Sciences Research Council, for providing funding.

A special note of appreciation also goes to my family and friends for their support and encouragement that helped me to stay focused and persevere when things did not go as expected.

CONTENTS

LIST OF FIGURES

ACRONYMS AND ABBREVIATIONS

AGI	locus identification code used by TAIR resource
AGRIS	Arabidopsis Gene Regulatory Information Server (http://arabidopsis.med.ohio-state.edu/)
AIC-MICA	Average Information Content of the Most Informative Common Ancestor, a measure of functional coherence for a set of proteins, fully defined in chapter 5
ALL	combined network of all evidence types, described in chapter 5
AMIGO	the official Gene Ontology browser and search engine (http://amigo.geneontology.org)
AMT	class of ammonium transporter proteins
API	Application Programmers Interface
ARACNE	algorithm for reconstruction of gene regulatory networks
AraCyc	A database of Arabidopsis Pathways
ARA-REF	Ondex dataset containing Arabidopsis genome and proteome information, fully defined in section 4.2.2
ArrayExpress	Microarray data repository hosted by EBI (http://www.ebi.ac.uk/arrayexpress/)
ASP	asparagine synthase
AspAT	aspartate aminotransferase
AtCISDB	Arabidopsis *cis*-regulatory element database by AGRIS (http://arabidopsis.med.ohio-state.edu/AtcisDB/)
ATH1-121501	the most recent version of the Affymetrix *Arabidopsis* oligonucleotide microarray
ATP	Adenosine triphosphate
AtRegNet	database of Regulatory Networks in *Arabidopsis thaliana* (http://arabidopsis.med.ohio-state.edu)
ATTED-II	database of *Arabidopsis* coexpression data (http://atted.jp/)
AtTFDB	Arabidopsis transcription factor database by AGRIS (http://arabidopsis.med.ohio-state.edu/AtTFDB/)
BERP	best entropy rank proportion, fully defined in section 5.2.5
BFRP	best fragment rank proportion, fully defined in section 5.2.5
Bioconductor	open-source R library for biocomputing
BioGrid	protein-protein interaction database (http://thebiogrid.org/)
BIOPAX	Pathway exchange language for Biological pathway data
Biopython	Python library for bioinformatics
BLAST	Basic Local Alignment Search Tool
BP	Biological Process, an aspect of Gene Ontology
C++	low-level programming language
CATdb	Complete Arabidopsis Transcriptome database (http://urgv.evry.inra.fr/CATdb/)
CATMA	Complete *Arabidopsis* Transcriptome MicroArray technology
CC	Cellular Component, an aspect of Gene Ontology
cDNA	complementary DNA
CEL	File format for storage of gene expression measurements from Affymetrix arrays

cHATS	constitutively expressed high-affinity transport system
COE	coexpression network, described in chapter 5
COEXPRESdb	database of coexpression networks for several mammalian species (http://coxpresdb.jp/)
Cytoscape	bioinformatics software tool for visualizing networks
CytoTalk	plugin for Cytoscape that allows access to R environment
DAG	directed acyclic graph
DATF	Database of Arabidopsis Transcription Factors (http://datf.cbi.pku.edu.cn/)
DNA	Deoxyribonucleic acid
EBI	European Bioinformatics Institute
ERF	Ethylene-Responsive Transcription Factor
EXP	Inferred from Experiment (Gene Ontology annotation evidence code)
FASTA	text-based format for storage of protein and nucleotide sequences
FTP	File Transfer Protocol
GAF	exchange format for Gene Ontology annotations
GDH	glutamate dehydrogenase
GDS	Generalized data structure, a type of attribute in the Ondex data model that can store more complex types of information
GeneChip	DNA microarray technology by Affymettrix
GEO	Gene Expression Omnibus, a warehouse repository for microarray data (http://www.ncbi.nlm.nih.gov/geo/)
GFP	Green fluorescent protein, refers to a SUBA database evidence code
GO	Gene Ontology
GOA	Gene ontology annotation
GOA-BP	Integrated Ondex dataset containing *Arabidopsis* Biological Process annotation, defined in chapter 4
GRN	gene regulatory network
GS	glutamine synthetase
GUI	Graphical User Interface
GZIP	one of the possible formats for file compression
HATS	high-affinity transport system
HUPO	Human Proteome Organisation, international consortium of proteomics research associations
IC	information content
IEA	Inferred from Electronic Annotation (Gene Ontology annotation evidence code)
iHATS	inducible high-affinity transport system
IntAct	protein-protein interaction database hosed by EBI (http://www.ebi.ac.uk/intact/)
IO	input/output system
IPI	Inferred from Physical Interaction (Gene Ontology annotation evidence code)
ISS	Inferred from Sequence or Structural Similarity (Gene Ontology annotation evidence code)
JA	jasmonic acid
Java	Programming language developed by Sun microsystems
JavaAssist	Java Programming Assistant, a byte code manipulation library for Java
JavaDoc	Documentation generator tool and standard for Java API, developed by Sun Microsystems

JavaScript	prototype-based scripting language
JNI	Java Native Interface, a framework that allows system-native code to be accessed from Java
JRI	library that allows to call R from the Java environment
Jung	Java Graph library
Jython	one of the implementations of Python scripting language written in Java
KEGG	Kyoto Encyclopaedia of Genes and Genomes, a database of biochemical pathways
LATS	low-affinity transport system
LIT	protein name co-occurrence in scientific literature abstracts network, described in chapter 5
Lucene	A Java-based text search and indexing solution
MCL	Markov Cluster algorithm
MF	Molecular Function, an aspect of Gene Ontology
MIAME	Minimum Information About a Microarray Experiment
MICA	Most Informative Common Ancestor (a term in the Gene Ontology with that is an ancestor of two query terms and has the highest possible information content)
mRNA	messenger RNA
N	nitrogen
NADH	Nicotinamide adenine dinucleotide
NADH-GOGAT	NADH-dependent glutamate synthases
NAR	Nucleic Acids Research journal
NASC-Arrays	microarray data repository based at the University of Nottingham (http://affymetrix.arabidopsis.info/narrays)
NCBI	National Center for Biotechnology Information
ND	No biological Data available (Gene Ontology annotation evidence code)
NetworkX	Python package for working with networks
NiR	nitrite reductase
NR	nitrate reductase
NRT	class of nitrate transporter proteins
OBO	exchange format for ontologies
Ondex	Data integration and graph visualisation framework (http://www.ondex.org/)
OXL	Ondex XML format, which is the main format used to serialise Ondex integrated graphs
Perl	high-level scripting language
PII	photosystem II
PlnTFDB	Plant Transcription Factor database at the Universitaet Potsdam (http://plntfdb.bio.uni-potsdam.de/)
PPI	protein-protein interaction
PPI	protein-protein interaction network described in chapter 5
PSI-MI	XML-based file format for exchanging the protein-proteins interaction data, maintained by HUPO
PubMed	database of scientific publications (http://www.ncbi.nlm.nih.gov/pubmed/)
Python	interpreted scripting language
R	statistical environment based on S4 programming language
RDF	Resource description framework format
rJava	library that allows Java to be called from the R envieronment

RMA	Robust Multichip Average, a method for normalisation of Affymerix GeneChip expression data
RNA	Ribonucleic acid
S4	statistical programming language
SBML	Systems Biology Mark-up Language
SEMEDA	Data integration system on which Ondex was originally based (Olson *et al.*, 1999)
SEQ	sequence similarity network, described in chapter 5
SGD	Saccharomyces Genome Database
SPARQL	query language for data in the RDF format
SPL	average shortest path length statistic, fully defined in section 5.3.5
SQL	query language for relation databases
STRING	database of functional protein association networks (http://string-db.org/)
SUBA	SUB-cellular location database for Arabidopsis proteins (http://suba.plantenergy.uwa.edu.au/)
SwissProt	manually annotated and reviewed subset of the UniProtKB database
TAIR	database on *Arabidopsis thaliana* genome (http://www.arabidopsis.org/)
Tera-BLAST™	A BLAST algorithm implementation by TimeLogic® company
TrEMBL	automatically annotated and unreviewed subset of UniProtKB database
UniProt	Universal Protein Resource (http://www.uniprot.org/)
UniProtKB	Protein knowledgebase at UniProt (http://www.uniprot.org/help/uniprotkb)
URL	Uniform resource locator, a character string for location an Internet resource
XML	Extensible Markup Language
ZIP	one of the possible formats for file compression

1 BACKGROUND AND INTRODUCTION

1.1 SUMMARY

Recent technological advances in biology have led to the increasingly fast pace of data accumulation and a multitude of resources and strategies to manage it and make it available to the research community. However research applications often require data to be combined from these different resources and representations in order to get a complete understanding of the living system. This need gave rise to the disciple of data integration, which researches strategies for effective management of the increasingly large body of experimental information and ways to ensure that consistency, provenance and intercompatibility between different resources is adequately realised. The Ondex system is specifically targeted at addressing the data integration requirements of the plant biology community. Ondex is designed around its graph-based unified data model, which is the basis both for the construction of the integrated datasets via Ondex workflow engine and their subsequent analysis/visualisation using Ondex front end. The ability to provide both the integration and visualisation capabilities as part of the same package, as well as the ability to effectively capture complex data in its data model are the strengths of the Ondex system that differentiate it from similar tools.

1.2 THESIS OUTLINE

The research presented in this work concerns the development of data integration and network-driven bioinformatics analysis methods for the study of a model plant species *Arabidopsis thaliana*. The practical relevance of the developed approaches was demonstrated through three independent use-cases presented in the chapters 4-6. Issues investigated therein include comparative analysis of various *Arabidopsis* information resources, identification and evaluation of functional modules and gene list dissection for the purposes of candidate gene prioritisation. The reminder of this introductory chapter covers the relevant background and introduces key concepts and formalisms. In particular the current paradigms and approaches used in biomedical data integration are outlined. The data integration strategy realised in the Ondex system is also explained. Ondex was the chosen platform for the implementation of the supporting code base for this work and this system was improved and had new capabilities added to it as a direct result of these efforts.

The developments to do with the improvements of the core Ondex functionality are presented in chapter 2. Originally, the Ondex project was founded to provide data integration capabilities to the plant research community, and at the moment it still remains the only non-commercial, open-source platform with such focus. Although the system was already developed for some time prior to the start of this project, it also had several shortcomings and limitations, which were identified and addressed as part of this work. These developments have served to greatly facilitate the analysis reported in the subsequent chapters and enabled seamless assembly of more complex analyses pipelines which would not have been possible otherwise.

Chapter 3 describes a pipeline that was developed for the construction of the coexpression networks from Affymetrix GeneChip microarrays. Its addition to Ondex toolkit provided a new and powerful method for the analysis and extraction of insights from large volumes of expression data. This development was further supported by the addition of the module detection and functional enrichment analyses, which were also added as part of the work on this thesis.

During this work, several integrated datasets of relevance to *Arabidopsis* and plant biology community were developed. In addition to their practical value, these datasets were used to study the strengths and shortcomings of the individual data types and information sources. To maximise the benefit from the integration, it is necessary to understand how different data relate to each and how the comparative analysis can be leveraged to evaluate their quality. To that end, a number of different information resources that provide the *Arabidopsis* data to the research community were evaluated and compared. This work is reported chapter 4 of this thesis. Additionally, it was also published in Briefings in Bioinformatics journal; a full version of this paper is enclosed in the Appendix.

Chapter 5 describes functional similarity metrics used for the analysis of functional modules identified in the networks constructed from different types of "evidence of relatedness" between genes. It also presents the work undertaken to evaluate a network-driven data integration strategy and its application to the detection of functional groupings of *Arabidopsis* genes. The results of the analysis demonstrated the benefits of considering multiple evidence types when attempting to recover groups of genes with similar functional annotation.

In chapter 6, the methods from the chapters 3 and 5 and the integrated datasets from the chapter 4 are brought together in an applied setting by using them to dissect a set of gene lists originating from an expression study. This example also presents the visualisation methods developed to support the interactive analysis of these data. By combining the coexpression, functional annotation and the gene set-driven analysis, it was possible to suggest several promising candidate genes that are likely to be of relevance for the understanding of the nitrogen uptake and response to wounding in *Arabidopsis.*

1.3 AIMS AND OBJECTIVES

To gain systems-level understanding of complex biological systems it is necessary to process together experimental data and prior knowledge into a unified model. If this problem is approached from a network-driven perspective, the task can be decomposed into:

- Definition of entities (nodes) of interest and identification of corresponding entities in relevant data sources
- Definition and establishment of relationships between them
- Relating the model back to the real biological system studied

However, the realisation of these tasks is hampered by the sheer volume and multi-level heterogeneity of potentially relevant information. Another challenge lies in the poor compatibility of the tools and analysis software for processing these data. The over-arching aim of this thesis is to address these shortcomings by contributing to the development of modular software architecture of inter-compatible integration and analysis methods. Due to the complex and multi-faceted nature of this task, it was broken down into a set of more focused objectives:

- Development of a framework to support the integration of data resources and the downstream analysis
- Creation of integrated datasets of relevance to the system-level understanding of *Arabidopsis* biology with particular emphasis of utilizing expression, functional annotation and protein-protein interaction data
- Develop an understanding of the structure and limitations of the individual data sources by establishing different evaluation procedures to assess the integration results from a variety of different perspectives
- Development of visualisation and analysis methods for extraction of biologically relevant insights from the integrated datasets produced
- Demonstrate the relevance of the resource, analysis methods and the framework as a whole by applying them in a set of common bioinformatics research scenarios

1.4 DATA INTEGRATION AND ITS ROLE IN BIOINFORMATICS

Recent technological advances have made it possible to generate vast amounts of biological data. At the genome level, the rate at which new sequencing data is being produced has proven to be a considerable both in terms of information management and downstream bioinformatics analysis (Metzker, 2010). A large number of diverse and separate resources have been developed to facilitate access and support analysis of collected data. The Molecular Biology Database Collection maintained by the journal Nucleic Acids Research (NAR) aims to maintain an up-to-date set of references to the most important databases for biological research. This repository listed just 226 resources in 2000 (Baxevanis, 2000); however this number increased to 858 (Galperin, 2006) when this project was started in 2006, and continued to grow to reach 1230 in 2010 (Cochrane and Galperin, 2010). The development of these resources responds to the need to make the large volumes of the 'omics data available to the research community. A number of 'omics approaches now exist that allow a large-scale sampling of cellular processes from a variety of different perspectives (Lee *et al.*, 2005), the most important of which are shown in the

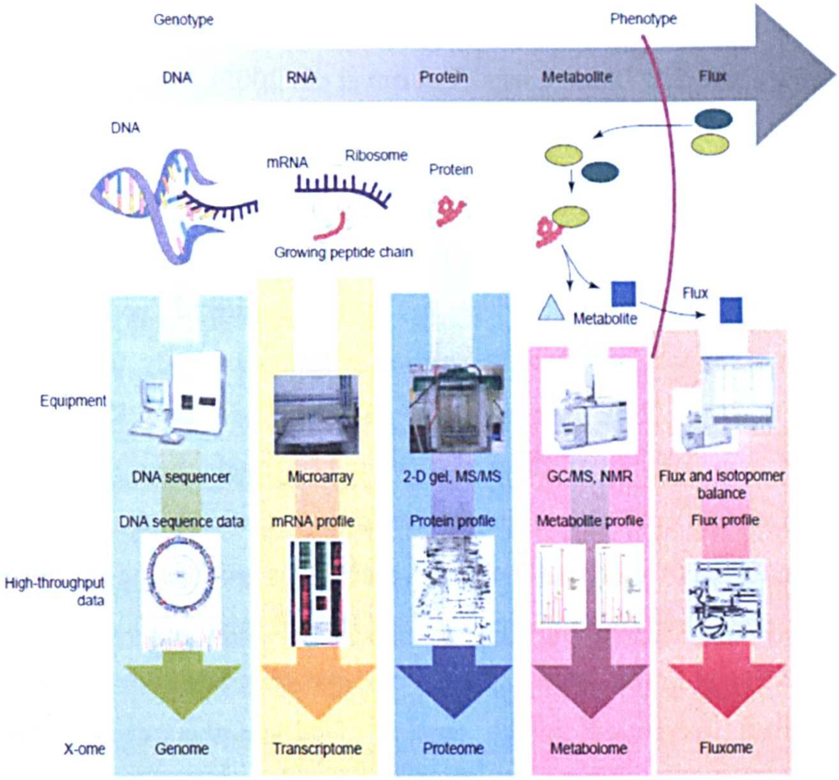


Figure 1.1 High-throughput technologies and corresponding 'omics approaches. Image from Lee *et al.* (2005)

overview in **Figure 1.1**.

Databases resources are often designed around particular sub-disciplines of biology that concern themselves with a particular aspect of research e.g. proteomics, metabolomics, transcriptomics, and interactomics (Goble and Stevens, 2008). However, in the beginning of this decade it became increasingly recognized that in order to gain further understanding into biological complexity, living organisms need to be considered across all levels of organisation (from molecules to organisms and up to ecosystems) and across all domains of study (Kitano, 2000, Ge *et al.*, 2003, Davidov *et al.*, 2003). This approach is known under the name of systems biology (Mesarovic, 1968). The main underpinning assumption behind it is that "the whole is more than the sum of its parts" and in order to understand living organisms the data collected using traditional approaches must be brought together and compiled into models, ultimately fully quantitative and predictive ones (Kell and Knowles, 2006). High-quality models can not only be used to make predictions (Ideker *et al.*, 2001) but also to explore the processes at higher levels of organisation – for example by building models at the scales of a cell (Slepchenko *et al.*, 2003) or a tissue (Swarup *et al.*, 2005).

As biological systems are hierarchical and highly inter-related in nature (Ayton *et al.*, 2007, Southern *et al.*, 2008), in order to characterise them large volumes of heterogeneous information are often necessary. However, it is widely recognised that the relevant data are often scattered across multiple independent resources and possibly buried within seemingly unrelated data (Joyce and Palsson, 2006, Ge *et al.*, 2003, Hernandez and Kambhampati, 2004). The combination of these issues makes conducting the necessary integration tasks manually increasingly foreboding, leading to the development of a large number of computational approaches for automating this process; reviewed in Sujansky (2001), Hernandez and Kambhampati (2004), Goble and Stevens (2008) and Sorani *et al.* (2010). This need gave rise to the discipline of biological data integration, which aims to facilitate the amalgamation of disparate experimental information and develop better strategies for representing and managing these data (Sujansky, 2001). As it is illustrated in

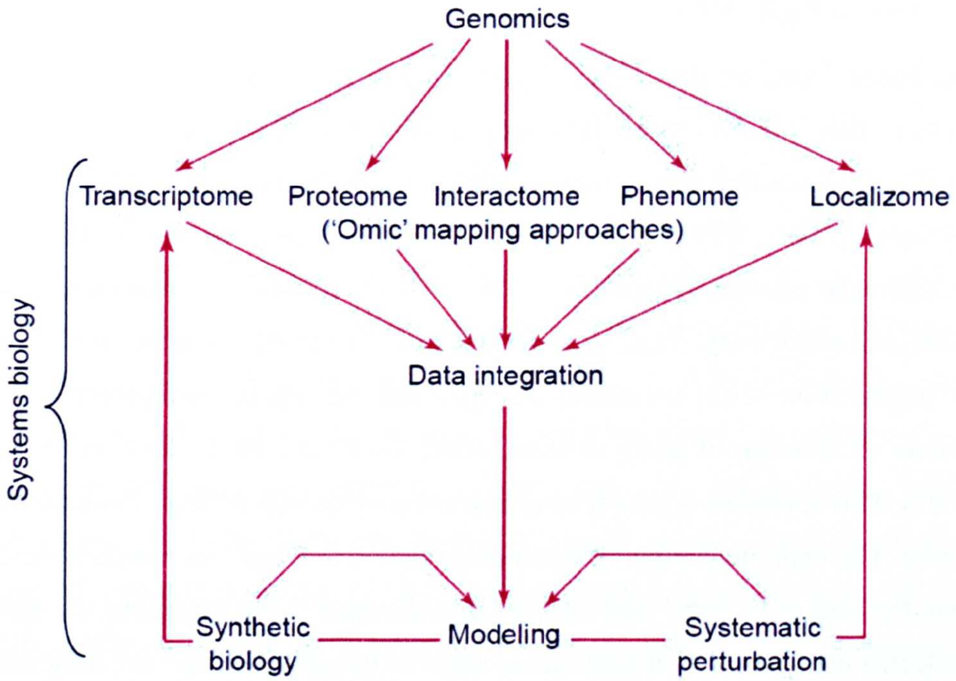


Figure 1.2 Data integration as an essential data flow mediator at the core of systems biology. Integration of experimental data allows construction of models from the 'omics data, which, in turn, are used to generate hypothesis for further experiments. From Ge *et al.* (2003).

the

Figure 1.2, integration is an essential intermediate step for the formulation of system-wide biological models (Ge *et al.*, 2003).

The widely recognised challenges to the data integration process include the size and variety of datasets, different types of heterogeneity and autonomy of various data providers (Hernandez and Kambhampati, 2004). The set of problems arising from the data variety include the potentially large size of records (e.g. genomic sequences or protein 3D structures) and the multi-domain and multi-scale organization of information. This makes it complex to define the appropriate relationships that correctly identify often abstract connections between these disparate data (Hernandez and Kambhampati, 2004). The heterogeneity of data is commonly divided into syntactic and semantic sub-types (Köhler, 2004). The syntactic heterogeneity refers to technical differences between resources, like formats, schemas and query interfaces (Köhler, 2004). The semantic heterogeneity refers to more fundamental differences in their conceptual representations, like different formalisms or levels of abstraction, scope-specific naming conventions and naming inconsistencies (Köhler, 2004). Lastly, the autonomy of data providers reflects the fact that differences between the needs of biological sub-disciplines and other sociological boundaries (such as between funding agencies) result in biological data being managed in a decentralized manner by a collection of largely independently operating bodies (Goble and Stevens, 2008). This means that data providers are free to ignore, re-interpret or re-invent data standards; to unilaterally change their schema or content or even to withdraw access to their data. Section 1.3.3 of this chapter explains how some of these issues are managed in the Ondex data integration paradigm.

Integrated data resources are characterised by their ability to provide a standard mode of access to information from a set of distinct heterogeneous data sources (Hernandez and Kambhampati, 2004). This task can be realized by either implementing a common data model (data warehousing) or a common query model (federated databases and mashups) (Goble and Stevens, 2008). User-driven on-demand approaches are also possible (integration workflows), where a consumer constructs an integration pipeline to answer specific questions from a declared set of components using a workflow interface to connect them (Curcin and Ghanem, 2008). Importantly, all data integration approaches assume that there will be shared, common entities or "touch points" between the integrated resources (Goble and Stevens, 2008). At some stage in the integration process these entities are identified and these links then allow the unified access to the information stored in these disparate resources. In other words, the integration is enabled by the unambiguous identification of entities or concepts being accessed. In those cases where there are established and stable common identifiers, the integration is also often realised by the data providers themselves – e.g. by hyper-linking each other's entries (link integration) (Goble and Stevens, 2008).

The set of formally defined terms to which the disparate entities can be mapped is called a controlled vocabulary (Lacroix and Critchlow, 2003). Often, it is also convenient to formally define the possible relationships between the terms themselves, thereby formulating an ontology. Although many possible definitions of 'ontology' currently exist, for the purposes of this thesis an ontology is defined according to Gruber (1993) as "a specification of a conceptualisation". The benefits of using ontologies for the formalization of knowledge in the biological domain are becoming increasingly recognized (Schulze-Kremer, 2002, Louie *et al.*, 2007, Bodenreider, 2008, Jensen and Bork, 2010). Their use facilitates the data integration process across different data providers by creating high-quality, reliable cross-links within the data as well as ensuring the internal consistency of the individual data resources themselves (Jensen and Bork, 2010).

In the field of plant biology, systems approach and data integration are now considered to be of great importance to the research community (Shinozaki and Sakakibara, 2009). The systems approach is also believed to be of great practical significance for development of new crop varieties. In a recent review by Mochida and Shinozaki (2010) the integrated, multi-omics approach was identified as an "effective strategy for clarifying molecular systems integral to improving plant productivity". For over 30 years, *Arabidopsis thaliana* has been a key model species for the study of plant biology (Meinke *et al.*, 1998). Its importance was boosted first by the sequencing of its genome in 2000 (The

Arabidopsis Genome Initiative, 2000) and then by the *Arabidopsis* 2010 initiative, which supported a wide range of projects with the overarching goal of assigning the functional roles to all of the *Arabidopsis* genes by 2010 (Shinozaki and Sakakibara, 2009). In 2008, the "The 1001 genomes" project was started that aims to describe the genomic variation in 1001 *Arabidopsis* accession lines (Weigel and Mott, 2009). One of the possible ways to support this research from the bioinformatics perspective is to consolidate and re-analyse the wealth of data produced by previous research efforts (Ferrier *et al.*, 2010, Vanholme *et al.*, 2010, Katari *et al.*, 2010). This project directly contributes to this task through the development of an open, re-useable integration and analysis tools that facilitate the exploration of these data all the way from source databases and experimental data to network models and insights into the underlying processes.

1.4.1 Networks as a tool for interpretation of complex biological data

As highlighted in the previous section, a number of integration approaches rely on a common data model for providing access to the combined body of information. The definition of a suitable data model is of pivotal importance for the success of any integration effort – as this model must be both accessible to the end-user and expressive enough to represent potentially complex agglomerations of data of different types from a range of sources. Graphs or networks are now a common formalism that many such models are built upon. A graph representation is commonly applied to describe protein-protein interactions, gene regulation networks and metabolic pathways (Junker and Schreiber, 2008). A graph representation has a number of advantages as it defines a formal framework through which biological systems can be explored by computational and statistical means. Graph theory is the field of mathematics concerned with networks, their properties and organization (Diestel, 2005) and it is this field that has provided a number of the relevant tools for the analysis of biological networks.

The nodes and edges of a network can be conceptually bound to a particular schema, often represented as an ontology, which can themselves be described in terms of graphs. One of the ontologies often used in bioinformatics is the Gene Ontology (GO) (Ashburner *et al.*, 2000), which provides a controlled

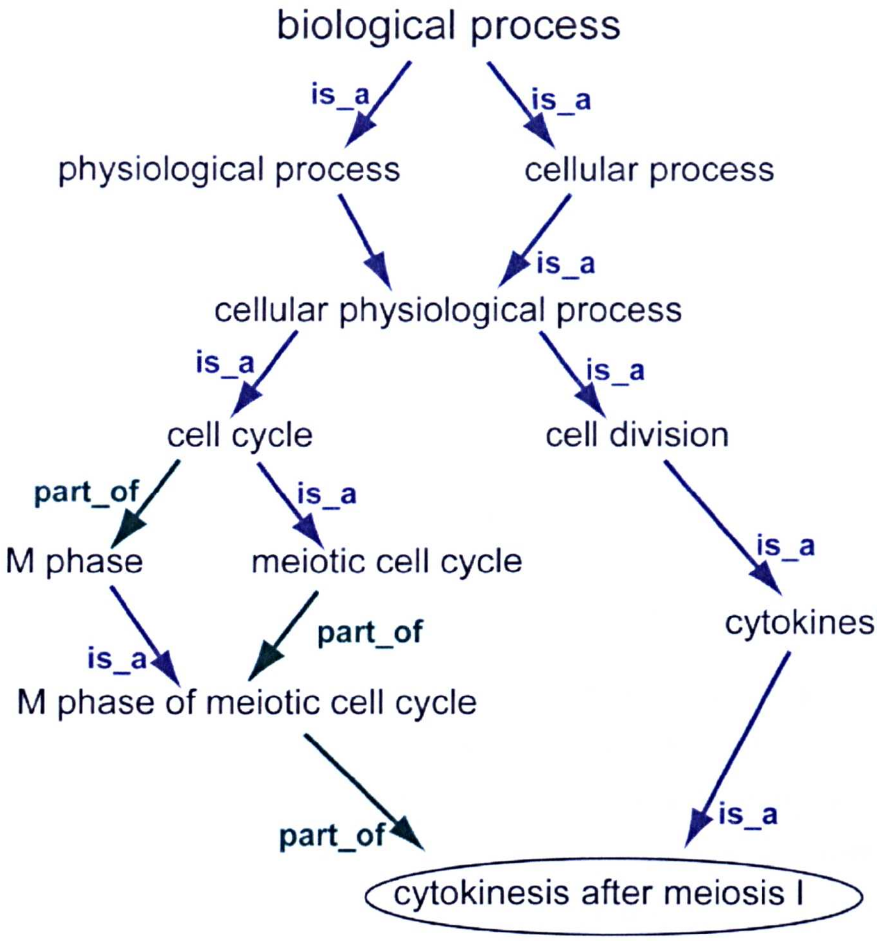


Figure 1.3 Biological function of a protein characterised using the Gene Ontology. Image source: *Saccharomyces* Genome Database website (http://www.yeastgenome.org).

vocabulary to describe the role of genes and proteins in terms of 'biological process' to indicate the biological purpose of the entity, 'molecular function' to identify its biochemical properties and 'cellular component' to specify its cellular localization or functional component (Ashburner *et al.*, 2000). In the GO ontology, each of these three branches is an independent taxonomy. Classified entities can have one or more assignments in each branch (**Figure 1.3**) and less specific terms higher in the hierarchy can be used if insufficient information is available for more precise assignment. This ontology is structured as a directed acyclic graph.

It has been demonstrated in a number of different studies (e.g. (Hwang *et al.*, 2005, Daigle and Altman, 2008, Troyanskaya *et al.*, 2003, Zhou and Liu, 2008, Lu *et al.*, 2005), that combining data from different types of high-throughput experiments and knowledge about properties of biological systems can increase accuracy, resolve ambiguity and help to get more useful information from the data. In such studies, the representation of the data as a network graph is now very commonly adopted as a convenient mathematical formalism both for managing the data and driving the analysis itself. In general terms, the aims of this approach are to determine the most probable biological network that fits the data and then to make inferences about the biologically relevant "unknowns" based on the available information.

Some types of biological data, like pathway information and protein-protein interactions naturally lend themselves to network representation. Other types of data, such as microarray expression profiling and proteomic assays, require additional analysis and interpretation steps to enable their results to be expressed in a graph formalism. For example, sets of gene expression measurements can be represented as a network of coexpressed genes once the similarities between expression signals have been identified and then converted into distances between the genes, which can then represent edges in the network. To maximise the scope of the data coverage offered by the data integration system it is often necessary to supplement the straightforward data conversion (or parsing) with more complex analysis methods. These methods can (re-)interpret the raw information and deduce secondary properties that are conformant to the common data model and can be added to the integrated dataset. However, once the effort to fit the data into the unified data model has been made, many of the post-integration data reduction tasks become easier. For example, many network clustering and topology analysis methods (such as measures of node centrality) that are used to group and identify important entities, are designed around the notion of a network.

1.5 DATA INTEGRATION AND FUNCTION PREDICTION

Integration of biological data is of particular importance for predicting the function of genes that have not yet been characterised experimentally (Re and Valentini, 2010). These methods often take advantage of multiple types of evidence both to increase the confidence in the assertions made and the number of genes for which a prediction can be derived. They also often rely on some form of a guilt-by-association principle, whereby function of a gene/protein is inferred on the basis of its association with other genes/proteins that have been functionally annotated. Such methods may rely on one type of data, like coexpression or sequence homology, or multiple types of data. In either case, data integration plays an important part in this process, as even in the cases where only one type of data is combined, it may still be advantageous to bring together data of the same type from multiple sources. The ability to predict function is particularly important for facilitating research into the species other than model organisms (Goodstein *et al.*, 2012), which are less well-studied and consequently often have relatively small numbers of experimentally determined functional annotations. This category includes crop species of plants, which are of immense importance commercially (Schoof and Karlowski, 2003). Even *Arabidopsis*, which is a main model species for plant biology, still has as much as 26% of all genes have no known functional annotation (TAIR, statistics derivation explained in chapter 4).

For this reason, computationally derived annotation methods are necessary for filling in the gap, where the functional characterisation methods are at present not capable of keeping up with the rate at which sequence data is being accumulated (Edwards and Batley, 2004). The simpler methods predominantly rely on the sequence homology information to derive predictions (Friedberg, 2006). Some prominent examples of such approaches include NetAffx (Liu *et al.*, 2003) pipeline, UniProt-GOA (Ensembl Compara (Vilella *et al.*, 2009)) and Blast2GO (Conesa *et al.*, 2005, Conesa and Gotz, 2008). The NetAffx pipeline is used to provide annotation for sequences represented on Affymetrix microarrays. Although predictions are based on the sequence-driven analysis only, it also integrates a wide variety of annotations from different sources, including Gene Ontology terms, protein domains, orthologous genes in other species and OMIM terms. Blast2GO focuses entirely on generation of Gene Ontology (GO) annotation based on statistical processing of BLAST output. The authors of the method also maintained an up-to-date resource of the results of Blast2GO analysis, which covered over 2000 different species. However, this resource has not been updated since 2010, when Blast2GO software was spun-out as a commercial service. The Ensembl Compara is a homology detection pipeline and an integral part of the Ensemble (Flicek *et al.*, 2012) framework. Transfer of GO annotations using orthology relations derived using this method is the main source for automatically generated GO annotations offered by UniProt and EBI. Both of these providers maintain functional annotation sets for multiple plant species.

A number of resources and tools were also created for predicting gene functions specifically for *Arabidopsis* and/or other plant species. Among them, some also predominantly rely on sequence homology methods to predict functional annotation. For example, PLAZA (Van Bel *et al.*, 2012) resource provides gene orthology information for 25 sequenced plant species derived using OrthoMCL algorithm (Li *et al.*, 2003), which is also used to derive GO annotations. More advanced methods and resource also use additional type of information, which can considerably expand the set of available annotations, as homology-based methods can only lead to a prediction in the cases where a similar gene was already characterised experimentally. In this way, using additional information of other types can increase coverage, but also improve accuracy – e.g. by identifying additional evidence to support functional annotations made. In particular, methods have been developed that can also use co-occurrence of gene names in literature (Li *et al.*, 2006), protein-protein interaction data (Kourmpetis *et al.*, 2011, Mostafavi *et al.*, 2008, Lee *et al.*, 2010, Bradford *et al.*, 2010), (co)expression (Kourmpetis *et al.*, 2011, Mostafavi *et al.*, 2008, Wabnik *et al.*, 2009, Li *et al.*, 2006, Lee *et al.*, 2010, Bradford *et al.*, 2010) and genetic context (Mostafavi *et al.*, 2008, Lee *et al.*, 2010, Bradford *et al.*, 2010) information for predicting gene function in plants. All of the methods listed here use multiple types of data for the analysis, and this set of examples demonstrates successful applications of both supervised (Kourmpetis *et al.*, 2011, Bradford *et al.*, 2010, Wabnik *et al.*, 2009, Li *et al.*, 2006, Lee *et al.*, 2010) and unsupervised (Mostafavi *et al.*, 2008) classification strategies for realising the guilt-by-association principle for functional inference.

1.6 The Ondex system

The Ondex system (Koehler *et al.*, 2005) is a realisation of a warehousing approach to data integration. The original implementation of the system was a re-imagining of the SEMEDA data integration system (Olson *et al.*, 1999) although since that time Ondex has undergone considerable re-structuring and re-development that has allowed it to take full advantage of the modern software tools, libraries and software engineering paradigms. At the time of writing, Ondex supports integration of data in excess of 40 different data formats. A number of the parsers, which enable this flexibility, were contributed as a result of the work undertaken to fulfil requirements from this thesis.

The import of the data from source files into the Ondex system is mediated via a range of parsers, which convert the data from their native format into the internal Ondex representation. The integration is achieved through the application of appropriate mapping method(s) (Weigel and Mott, 2009), allowing for the subsequent merging of equivalent entities. Another feature of Ondex is that the integration process and subsequent analysis can be formalised in the form of an Extensible Markup Language (XML)-formatted workflow. The use of workflows ensures both the transparency and simplification of the integration process through a user-friendly graphical interface. Ondex is implemented in Java and this means that it can be run on a variety of computer systems and the size of the datasets that can be effectively manipulated is only limited by the available system resources, mainly available memory. User interaction is further facilitated by the Ondex front end (Kohler *et al.*, 2006) application, which allows both interactive analysis and visual exploration of integrated Ondex datasets. The integrated datasets can be serialised in a proprietary format called OXL (Taubert *et al.*, 2007) or exported in a variety of commonly used representations such as Systems Biology Mark-up Language (SBML) and tab-delimited files.

Ondex was selected as both the main tool for the analysis and as the basis for the majority of the development efforts during the course of this project. Its strengths and limitations have greatly influenced the work described in subsequent chapters of this thesis. To put this work into an appropriate context, key aspects of the system are introduced in the following three sections which describe the organisation of the software, the data model and the capabilities of the Ondex front end. An additional, in-depth introduction from a programming perspective is also included in chapter 2, as it was felt this was necessary to explain important details in the work described therein.

1.6.1 The Ondex data model

Ondex uses a directed, typed multi-graph as the foundation for its data storage model. The model imposes a restriction on the graph that postulates that only one edge of the same type and the same direction can connect two concepts. Additionally, the concepts can be assigned as a "tag" to a set of other concepts or relations. This type of relationship is not visualised, but instead is used to represent set-type relationships between parts of the graph without introducing clutter. This formalism provides convenient way of selecting specific parts of the whole network (e.g. pathways or user-specified lists on nodes) as well as providing a handle for set-driven analysis (e.g. intersection, union and negation types of operations on parts of the graph).

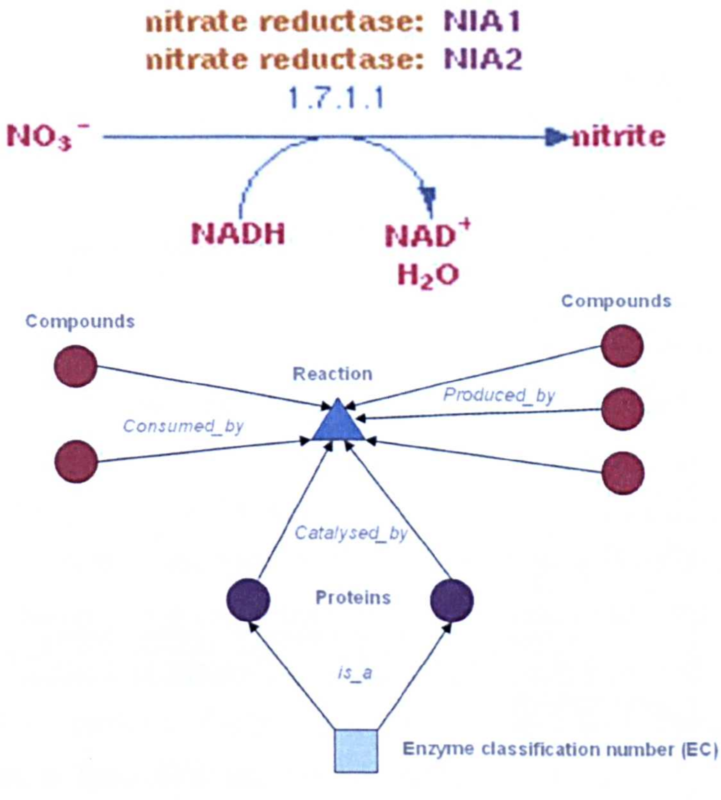


Figure 1.4 A reaction from AraCyc (top) represented as an Ondex graph (bottom) of information concepts and relations between them. The colors are consistent between the two panels and indicate equivalent types of data captured in the same concept classes.

To be converted into this formalism, the information in the source data resource is decomposed into a set of concepts and relationships between them by a parser plug-in module. To ensure the compatibility of different parsers and achieve some consistency in the way in which data are transformed into the Ondex core data model, there are pre-defined sets of concept classes and relationship types as well as accompanying guidelines for their use distributed with the main Ondex application. Concept classes and relationship types are arranged into a tree structured ontology with implied "is a" relationships between them. The "Thing" concept class and "related to" relationship types act as root terms for their respective ontologies. Each concept is only allowed

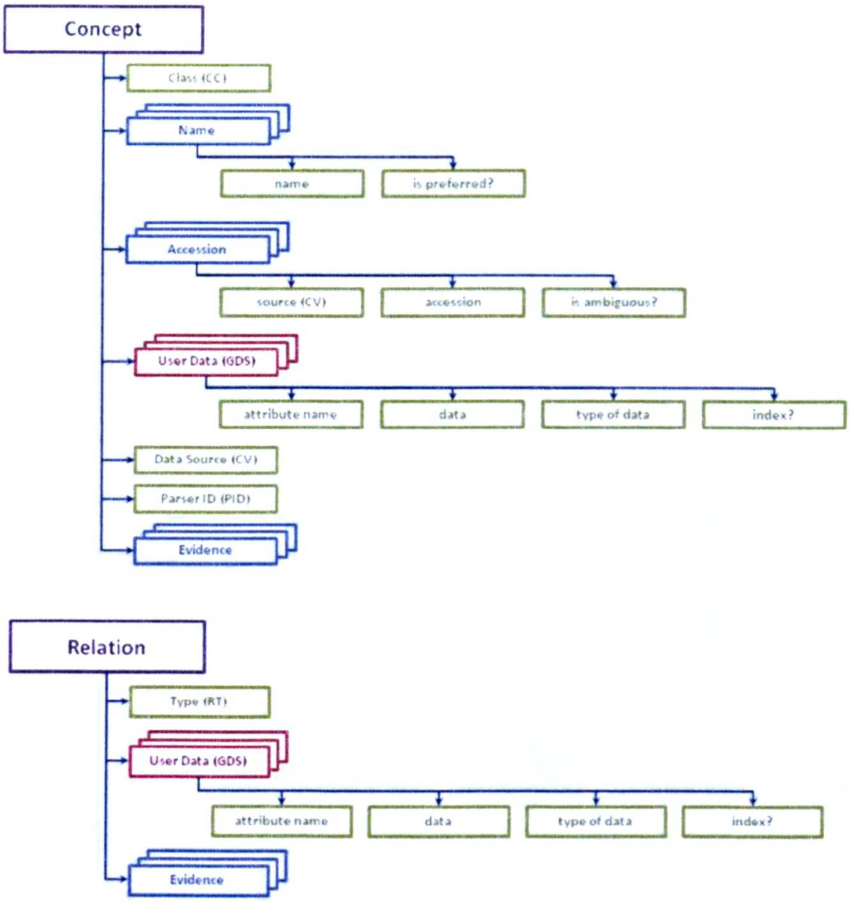


Figure 1.5 Overview of Ondex attribute model. This diagram shows the possible attributes on Ondex concept and relations, as well as the inner organization of the more complex elements in the core data model. Multiple boxes indicate that multiple instances of a particular attribute are allowed. Generalised data structure (GDS) attribute is a special case, as multiple instances are only allowed if they have different Attribute Name.

to have an outgoing relation to one parent. The relation type imposes a restriction on the multi-graph, in that just one relation of the same type and direction is allowed between any two concepts. **Figure 1.4** gives an example of how biological pathway data can be interpreted in terms of concepts and relations.

In addition to type-attributes, both concepts and relations can support a selection of additional attributes that allow storage of additional information about those entities. Some of these attributes are complex and are composed of several fields – the complete set of allowed attributes is shown in **Figure 1.5**. To facilitate integration, it is important to unambiguously identify pieces of information and certain fields that are key to the integration process are bound to specialised controlled vocabularies – those attributes are Evidence Types, Data Sources (both fields that capture provenance of a concept and identify an accession if it is present) and General Attribute Names. A set of general attributes allows some flexibility in the storage of additional information. Any number of attributes of this type is allowed, the only restriction being that the Attribute Names must only be used once within a set associated with the same concept. The collection of concept class and relation type ontologies and three controlled vocabularies for their attributes are collectively known as "Ondex metadata". Although, as mentioned above, a base metadata is provided with the main application, a user- and application-case-specific extension can be made as and when necessary during the integration process. For this reason, a separate, independent copy of the metadata is always associated with each graph instance – both in the in-memory and in the OXL-serialised versions. The graph structure, metadata and all of the information stored in various attributes constitutes a single instance of the Ondex integrated dataset. The data model itself is realised as a set of Java interfaces (implementation-independent contract declarations). This software architecture allows developers to build different implementations of system components which can still be seamlessly substituted within the same framework. This allows customized versions of Ondex to be built that extend or optimise aspects of performance to support particular applications. For example, two separate implementations of the Ondex data model can be used: an in-memory implementation, which is optimised for performance and a persistent implementation, which minimises the amount of memory resources used but is slower because it uses a database (Berkeley Database – Olson *et al.* (1999)). An optional indexing layer powered by Lucene (Prasad and Patel, 2005) is also provided for accelerated searching required by some of the analysis and integration methods.

1.6.2 Data integration, workflow engine and plug-ins

The Ondex data integration framework is made up of independent modules (plug-ins), which can be chained together to form workflows that are executed to realise required tasks. A new interface for composing, executing and storing workflows was designed as part of the work on this thesis and is documented in detail in chapter 2. This section introduces the five types of plug-ins allowed in the Ondex system and their roles. It also provides an outline of the way data

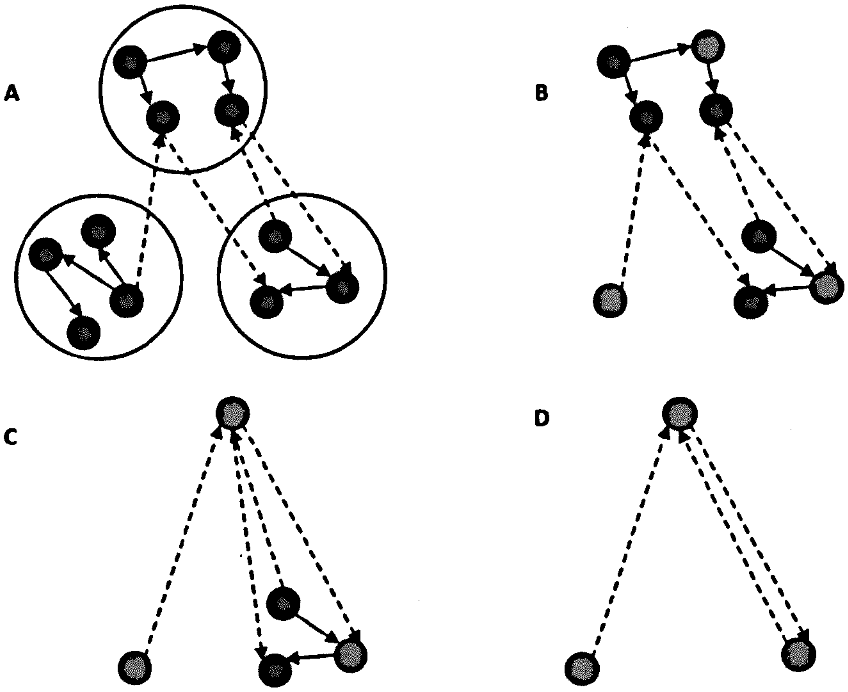


Figure 1.6 An illustration of the sequence of operations performed by the "relation collapser" transformer. (A) a network showing three connected components with respect to the edge type (solid line) that will be used to collapse the nodes. (B) the first group has been collapsed with all attributes from the removed nodes re-assigned to the remaining one. (C) the second cluster has been collapsed, note that in this case multiple edges have been re-assigned to remaining node.(D) the last remaining cluster has been collapsed, the two incoming edges have been merged to one edge.

integration process is realised in Ondex.

The integrated graph schema is populated by the execution of parsers. Once all of the data is converted into a common representation, the commonalities in it can be identified by the application of mapping methods (Lysenko *et al.*, 2009c). Mapping methods create the new relationships between the concepts (nodes) within the networks by evaluating the values of the attributes on those nodes (or in some cases – on the neighbouring entities). Some examples of mapping methods currently implemented in Ondex include accession-based (matches database accession identifiers on concepts), BLAST-based (creates relations by extracting the sequence attributes of nodes, passing them to the BLAST sequence comparison algorithm and parsing back the results).

Concepts identified as being equivalent entities through mapping methods can be aggregated by the application of transformers. For example, the "concept collapser" transformer copies all attributes and relationships from one concept to another and removes the redundant concept entries from the graph. When used in combination with appropriate mapping methods, this transformer can be a powerful tool for resolving complex patterns of redundant data both in terms of nodes, edges and their attributes. The illustration of this principle is shown in **Figure 1.6**. The graph in this example contains two types of edges - the solid line identifies the type of edges that is being "collapsed" (e.g. could be an edge type indicating equivalence), whereas the dashed line indicates another type of edges. At the first step, the transformer identifies all of the connected components with respect to the edges of the type to "collapse" (A). At the second step (B), a core node is created for each group that will inherit all of the attributes and edge associations of all group members. The transformer then proceeds to process each group at a time (panels B through D). First, the attributes of all other group members are copied to the core node, then the same is done for edges. Note that as Ondex data model only allows one edge of the same type and direction between the same pair of concepts, it is possible for the redundant edges to occur as well. In these cases, likewise only one core edge is retained per such group of edges that inherits all of their attributes. During the attribute-copying process the necessary checks for uniqueness are performed as required by the Ondex attribute data model (outlined in the **Figure 1.5**) and all duplicates are discarded (e.g. in the cases of data source or evidence type) or assigned a new, unique identifier (e.g. multiple, but different protein sequences from two different data source that contribute the two different concepts being merged). The combination of "accession-based" mapping that creates equivalence relations between concepts, followed by the "collapsing" on the "equivalent to" edge is one of the most often-used types of graph transformations in Ondex and was used for the construction of the datasets in chapters 4-6.

An additional use for transformers is to realise data abstraction. When this is the case, the entities being collapsed are not necessarily semantically equivalent. For example, if one data source annotates genes with GO terms and another source assigns them to proteins a complete set may be obtained by merging "Gene" and "Protein" concepts, given that one of the imported data sources provides the "encoded by" relation to identify the connection between them.

When Ondex is used for the construction of an integrated knowledge base, the integration normally entails the application of the parsers, mapping methods and transformers only. However, Ondex also supports a range of additional transformers and filters that allow further analysis and data reduction of the Ondex integrated schema. Some examples of tasks realised by such analyses implemented as transformers are graph analysis methods and clustering methods. Ondex filters allow data reduction by selectively removing entities from the network that meet a set of criteria specified by the user - e.g. by matching a combination of concept types or attributes or even by considering the attributes of the neighbouring entities in the network.

The final step of the workflow usually involves exporting the results. This can be done in the specialist OXL format, in which case the graph can be re-used in future analysis, integration or explored interactively in the Ondex front end. A range of other exchange formats can also be exported. For example, tab delimited formats, Resource description framework (RDF) or SBML. Some of the export plug-ins generate reports of the analysis performed on the graph, for example summary statistics of the number of nodes and edge or core graph-theoretic properties of the network. An overview of the key integration steps and parts of the system involved are shown in **Figure 1.7**. This example shows the parsers and mapping methods used in Pesch *et al.* (2008), which used Ondex to compare different approaches to transient mapping across different data sources for the functional annotation of *Arabidopsis* proteins.

1.6.2.1 Ondex front end

The Ondex front end graphical user interface supports visualisation and analysis of the graph structure of an Ondex knowledge base. The visualisation engine uses the Jung graph library (White *et al.*, 2004), which allows both customisation of appearance for various graph elements and mediates the user interaction with the graph. Similar to the Ondex workflow, the interactive visualisation and analysis components are also realised as a set of plug-ins. The front end allows some of the workflow plug-ins to be re-used – for example, all of the filters are shared between the workflow engine and the front end. However, in the front end the filters just change the visibility of graph elements, whereas in the workflow they remove these elements from the network. The tasks of changing colours, shapes and labelling of graph elements is handled by a front end-only type of plug-ins called "annotators".

As some of the networks are too large to visualise effectively, the Ondex front

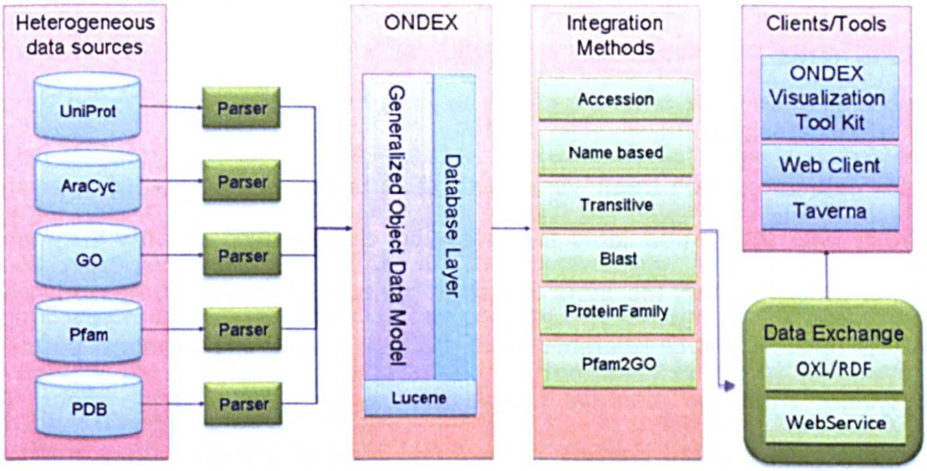


Figure 1.7 The organization of a typical Ondex workflow. This diagram breaks down a typical Ondex data integration pipeline. Databases are imported via the specialised parsers into the Ondex graph representation. The data can then be further manipulated using the data integration methods, exported and visualised in Ondex front end. Image from Pesch *et al.* (2008).

end also supports a "distilled" view that displays useful summary information from the knowledge base to be displayed as another as a network layout called a "metagraph". The metagraph only shows the types of concepts and relations found in the graph and the relationships that exist between them. An edge is drawn in the metagraph between pairs of nodes (representing concept classes) if there is at least one instance in the actual graph of the edge of matching type, direction and concept classes of source and target in the actual network. The actual numbers of each occurrence are also counted and can be read by clicking on respective graph elements. The counts of all the attributes are also listed in tabbed panes, accessible from the main metagraph window. The metagraph uses shapes and colours for concepts and relations that are consistent with the main graph layout window and so it plays a supporting function as a key to the main graph.

Another way to interact with the contents of an Ondex knowledge base is through a command-line console. This interface was introduced during the work on this project and is describe in detail in chapter 2. The command console now supports approximately 250 different function calls. The advantage of the command-line interface in the console is that it provides a command scripting environment for more experienced users and also gives the user the flexibility to customise the visualisation commands to suit their specific needs.

1.7 Discussion

Ondex data integration framework exists in an increasingly crowded ecosystem of other data integration tools for biological data. However, only very small number of tools offers the same degree of generic applicability, flexibility and analysis options at a comparable level. Perhaps one of the strongest points of the system from the practical perspective is the ease of deployment. The system is implemented entirely in Java and none of the core functionality is reliant on any external dependencies. This means that the installation is as simple as unpacking the distribution and is completely platform-independent. The Ondex system also includes both data integration and graph visualisation components, which are designed to be interoperable and capable of exploiting the same graph-based data model. Because Ondex offers both of these functionalities it actually competes with two different types of tools – data integration/workflow environments and graph visualisation software. In the first category, the two most prominent solutions that offer similar degree of analysis power and flexibility whilst still offering a degree of platform neutrality are Taverna (Hull *et al.*, 2006) and Galaxy (Giardine *et al.*, 2005). The prominent examples in the latter category are Cytoscape (Smoot *et al.*, 2011) and Gephi (Bastian *et al.*, 2009).

Taverna is a workflow management and execution system, which allows construction of workflows from the web service based components. The system is split up into three components Taverna Engine (workflow execution), Taverna Workbench (GUI client) and Taverna Server (remote workflow execution). The Taverna environment is more advanced than the one offered as part of the Ondex system, and allows much greater complexity of constructed workflows, which can be easily viewed and monitored via interactive graph visualisation in the Taverna Workbench client. As the plug-in components are webservice-based the overhead of implementing and deploying them can be lower. This is because only generic and widely used set of technologies are needed to implement them, as opposed to tool-specific application programming interface (API), like Ondex. However, Ondex workflow components can be packaged and distributed as files and run locally on the user's system. This may be advantageous in the situations where greater security is necessary or large volumes of data need to be processed, as it does not necessitates transfer of information via the Internet to pass it between different remote services. Also, as Ondex plugins predominantly operate on the Ondex graph representation, they are intercompatible with each other by default, which makes it easier to re-use components between application cases and reduces the overhead – it is normally not necessary to write converters to bridge inputs and outputs of different components, as is the case in Taverna.

Galaxy offers a browser-based environment for workflow execution, with the interface design largely similar to that of Ondex Integrator. Although it can be argued that this makes the Galaxy server more difficult to deploy, from the user's perspective the access to the system is largely seamless and platform-independent. Galaxy workflow components are very loosely restricted in terms of the types of data they operate on and place no format requirements at all in terms of its format. This is a very different approach compare to Taverna (data passed between workflow components must be wrapped in XML) and Ondex (all data must be imported into Ondex graph data model). In fact, the workflow components used in Galaxy are often just wrappers that delegate to various external tools or even scripts. This can cause considerable complications when it is necessary to make components available on a different server, as the overhead would effectively be the same as having to re-deploy the underlying tool or script. The loose typing also imposes considerable overhead of having to write format converters.

In terms of graph visualisation tools, a very similar approach is taken by Gephi, Cytoscape and Ondex frontend. All tools provide ways to visualise and annotate networks (colour, shape and size), search for particular components and alter their visibility. All three systems also provide a way to extend the basic functionality by adding in additional plugins, which offer other analysis methods. Gephi is unique in terms of being the first major Java-based graph visualisation tool to implement full hardware acceleration, which is essential for ensuring adequate performance when working with large networks. Cytoscape has a very wide variety of plugins and is supported by a large community of users. However, both of these tools operate on a very loosely typed graph that only describes a basic graph structure and does not formally specify any metadata about the attribute structure of nodes, edges or their properties. It can be argued that the more complex data model of the Ondex graph can more effectively model complex datasets and ensures greater intercompatibility of graphs between different application cases.

Because of the investment into both workflow driven integration and graph visualisation domains Ondex system has functionality that can complement other tools in both of these domains. In 2008 further development of Ondex was funded by the BBSRC SABR grant (BB/F006039/1). This allowed further expansion of the system and part of these developments was to add greater interoperability with other major bioinformatics tools. To that end, an extension was developed that allowed Ondex workflow components to be wrapped as Taverna-compatible web services. This enabled both the execution of Ondex-specific workflows in Taverna and the re-use of Ondex workflow components independently of the rest of Ondex. From the perspective of Taverna, Ondex was able to offer a persistent graph-based data management, which was previously not readily possible as part of the system. The results of this work are still being prepared for publication at the time of writing. There was also a plugin developed that allowed the use of Ondex graphs in Cytoscape (Weile *et al.*, 2011). This implementation took advantage of Ondex typing system for nodes and edges for constructing abstracted network views in Cytoscape environment.

As is demonstrated by these two examples, Ondex system has additional functionality that can complement other existing tools. However, there is also one additional component of the system that differentiates it from similar tools in both domains. This component is the underlying graph-based data model, which is backed by a controlled vocabulary and supports typing of attributes, nodes and edges. The assignment of the formally defined types provides a common point of reference both for the data integration components of the workflow system and plugins in the Ondex frontend. This simplifies the design of complex operations on the graph, where it is often necessary to evaluate graph elements based on their meaning. For example, different types of operations would be meaningful for nodes representing individual pathways as opposed to those representing genes. Further discussion about Ondex data model and its technical specification is included in chapter 2 of this thesis.

2 EXTENSION OF ONDEX DATA INTEGRATION SOFTWARE

2.1 SUMMARY

The work presented in this thesis has two main components. The first component is the development of tools and methods to facilitate the integration and analysis of biological data, which were realised as part of an open-source software framework called Ondex. The second component demonstrates the relevance of the methods developed through three different research application cases. This chapter presents the two major technical contributions made to the Ondex system as part of this work and explains how they fit into the rest of the Ondex software architecture. The first such contribution was the development of the GUI-driven tool for workflow construction. This development was essential for effective management of a large and complex collection of workflows necessary to generate and keep up-to-date the datasets used in the subsequent chapters. The second contribution was the scripting environment for the Ondex front end. The introduction of the scripting functionality allowed analyses to be done on a much finer level of granularity and with increased flexibility than was previously possible. This was particularly important for the efficient interrogation of the integrated datasets in Ondex front end.

2.2 INTRODUCTION

The Ondex data integration system (Kohler *et al.*, 2006) supports the import of a variety of data sources and exchange formats and offers a wide and ever-increasing suite of tools to analyse, query and visualise these data. The work described in this thesis has relied on many pre-existing features of the system. However, there were also many areas where new functionality was necessary to import additional sources of biological data, add novel analysis methods or to improve the usability of the software itself. Many new features and improvements were therefore introduced to the Ondex system as a direct requirement of the work carried out during this project. Contributing to a project under active development has also incurred some costs in the form of having to provide support for the new features, contribute to its end-user documentation and update it to keep up with the changes of the code in the main project it relies on. As Ondex is open-source software with a number of academic users, these software development achievements constitute an important contribution both to the wider research community and to the field of tool development for bioinformatics as a whole.

The full extent of the changes made to the Ondex system during this study is too broad and interwoven with the work of other developers working on Ondex to be fully covered within this thesis. Therefore, only the developments that constitute a complete and fully functional additions to the system, which were initiated and realised as part of the work for this thesis, will be mentioned here. For simplicity, the implementation of the more specific methods related to particular aspects of the application cases are described in the corresponding chapters. This chapter covers the extensions of general relevance only, such as the analysis methods that are applicable to a wide range of data integration problems or improvements that make the system more reliable and easier to use. In this category, two important contributions were made to the Ondex system: a new workflow management framework and the development of Application Programmers Interfaces (APIs) to other programming environments. Other smaller developments include a library of tools to facilitate data exchange in the tabular format, integration of clustering tools, statistical analysis methods and new visualisation approaches.

The needs of the application case have also pushed the limits of what was technically possible in the Ondex system and this motivated the introduction of a large number of new data representation formalisms and analysis methods that went far beyond the original remit of Ondex as primarily a data integration platform. Therefore, the work to extend the core functionality was chiefly motivated by the need to be able to express and manage this newly introduced complexity. For example, a more flexible workflow management solution was required to effectively deal with the number of workflows and plug-ins available, and scripting functionality provided by the interface to JavaScript allowed an interactive creation of fine-grained analysis scripts tailored to specific problems. Another important motivation was to increase the productivity – if the repetitive and error-prone tasks can be automated, more time may be spent on the actual data analysis. Addressing these issues resulted in a net gain in amount of work that was done on the biological application case during the course of this project - although some time was spent on development and support of these features, even more time was saved as a result of having them in place.

2.3 Development of a new workflow management and execution system

It was identified very early in the project that an effective approach was needed for creation and management of a large number of workflow files. As all of the interactions with the data integration part of the Ondex system are mediated by XML workflows, in excess of eight thousand workflows were created to support the data integration, analysis and testing during development described in this thesis. The primary reason for such a large number was the need to accommodate the need of being able to refer back to the analysis done with a particular version of Ondex or plug-in. Simplifying the management of this complexity and making the workflow creation process simpler and less error-prone was one of the main motivations for extending the existing Ondex system. Another shortcoming of the original implementation was the severe restrictions on the types and formats of data that can be operated on by a plug-in. The former issue was dealt with by creating a user-friendly tool for creating and editing workflows. This became known as the Ondex Integrator. The latter issue was addressed by defining and implementing a generic Ondex workflow API that generalised the original plug-in API and could be used to define a more advanced workflow enactor system.

The original Ondex system offered basic functionality to script workflows using a simple XML format. The workflow execution was handled with one Java class that parsed this XML file, instantiated appropriate plug-ins and executed them. The workflow parser relied on a specific package structure in order to resolve the class names and locate the necessary plug-ins. Although this solution allowed some data integration tasks to be performed, it also came with a number of drawbacks that resulted in a considerable overhead.

Throughout this project, the Ondex system was under very active development and the behaviour, naming and arguments of the plug-ins changed frequently. This made working with Ondex as a user and developer a challenge, and the situation was made more challenging by the fact that the documentation was not ready and was generally only to be found in the source code accessible as comments or values of specific arguments. If, for example, the behaviour of the plug-in was changed, the only symptom was usually the failure of a workflow to complete successfully or, worse still, incorrect modifications or artefacts in the graph that were only evident in the later stage of the analysis – and resolution of these issues would usually be very time consuming.

Another set of complications arose from the manual editing of the workflow definition files. This process necessitated that the user either commits to memory all of the correct identifiers of Ondex plug-ins and their arguments or refers to the source code in order to find the correct parameters. However, the number of Ondex plug-ins is now well past the one hundred mark and on average they have around eight configurable options. Manual entry of such a large number of parameters was also found to be particularly error-prone.

From the developer perspective, the close coupling of the workflow execution to the process of parsing the workflow description file made the system inflexible and difficult to improve, extend or debug. Although the API for the definition of plug-ins allowed them to be easily executed programmatically, this was not the case when the task necessitated the construction of a workflow programmatically. It was clear that the implementation also constituted bad practice from the object-oriented software design perspective, as it breaks two of its core principles - encapsulation and separation of concerns (Wu, 2006). The encapsulation principle calls for all the code and variables that are needed to carry out a particular task to be enclosed within their modules, with only a minimal number of well-defined inputs and outputs passing between the modules. Separation of concerns requires each individual module of code to be designed to carry out one particular task.

2.3.1 Overview of the new system architecture

The tasks of creating, input/output (IO) and executing workflows are realised by three top-level modules. To support this architecture an additional level of workflow API is required – one set of classes that stores the definition of available workflow components and another set that holds the bindings of selected modules to the values of their arguments once they have been configured. These sets of classes become the main form in which the information is exchanged between three parts of the system. An additional channel of communication is also necessary in order to capture and handle possible errors that could arise in any of the parts, and wherever possible are collected and reported to the user by a unified set of error-handling classes.

The module that manages workflow creation also builds and maintains the complete index of all available plug-ins, the associated documentation and the name and specification of their arguments. A sub-module deals with keeping track of individual workflows that were created, which take the form of a list of selected components and the values of the arguments that have been set. Although this module holds the references to all of the available components, they are not instantiated until needed in order to keep the memory foot-print low and keep the time needed for indexing to the minimum. This module also has the functionality to carry out a set of input validation tasks – like checking the integrity of the workflow structure (e.g. that there is no plug-in scheduled to be executed that needs an Ondex graph instance before a plug-in that creates one) or verification of individual arguments against their specifications (e.g. that all of the required arguments have been set for all plug-ins and the correct type of value was supplied for them). As this module is not coupled to a particular input format, it allows the functionality of constructing the workflow either dynamically or from a workflow description file. The information about what plug-ins to run and the associated options is held internally in a task description class.

The task description class holds the information in a form that is suitable for interactive editing or being saved in the file. Internally, this representation forms a tree data structure of the various elements corresponding to the levels of organisation a workflow – e.g. workflow (root), workflow component, workflow component parameter. The interactive creation of the workflow is handled by populating the task description via the Ondex Integrator graphical user interface (GUI). The task description format is also used by the export and parser classes that save and load the workflows from the XML-formatted Ondex workflow file format. These two groups of classes constitute the workflow creation and IO modules.

The task description representation is converted to the workflow instance representation that is used by the workflow enactor and also creates and maintains the reference that links the components in the workflow instance to the original specifications in task description. This reference can be used to produce meaningful error reports, which relate users' inputs to the problems identified, and to provide feedback about the workflow execution progress.

To accommodate this new workflow representation, a new XML format to describe Ondex workflows was developed. The implementation allowed for backwards compatibility, and supported reading of the original workflow XML format by parsing in to a workflow that conformed to the new formalism, but realised original type of behaviour. This was possible because the new format retained the same basic set of information needed to re-construct the workflow. The only types of information that the older format did not provide were filled in with by assuming that only one Ondex graph was being used and that (with the exception of the Ondex graph) only the simple data types (e.g. string and numbers) where passed to the workflow plug-ins.

2.3.1.1 Ondex Plug-in API and pre-existing functionality

The Ondex plug-in API defines five types of plug-ins all of which implement an `AbstractONDEXPlugin` interface. From the workflow enactor point of view, the plug-in is seen purely in the terms of inputs that it requires and outputs that it produces. The `AbstractONDEXPlugin` interface describes the unifying features of all Ondex plug-ins – i.e. the inputs and outputs that all of them must have. Therefore, all plug-ins need one instance of `ONDEXGraph` and an object that holds the collection of arguments for that plug-in (`ArguemrntDefinition` class) as inputs. The interface does not define any inputs, so these must be handled based on a type of a plug-in. Some of the code needed to run specific types of plug-ins is also externalised and is located in another class. According to this scheme, the following steps are taken to successfully execute an Ondex plug-in: create and instantiate an instance of a plug-in class, get a plug-in specific argument container from it, populate the container with appropriate configuration arguments, pass the container back to the plug-in, pass an instance of Ondex class to the plug-in, run the pre-amble external code, execute the plug-in and run the post-amble external code.

2.3.1.2 Plug-in registry and workflow description API

Ondex plug-ins are available as separate optional modules and for this reason any particular Ondex installation will not have the information in advance about which plug-ins (and which versions) will be available. The loading of plug-ins is therefore managed by the `PluginRegistry` class. Upon the initialization of the `PluginRegistry`, specified plug-ins directories are scanned for plug-ins and an index is assembled of everything that is available. All plug-ins in the index are referenced using a special unique identifier field and full record also holds relevant documentation, the list of arguments, argument types and restrictions on them. This representation is a descriptive read-only record that makes it possible to discern the correct set of configuration options for that plug-in. The workflow is assembled by creating a task entry instance that holds a reference to the original type of the plug-in as well any configuration arguments supplied. The list of all tasks is then deposited in the order they are to be executed in the Task class.

The Task class, task entry and plug-in descriptor are all generic and do not require the instantiation of any of the actual Plug-in classes in order to work. This allows the separation of the workflow execution from the workflow assembly, configuration and storage tasks. In theory, using this API it is possible to define a `PluginRegistry` implementation that resides on a server and provides the plug-in descriptors to the client on a different computer. The workflow can then be assembled and configured on the client-side and task description sent back to the server for execution. As this representation is independent form the plug-in execution API, the user interface layer that uses it is not directly affected by the changes to the workflow enactor and vice versa.

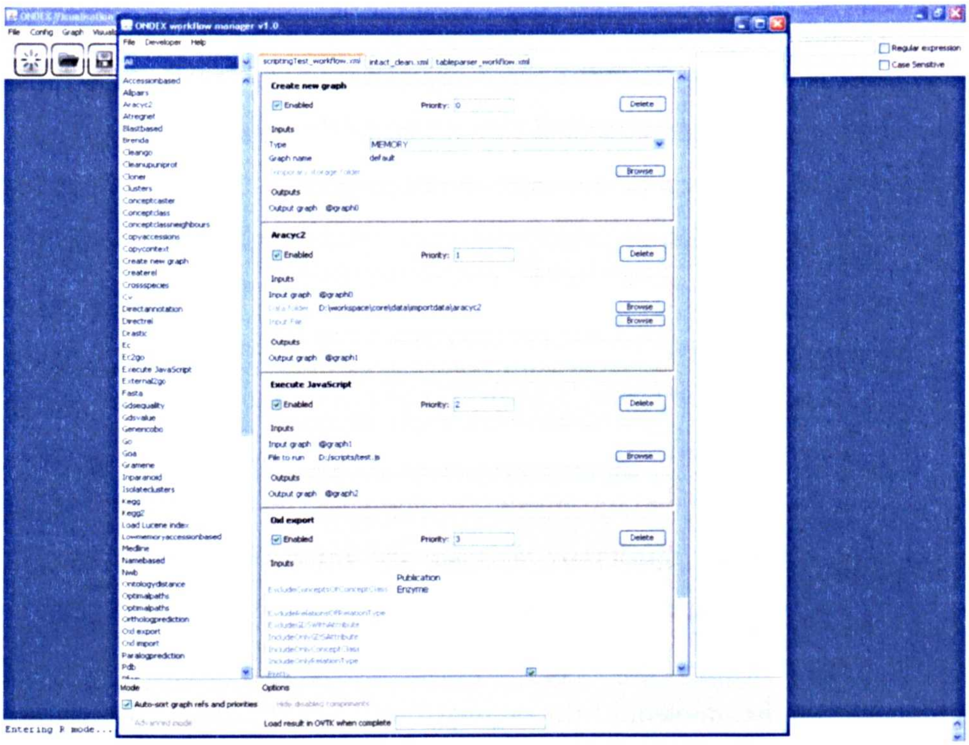
2.3.1.3 Workflow enactor

The Ondex workflow enactor system introduces an abstraction layer that allows most the complexity of the plug-in execution process to be bypassed. Like many other workflow implementations, the basic building blocks of the workflow enactor model are processors and data links. Task processors are components of the workflow that carry out specific tasks and have multiple inputs and outputs, whereas data links determine how outputs of one processor are matched to the inputs of the other. When defined in this way, all of the plug-ins can be executed in exactly the same manner and all the workflow system needs to do is to do it in the correct order and pass the output and input objects to the correct task processors. Although a much better solution would have been to refactor the Ondex plug-in API to conform to this simple model and fully encapsulate pre- and post-amble code inside the respective plug-ins, this was not possible due to the amount of time that would have been needed for such an extensive refactoring. Instead, the original system of plug-in execution was modified to become more modular, the original Ondex plug-in API left intact and the processor interface was added as a wrapper. This has partially solved the problem of the original specification lacking some of the necessary functionality – like the ability of plug-ins to create new data objects as it became possible to define a new type of plug-in using the processor interface directly.

2.3.1.4 Ondex Integrator tool

The Ondex Integrator is a GUI that facilitates the tasks of workflow creation and editing. It provides an easily accessible way for the user to browse through the plug-ins that are currently available in their particular installation of Ondex and an intuitive interface for maintaining and modifying the XML files, which store Ondex workflows, It also allows workflows to be validated and executed (**Figure 2.1**).

The first version of the workflow management GUI was released in 2007 under the name of "Ondex Workflow Launcher". This tool was the first user interface ever created for the Ondex data integration back end. This version was integrated with the Ondex front end and allowed seamless exchange of the graphs between the back end and front end parts of the Ondex system. It was possible to run the analysis directly on the graphs loaded in the front-end or to immediately visualise the results of the integration without the need for a lengthy procedure of saving the graph on disk using the Ondex back end and loading it in again in the front end.



scriptingTest_workflow.xml | intact_clean.xml | tableparser_workflow.xml

Create new graph
☑ Enabled Priority: 0
Inputs
Type MEMORY
Graph name default
Outputs
Output graph @graph0

Aracyc2
☑ Enabled Priority: 1
Inputs
Input graph @graph0
Data folder D:\workspace\core\data\importdata\aracyc2
Input File
Outputs
Output graph @graph1

Execute JavaScript
☑ Enabled Priority: 2
Inputs
Input graph @graph1
File to run D:/scripts/test.js
Outputs
Output graph @graph2

Oxl export
☑ Enabled Priority: 3
Inputs
ExcludeConceptsOfConceptClass Publication
Enzyme
ExcludeRelationsOfRelationType
ExcludeGDSWithAttribute
IncludeOnlyGDSAttribute
IncludeOnlyConceptClass
IncludeOnlyRelationType

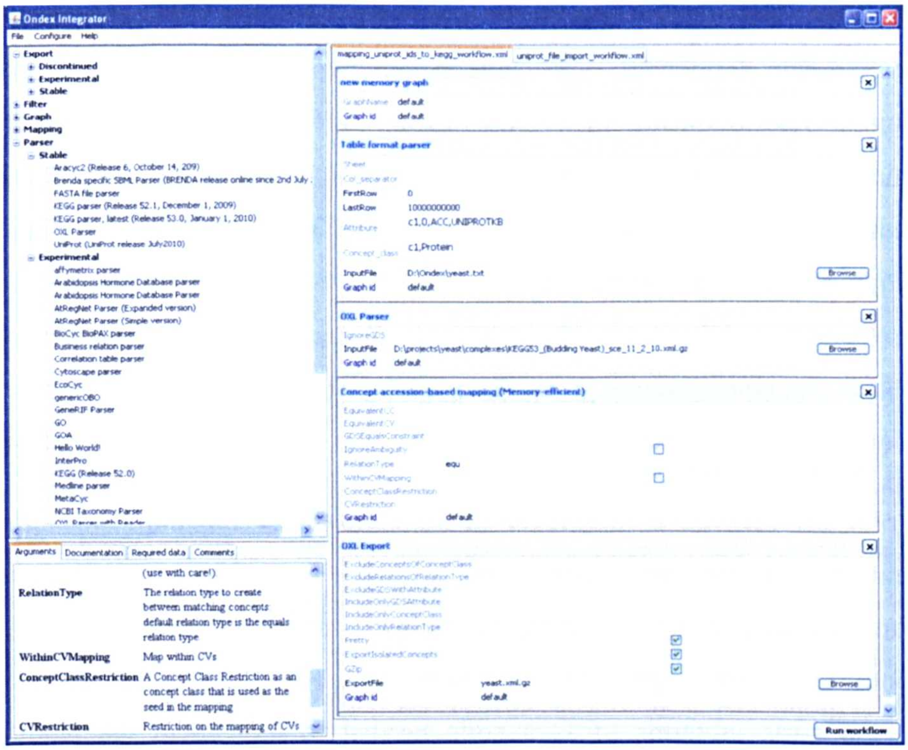
Mode
☑ Auto-sort graph refs and priorities
Advanced mode

Options
Hide disabled components
Load result in OVTK when complete

Figure 2.1 Ondex Integrator v1.0. This is a screenshot of the first version of the Ondex workflow management tool, released in 2007. The panel on the left lists all available plug-ins, which can be filtered using a combo box above it. In the centre is a workflow editing tabbed pane. It lists currently loaded workflow files in separate tabs and lists individual components and their configuration. New elements can be added by selecting them from the list on the left, by dragging and dropping them from another position in the workflow or by copying and pasting existing elements. A list on the left holds the references of all previously created graphs. These references can be used in subsequent workflows or loaded for viewing in Ondex front end.

The graphical user interface works directly on the task description representation. The advantage of this approach is that, since this representation does not depend on any of the actual plug-in implementation classes, the configuration and arguments can still be recovered even when the workflow file is no longer in sync with the version of the plug-ins installed on the system.

Workflows can be constructed by selecting a type of plug-in from the list – a plug-in of that type is than added to the workflow at the position specified. Each plug-in entry shows all of the valid arguments, their default values and allows access to the available documentation. Any available documentation about the plug-in itself and each of its arguments was displayed in the tooltips of corresponding user interface elements. The GUI supports all of the common


Ondex Integrator
File Configure Help
Export
Discontinued
Experimental
Stable
Filter
Graph
Mapping
Parser
Stable
Aracyc2 (Release 6, October 14, 209)
Brenda specific SBML Parser (BRENDA release online since 2nd July .
FASTA file parser
KEGG parser (Release 52.1, December 1, 2009)
KEGG parser, latest (Release 53.0, January 1, 2010)
OXL Parser
UniProt (UniProt release July2010)
Experimental
affymetrix parser
Arabidopsis Hormone Database parser
Arabidopsis Hormone Database Parser
AtRegNet Parser (Expanded version)
AtRegNet Parser (Simple version)
BioCyc BioPAX parser
Business relation parser
Correlation table parser
Cytoscape parser
EcoCyc
genericOBO
GeneRIF Parser
GO
GOA
Hello World!
InterPro
KEGG (Release 52.0)
Medline parser
MetaCyc
NCBI Taxonomy Parser
Arguments
Documentation
Required data
Comments
mapping_uniprot_ids_to_kegg_workflow.xml
unprot_file_import_workflow.xml
Browse
Run workflow

	(use with care!)
RelationType	The relation type to create between matching concepts default relation type is the equals relation type
WithinCVMapping	Map within CVs
ConceptClassRestriction	A Concept Class Restriction as an concept class that is used as the seed in the mapping
CVRestriction	Restriction on the mapping of CVs

new memory graph ☒
GraphName default
Graph id default

Table format parser ☒
Sheet
Col_separator
FirstRow 0
LastRow 10000000000
Attribute c1,0,ACC,UNIPROTKB
Concept_class c1,Protein
InputFile D:\Ondex\yeast.txt
Graph id default

OXL Parser ☒
IgnoreGDS
InputFile D:\projects\yeast\complexes\KEGG53_(Budding Yeast)_sce_11_2_10.xml.gz
Graph id default

Concept accession-based mapping (Memory-efficient) ☒
EquivalentCC
EquivalentCV
GDSEqualsConstraint
IgnoreAmbiguity □
RelationType equ
WithinCVMapping □
ConceptClassRestriction
CVRestriction
Graph id default

OXL Export ☒
ExcludeConceptsOfConceptClass
ExcludeRelationsOfRelationType
ExcludeGDSWithAttribute
IncludeOnlyGDSAttribute
IncludeOnlyConceptClass
IncludeOnlyRelationType
Pretty ☑
ExportIsolatedConcepts ☑
GZip ☑
ExportFile yeast.xml.gz
Graph id default

Figure 2.2 Ondex Integrator v2.0. In this version of Ondex Integrator the user interface was streamlined by removing unused elements. The documentation was moved from tooltips to a separate set of tabbed panes (bottom-left). The plug-ins display was also changed to a tree, which made it possible to sort plug-ins by status as well as type. The workflow execution progress is reflected by the changes in the colour of the workflow component elements (displayed in the tabbed pane on the right).

types of arguments currently defined as part of the Ondex plug-in API and can provide a different input control that corresponds to each type – e.g. a multi-line list where there can be several arguments of this type or a check box if the argument can only take a value of "true" or "false". The inputs and outputs of plug-ins can also be assigned identifiers that determine the flow of resources between different workflow components. The GUI interface delegates to the other parts of the workflow API for validation of the workflows created, their execution and saving and loading of files.

2.3.1.5 New features implemented in the Integrator 2.0

Ondex Integrator v2.0 (**Figure 2.2**) was released in 2009 and featured a number of improvements to the interface. This version was no longer integrated into the Ondex front-end and was instead released as a stand-along tool. The interface was updated to make the best use of the new plug-in annotation framework, which was contributed by other developers on the Ondex project. The annotation framework uses Java doclet technology to allow a more structured description of Java class files. Unlike the standard JavaDoc comments, this information can still be accessed at run-time. By introducing this annotation, it was now possible to easily keep track of the plug-in development state and sort them into "stable" and "experimental" categories in the user interface. When more documentation is available, it is presented to the user in several tabbed frames at the bottom-left of the main window.

2.4 SCRIPTING API FOR ONDEX

The plug-ins provide a high-level of customisability for the Ondex data integration and analysis pipeline by allowing a number of commonly encountered tasks to be solved by combining a number of generic reusable components. However, packaging the necessary code as a plug-in introduces additional costs in terms of Java code needed to implement the required interface methods, declare required inputs and unwrap the arguments. In some cases the amount of code required to set up an Ondex plug-in actually exceeds the amount of code that actually tackles the task itself several fold.

To address this problem, the need for a lower-granularity interface was identified. This would be more suited to tackling a wider variety of tasks using more specific and simpler reusable components. Such capabilities can be supplied by a scripting language and a library of appropriate functions. Functions can easily be chained together in much the same way the workflow components can, but are usually designed to be much more specific and have much lower implementation overhead. For this reason, functions are a much better choice for the implementation of application-case specific processing and analysis routines. An additional advantage conferred by the scripting interface is the ability to interactively access the code written in other programming languages, without the need to re-implement it in Java or the need to develop a specialized exporters and parsers to allow the round-trip data exchange between Ondex and other tools.

The need to allow this type of interaction between Java and other programming environments is well-recognised in the Java developer community and there are a number of interpreters and interfaces for other languages available for Java. Nevertheless, realising a general scripting interface is still less than straightforward, as different programming languages frequently have incompatible semantics and mode of operation. The scripting language libraries often only support part of the functionality - either in the core implementation of the scripting language or in the way Java objects are handled within it. This necessitates a creation of appropriate wrappers for the set of classes to be scripted, which re-package the original Java representation to make it compatible with the scripting library of choice. As creation and maintenance of a complete set of wrappers is prohibitively time-consuming and the types of the modifications needed are usually clearly defined, this process was automated using a byte code generation solution (JavaAssist v3.12.0). A byte code generator provides a way of creating and modifying Java classes at run-time and can therefore provide a way of re-generating a set of wrappers without the need to change the main application in any way. Therefore, the source classes may be modified by other developers without the need to update the bindings to the scripting interface, which will always be in sync automatically. It also makes it possible to add additional functionality for the scripting environment as add-on modules even after the main application has been built.

The API defines and manages the execution of a set of abstract tasks needed to load and maintain a scripting solution, like the initialisation of the scripting environment, generation of wrappers, clean-up, user interaction and error reporting. As the API itself only defines the generic structure of relating a scripting solution to a set of core Ondex classes and provides some utility methods to facilitate the process of defining an appropriate set of wrappers, a number of different scripting solutions can be seamlessly supported within the same framework. Any of the implemented scripting environments can be accessed either by supplying a script as an argument to a scripting plug-in in the Ondex workflow, or interactively, using a command console in the Ondex front end.

2.4.1 JavaScript

JavaScript syntax and semantics are very similar to Java and since Ondex is

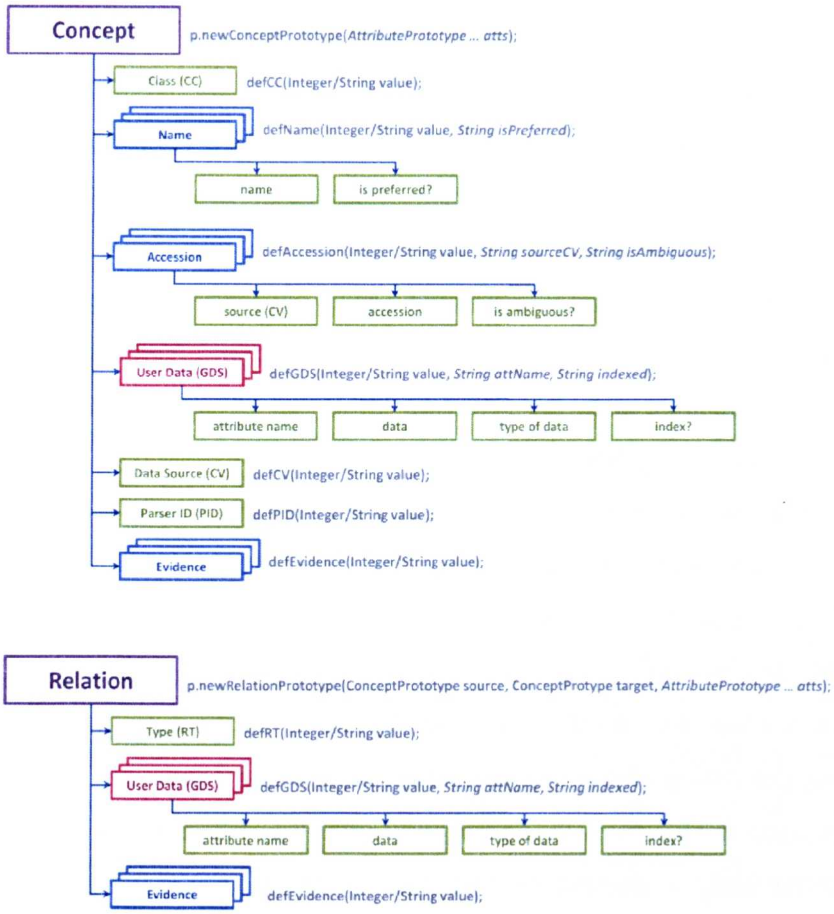
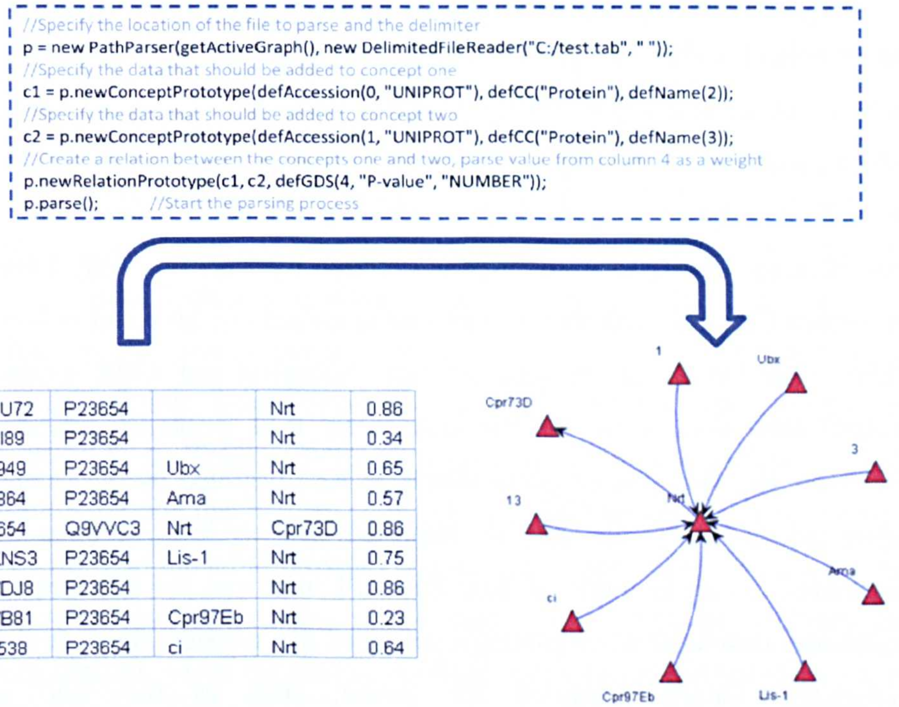

Concept
p.newConceptPrototype(AttributePrototype ... atts);
Class (CC)
defCC(Integer/String value);
Name
defName(Integer/String value, String isPreferred);
name
is preferred?
Accession
defAccession(Integer/String value, String sourceCV, String isAmbiguous);
source (CV)
accession
is ambiguous?
User Data (GDS)
defGDS(Integer/String value, String attName, String indexed);
attribute name
data
type of data
index?
Data Source (CV)
defCV(Integer/String value);
Parser ID (PID)
defPID(Integer/String value);
Evidence
defEvidence(Integer/String value);
Relation
p.newRelationPrototype(ConceptPrototype source, ConceptProtype target, AttributePrototype ... atts);
Type (RT)
defRT(Integer/String value);
User Data (GDS)
defGDS(Integer/String value, String attName, String indexed);
attribute name
data
type of data
index?
Evidence
defEvidence(Integer/String value);

Figure 2.3 Accessory methods from the delimited file parsing API. The accessory methods provide a simple way of binding specific columns or constant values to the entries in the Ondex data model. There is a method for every possible attribute of concept or relation. To simplify the task for the user, the entries from the Ondex controlled vocabulary are automatically resolved from string values or added to it, if missing. Arguments shown in italic can be omitted from the input, in which case the default values will be inserted automatically (as all of the fields of the attributes always need to be present).

written in Java, it is one of the easiest languages to integrate. Although there are several JavaScript scripting libraries for Java, currently one of the most advanced and flexible solutions is Mozilla Rhino (http://www.mozilla.org/rhino/), which now supports all of the features of JavaScript 1.7. This implementation allows direct access to Java objects and classes as well as "native" classes defined in JavaScript. The robust performance of the framework is ensured by the compilation of the JavaScript classes into Java byte-code. Therefore, once a JavaScript has been compiled using Rhino, it becomes Java and can offer similar levels of performance. For these reasons, this scripting solution was chosen to be one of the first to be added to the system via a newly developed scripting API.

A JavaScript wrapper implementation provides a straightforward method for specifying a subset of classes to be made accessible via the scripting interface. The only input required is the fully qualified name of the class to wrap. The wrapper generator then scans all of the methods of this class and compiles a list of other non-default classes it relies on either as an argument or a return object from a method. These classes are then scanned in the same manner, until all types required are resolved to the types natively supported by the Rhino library. Optionally, it is possible to customise this process by specifying the set of methods of the base class that need to be implemented or ignored in the wrapper, as well as the new names for these methods or for the wrapper class itself. All static methods encountered by the scanner are mapped to functions by collecting all of them in a single class, which is then made available in the global scope. Because JavaScript, unlike Java, is a loosely typed language, wrappers also contain the code necessary to correctly resolve Java generics to and from JavaScript representations and handle the type casting errors.



```
//Specify the location of the file to parse and the delimiter
p = new PathParser(getActiveGraph(), new DelimitedFileReader("C:/test.tab", " "));
//Specify the data that should be added to concept one
c1 = p.newConceptPrototype(defAccession(0, "UNIPROT"), defCC("Protein"), defName(2));
//Specify the data that should be added to concept two
c2 = p.newConceptPrototype(defAccession(1, "UNIPROT"), defCC("Protein"), defName(3));
//Create a relation between the concepts one and two, parse value from column 4 as a weight
p.newRelationPrototype(c1, c2, defGDS(4, "P-value", "NUMBER"));
p.parse();        //Start the parsing process
```

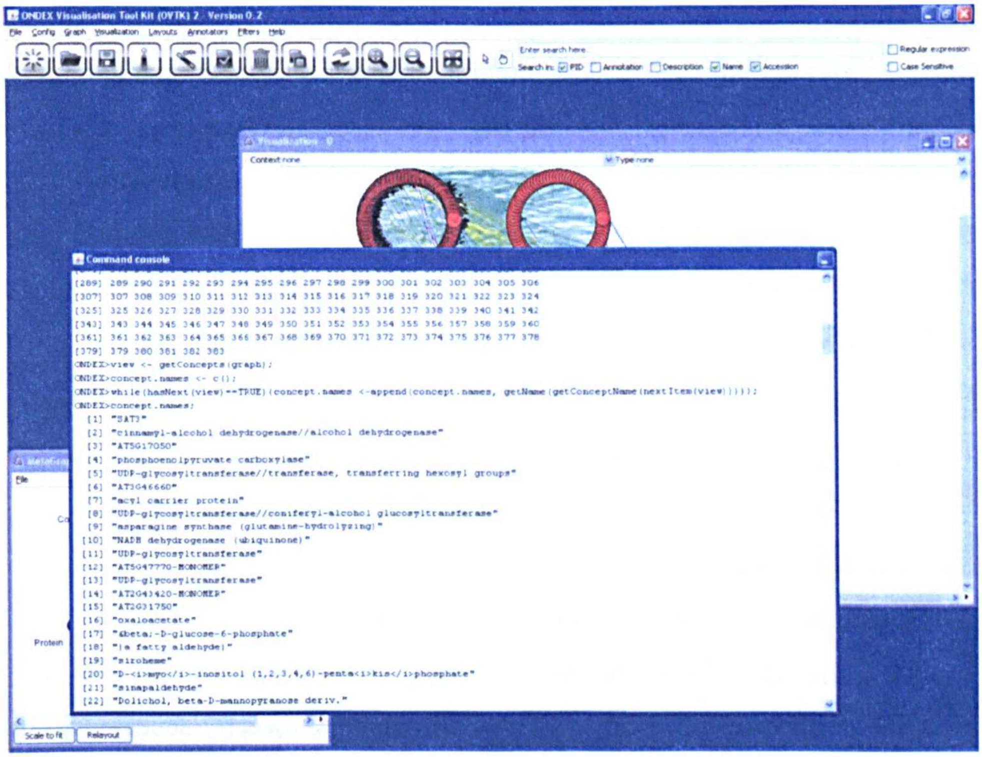
Q9VU72	P23654		Nrt	0.86
Q9VI89	P23654		Nrt	0.34
P83949	P23654	Ubx	Nrt	0.65
P15364	P23654	Ama	Nrt	0.57
P23654	Q9VVC3	Nrt	Cpr73D	0.86
Q7KNS3	P23654	Lis-1	Nrt	0.75
Q9VDJ8	P23654		Nrt	0.86
Q9VB81	P23654	Cpr97Eb	Nrt	0.23
P19538	P23654	ci	Nrt	0.64

Figure 2.4 Example of using delimited file parsing API. The small sample file (bottom-left) was converted to the Ondex network (bottom-left) using the parsing script (top). The example and image were adapted from Ondex tutorial.

2.4.1.1 Example: delimited file parsing

One of the best examples of how scripting interface can be used to simplify time-consuming tasks is the API for parsing delimited files. Delimited files still remain one of the most commonly used exchange formats and the vast majority of the data used during the work on this thesis was accessible in this form. Despite their simple formalism, many delimited formats have additional features or complex relationships between their elements and, for this reason, developing a generic parser that supports most of the possible formats remains a difficult task. Another level of complexity is introduced by the requirements of normalization to a standard conceptual schema – a parser must resolve any semantic or syntactic heterogeneity issues when data is imported into a unified representation. In Ondex, this is done by matching the data elements in the original data source to the entries in the controlled vocabulary and sorting data into semantically defined fields.

Delimited file parsing API simplifies this process by allowing fast construction of format-specific parsers from a set of simple components (**Figure 2.3**). The set of methods allows the definition of concepts, relations and relationships between them. Additionally, all values can be piped into appropriate attributes on these nodes and edges. Where a field is subject to the restrictions of the Ondex controlled vocabulary, it is possible to specify the correct static value to be used or to dynamically fill it in by creating a look-up between the Ondex controlled vocabulary term and regular expression patters that it must match in the source file. Although this API is written entirely in Java and is also used within other Ondex parsers (among them AtRegNet and TAIR interactome parsers) that work with the delimited files, it is primarily intended for interactive use from the scripting interface. As illustrated by the example in **Figure 2.4**, a typical file may be imported with as little as three to five commands. As an example of how this API improves the efficiency of an Ondex user/developer when parsing a delimited file, consider the code size of a pre-existing Ondex delimited file parser, such as the 'tab' parser (v10.03.2008). This native Java parser implementation is over three hundred and fifty lines long. By comparison, just four commands (lines) were needed using the delimited file parsing API to handle exactly the same file format.


ONDEX Visualisation Tool Kit (OVTK) 2 Version 0.2
File Config Graph Visualization Layouts Annotators Filters Help
Enter search here
Search in: ☑ PID □ Annotation □ Description ☑ Name ☑ Accession
□ Regular expression
□ Case Sensitive
Visualization - 0
Context none
Type none
Command console
Protein
Scale to fit
Relayout

```
[289] 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306
[307] 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324
[325] 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342
[343] 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
[361] 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378
[379] 379 380 381 382 383
ONDEX>view <- getConcepts(graph);
ONDEX>concept.names <- c();
ONDEX>while(hasNext(view)==TRUE){concept.names <-append(concept.names, getName(getConceptName(nextItem(view))))};
ONDEX>concept.names;
  [1] "SAT3"
  [2] "cinnamyl-alcohol dehydrogenase//alcohol dehydrogenase"
  [3] "AT5G17050"
  [4] "phosphoenolpyruvate carboxylase"
  [5] "UDP-glycosyltransferase//transferase, transferring hexosyl groups"
  [6] "AT3G46660"
  [7] "acyl carrier protein"
  [8] "UDP-glycosyltransferase//coniferyl-alcohol glucosyltransferase"
  [9] "asparagine synthase (glutamine-hydrolyzing)"
 [10] "NADH dehydrogenase (ubiquinone)"
 [11] "UDP-glycosyltransferase"
 [12] "AT5G47770-MONOMER"
 [13] "UDP-glycosyltransferase"
 [14] "AT2G43420-MONOMER"
 [15] "AT2G31750"
 [16] "oxaloacetate"
 [17] "&beta;-D-glucose-6-phosphate"
 [18] "|a fatty aldehyde|"
 [19] "siroheme"
 [20] "D-<i>myo</i>-inositol (1,2,3,4,6)-penta<i>kis</i>phosphate"
 [21] "sinapaldehyde"
 [22] "Dolichol, beta-D-mannopyranose deriv."
```

Figure 2.5 Example of Ondex-R integration. In this example a graph was loaded into the Ondex front end and the console was switched to R mode. Then an R view object is created it holds a reference to an Ondex view object that contains all concepts currently in a graph. This object can then be queried via a number of pre-defined functions to get information that can be used in a subsequent analysis in R – in this example it is protein names.

2.4.2 R statistical environment

As a cross-disciplinary research field, bioinformatics often draws upon the methodology of other disciplines to solve biologically relevant problems. In particular, statistical analysis is often necessary to evaluate the significance of the findings or to formalise evaluate the sources of variation in the data being used or the models being developed. The R software environment (R Development Core Team, 2008) is a popular statistical computing and visualisation solution that is often called upon to fulfil this need. The relevance of this platform to the bioinformatics community is particularly evident by the amount interest in Bioconductor (Gentleman *et al.*, 2004), an R library for genomics analysis. From the data in PubMed, this original methodology paper for Bioconductor (from 2004) has been cited 3904 times by April 2012. The R environment is implemented in C++, but also has its own high-level language based on the S4 specification, although many analysis methods are implemented in C++ directly and only use S4 for linking with other functionality in the R environment. Ondex and R, however, have different and complementary strengths – Ondex facilitates data acquisition and defines a unified generic schema for wide range of biologically relevant information types, whereas the focus of R is primarily on the downstream analysis of numerical data. An interface between these two tools would simplify the exchange of data necessary to bring together these tasks and allow more complex analysis pipelines to be developed with minimal effort.

As many analysis methods in R are implemented in C++ libraries, just having a Java interpreter for S4 would not be sufficient to gain access to those libraries – as those links require a fully functional R environment itself in order to work. In order to access the full R functionality a link between Java and the R implementation in C++ is required. Programs written in C++, however, are compiled to native code before they are executed, which is operating system and hardware specific. The only way this code can be directly accessed from Java is through a specialised interfacing framework called Java Native Interface (JNI).

Although there are other R-to-Java interfacing libraries currently available, the twin libraries rJava and JRI are the only ones that use the JNI framework and therefore offer the best performance. The rJava library allows calls to Java to be executed from the R environment. Each instantiated class is wrapped in an R object which maintains a reference to it. The method calls are possible by calling a special function that takes in the class instance reference object, name of the method and its arguments as input. This function only uses a base class of the argument to correctly construct a method signature, so the class of the argument often needs to be changed, which can be done using a casing function. Calls to appropriate functions are also required in order to convert the primitive data types returned by Java into their R equivalents. The JRI library is the opposite of rJava and allows R environment to be accessed from Java. It allows R commands to be executed from Java and makes the results of this evaluation available from the Java program. When used in combination, these libraries allow two-way communication between Java and R environments – a 'call-back'.

Although rJava-JRI libraries are sufficient to realise Ondex-R integration, there are still considerable usability issues associated with using the rJava interface directly. The functions it offers are very low-level and calls to Java methods are very verbose and have a complicated syntax. The need to cast objects to the correct type and to use conversion methods to convert return values to R data types also add an unwelcome layer of complexity. This issue is dealt with by using an additional set of wrapper object on the R side, which deal with all of this complexity and present them to the user as a set of native R objects and functions. Similarly, on the Java side when JavaScript-wrappers are generated an additional matching S4 class is generated for each of them by executing the S4 class creation commands using the JRI interface. Every method on the class is then wrapped as an R function that takes this class as its argument. When this interface is used, all objects necessary for accessing Ondex are automatically wrapped as S4 classes at the point they are accessed through the R interpreter. As each instance of the S4 class only holds a single reference to the complementary instance of the Java class, this implementation is also memory efficient. All data is still stored on Java-side and is only moved across to R side upon request. Only supported data types can be moved to R – as the wrapper generator completely resolves all Java class dependency trees, all Java classes returned by method calls are guaranteed to be contained within an S4 wrapper. If a return type is supported, it will be automatically converted to the matching R data type when it is returned by the function.

When using this interface a user is presented with a direct link to R via the JRI interface (**Figure 2.5**). The console looks and functions exactly like an R console would. The only difference is that it is actually an integral part of the Ondex front-end, has an additional set of S4 classes to mediate interaction with Ondex and all objects in the main application, such as graphs and views, can be accessed and manipulated using the R environment syntax.

The integration of R in Ondex provides a far superior functionality compared to other graph visualisation or data integration tools where such a link has been realised. For example, a link to R was available in older version of Cytoscape via the CytoTalk plug-in (Reiss *et al.*, 2005). With this extension it was possible to access the Cytoscape API from an R console, but this implementation was relatively low level, has a complex syntax and is no longer available in the latest version of Cytoscape. A better implementation was realised in GUESS (Adar, 2006), a robust graph analysis tool for social networks, which also has its own scripting language based on Jython, but no data integration capabilities. This implementation allows switching between the R and native scripting language, it is possible to send the data to R and receive it back, but the graph can be queried or updated with the changes only when in native mode. Both of these implementations use socket-based approach, which is much slower than passing the data directly through the JNI.

2.4.3 Jython/Python

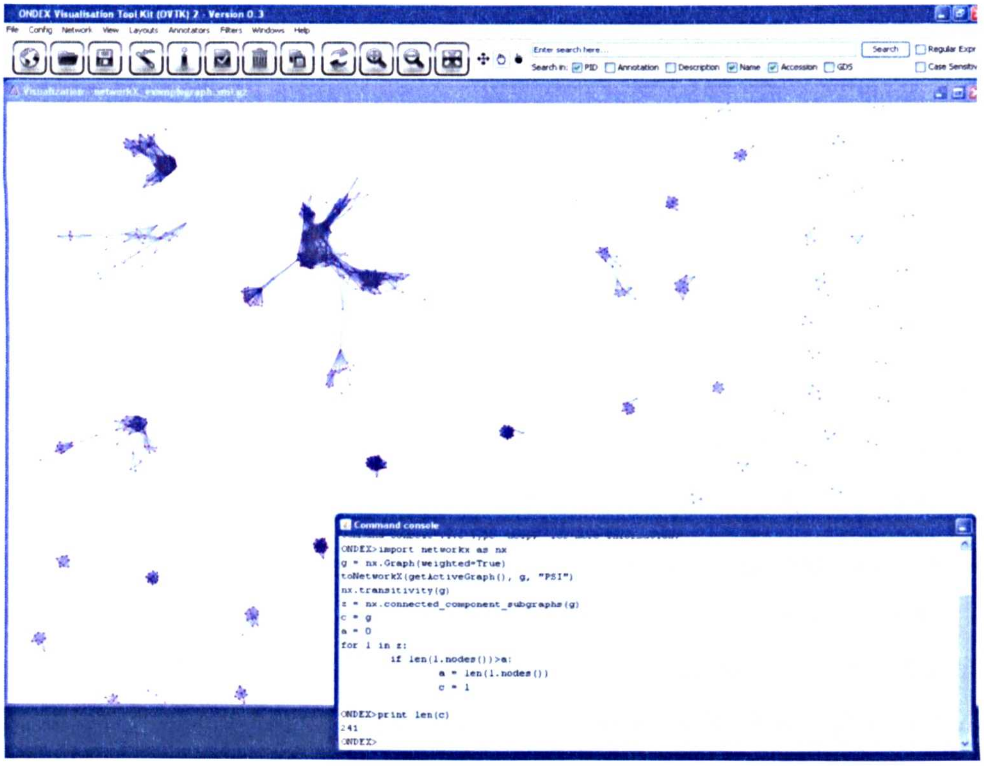
Python is a powerful and versatile scripting language with a very active bioinformatics user community. Unlike the situation with JavaScript, Python syntax and formalisms are very distinct from Java, but at the same time is more succinct (e.g. more can operations can be performed with less code) and comes with an extensive collection of libraries. Several of these, are aimed at addressing the needs of the bioinformatics research community. These include NetworkX (Hagberg *et al.*, 2008a) for graph analysis and visualisation and NumPy/SciPy (Peterson, 2009) which is a library of scientific mathematics, science, and engineering numerical analysis methods. Python is particularly popular within the bioinformatics community where it has overtaken Perl as the preferred scripting language. This was revealed in a 2007 survey conducted by Bioinformatics Organization, Inc. (www.bioinformatics.org). They found that 23% of bioinformatics researchers questioned were interested in learning Python, compared to 19% for Perl and 16% for Java. A number of Python-based projects are specifically developing tools to support bioinformatics research - for instance, Biopython (an extensive open source library for computational molecular biology) (Cock *et al.*, 2009), GenomeDiagram (a toolkit for visualisation of large genomic datasets) (Pritchard *et al.*, 2006), PySCeS (modelling solution for cellular systems) (Olivier *et al.*, 2005), Sarment (hidden Markov model implementation) (Gueguen, 2005) and SIR (collection of tools for working with biological databases) (Ramu, 2001). By adding a Python environment to the Ondex system it was possible to take advantage of these tools and analysis methods.

Currently two solutions allow interoperability between Python and Java. The first one is JPype, which integrates the two environments at a virtual machine level. Although promising, this project is still at a very early development stage and is primarily targeted at web developers. Another solution is Jython (formerly known as JPython), which is a Python interpreter implemented in Java. Jython allows full access to Java classes from the Python environment and allows them to be used alongside the Python data structures. In this way all of the functionality of Java can be accessed from within a Python-like coding environment. As a pure Java implementation, Jython can also be very easily added to any Java application. For these reason Jython was considered to be the better choice of interpreter for use within the Ondex system.

Jython scripting environment in Ondex was implemented in a very similar way to that used for JavaScript. The entry point for the Jython scripting environment is an interpreter class that wraps the actual interpreter from the Jython. This class handles errors and mediates interactions with the core Ondex classes and methods accessible from the scripting environment. A wrapper generator is used to wrap these classes and methods to be more compatible with the Python environment. In Jython, the Java collections API maps to native data structures, whereas in Ondex API, arrays are used more commonly than collections. To improve the usability, wrappers perform backwards and forwards conversion between array and list data structures. As the example in the next section illustrates, the implementation also allows import of external libraries and their use in conjunction with the Ondex data model.

2.4.3.1 Example: interaction with the NetworkX v0.99 library

NetworkX is a graph analysis and visualisation library for Python. Its primary goal is to allow fast and flexible construction of classical graph representations from the source data. These graphs can then be analysed using a number of standard network analysis algorithms or converted to publication-quality vector images. The NetworkX data structure is designed to be very simple and straightforward to use and therefore it is much quicker to prototype and test analysis algorithms than working with the native Ondex schema. NetworkX also has many more graph analysis methods than currently available in Ondex itself. The real advantage that is gained from using NetworkX, however, comes



ONDEX Visualisation Tool Kit (OVTK) 2 - Version 0.3
File Config Network View Layouts Annotators Filters Windows Help
Enter search here...
Search
□ Regular Expr
Search in: ☑ PID □ Annotation □ Description ☑ Name ☑ Accession □ GDS
□ Case Sensitiv
Visualization - networkX_examplegraph.xml.gz
Command console

```
ONDEX>import networkx as nx
g = nx.Graph(weighted=True)
toNetworkX(getActiveGraph(), g, "PSI")
nx.transitivity(g)
z = nx.connected_component_subgraphs(g)
c = g
a = 0
for l in z:
        if len(l.nodes())>a:
                a = len(l.nodes())
                c = l

ONDEX>print len(c)
241
ONDEX>
```

Figure 2.6 Using the NetworkX Python library to Analyse an Ondex graph. The console queries in Jython mode were used to find the largest connected graph component. An empty NetworkX graph was created and populated with a subgraph of all nodes that are connected by a relation with "PSI" attribute – the same subgraph is also visible in the background. A method on the NetworkX graph was then used to get all connected components, and the largest one was identified by iterating through them.

from the ability to easily switch between different graph types. For example, in different situations it may be preferable to represent a network as a graph with direct or undirected edges or to assign a weight to them. Some of the analysis methods, like shortest path, will produce different results depending on the type of the graph they are applied to. In order to capture the full complexity of biological data, the Ondex graph representation is one of the more complex types and has a number of non-standard extensions that make it more than a pure mathematical graph. If the Ondex graph data structure is mapped onto one of the more basic representations provided by NetworkX library certain types of analysis can be carried out that are specific to that network representation. Conversion to the required formalism makes subsequent analysis more transparent, easier to follow and, therefore, less error-prone.

To use any non-default Python libraries from the Jython scripting environment they must first be installed on the user's system. This process is identical to the installation of libraries for a standard Python distribution – they all need to be correctly deployed in a directory (usually "lib" under the Python installation) along with any dependencies. In order to be accessible from Jython this directory must be provided to the Java application as part of its command line arguments. After that, the library may be imported via a command in the interpreter. The example in **Figure 2.6** illustrates how NetworkX can be used to interactively find a largest connected component in an Ondex graph using functionality from the NetworkX library. The first line creates an undirected weighted graph view of the Ondex network. To do this, an attribute needs to be specified, which will be used as a weight for edges. The view created uses the original Ondex identifiers for all nodes and edges. After the analysis is run, these identifiers can be used to refer back to the Ondex entities in the largest connect component and may be employed for further processing.

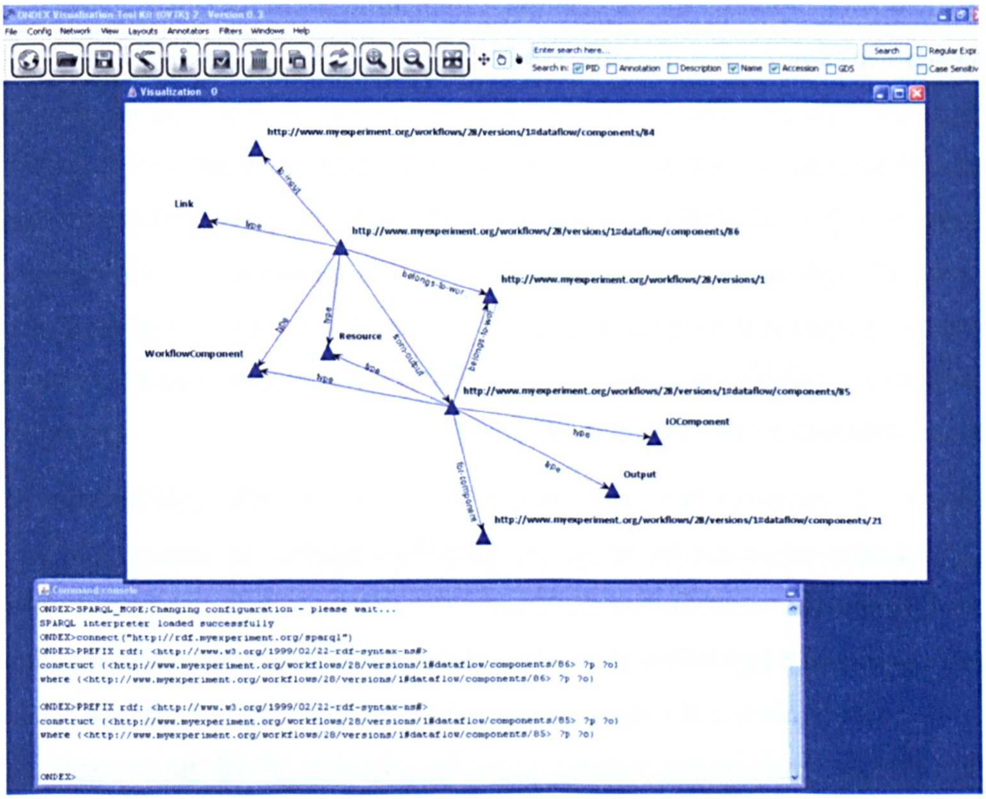
This example illustrates how simple set functions can be used to map an Ondex network to any of the graph types supported by NetworkX. This representation can then be used to interactively analyse the network. The results of the analysis can then be used to generate analysis reports or written back to the Ondex graph. Alternatively, a graph can be exported using one of the NetworkX exporters and analysed further using other graph analysis tools.

2.4.4 SPARQL and semantic web

As biological data are becoming available in ever-increasing quantities, one of the challenges faced by the research community is to efficiently mine and share the accumulated knowledge. Although web technologies are now widely used as a means of providing easy access to this information, integration and computer-driven analysis over large number of heterogeneous resources still remains an on-going challenge. One of the possible ways to enable such an analysis is through the use of the Semantic Web technologies, which define a framework for unambiguously identifying and categorising resources available on the Internet. The Resource Description Framework (RDF) format (Lassila *et al.*, 1998) is used to make statements about these resources in the form of triples (subject, predicate and object). These statements can be used to express relationships between the resources in a format suitable for computational processing. Data represented in RDF can be interpreted as a graph where RDF resources (nodes) are linked to each other by predicates (edges). RDF also enforces the use of globally unique identifiers and has an option of binding data to a structured schema by creating references to appropriate ontologies. A number of standards and ontologies for representing biological data have now been developed and several prominent biological data providers provide their data in RDF format.

Semantic Web approaches address a similar set of problems to those tackled by Ondex, in the sense that they aim to provide a framework to integrate and analyse disparate data. However, Semantic Web technologies were primarily designed to work with the federated approach to data integration, where data is distributed across potentially many online resources and is queried and linked dynamically upon request. For this reason, the Ondex approach of importing data via a set of parsers and working on a well-defined and usually local set of source data files is not readily compatible with the real-time and unbounded nature of RDF data stores. One possible way to reconcile these approaches and bring some of the functionality of Ondex to semantic web based resources is to make the parsing process in Ondex query-driven. By having a generic parser for the results of a query against a set of online RDF data stores, a much wider variety of resources can be imported with minimal effort. An additional benefit of this approach is that the data imported in such way would be compatible with the RDF specification by default and can be worked on using the same set of tools, thus also addressing the need for a query language for the Ondex graph itself. The ability to query the Ondex graph via a formalised and efficient language has the potential to both improve the usability of the system and facilitate sharing of the Ondex-based datasets with the wider research community. SPARQL Protocol and RDF Query Language (SPARQL) (Prud'Hommeaux and Seaborne, 2006) is a language for querying RDF graphs. SPARQL allows graph patterns to be defined using an SQL-like syntax, which can then be resolved against the content of an RDF data store. The result of a query can be either a collection of data fields or a sub-network in RDF format.

In order to be able to use the data available on the Semantic Web, an application needs to be able to connect to the external resources, support the construction and execution of SPARQL queries against them and interpret the results. In the Ondex implementation, the former two requirements were addressed by linking the Jena library to the Ondex scripting environment. Jena is a Java framework for construction of Semantic Web applications (McBride, 2001). As well as SPARQL engine, Jena also provides a Java API for working with RDF data, IO capabilities and deployment of RDF resources. As the Ondex SPARQL engine implementation required a direct access graph model whereas the other scripting solutions only needed to exchange the data via a set of pre-defined methods, the design of this API is very different from the previously described Ondex scripting solutions. Rather than allowing access to a set of Ondex objects and functions, this implementation is in essence an on-demand parser, which executes SPARQL queries and imports results into an Ondex graph. This is possible because, with the exception of some advanced


ONDEX Visualisation Tool Kit (OVTK) 2 Version 0.3
File Config Network View Layouts Annotators Filters Windows Help
Enter search here...
Search
□ Regular Expr
Search in: ☑ PID □ Annotation □ Description ☑ Name ☑ Accession □ GDS
□ Case Sensitiv
Visualization 0
http://www.myexperiment.org/workflows/28/versions/1#dataflow/components/84
to-input
Link
type
http://www.myexperiment.org/workflows/28/versions/1#dataflow/components/86
belongs-to-wor
http://www.myexperiment.org/workflows/28/versions/1
type
type
Resource
from-output
belongs-to-wor
WorkflowComponent
type
type
http://www.myexperiment.org/workflows/28/versions/1#dataflow/components/85
IOComponent
type
type
Output
for-component
http://www.myexperiment.org/workflows/28/versions/1#dataflow/components/21
Command console

```
ONDEX>SPARQL_MODE;Changing configuaration - please wait...
SPARQL interpreter loaded successfully
ONDEX>connect("http://rdf.myexperiment.org/sparql")
ONDEX>PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
construct (<http://www.myexperiment.org/workflows/28/versions/1#dataflow/components/86> ?p ?o)
where (<http://www.myexperiment.org/workflows/28/versions/1#dataflow/components/86> ?p ?o)

ONDEX>PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
construct (<http://www.myexperiment.org/workflows/28/versions/1#dataflow/components/85> ?p ?o)
where (<http://www.myexperiment.org/workflows/28/versions/1#dataflow/components/85> ?p ?o)


ONDEX>
```

Figure 2.7 Dynamic import of RDF data using SPARQL. The Ondex SPARQL query engine was used to fetch data from a MyExperiment RDF Endpoint. Two queries were made to get the entities associated with dataflow components 85 and 86. The graph loaded in the background shows the results of the import. Note the entities common to both queries were only imported once ('http://rdf.myexperiment.org/workflows/versions/1', 'WorkflowComponent' and 'Resource') had run and the results of both queries were merged using the unique identifiers on these entities.

features, an Ondex graph is compatible with the RDF formalism. Each RDF resource can be interpreted as an Ondex concept, with literals (e.g. strings and numbers) as attributes and predicates as relations between them. As RDF identifiers are designed to be globally unique, if an RDF identifier is encountered that is already in the Ondex graph, this pre-existing node will be used and linked to any new data contributed by the query. In this way an application-specific Ondex dataset can be created from the information collected from different RDF data stores.

Another capability gained by adding SPARQL to the Ondex system is the ability to query standard Ondex graphs in a generic manner. Prior to this development, the only option for extraction of data from them involved creation of specialised exports and transformers that employed a set of simple API methods to realise each operation. Using an exporter to RDF (contributed by another Ondex developer) any Ondex graph can be exported in a SPARQL-compatible format. The Ondex SPARQL query engine can then be used to mine these data and visualise the results in the Ondex user client. Although it is also possible to load this data into other SPARQL-enabled environments, the Ondex client is one of the few tools that allows the results of the RDF queries to be visualised and also supports a wide range of other analysis methods for graphs not available on other platforms. Data imported in this manner is also compatible with Ondex Integrator plug-ins.

Figure 2.7 illustrates how information from several RDF queries can be dynamically integrated by executing SPARQL queries in the Ondex user client. The implementation of the Ondex SPARQL query engine is still at an early prototype stage and it is recognised that the necessity to export the data before the queries can be run on it constitutes a suboptimal solution. Nevertheless, it does demonstrate the utility of using SPARQL for working with Ondex graphs and provides a means to query the data - which was not possible prior to the introduction of this engine. As will be demonstrated in the subsequent chapters, an ability to find matching patterns in the graph underpins many of the analysis methods implemented for this thesis. Currently, efforts are underway to bring Ondex and RDF data models more closely together and, when complete, the export step will no longer be necessary. However, due to the major re-engineering required, this refactoring is beyond the scope of this project.

2.5 Discussion

The Ondex framework addresses many of the challenges of dealing with large and complex biological information. The Ondex approach is built upon two key principles – configurability and reusability. Rather than encoding a set of pre-defined solutions to common data integration problems, the system provides a large number of generic modules from which application-specific workflows can be assembled. The development of a workflow management API and the Ondex Integrator tool have simplified the process of managing these modules from the perspectives of both developers and users. The Workflow API was developed to overcome the shortcomings of the Ondex Plug-in API by defining a unified interface for all possible types of Ondex plug-ins, which helped to standardise and optimise execution of Ondex workflows. The plug-in descriptor/task entry and supporting classes also allowed all meta-data about the plug-ins to be assembled in one place from which it can be made readily available to the end-user through the GUI. Based on these two developments, an Ondex Integrator tool was built, which has greatly simplified the process of workflow creation and management. Together, these developments allowed for greater productivity when using the system and enabled more complex analyses to be realised within Ondex workflow than was practically achievable beforehand.

The development of an Ondex scripting framework made the system just as customisable at a lower level of component granularity. It supported the introduction of functions, designed to handle a simpler set of data integration and analysis tasks that were too small to be sensibly realised using the plug-in architecture. An analysis script could then be built from these functions by executing them consecutively and linking together their inputs and outputs in a way that is more accessible and intuitive for many bioinformaticians. On top of this functional layer, a set of other capabilities could then be implemented, where native Java components could be seamlessly combined with those implemented in a less restrictive scripting environment like Python or JavaScript. It was also possible to take advantage of ready constructed bioinformatics analysis tools available as libraries built in these languages and use them as part of Ondex-driven analysis pipeline. It has also enabled the use of a SPARQL interpreter to dynamically query Semantic Web resources and thus successfully combine the warehousing-based and federated-based data integration approaches in the same system. The SPARQL implementation has also provided a general mechanism and query language for interrogating an Ondex graph. This fulfils a requirement that has been considered by many developers as a major shortcoming of the Ondex system.

All of these developments have allowed easier management of complex analysis pipelines developed for this thesis and also increased the productivity when using the system. Through use of clearly defined formalisms, like plug-ins and functions analyses were made more transparent, easier to understand and reproduce. By using these reusable components, the analysis methods can also be more readily reconfigured for use on species other than *Arabidopsis* or on different datasets. From the point of view of this project, the greatest benefit from these developments was that they have made it possible to manage data in a proactive way. Biological data is constantly updated, new data providers enter the scene and new formats are defined for exchange of these data. Therefore, during the four years of this project one of the major challenges was to keep up with these changes and to update the datasets used for this project accordingly. This process required changes to be made to some parts of the data integration pipeline and analysis to be re-done on a regular basis. Through the use of functions and plug-ins these changes could be restricted to the set of affected components, thus making the update process more manageable.

The usability improvements have also helped to expand the Ondex user community and support collaborative efforts of other developers. The Ondex Integrator user interface has significantly improved access to the system and has simplified the process of getting to grips with Ondex for the new users. For example, the Integrator interface shows all of the available plug-ins, information about them and all valid configuration options, whereas previously this type of information could only be accessed by looking through the Java source code of appropriate plug-ins. The addition of Python, R and SPARQL have also opened up Ondex to the users of these languages – as well as enabling a wider range of bioinformatics problems to be tackled by bringing in the functionality developed under those environment into the system. Both the Ondex Integrator and scripting environment have been a core part of the Ondex tutorial since their introduction in 2007; this reflects their importance for the Ondex user community.

3 COEXPRESSION NETWORK CONSTRUCTION

3.1 SUMMARY

The expression levels of multiple genes can be measured simultaneously using a number of different DNA microarray approaches. Due to the large number of measurements, often under a set of different conditions, it is often useful to summarise such data in the form of a coexpression network. In such representation, the nodes represent individual genes and links represent a measure of similarity between their expression profiles. To allow incorporation of expression data as part of the integrated dataset used in the Ondex system, a coexpression analysis pipeline was implemented as part of this work. The analysis pipeline combines several well-established methods for each step of the coexpression analysis, automates the handling of bad data entries and mediates the flow of information between different analysis steps. Java-based, parallelised implementations for the calculation of weighted Pearson correlation and a network structure based threshold selection were also produced as part of this work.

3.2 INTRODUCTION

Changes in the types and quantities of proteins in the cell (proteome) are fundamental ways that living organisms use to respond to changes in the environment and realize their progression through the lifecycle (Kitano, 2002). One of the possible ways to control protein levels is at the stage of transcription, by regulating the number of mRNA copies for the particular genes (Schena *et al.*, 1995). Amounts of the specific mRNA types (transcriptome) can serve as an indicator of the quantities of corresponding proteins present in the cell (Gygi *et al.*, 1999). Because the transcriptome is more amenable to quantification using current technologies than the proteome, this trend has been of great importance and was the main reason that DNA microarrays were adopted so enthusiastically. DNA microarray-based approaches have been actively used since mid-1990s (Rockett and Hellmann, 2004) and by re-analysing the these data it is often possible to extract novel insights that were not an intended target of research in the original microarray experiments (Jen *et al.*, 2006).

Most of the approaches for analysis of expression data rely on the principle of "guilt-by-association", whereby genes that have corresponding levels of expression across multiple conditions are likely to be biologically linked (Wolfe *et al.*, 2005). However, this link may be an indication of one or more different types of associations. Possible interpretations include involvement in the same metabolic pathway (DeRisi *et al.*, 1997), protein-protein interaction of the respective gene products (Ge *et al.*, 2001) and an association with a common regulatory mechanism (Ideker *et al.*, 2002) or biological process (Stuart *et al.*, 2003). The groups of co-expressed genes are commonly recovered from all-versus-all coexpression matrixes using clustering and principle component analysis methods to yield gene lists for further study (Korenberg, 2007).

Coexpression data can also be conceptualised as a network, where nodes are genes and edges indicate similarity of expression profiles. The steps commonly undertaken to construct such a representation are explained in detail in the Section 6.1.1. Network representation is suitable for clustering as well as for application of methods from graph theory (Butenko *et al.*, 2009) and may be leveraged to allow the interactive visual exploration of large and complex biological datasets (Shannon *et al.*, 2003, von Mering *et al.*, 2003, Kohler *et al.*, 2006). Network visualisation is potentially important, as it allows presentation of extensive datasets in an intuitive and easily accessible form. This makes it possible for non-technical experts (e.g. experimental biologists) to more easily benefit from the results of data integration and bioinformatics analysis. Therefore, networks can serve as a useful communication tool between biologist and bioinformatics researchers and facilitate cross-disciplinary research. Such interaction is particularly important as, at present, most biological knowledge is still not available in a structured form; therefore facilitating easier access to the data can enable discoveries not attainable by purely computational means (Kohler *et al.*, 2006).

As a means of visualising networks was already provided as part of the Ondex system, only limited extensions to the existing visualisation methods were necessary to enable viewing of expression networks. However, what was missing from the system were the straightforward methods for import of the expression data and its interpretation as a network. This chapter describes the work that was done to develop such an analysis pipeline and explains all of the steps involved in the process. The next section introduces the microarray technology and reviews the selection of the relevant current methods with the view of providing a justification for the selection of the individual analysis components to be part of this pipeline.

3.2.1 Transcriptome analysis using microarrays

DNA microarray technologies exploit the property of DNA hybridisation, whereby the complementary single DNA strands will form a double helix under a particular set of conditions (Deonier *et al.*, 2005). As described by Deonier *et al.* (2005), the analysis usually involves the following steps. In most approaches for transcriptome profiling the mRNA in the sample is converted to the complementary DNA (cDNA). These single-stranded cDNA molecules (targets) hybridise to complementary components fixed to a solid substrate (probes). The probes are densely grouped at particular locations (spots), so that each group only contains the probes with an identical sequence. The array is brought into contact with the sample to allow probes to hybridise with their targets, after which all unhybridised cDNA is washed away. The targets are integrated with a fluorescent marker that allows their relative abundance to be evaluated by measuring the intensity of the fluorescence at a particular location on the array.

Presently, there are two widely used types of microarrays for the profiling of gene expression – spotted cDNA (Schena *et al.*, 1995) and oligonucleotide-based (Pease *et al.*, 1994) arrays. The spotted arrays use longer probes of about ~200 nucleotides-longs, usually with one probe sequence per matched target sequence. In oligonucleotide arrays each target is matched to a set of shorter probes that match different parts of the target sequence, between 25 and 60 nucleotides in length (Deonier *et al.*, 2005). The oligonucleotide arrays are now more common, although both types are still currently in use (Kawasaki, 2006).

The raw fluorescence measurements are affected by noise both from the technical and biological steps of the protocol (Kohane *et al.*, 2003). Therefore

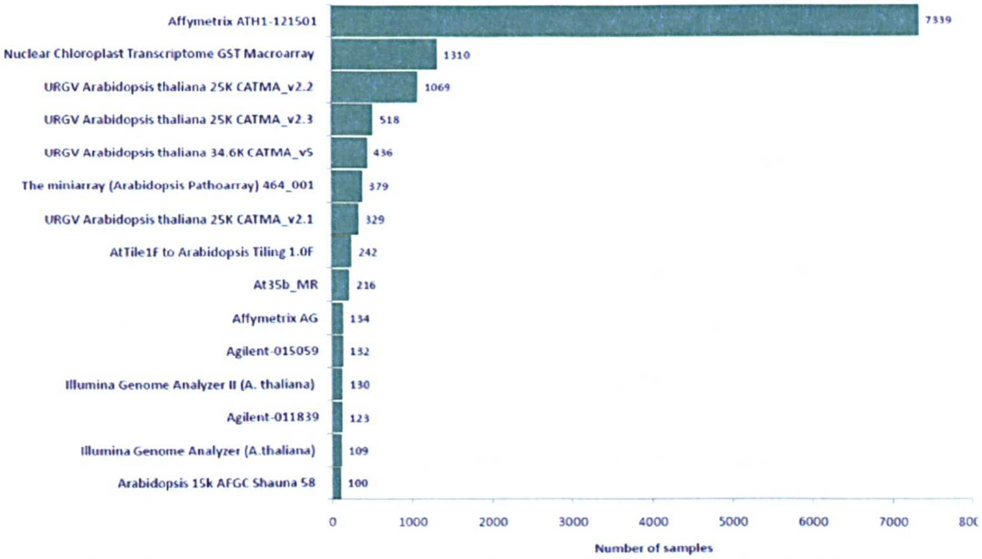


Figure 3.1 Number of samples (individual slides) in the GEO database for all platforms used to study *Arabidopsis thaliana* that have more than 100 samples.

statistical processing is commonly applied to the reported values in order to account for these effects, commonly referred to as "normalisation" (Kohane *et al.*, 2003). The work in this thesis primarily concerns the analysis after this stage and only uses well-known and established methods for microarray normalization. Therefore, the detailed description of the normalisation methods or their benefits and drawbacks is not provided here because of their limited relevance. The choices used were primarily guided by the works of Reimers (2010) and Korenberg (2007) and the references for the selected normalisation approaches and implementations are included in the appropriate method sections.

The absolute measurements of expression levels from the oligonucleotide microarrays (of the same platform) tend to be consistent between different experiments (Shippy *et al.*, 2004, Petersen *et al.*, 2005, Piper *et al.*, 2002). Consequently, the data from them can be more readily combined and tend to be less affected by the intra-experimental discrepancies. With its ability to detect over 23 750 different transcripts, the Affymetrix *Arabidopsis* oligonucleotide microarray ATH1-121501 (Redman *et al.*, 2004) provides very good genome coverage when compared to other platforms. As illustrated in **Figure 3.1**, this platform is currently the most widely used for studying expression in *Arabidopsis thaliana*. For the reasons outlined above, the ATH1-

121501 platform was predominantly used in this work. The results from the selected experiments using other microarray types were also integrated for particular use-cases in the form of the differentially expressed gene lists. Although the choice to base the coexpression analysis pipeline on the Affymetrix oligonucleotide array was made in order to get access to the largest possible set of expression data for *Arabidopsis,* the data importer for the Ondex system was implemented in a generic manner and the pipeline can be used for the analysis of other Affymetrix oligonucleotide arrays.

3.2.2 Construction of networks from expression data

3.2.2.1 Profile similarity functions

Expression levels of genes under a set of different conditions (gene expression profiles) can be interpreted as a network where the nodes are genes and edges represent the similarity between their expression profiles (Stuart *et al.*, 2003). When supported by other types of data, coexpression networks can be a powerful tool for the interpretation of microarray data (Eisen *et al.*, 1998, Marcotte *et al.*, 1999). In order to construct such a representation from the vectors of raw gene expression values, a function is required to produce a similarity (distance) measure for every pair of vectors in the dataset. Most commonly used measures include Pearson correlation, Euclidean distance, Spearman rank correlation and mutual information (Steuer *et al.*, 2002, Butte and Kohane, 2000). The Pearson correlation metric can differentiate between negative and positive associations, whereas mutual information and Spearman correlation can also recover non-linear dependencies between the vectors. A number of studies have also suggested refinements (Zhang and Horvath, 2005, Cherepinsky *et al.*, 2003, Watson-Haigh *et al.*, 2010, Balasubramaniyan *et al.*, 2005) or novel metrics (Yona *et al.*, 2006, Kim *et al.*, 2007, Nguyen and Lio, 2009) specifically tailored for evaluating gene expression profiles.

From the perspective of this work, there were clear advantages in using one of the more widely adopted metrics, both in terms of the more straightforward comparison to other works, increased confidence in the approach, greater relevance of findings to the research community and being able to benefit from the applicable methodology refinements. Although the studies introducing new metrics tend to present some sort of evaluation to illustrate their superior performance, such evaluations are usually limited in scope and the uptake of such new measures by the research community remains low. As was recently highlighted by Boulesteix (2010) and Jelizarow *et al.* (2010), independent and large-scale evaluations are imperative for determining the real benefit of novel bioinformatics analysis methods. This is especially true in the case of selecting the most suitable measure for expression profile similarity – as the performance of different distance measures was shown to be very sensitive to the choice of the microarray evaluation set (Yona *et al.*, 2006, Li and Wang, 2009, Daub *et al.*, 2004). However, so far there have been no such comprehensive studies to compare the performance of the newly developed measures and conducting one was considered to be outside the scope of this work. For these reasons, it was decided that the best strategy was to adopt one of the more established and better-understood metrics.

The more commonly used measures have now been evaluated in several independent studies. Among them, the comparisons performed by Yona *et al.* (2006), Li and Wang (2009) and Daub *et al.* (2004) appear to be among the most comprehensive ones. However, there appear to be some differences between the results obtained. Most notably, the performance appears to vary greatly depending on the choice of the microarray set. Nevertheless, a number of useful insights can still be derived from these works. In particular, it is possible to observe that, depending on the dataset, Spearman correlation and the Euclidean distance often either massively over- or under perform other metrics (Yona *et al.*, 2006, Li and Wang, 2009). Mutual information and Pearson correlation tend to perform more consistently and were not found to under-perform as frequently as the former two measures (Daub *et al.*, 2004). Daub *et al.* (2004) has concluded that there was no difference in the performance between the latter two measures. For more than half of all the microarray sets investigated in these three works there were negligible differences between most metrics, however very substantial differences were observed in a minority of cases – but no clearly superior metric or a strategy for selecting one was apparent for those instances. Based on this information, Pearson correlation was chosen for this work, as it tends to perform consistently and can be calculated faster than the mutual information methods.

3.2.2.2 Threshold selection approaches

As any two vectors have a distance value, a coexpression network is a fully connected, weighted graph. Although such a representation can also be used directly (Zhang and Horvath, 2005), analysing data in this raw form may become very computationally intensive as many graph analysis algorithms work faster on the more sparse graph representations. For this reason, it is a common practise to remove some of the edges from the network, leaving only the ones that capture the most biologically meaningful connections. To that end, a number of filtering strategies have now been developed, which can be loosely grouped in several categories. The simplest of the methods involve an *ad-hoc* selection of an arbitrary stringent threshold (Zhou *et al.*, 2002), applying an arbitrary cut-off to a rank-transformed coexpression values (Obayashi and Kinoshita, 2009, Ruan *et al.*, 2010) or evaluating the significance of the detected similarities (Lee *et al.*, 2004a). Other, more complex approaches rely on a statistical analysis of the data, whereas others draw upon other knowledge.

Purely statistical approaches focus on the analysis of the set of expression values themselves to identify the edges to be retained (Markowetz and Spang, 2007). These methods are based on the notion of determining statistical independence of individual profiles (Markowetz and Spang, 2007). The tools implementing this type of analysis include BANJO (Yu *et al.*, 2004), ARACNE (Margolin *et al.*, 2006), NIR/MNI (Gardner *et al.*, 2003, di Bernardo *et al.*, 2005), BNarray (Chen *et al.*, 2006a), GNA (de Jong *et al.*, 2003) and BNFinder (Wilczynski and Dojer, 2009). These methods make it possible to infer an underlying gene regulatory network (GRN) and even recover the direction of the regulatory relationships between the genes. However, their utility is often limited by the type and amount of available data. From the theoretical perspective, a 'perfect' resolution of the GRN is only possible if the number of measurements is greater than the number of genes being studied (Markowetz and Spang, 2007). As this is rarely the case in microarray profiling studies, additional simplifying assumptions or workarounds are often necessary to compensate for it (Markowetz and Spang, 2007). Even with these strategies, the performance of such algorithms is often low when the number of genes considered is >5000. For example, of the three different network inference methods evaluated by Bansal *et al.* (2007), the best performing (ARACNE) achieved only 0.14 precision and 0.35 sensitivity on a set of 7907 genes, whereas in another study PCIT algorithm was reported to out-perform ARACNE with the score of just 0.08 precision and 0.2 sensitivity on a set of 7750 genes (Reverter and Chan, 2008).

The thresholding strategies that rely on prior knowledge work try to optimise the number of links in the network that are known to correspond to meaningful biological relations. Associations commonly used for such verification include confirmed transcription factor-target relationships, pairs of proteins of similar function, proteins known to interact or assigned to the same metabolic pathway. Another possible strategy is to optimise the threshold according to some properties derived from the dataset, which are known to be representative of such associations. For example, in Elo *et al.* (2007) the threshold was chosen according to the clustering coefficient (defined further down) of the resulting network, whereas Zhang and Horvath (2005) have advocated the use of the scale-free topology property as such an indicator.

For this work, the method proposed by Elo *et al.* (2007) was selected. It is clearly superior to simpler, *ad-hoc* approaches as it attempts to maximise a graph property demonstrated to be a good indicator of biologically meaningful relationships. The study presents convincing evidence to that effect, both on real and simulated data and shows that this method makes it possible to achieve the best balance between true and false positive rates. Another possible alternative was to use the functional similarity of genes directly to derive the cut-off threshold. However, as was reported in chapter 3, only about 60% of the *Arabidopsis* genes have at least one functional annotation. It is also challenging to ascertain whether the currently known annotation sets capture all of the real functions for particular genes and there is also no manually reviewed negative control datasets of sufficient size currently in existence. Under these circumstances, the use of functional annotation is likely to lead to many false-negatives due to the missing information.

3.2.2.3 Extraction of insights from coexpression datasets

Once the coexpression relationships between the genes have been determined, the next step is to relate these patterns back to the underlying biological processes being investigated. This step is often very open-ended and may be less formalised because of the need for human input for the interpretation of the more complex patterns observed. Therefore, a range of approaches have been developed ranging from the computationally driven ones to the ones that focus on enabling user-driven interactive query and evaluation.

One of the simpler and often-used methods for summarising these data are expression plots that help to identify related genes or sets of condition where the link between them manifests itself. A number of expression data resources offer this functionality, for example NASC-Arrays repository (Craigon *et al.*, 2004) offers the two-gene scatter plot functionality and ACT (Manfield *et al.*, 2006, Jen *et al.*, 2006) supports the construction of gene co-correlation plots. Other commonly applied types of analysis involve the detection of modular structure, like clique-finding (Shi *et al.*, 2010, Zheng *et al.*, 2010, Manfield *et al.*, 2006, Jen *et al.*, 2006) and clustering approaches (Eisen *et al.*, 1998, Mao *et al.*, 2009, Wu *et al.*, 2002). In Ondex, the clique-finding and other network query methods were added by enabling use of the NetworkX library from the console in the Ondex front end. The implementation of this link was developed as part of the work for this thesis and was presented in chapter 2. Additionally, an implementation of the Markov Cluster algorithm (MCL) was also wrapped in Java and made accessible both in the form of an Ondex workflow plug-in and a function from the console in Ondex front end.

Individual coexpression links and modules often need to be related to the other types of data for their interpretation. STRING (von Mering *et al.*, 2003) and ATTED-II (Obayashi *et al.*, 2007) resources provide two contrasting examples of different strategies for managing and integrating this supporting information. The STRING representation has multiple, typed links supported by the different sources of evidence such as interaction, coexpression or pathway membership. These evidence types can be combined using a special scoring system to attain a confidence value for the association. ATTED-II shows the data from other sources (presented as a network) alongside the coexpression, however it is left up to the user to manually review and relate this information to the coexpression patterns. In the Ondex system all data is integrated into a generic schema, therefore there are no restrictions on the representations and types of the information combined with the coexpression networks. Consequently, once integrated, data can be transformed into either of these representations, if and when needed. The interactive exploration of the network used in this work was supported by the pre-existing filtering methods of Ondex for the selection of particular gene sets and making the comparisons between them. A new type of analysis was also implemented to identify and process the semantic motifs that correspond to the transcription factor-target coexpression patterns. Visualisation functionalities in the Ondex front-end were also extended and used extensively for manual examination of the networks constructed.

Another commonly used approach is to apply summarisation methods to this information, e.g. by determining the enrichment of particular functional role(s) in a module (Mentzen and Wurtele, 2008, Shi *et al.*, 2010, Mao *et al.*, 2009). One such method was developed for this work and was presented in the chapter 3 of this thesis. For the application cases presented here, it was used in combination with Fisher's enrichment analysis to identify the predominant and statistically overrepresented GO functions in the modules respectively.

3.3 IMPLEMENTATION OF THE COEXPRESSION ANALYSIS PIPELINE

To support the coexpression analysis, the Ondex data integration system was extended with a new set of parsers that can import the expression data in various formats. In particular, one of the goals was to investigate whether the selection of an appropriately targeted subset of expression studies can result in a larger number of links relevant to the set of responses of interest. As the main biological focus of this thesis is to explore the regulatory mechanisms of responses to nitrate, the relevance of the dataset was evaluated by comparing the number of known relevant genes which were connected by edges in different coexpression networks. To carry out this comparison, a parser was created to import data from two databases that allow bulk download of coexpression data – ATTED-II and COEXPRESdb (Obayashi and Kinoshita, 2011). Although, as was mentioned in the previous section, other resources also provide coexpression data for *Arabidopsis*, they only allow a limited number of coexpression values to be obtained at a time, which was found to be prohibitively slow and therefore unsuitable for the purposes of this work.

Alongside the support for these databases, a new analysis pipeline was developed to construct coexpression networks from the raw expression data. Coexpression analysis can be both memory and CPU-intensive, therefore this pipeline was implemented in two different implementations – a stand-alone Java-based program, which can export datasets at various stages of the analysis (coupled with a set of parsers to import this data into Ondex), and an Ondex plug-in that encapsulated the same analysis routines and could be run as part of an Ondex workflow. The rationale behind this design was that the re-usability of the analysis pipeline was maximised and the development process was simplified. Ondex is more complex to build and assemble into an executable program, whereas the stand-alone version could be compiled and deployed very easily and was more suitable for prototyping and rapid development.

3.3.1 Implementation overview

An overview of the pipeline is provided in **Figure 3.2**. To start the analysis, a list of microarray experiments from one of the three supported databases needs to be provided by the user. Currently three prominent microarray data warehouses are supported – NASC-Arrays (Craigon *et al.*, 2004),

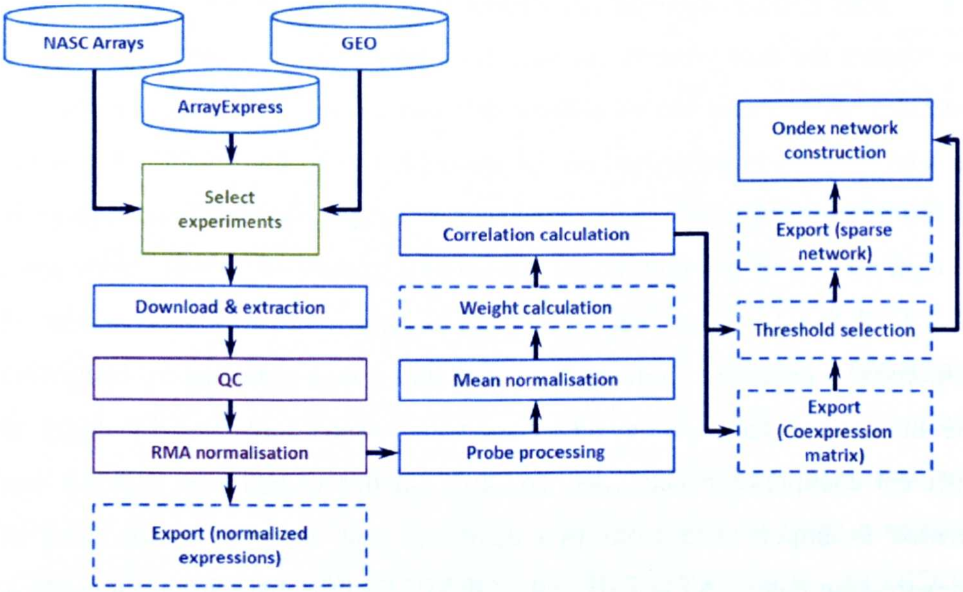


Figure 3.2 Overview of the coexpression analysis pipeline. Optional steps are highlighted with the dashed outlines, manual steps – in green and R steps – in violet.

ArrayExpress (Brazma *et al.*, 2003) and GEO (Barrett *et al.*, 2005) . The analysis is designed to work from the raw expression data in the Affymetrix .CEL file format. For each of the experiment identifiers the appropriate FTP URL is constructed and the associated .CEL files are downloaded and decompressed from archives of ZIP or GZIP format from one of the three providers.

The download step is followed by the normalization, after which a table of normalized gene expression values can be saved to a file for further analysis. After that, the Pearson correlation coefficients are calculated, a complete matrix of which can also be optionally be saved. The final step of the process is import of data into Ondex. This can be done either directly from the in-memory representation produced at the end of this analysis or from a previously created file.

This pipeline implements and combines a number of established methods for analysis of coexpression data. All of the methods that were implemented *de novo* as part of the work on this thesis are described in the sections 3.3.2 and 3.3.3 below.

3.3.2 Calculation of correlation values

Calculation of the correlation values follows the protocol used by the COEXPRESdb and ATTED-II databases, as described on their websites. The array normalization was conducted in R/Bioconductor (Gentleman *et al.*, 2004), where each of the downloaded .CEL files is verified, loaded into the expression set object and normalised using Robust Multichip Average method (RMA) (Irizarry *et al.*, 2003). The main Java application delegates to R by generating a necessary script in the S4 language according to the user-specified parameters, which is then passed to the R environment. The particular steps performed on the R side include the initialisation of the required `affy` library (Gautier *et al.*, 2004), loading of the .CEL files, detection and exclusion of the problematic samples and running the RMA analysis itself. The implementation produces a combined table of normalized expression values for all of the .CEL files in all of selected experiments. This table is saved by the R part of the pipeline, which then passes the control back to the main Java application where this table is read into a Java-based array data structure. At this stage, the dataset is centred by calculating the average expression of each gene and subtracting it from individual expression values. The Affymetrix probe set identifiers are resolved using the corresponding mapping file for the array platform and the measurements for the ambiguous probe sets can either be pulled together and averaged or excluded from the subsequent steps of the analysis.

Optionally, this step can then be followed by a calculation of a redundancy weight for each of the slide. This weight can be used to reduce the effects of replication on the Pearson correlation values. Depending on this choice, either the standard or weighted Pearson correlation coefficient is calculated for all gene pairs. In the former case the correlation coefficient is calculated using the following formula:

$$r_{k,l} = \frac{\sum_{i=1}^{n}(k_i - \bar{k})(l_i - \bar{l})}{\sqrt{\sum_{i=1}^{n}(k_i - \bar{k})^2 \sum_{i=1}^{n}(l_i - \bar{l})^2}} \qquad (3.1)$$

In this instance, k and l represent the expression vectors (n samples in length) of the two different genes and $\bar{k}$ and $\bar{l}$ are the corresponding mean values. The correlation coefficient is calculated for all possible pairs of profiles, resulting in a symmetric matrix. If the option to select a weighted version of the correlation coefficient is chosen, first a matrix of similarities of individual samples is calculated. This is done using the same formula, however k and l become two different sample expression profiles, which are n genes long. The weight w_p, for the sample profile p is calculated by evaluating the following equation across all possible pairings of p and all other samples in the set (Obayashi *et al.*, 2007):

$$w_p = \frac{1}{\sqrt{\sum_{j=1}^{m} \frac{\max\left(0, r_{p,s_j} - C\right)}{1 - C}}} \qquad (3.2)$$

The constant C was set to 0.5, as this value was reported to be optimal in the original study. The weight can then be incorporated into the original equation for Pearson coefficient of correlation in order to reduce the effects of the very similar samples (i.e. likely replicas) on the statistic (Obayashi *et al.*, 2007):

$$R_{a,b} = \frac{\sum_{j=1}^{m} w_j (a_j - \bar{a}_{w_j})(b_j - \bar{b}_{w_j})}{\sqrt{\sum_{j=1}^{m} w_j (a_j - \bar{a}_{w_j})^2 \sum_{j=1}^{m} w_j (b_j - \bar{b}_{w_j})^2}} \qquad (3.3)$$

As this step of the analysis is computationally intensive, the implementation was designed to be executed in parallel in order to take advantage of all available computational power. Both the redundancy weight for each possible pair of slides as well as the correlation values itself between every possible pair of genes can be calculated independently from other pairs in respective categories, although as can be evident from the formula, the calculation of weights does need to be finished first. This was achieved by partitioning the total set of vector pairs across several queues, which are worked on by different threads. The result is written into the correct position of the right triangular results matrix object within the thread where the calculation was performed, which can then be exported as a compressed, tab-delimited file.

3.3.3 Threshold selection

The threshold selection approach that was implemented for the coexpression analysis pipeline was first proposed by Gupta *et al.* (2006) and further refined by Elo *et al.* (2007). In the latter study the method was evaluated both on real and simulated datasets and it was found that this strategy of threshold selection has performed comparatively better on real datasets and has consistently matched the best precision/recall trade-off in the simulated data. One of the clear advantages of this method it that it was demonstrated to maximise the number of biologically meaningful links without the need for a hard-to-get "gold standard" to derive the optimum cut-off value. Unless specified otherwise, the implementation of the Elo *et al.* (2007) was used for the selection of optimum threshold in all analysis described in this thesis unless otherwise stated. One of the graph properties suggested to be useful for determining a suitable cut-off is the clustering coefficient. The local clustering coefficient C_i of a node is defined as a ratio of existing and all possible fully connected triples in a connected neighbourhood of that node. The global clustering of a network, $\bar{C}$ was defined by Watts and Strogatz (1998) as the average of all local clustering coefficients in the network.

One of the refinements introduced by Elo *et al.* (2007) was intended to eliminate an error-prone and non-automatable step of manually selecting the cut-off based on the shape of the plot of clustering coefficient against different cut-off values. This was done by suggesting a fully automated and unbiased approach of locating the local minima based on the changes in real clustering coefficient compared to a change in the randomised control as the cut-off threshold was gradually increased. Under this scheme, the control is a randomly generated network with the same node degree distribution as the real network, but with randomly reassigned edges. An expected clustering coefficient in such a network can also be determined using the following formula from Elo *et al.* (2007):

$$C_0 = \frac{(\bar{k}^2-\bar{k})^2}{\bar{k}^3 N} \qquad (3.4)$$

were $\bar{k} = \frac{1}{N}\sum_{i=1}^{N} k_i$ and $\bar{k}^2 = \frac{1}{N}\sum_{i=1}^{N} {k_i}^2$

To minimise the effect of possible noise, the values are put through a median filter before the comparison is made. The threshold value is gradually increased in 0.01 increments and the optimal cut-off threshold (t) is defined as the one that resulted in local minima in the difference, formally defined as:

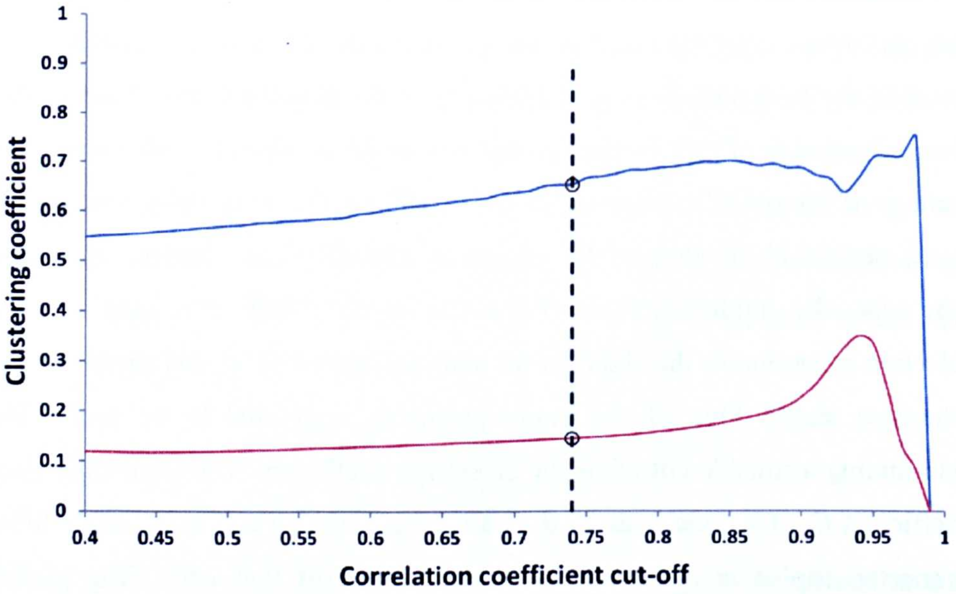


Figure 3.3 Threshold selection example. The graph presents the output from the threshold select algorithm applied to the ATTED-II data. The real data is shown in blue and simulated data is shown in red.

$$t = min_j\{r_j : \bar{C}(r_j) - \bar{C}_0(r_j) > \bar{C}(r_{j+1}) - \bar{C}_0(r_{j+1})\};$$
$$0.5 < r < 1.0; r_{j+1} = r_j + 0.01 \tag{3.5}$$

This approach was implemented in Java as one of the optional components of the coexpression analysis pipeline. A typical output from this analysis can be seen in **Figure 3.3**, which illustrates the process of cut-off determination for the in ATTED-II database. It is possible to see that as the cut-off level gets higher relatively more cliques are recovered from the real network compared to its random counterpart. The line also becomes more uneven as more features are present in the network and the analysis method is designed to identify the very first such occurrence. The original study demonstrated that this method was successful in finding a cut-off value that provided the best trade-off between false and true positives for proteins that share the same function both for real and simulated data (Elo *et al.*, 2007).

3.3.4 Coexpression data in an Ondex representation

As coexpression datasets can get very large, the conversion process can optionally use the information already present in the network during the loading process in order to reduce the time and memory needed to integrate the coexpression data. This is achieved by indexing all of the nodes on the same type of accession as the one used in the coexpression network. This allows to selectively create edges that have corresponding nodes already present in the graph. Another possible additional condition is to only create the coexpression edge in the cases where there already is another type of edge already in existence. The default approach is the threshold-based network construction, whereby coexpression edges are created only when an absolute value of Pearson correlation coefficient is above the specified threshold and nodes are created for genes that have at least one coexpression edge. Optionally, this subset can be constrained by forgoing the creation of new nodes and only creating the links between the nodes already present in the graph (i.e. by matching the user-specified gene accessions). Another option is to restrict the dataset even further and only create a coexpression edge if there is already an edge of a particular type linking the nodes. The latter approach may useful for the application cases where it is necessary to look at the relationship of coexpression and another shared property – for example, shared pathway or protein-protein interaction.

Once the coexpression data is loaded into the Ondex system, it can be combined with additional information, mined further using clustering approaches and graph analysis methods and explored interactively in the Ondex front end. To facilitate visual exploration of the coexpression data, a "colour by value" annotator was extended. The updated version of the annotator allows the size and colour of the edge to be changed to represent the magnitude of the coexpression and also incorporates a number of other convenience features, like filtering or re-colouring of unrelated graph entities.

The analysis pipeline presented in this chapter describes how a set of different pre-existing methods for the construction of the coexpression networks were combined in a novel and flexible way. Together with the time-saving benefits from the encapsulation and automation of the several time-consuming steps necessary to acquire and process expression data, an additional benefit of this work was to make the coexpression data readily available in a semantically consistent representation adopted by the Ondex system. From this format, it can be easily combined with other relevant data (e.g. pathway and ontology annotation) or exported further into other data exchange formats like RDF or OXL. The next chapter will further illustrate how the networks produced using this method can be combined with other integrated datasets constructed for this thesis in order to gain better understanding of nitrogen-responsive processes in *Arabidopsis*.

3.4 CONCLUSION

Microarray data available in open-access repositories like NASC-Arrays and ArrayExpress contains observations of *Arabidopsis* transcriptome under a diverse range of experimental conditions. This data has the potential to provide even more information about how gene expression is regulated. The widely used oligonucleotide array platforms are of particular importance for this task, as they allow expression levels of individual genes to be compared between different studies. To this extent, a number of strategies have been developed to mine and summarise this data, for example Gene Expression Atlas

(Kapushesky *et al.*, 2010), and a number of other resources that allow coexpression values to be calculated for a small set of genes. The coexpression network for the whole set is also available from the ATTED-II database.

However such resources, like ATTED-II, that combine a wide range of measurements from a wide range of large-scale experiments may result in some of the important observation being missed. As some coexpression relationships only come into play under very specific conditions, they may be drowned or drowned out in a larger, more general datasets. The examples presented in this chapter confirm this hypothesis – the ATTED-II dataset provided far fewer coexpression links relevant to the gene list from the *nar2.1* study than the more specialised dataset constructed for this work.

However, the approach that involves construction of more focused coexpression datasets does come with its own problems and disadvantages. Compare to one general, publicly accessible coexpression resource, it can result in greater computational cost (as datasets need to be generated for specific application cases), semantic and syntactic compatibility of data (if different methods are implemented by different research groups), and heterogeneous levels of accuracy (e.g. if different critical value cut-offs are used). The implementation described here addresses these difficulties by leveraging the capabilities of the Ondex system for managing different data formats and takes advantage of the latest developments in the study of coexpression. The resulting method provides an optimum trade-off between scalability, accuracy, portability and consistency.

The scalability was achieved through extensive use of analysis parallelisation and delegation to more efficient implementations, like R/Bioconductor, where it was applicable. Accuracy was ensured by addressing the possible biases – namely by using slide redundancy weighting and network topology-driven approach for threshold selection. Portability of the implementation and the datasets was mainly addressed by incorporating the analysis pipeline into the Ondex framework. As Ondex can operate on a variety of different platforms and is relatively easy to deploy, the analysis pipeline can be installed and run with minimal effort. Output format produced can also be parsed into Ondex, which can then convert it into a wide range of other representations – OXL, RDF, delimited file or even allow direct access to the data via a range of Taverna-compatible web services. As a lot of analysis steps are optional or configurable, the analysis can be adapted to suit a wide range of possible user preferences.

4 INTEGRATION AND EVALUATION OF THE RELEVANT DATA SOURCES

4.1 SUMMARY

The development of a systems based approach to problems in plant sciences requires integration of existing information resources. However, the available information is currently often incomplete and dispersed across many sources and the syntactic and semantic heterogeneity of the data is a challenge for integration. This chapter explains how the Ondex system can be used to study and quantify the differences between resources and dissect different aspects of complex biological data. This analysis is presented in the context of designing the optimal data integration strategy for each combination of resources and types of data used in this thesis. Key genomic, proteomic, functional and localisation datasets used in the subsequent chapters are also presented and the steps and decisions taken during the integration process are explained.

4.2 INTRODUCTION

As was outlined in the introduction to this thesis, a data integration process allows heterogeneous data to be brought together through the identification of common identifiers and creation of mappings between them. The interpretation of the data and resolution of heterogeneities between different data sources are essential prerequisites to this process. Better understanding of the different ways to represent biological data and how they can be reconciled is vital for the continued improvement of the standards and frameworks for management of ever-increasing quantities of biological data. This type of analysis provides valuable insight for the development of tools and approaches to characterize and manage diverse assortment of information, which is currently identified as one of the major unsolved problems in bioinformatics research (Hamdi-Cherif, 2010). The work presented in this chapter has made a contribution to this area of research and some of these findings have now been published in a the Briefings in Bioinformatics journal (Lysenko *et al.*, 2009). Full version of this publication is included in the Appendix. Additionally, several of the datasets described here were used to support other analysis described in the subsequent chapters.

4.3 *ARABIDOPSIS THALIANA* GENOMIC REFERENCE DATASET

4.3.1 *Arabidopsis thaliana* genomic and proteomic data

Although the *Arabidopsis thaliana* genome has now been sequenced, the identification of genes and their correspondence to proteins is still an on-going process. The *Arabidopsis thaliana* genome information resource, maintained by the TAIR initiative (Rhee *et al.*, 2003) is still being continuously refined and updated and at the time of writing, the 10th release of the genome was being prepared for release. Although these refinements are necessary, they are also posing an additional set of challenges when attempting to manage a set of integrated datasets. TAIR gene and splice accessions are often used as primary identifiers for *Arabidopsis* gene and protein sequences. In different releases of TAIR, some of the identifiers used for gene loci and protein splice variants where often updated. However, the process of updating the identifiers by the other providers is often delayed and in some instances may not be possible altogether.

It was clear from the very early stages of this project that in order to support the rest of the integration process, a strategy was needed for constructing a base reference dataset for the *Arabidopsis* genome and proteome. TAIR and UniProt (Apweiler *et al.*, 2004) were identified as the two resources that provided accession numbers that were widely used by the other data providers. In particular, TAIR provides gene locus and splice variant identifiers, while UniProt maintains its own set of protein sequence accession numbers. An important feature of both of these resources is that they also provide cross-references to other major databases.

Additionally, some heterogeneities of semantic nature also frequently occur - for example, in different resources, GO terms may be linked to either gene or protein records. Yet another complication arises from the existence of splice/sequence variants and mutations, which can lead to several protein products being associated with the same gene locus.

4.3.2 Reference set construction

Import of the data was done by using two of the pre-existing Ondex parsers for TAIR and UniProt resources. TAIR parser imports data from the flat files in delimited and FASTA formats, specifically it uses the files holding the publication, protein-coding cDNA and protein sequences, domain mapping, locus history and mappings from TAIR AGI locus/protein identifier to UniProt and NCBI ones. For this work, the TAIR9 release of the resource was used and all of the necessary files were imported from the ftp://ftp.*Arabidopsis*.org/. The import has produced a set of 33410 protein-coding genes associated with a unique TAIR locus identifier. Additionally, these genes were also annotated to publication, domain and protein concepts. All of the entities except protein and gene concepts were removed by using a concept class filter, which removed all entities of a particular type from the graph. Then these two sets of concepts were merged by combining all connected groups on the "encoded by" relation. This step has produced a graph containing a set of 29271 merged concepts with a unique TAIR locus identifier (from the "gene" concept) and one or more TAIR protein/UniProt identifier (from the "protein" concepts).

The UniProt data was imported into the graph by using an Ondex UniProt XML format parser. The data file containing both TrEMBL and SwissProt parts of the database was produced by using the web interface of the UniProtKB website and selecting all protein records corresponding to the NCBI taxonomic identifier of 3702. The UniProt parser creates protein concepts with one primary UniProt identifier and zero or more secondary identifiers, as well as concepts holding additional annotation pertaining to that protein, e.g. publications, enzyme commission numbers, Gene Ontology terms and Pfam protein domains. All of the entities of types other than "protein" were removed using a concept class filters. For *Arabidopsis* entries, UniProt also provides TAIR splice variant identifiers and cross-references to the TAIR loci, which are also imported by the Ondex UniProt parser. However, the presence of these additional identifiers is not guaranteed.

The corresponding entities between the TAIR and UniProt parts of the dataset were identified using an accession-based mapping. Three passes were performed, matching on UniProt, TAIR locus and TAIR splice variant identifiers. The accession-based mapping created a relation of type "equivalent to" between all concepts that share matching identifiers of particular type. After that, all of the concepts from UniProt that did not match any concepts

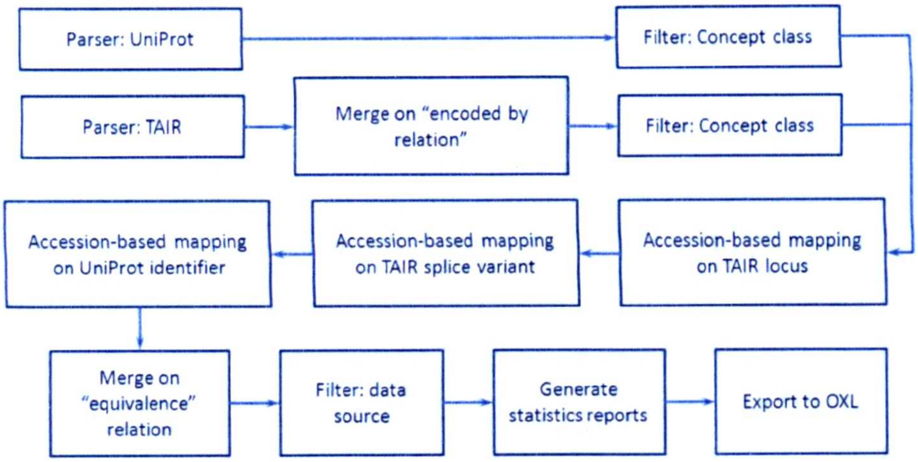

Parser: UniProt
Filter: Concept class
Parser: TAIR
Merge on "encoded by relation"
Filter: Concept class
Accession-based mapping on UniProt identifier
Accession-based mapping on TAIR splice variant
Accession-based mapping on TAIR locus
Merge on "equivalence" relation
Filter: data source
Generate statistics reports
Export to OXL

Figure 4.1 A schematic representation of the workflow processing steps for ARA-REF set creation and analysis.

from TAIR on any of these accessions were removed from the network. The combined entities were created by applying a "relation collapser", which has merged all entities within connected components with respect "equivalent to" relations. In this way, all entitles remaining in the graph had at least one, unique TAIR locus identifier complemented by a list of (also unique) UniProt and TAIR splice variant identifiers associated with this locus entry in both TAIR and UniProt databases. The outline of the integration process is provided in **Figure 4.1.**

The integration process resulted in 2.9% of concepts that had more than one TAIR locus identifier. This was due to a small number of UniProt-TrEMBL entries that matched several possible entities with a unique TAIR locus during one of the accession-based mapping applications. As the number of such entries was relatively small and they were unlikely to cause major adverse effects for subsequent analysis, it was decided to retain them in the datasets to preserve the idea that these represented a complete current proteome set as captured by both by TAIR and UniProt databases and contained a full set of representative accession numbers from both resources. In total, this dataset had 26937 concepts and from this point on is referred to as ARA-REF.

4.4 PROTEIN-PROTEIN INTERACTION DATA

Protein-protein interactions (PPI) are the foundation of many essential regulatory processes and define higher levels of organisation of individual proteins into complete functional units. PPI data are provided by a number of sources, but only one of them (curated TAIR interactome) specialises in *Arabidopsis*. There is a great deal of interest in finding methods for understanding the relationship between protein interactions and coexpression among genes as the basis for making more accurate predictions of biological function from high throughput experiments and for easier identification of metabolic and regulatory networks that underlie biological responses (e.g. to disease, environmental stress etc.). This investigation concentrated on the three most relevant PPI databases and has assessed the coverage they provide in terms of both individual interactions and protein content.

Interactions from the following data sources were integrated using methods supported in the Ondex system: IntAct (Kerrien *et al.*, 2007), The *Arabidopsis* Information Resource (TAIR) (Swarbreck *et al.*, 2008) and BioGrid (Breitkreutz *et al.*, 2008). Although STRING (von Mering *et al.*, 2005) and Bind (Bader *et al.*, 2003) databases also include *Arabidopsis data*, they could not be considered here due to very restrictive licensing and access policies implemented by the data providers. The data from IntAct and BioGrid was imported into Ondex using a dedicated PSI-MI format parser, which was created as part of this work. The PPI data from TAIR were provided in tabular format, and was imported using the tab-delimited API of the Ondex scripting interface, which was described in chapter 2.

4.4.1 PPI dataset construction.

Two of the data sources currently support a PSI-MI XML format, which is an established format for the exchange of the data for protein-protein interaction experiments. The import of this data into Ondex was mediated by a new PSI-MI parser, which was created as part of this work. Internally, the parsing and validation of the source file is delegated to the PSI-MI XML 1.0-beta4 library, which is maintained by the Proteomics Standards Initiative and is freely available for download from http://sourceforge.net/psidev website. The parser itself only handles the index and mapping of the fields in the file to the Ondex data model. Briefly, in the PSI-MI model, data is grouped into experiments, which can have one or more different interactions that can have one or more different participants (proteins as well as other biological entities). The all of these groups can be linked to various metadata that specify provenance, types (of experiments/interaction and interaction participants) and cross-reference to relevant resources.

The parser mediates the transformation of data from this experiment-centric perspective to the network-centric one, where it is decomposed into a set of interacting entities and interaction edges between them. To allow the flexibility of transformation, the parser allows to specify different levels of verbosity – for example, it is possible to create edges of types specifying different types of interactions or just one type of edges of type "interacts with". This may be desirable where the end-goal is a homogeneous network, as the common type will greatly simplify subsequent processing and analysis steps. It is also possible to specify a "spoke" versus "clique" model of representing interactions, which are discussed in more detail further in this section. The PSI-MI parser was used to import the IntAct and BioGrid sets of interaction data for *Arabidopsis*, which was downloaded from the respective resources on 16/08/2009.

The PSI-MI parser also created a publication concept where the interaction was reported and created different typed concepts for different types of interaction participants, (e.g. protein, DNA, RNA, small molecule etc.). As these entities were not required for the planned analysis they were removed by applying a concept class filter.

TAIR curated interactome file "TairProteinInteraction" was downloaded from ftp://ftp.Arabidopsis.org, as this file is tab-delimited, it was parsed using the scripting console functionality. The most recent version of this file available at the time was used for this work, which was dated 27/05/2009. The parsing process generated the identical data representation with the one produced by the PSI-MI parser in order to allow comparison and one general "interacts with" edge was created for every pair of proteins in the source file.

Additionally, *Arabidopsis* protein data was imported from the TAIR resource, similar to the way already described in 4.3.2, except that in this case, the collapsing on the "encoded by" relation was not performed. Instead, after the import stage, all of the concepts except the protein concept containing a TAIR

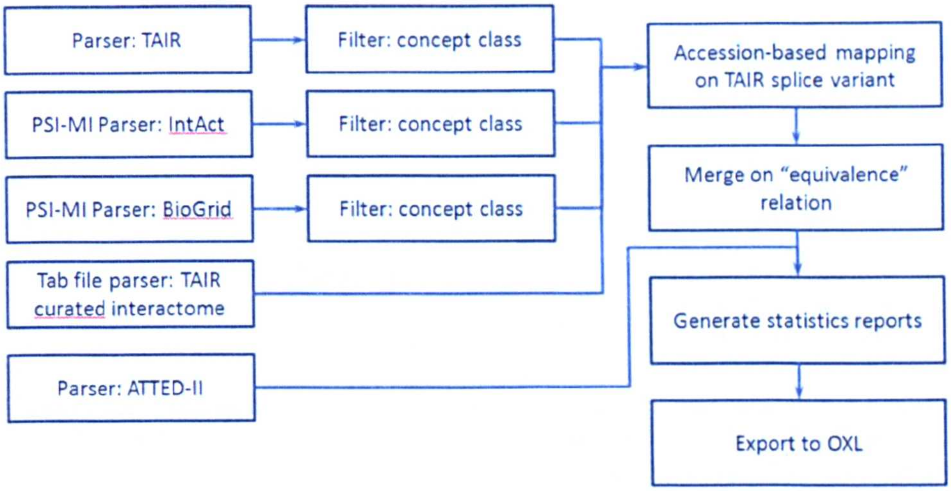


Figure 4.2 Generation of the protein-protein interaction and coexpression dataset.

protein identifier were filtered out. This was done because the TAIR parser also captures the locus history information and adds the now-obsolete identifiers to the protein concepts as secondary, cross-reference accessions.

The last resource to be parsed was ATTED-II coexpression database. This resource provides the entire coexpression matrix calculated over 1388 arrays for download as a set of five compressed files. A new parser was created to allow import of this data into Ondex. The ATTED-II database uses the TAIR locus identifiers and provides an option of parsing data in a content-aware mode. In that mode, when the parser is started, the graph is queried for the existing TAIR loci accession and the coexpression edges are created only for the concepts that have a matching accession. In addition to this option being used, the ATTED-II parser was also configured to only create edges for the cases where an absolute value of Pearson correlation exceeded 0.6. Similar to the previously describe procedure; the data was integrated using a combination of accession-based mapping and merging of equivalent concepts identified. The simplified sequence of steps for this process is shown in the **Figure 4.2**.

4.4.2 Overlap of protein interaction data sources

The intersection between the data from these three data sources is shown in **Figure 4.3**. The number of proteins (nodes) in the integrated network was 2741 but only 503 out of 5480 interactions in the integrated protein-protein interaction network are common to all 3 sources, with the IntAct database

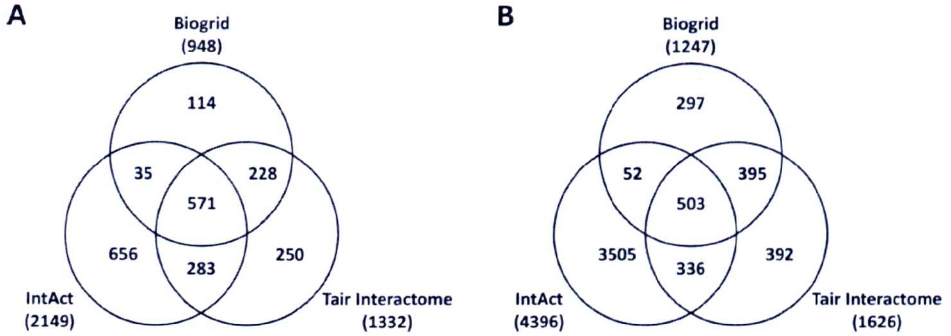


Figure 4.3 The number of protein identities (A) and interactions (B) found in three major protein-protein interaction resources for *Arabidopsis* (IntAct, Biogrid and TAIR Interactome).

contributing many more proteins than either TAIR interactome or BioGrid.

It is apparent from **Figure 4.3** that each of these sources makes a significant unique contribution to the complete network. The presence of a non-redundant component of protein interactions in each of the sources indicates that data from different subsets of PPI publications has been curated by each of the resources and highlights the value of developing an integrated dataset for maximum coverage of a data domain.

An important consideration when analysing protein-protein interaction data is

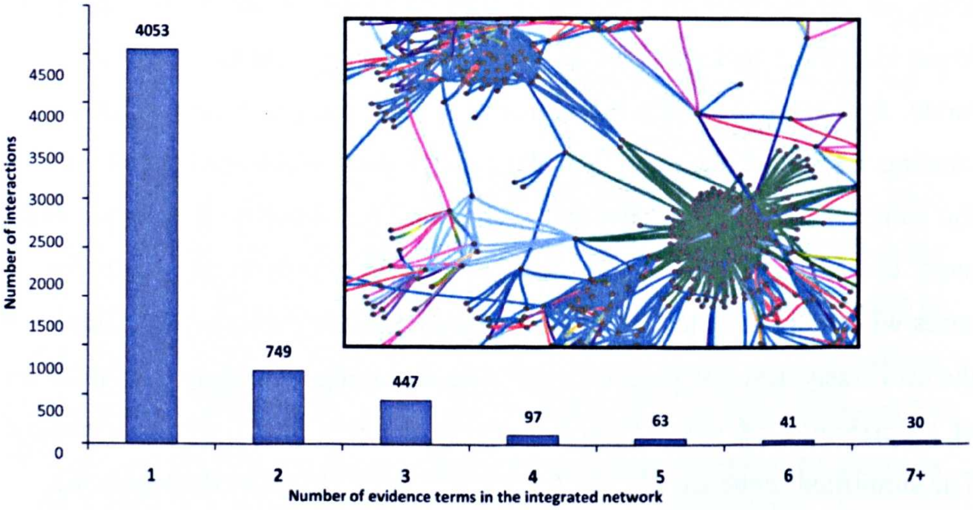


Figure 4.4 The frequency distribution of protein interactions associated with named experimental methods taken from the integrated data from IntAct, BioGrid and TAIR Interactome databases. The upper panel shows how the experimental method used to establish the interaction can be represented by the edge colour. Multiple colours in the same edge show where data from more than one experimental technique is available.

the range of experimental methods that have been used to identify a protein interaction. In the integrated dataset each experimental method used in the source database is represented as a type of evidence, which is stored as a property on the edges (relationships) of the graph. **Figure 4.4** shows the frequency distribution of the number of evidence types in the integrated database. It is evident that most interactions have been confirmed by just one experimental method. The example shown in the upper panel of **Figure 4.4** offers an illustration of how this type of data can be visualized as a network using the Ondex front end tool. The largest connected component of the integrated network has been selected to show how the experimental method used to establish the interaction can be represented by the colour of the edge. Multiple colours in the same edge show where data from more than one experimental technique is available. It is possible to see that one prominent network cluster (green edges, lower right) is supported by the same evidence type. This pattern is indicative of data from a targeted (or fishing) study devoted to finding all possible interactors for a limited number of bait proteins.

The frequency of the various evidence types found in the Ondex database is shown in **Figure 4.5**, which illustrates how integration reveals an inconsistent

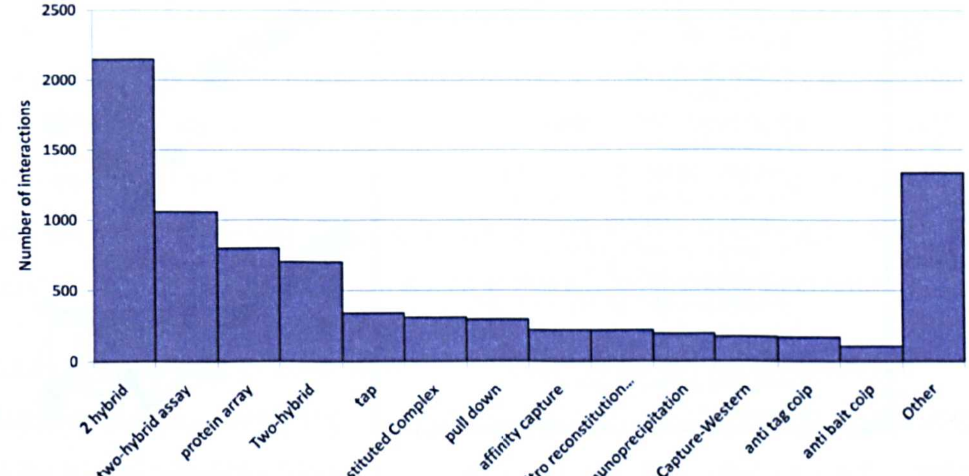
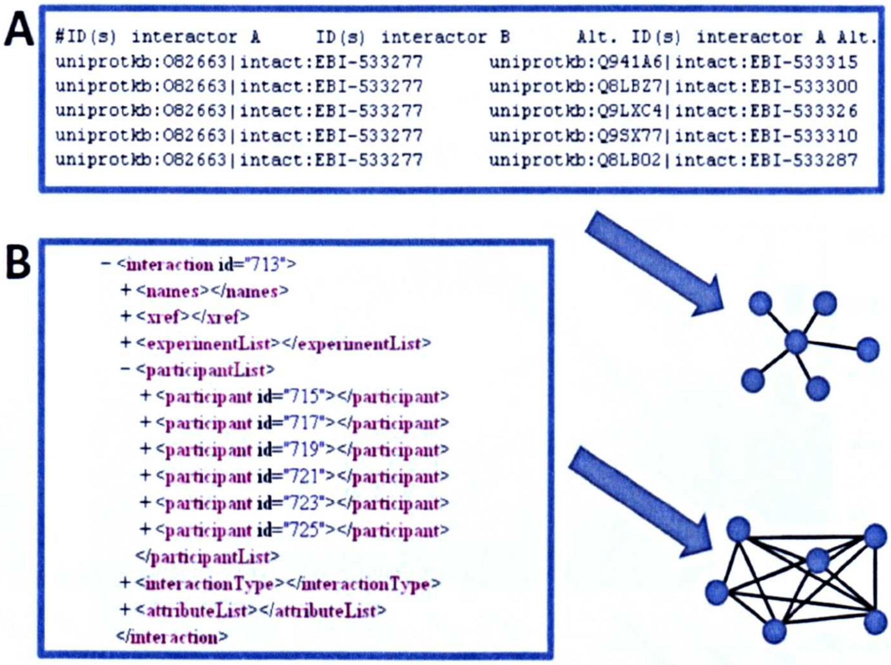


Figure 4.5 The number of protein interactions with a particular evidence type as indicated in the source database calculated for the whole integrated PPI network. Only the 12 most frequent evidence types are shown but in total there are 66 distinct controlled vocabulary terms.

use of controlled vocabularies. Although the vast majority of the interactions among all three sources were established using the yeast two-hybrid method, these are not named consistently among the databases. For example, it is recorded as "2 hybrid" in IntAct and "yeast two hybrid assay" in the TAIR curated interactome. The term "2 hybrid" used in the IntAct controlled vocabulary is formally defined in PSI-MI ontology (MI:0018), whereas the term "yeast two hybrid assay" in TAIR interactome is not formally defined and appears to be used in a broader sense to specify both classical two-hybrid system and a wider range of related techniques. Therefore, it is not a semantically exact match to the definition in IntAct. An important aspect of the different experimental methods is their reliability at detecting a protein interaction. Although this topic is outside of the intended scope of this work, others have developed network analysis methods that take this into account (see for example (Deane *et al.*, 2002)).



```
#ID(s) interactor A      ID(s) interactor B      Alt. ID(s) interactor A Alt.
uniprotkb:O82663|intact:EBI-533277      uniprotkb:Q941A6|intact:EBI-533315
uniprotkb:O82663|intact:EBI-533277      uniprotkb:Q8LBZ7|intact:EBI-533300
uniprotkb:O82663|intact:EBI-533277      uniprotkb:Q9LXC4|intact:EBI-533326
uniprotkb:O82663|intact:EBI-533277      uniprotkb:Q9SX77|intact:EBI-533310
uniprotkb:O82663|intact:EBI-533277      uniprotkb:Q8LB02|intact:EBI-533287
```

```
- <interaction id="713">
  + <names></names>
  + <xref></xref>
  + <experimentList></experimentList>
  - <participantList>
    + <participant id="715"></participant>
    + <participant id="717"></participant>
    + <participant id="719"></participant>
    + <participant id="721"></participant>
    + <participant id="723"></participant>
    + <participant id="725"></participant>
    </participantList>
  + <interactionType></interactionType>
  + <attributeList></attributeList>
  </interaction>
```

Figure 4.6 An example network derived from data from the same experiment represented in two different formats exported from the IntAct database (A) – tab delimited, (B) PSI-MI v2.5 (XML) version 2.5. It illustrates that different formats can sometimes lead to different interpretations of the same information. If the tab delimited representation is used (A) the network consists of only five binary interactions with one hub node, whereas in (B) all six proteins are grouped in the same interaction element, so interactions between all of the members are inferred.

In addition to the issue of reliability, the experimental methods for detection of protein-protein interactions can have an impact on the number of relations and overall network structure. The interpretation of integrated datasets is further complicated by the fact that some experimental techniques do not establish the actual interactions between individual proteins, but rather their membership in a particular protein complex. This poses problems for how to interpret such information in terms of binary protein-protein interactions, as the true interaction pairs are unknown. In some cases, where all of the proteins in the complex form a long-term stable interaction, a fully connected cluster of interactions may be an appropriate representation. In addition to the usual challenges of technical or semantic heterogeneity between the data sources, different export file formats from the same database can lead to different interpretations and can potentially result in the incorrect representation of the experimental interactions. **Figure 4.6** illustrates how this situation can arise because of the different data formats used to extract the data about a particular PPI experiment. The figure shows information from Eubel *et al.* (2003) downloaded from IntAct in both PSI-MI and tab-delimited file formats. The PSI-MI representation groups all of the proteins in the same interaction element, which according to the relevant documentation is interpreted as a clique. In tab-delimited format the same information is represented as a set of five binary interactions where O82663 interacts with all of the other proteins. Both of the representations are actually misleading, as the original paper only identified these proteins as a complex, but did not measure any interactions between them. In general the clique representation may well be acceptable, if the definition of interaction is expanded to include the indirect interactions.

4.4.3 Combining protein interaction and coexpression information

Bringing together multiple types of biological data can aid in the construction of functional networks (Lee *et al.*, 2004b), since proteins involved in the same functional role should be linked by evidence from more than one class of biological information. However, the utility of these approaches is dependent on the information available. For *Arabidopsis*, there are large collections of data from gene expression studies, and resources such as the ATTED-II database (Obayashi *et al.*, 2009, Obayashi *et al.*, 2007) provide information on coexpressed *Arabidopsis* genes from some 58 microarray experiments. There is, however, much less information available on protein-protein interactions from *Arabidopsis* and the integrated dataset constructed included only 2741 proteins, with 5480 interaction pairs in total. This set was integrated with the coexpression information in order to explore the extent to which interacting proteins also display similar expression profiles.

From a total of 5157 edges in the integrated PPI network that were considered coexpressed only 253(4.9%) edges in the integrated dataset were both coexpressed and involved in a protein-protein interaction. This number is somewhat contradictory to the previous work by other researchers who have demonstrated that coexpression to be a strong predictor of protein-protein interactions (Kemmeren *et al.*, 2002, von Mering *et al.*, 2002). However, a permutation test would be necessary to conclusively prove that the result observed here is statistically significant, though such a test could not be done in this case as the data was no longer available at the time this thesis was written. Another possible explanation for this observation could be that it reflects a high number of transient interactions recorded in the dataset. In Jansen *et al.* (2002) it was found that no transiently interacting proteins had an average correlation coefficient higher than 0.4; which is below the threshold of 0.6 that was used for coexpression network construction. Evaluating the influence of different thresholds on the structure of the integrated data set is deferred to future work.

Constructing functional networks in plants is currently limited by the lack of data for some classes of biological information such as protein-protein interactions, where few experiments have been conducted. Such approaches, however, do have the potential to provide additional insight by suggesting new relationships between proteins, especially when complemented by visualisation tools that facilitate manual inspection of the resulting networks and dissection of the sources of evidence that contribute to suggesting putative functional modules. This application of the coexpression and PPI data is further explored in the chapters 5 and 6 of this thesis.

4.4.4 Conclusions

In this case study, several protein-protein interaction resources providing *Arabidopsis* data where integrated and compared. They were expected to be more typical of independently developed databases and this was indeed the case. During the analysis, the most obvious of semantic integration problems were identified – that of inconsistent use of terminology to describe the experimental methods by BioGrid, IntAct and TAIR Interactome. This type of heterogeneity is difficult to deal with automatically. While it would be easy to resolve inconsistent naming such as "2 hybrid" and "yeast two hybrid assay", some of the other methods can have multiple variants and different names and will require someone with expert knowledge to identify these correctly. This example illustrates first-hand the importance of using common ontologies for representing common entities. If these three databases followed the ontology for describing the experimental methods, there would not have been the diversity of terms used to name the yeast two-hybrid method in **Figure 4.5**.

All three databases considered hold information about PPI experiments gathered or supported by the scientific literature. The selection of the literature and curation methods inevitably creates differences between the databases. Furthermore, there is a difference between what has been established in an interaction experiment and what is considered as an established fact. For example out of 12 proteins listed as members of the *Arabidopsis* RNA polymerase II complex by KEGG (accessed via the web interface) only 5 were found in the integrated PPI database from all three sources.

Given the differences between the data collection methods used in the three interaction databases, it was notable that the data integration process generated a more complete resource with the number of proteins catalogued as involved in interactions increasing by 27% over the single most comprehensive database, which was IntAct. The number of interactions was also increased by a similar amount relative to IntAct (25%). This clearly demonstrates the potential advantage of integration in this data domain.

It was interesting to note that a relatively small number of proteins were present in all three databases (20%) and an even smaller number of interactions were found in common (11%). One possible explanation of this observation may be that it reflects differences between the data collection and curation strategies of the three databases however, other, more systematic, differences cannot be discounted either, without further investigation.

Another potential benefit of integration of data across multiple datasets is to increase confidence in noisy data by combining multiple 'hints' from independent sources. This is especially relevant for protein-protein interactions, as many of the currently used detection methods have limited accuracy. This analysis showed that relatively small numbers of interactions are supported by multiple sources of evidence. The presence of these multiple evidence can be visualized in Ondex front end environment in order to provide an easy overview of interaction relationships and how specific patterns emerge from the data using particular approaches, such as targeted interaction fishing.

There is an active research interest in Bioinformatics for using indirect evidence that could be used to indicate interactions, including gene coexpression (Jansen *et al.*, 2002, Bhardwaj and Lu, 2005) and inference of interactions from sequence homology (Goffard *et al.*, 2003, Huang *et al.*, 2004). The problem of introducing such indirect evidence is that some numerical measure of confidence, like accuracy of particular interaction detection methods, is required and it is often not provided by the source databases. Another difficulty lies in resolving the provenance of data in order to avoid counting the same piece of evidence captured by multiple sources several times. This is a promising direction for follow-up to this work and therefore maximising the set of protein interactions supported by multiple direct measurement methods is a useful resource for calibrating the methods for combining computationally predicted and measured interaction data.

4.4.4.1 Implications of PPI detection methods on data interpretation

Due to their importance for understanding the behaviour of biological systems, the discovery and characterisation of protein-protein interactions is a subject of intense research interest. A number of different experimental methods have been developed that allow detection of interaction events and the identification of the participating proteins. Several of these approaches, like "yeast two hybrid" (Fields and Song, 1989) and co-immunoprecipitation (Phizicky and Fields, 1995) can now be applied in a high-throughput manner. However, there is now some evidence that some of the methods are likely to produce substantial amount of false positive detections (Deng *et al.*, 2003).

The interpretation and management of experimental protein interaction data poses a considerable challenge for bioinformaticians. A number of different resources have been established to host this type of data for *Arabidopsis* and make it available to the research community. Another major achievement in this area was the development of a set of standards and an exchange format that allows unambiguous documentation of PPI experiments – i.e. the PSI-MI format defined by the Human Proteome Organisation (HUPO). However, many protein interaction resources have not adopted the PSI-MI standard and still provide data in a tabular format, which may not always capture adequate information about the findings from the experiment.

An important complexity in protein interaction data arises because some experimental techniques cannot completely resolve the nature of the interactions. In some methods, like co-immunoprecipitation, a "bait" protein is tagged and extracted together with all of its binding partners. In such methods, it is not possible to unambiguously resolve the direct binary interactions between multiple interaction partners using this method alone. This introduces a need for further analysis to interpret these results as well as a requirement for a suitable descriptive framework capable of modelling potentially complex information about what is actually known about any given interaction.

It is recognised that the data currently captured in PPI databases only describes the finding of the experiments rather than the true links in the protein-protein interaction network, and this may have consequences for downstream computational analysis. However, this problem cannot be adequately addressed based on the information currently captured in the PPI exchange formats. In this work this ambiguity of direct versus transient interactions was partially addressed by using the "spoke" and "matrix" models of (Bader and Hogue, 2002). In a spoke model the assumption is that only one protein has a link to each of the partners for the methods where bait protein can be identified. A "matrix" model is used to represent the assumption that every interaction participant has a link to every other participant for methods where no "bait" protein is used and a set of all proteins is extracted instead.

4.5 FUNCTIONAL ANNOTATION DATASET

Annotation of genes and proteins with their functional role and cellular localisation information is an essential step both for validating the results of analyses and making new inferences from data. It is therefore of great importance that any annotation datasets used are as accurate and as complete as possible. For this work, the functional dataset was constructed primarily by bringing together data from different providers. Although other means of expanding the existing datasets using computational approach are also possible, they were not attempted in this project. The rationale behind this decision was that current bioinformatics resources are supported by a number of diverse and sophisticated analysis pipelines and specialised curation teams and it is unlikely that it will be possible to match this level of quality with the resources of this project

The functional annotation dataset presented here was assembled from the GO Biological Process annotations provided by TAIR, GOA-EBI (Barrell *et al.*, 2009) and UniProt and transcription factor annotation from DATF (Guo *et al.*, 2005), AtTFDB (Palaniswamy *et al.*, 2006) and PlnTFDB (Riano-Pachon *et al.*, 2007). The protein localisation dataset combined the experimentally determined GO Cellular Component annotation from TAIR, GOA-EBI and UniProt, as well as annotation from the SUBA database (Heazlewood *et al.*, 2007).

4.5.1 SUBA database

Subcellular localisation for Arabidopsis proteins database (SUBA) (Heazlewood *et al.*, 2007) is an integrated resource that collects cellular localisation data from compiled from external sources literature-curated annotations (Swiss-Prot, AMIGO and TAIR), inferred locations from gene descriptions as well as providing data from original localisation studies that use either chimeric fluorescent protein fusion and mass spectrometry studies. This resource is highly focused both in terms of species and type of information and provides non-derived and possibly unique information, but at the same time uses a very simple data model both for defining a component (14 categories including "unclear" and "any location" terms) and capturing provenance information (5 possible provenance codes). The annotations offered by SUBA are equivalent to the more widely used GO Cellular Components ones, which are also provided by such prominent resources like TAIR and UniProtKB. From this point of view, it is a quite common example of a smaller, highly specialised resource and as such it offers an interesting example for a case-study. One possible question considered here are to quantify if SUBA provides additional annotation not captured by other larger, more organised but at the same time less focused annotation projects. And the other point of interest is to use the data integration capabilities of Ondex to create a mapping between the SUBA-GO and SUBA-Ondex provenance capture system and dissect the differences between SUBA and other resources in a greater detail.

4.5.2 Gene Ontology annotation formalism

Gene Ontology provides one of the most commonly used controlled vocabularies for unambiguous annotation of genes and proteins. It is structured as a directed acyclic graph (DAG) of terms organised in three independent aspects: "Cellular Component" (CC), "Biological Process" (BP) and "Molecular Function" (MF), these names also correspond to those of the root term for each of those aspects. The edges of the graph are typed according to the nature of the relationships between the terms, which include "*is a*", "*part of*", "*regulates*", "*negatively regulates*" and "*positively regulates*". The "*is a*" type of edge indicates a sub-typing association between the terms. Each of the terms must be connected to at least one other terms via an "*is a*" type of relationship, and due to the formalism of a DAG it also must be transitively be connected to the root term and each term can have more than one parent and child. The root term is considered to be most general in the ontology, and the specificity of the terms increases with their distance from the root. The "is a" edge is transitive – therefore, if an entity is annotated to a child term, it is by extension considered to inherit the annotation of all of its parents.

The hierarchical structure of GO introduces several complications when it is necessary to evaluate the quality of annotation provided by a particular source. As "is a" relationship is purely semantic the distance from the root only provides a very rough indication of terms accuracy. This problem can be addressed by quantifying the accuracy of terms by the information content,

Table 4.1 GO evidence codes arranged by type. The indent level is used to indicate the codes which are a specialisation of another code.

Experimental evidence	EXP	**Inferred from Experiment**	
		IDA	Inferred from Direct Assay
		IPI	Inferred from Physical Interaction
		IMP	Inferred from Mutant Phenotype
		IGI	Inferred from Genetic Interaction
		IEP	Inferred from Expression Pattern
Computational analysis	ISS	**Inferred from Sequence or Structural Similarity**	
		ISO	Inferred from Sequence Orthology
		ISA	Inferred from Sequence Alignment
		ISM	Inferred from Sequence Model
	IGC	Inferred from Genomic Context	
	IBA	Inferred from Biological aspect of Ancestor	
	IBD	Inferred from Biological aspect of Descendant	
	IKR	Inferred from Key Residues	
	IRD	Inferred from Rapid Divergence	
	RCA	inferred from Reviewed Computational Analysis	
Literature-based	TAS	Traceable Author Statement	
	NAS	Non-traceable Author Statement	
	IC	Inferred by Curator	
Unreviewed	IEA	Inferred from Electronic Annotation	
	ND	No biological Data available	

which is derived from the probability (p) of encountering that particular annotation using the formula: $-\log(p)$ (Resnik, 1999). In this case, the probability can be determined by considering the frequency of encountering a term in a combined set of all annotations of a relevant context. The annotation of GO terms with information content was implemented as one of the analysis methods in Ondex as part of this work.

As well as maintaining the ontology itself, the GO consortium has also defined a tabular exchange format for the annotation of genes and proteins to it (GAF currently - v2.0). This format allows an arbitrary external accession to be linked to a GO identifier and also allows to capture some information about the nature of identifier, species, aspect of GO and provenance to be captured.

One of the ways in which provenance is captured is by associating each entry with one of the evidence codes from GO controlled vocabulary (**Table 4.1**). The GO consortium annotation guidelines only attempt to capture the very broad, qualitative properties of evidence. The evidence codes can be divided into four categories, of "experimentally determined", "computationally inferred, curator-reviewed", "curator/author inferred" and "computationally inferred non-reviewed". Although there is some general agreement about the reliability of these four categories, at present there is no general agreement about the accuracy of the more specific ones. It was also identified in Jones *et al.* (2007), there is also some variation in accuracy between the different codes for the information from different data providers. The problem with quantifying this confidence lies in a requirement for a "gold standard" dataset of the correct functional assignments and since the GO annotation process uses expert curation, a reference standard that surpasses it is difficult to find. For this reason, when quality of annotation was compared, the actual evidence codes where evaluated based on the comparison between the categories they belonged to.

By considering these characteristics of GO and the way annotation is structured described in this section, different annotation resources can be compared in terms of: (i) coverage, (ii) specificity of annotation and (iii) quality of supporting evidence.

4.5.3 Data integration methodology

The key integration steps for creation of the combined annotation dataset are shown in

Figure 4.7. The first part is identical to the integration done to create the ARA-REF dataset, with the exception of the filtering step of the UniProt data. As UniProt parser creates the concepts to represent GO terms and relations connecting them to proteins, this step was adjusted to retain them. The other two GO annotation sets were imported using a pre-existing GAF 2.0 parser. A limitation of the Ondex data model is that it only allows one set of unattributed evidence for edges in the "Evidence Type" attribute. As this was the way GO evidence codes were stored it was necessary to extract this information into a

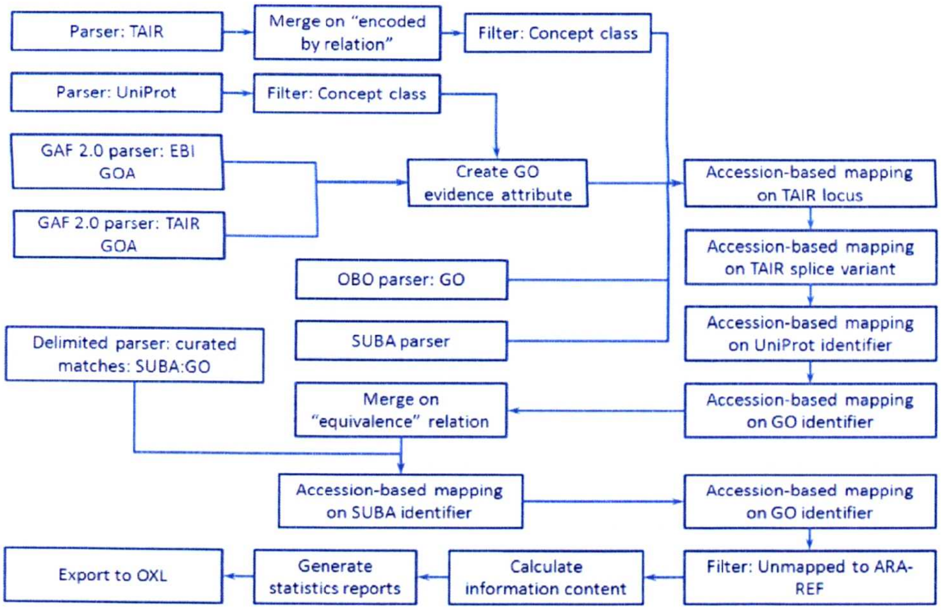

Parser: TAIR
Merge on "encoded by relation"
Filter: Concept class
Parser: UniProt
Filter: Concept class
GAF 2.0 parser: EBI GOA
GAF 2.0 parser: TAIR GOA
Create GO evidence attribute
Accession-based mapping on TAIR locus
Accession-based mapping on TAIR splice variant
OBO parser: GO
Delimited parser: curated matches: SUBA:GO
SUBA parser
Accession-based mapping on UniProt identifier
Merge on "equivalence" relation
Accession-based mapping on GO identifier
Accession-based mapping on SUBA identifier
Accession-based mapping on GO identifier
Export to OXL
Generate statistics reports
Calculate information content
Filter: Unmapped to ARA-REF

Figure 4.7 High-level overview of the workflow steps used to generate the *Arabidopsis* functional annotation dataset.

special, general-purpose attribute. This was step was done by creating a new type of Ondex plug-in specifically for this task. As the GAF 2.0 format only stores the links between the GO terms and gene/protein identifiers and not the relationships between the terms, the structure of the ontology was imported via an OBO format (GO, 2004) parser. Additionally, SUBA database was imported using a specially written parser and a manually created mapping file of matching SUBA compartments to GO Cellular Components was imported using a tab-delimited parser.

As each of these resources created its own set of gene, protein, GO terms and SUBA compartment concepts, the resulting network was subject to considerable redundancy. This redundancy was resolved through the multiple applications of accession-based mapping and equivalence merging plug-ins in a sequence shown in

Figure 4.7. The SUBA:GO mapping file was imported after this step to preserve the unique identity of these concepts but allow the correspondence to be represented via the equivalence relations. At the end of this process all gene and protein entities that were not mapped to the ARA-REF component of this dataset were filtered out. After that, all GO and SUBA terms were annotated

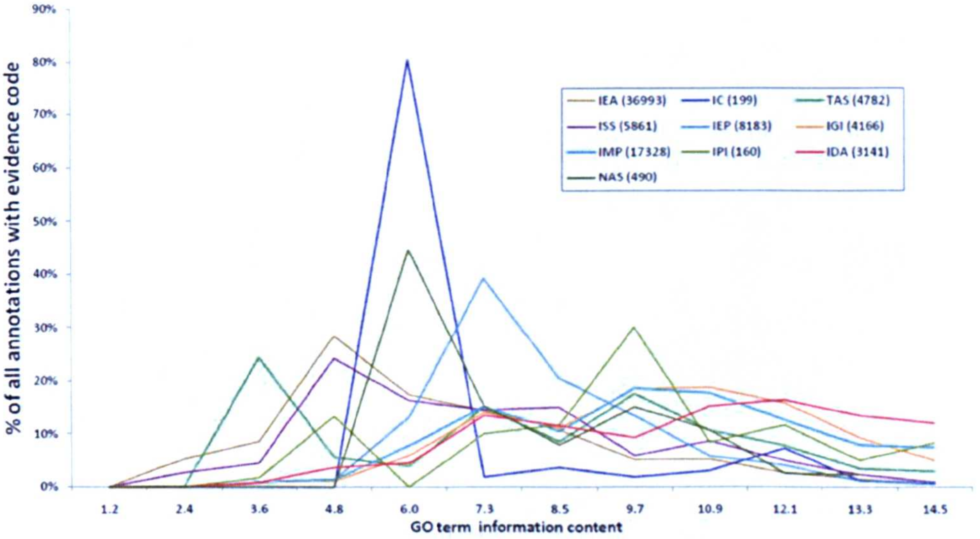


Figure 4.8 Relationship between the evidence type and the information content of GO terms annotated by it.

by creating an information content attribute, which was calculated using a complete, non-redundant set of all annotations to ARA-REF. Then, a specially written analysis/report plug-in was run to collect the range of statistics presented in the next section.

4.5.4 Results and discussion

GO allows for the simultaneous existence of multiple annotations that may have either different levels of specificity or different level of confidence. Therefore, when several sources of annotations are considered, they are likely to differ not only due to the numbers of annotated entities and instances, but with respect to these other factors as well. This introduces an additional level of complexity when comparing these resources – as they could be different with respect to all of these factors and the decision about which one is more important is likely to have some effect on all the others.

To gain a better understanding of the relationship between the specificity of annotations and the quality of the supporting evidence for different annotations, a measure of information content (IC) was calculated for all the terms in the BP aspect of GO using the combined set of all annotations. The proportion of annotations for each evidence code that fall within a particular IC range is presented in **Figure 4.8**. It is possible to see that experimental evidence types tend to be associated with the more informative whereas computationally

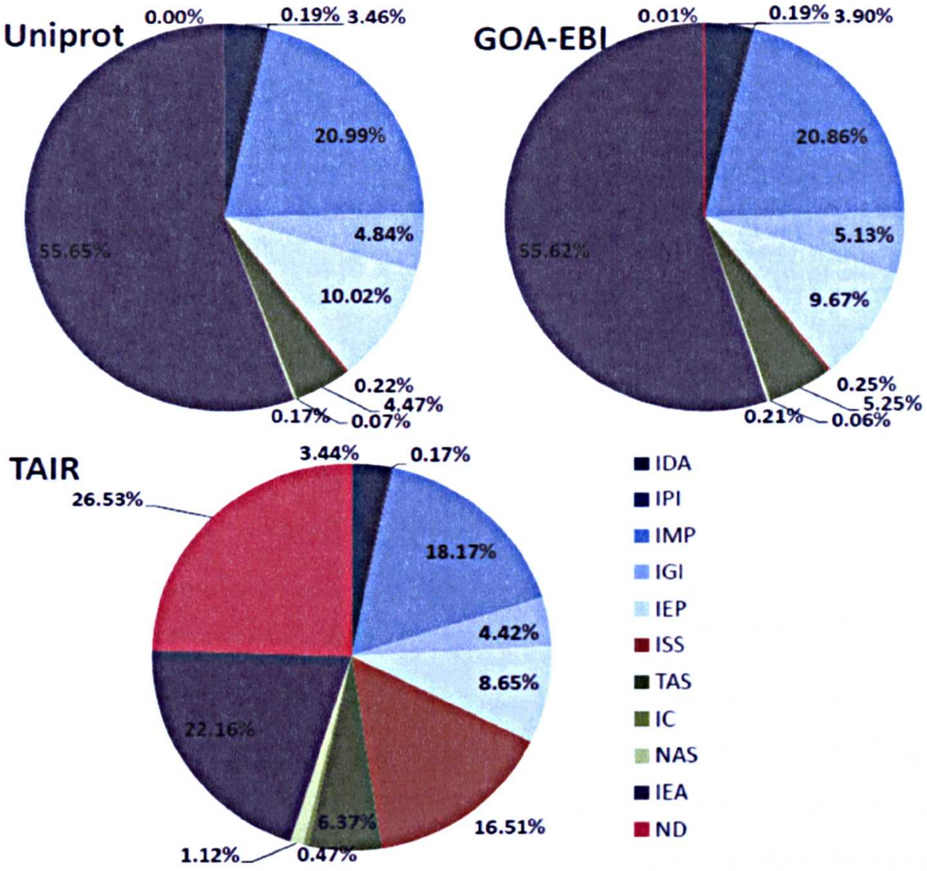


Figure 4.9 Use of the different evidence codes by the three GO annotation resources. The experimental codes are shown in blue colors and literature-derived ones – in green. The red segment is "No data" code, used to indicate that a search for reported functional annotation was done by a curator but has returned no results.

determined evidence types correlate with the less informative terms. Evidence types from author and curator inferences do not show any obvious bias. However, it is important to note that the number of annotations is also very different (shown in brackets in the figure legend) – and even though most of the IEA annotations are associated with the terms in the 3.6-4.8 IC range, there are still some annotations in the high IC range as well and, in absolute terms, this number is much greater than the number of experimentally established annotations in the same IC range.

Figure 4.9 shows the distribution of these evidence types from the UniProt, GOA-EBI and TAIR resources, which have 24686, 27967 and 34743 annotations respectively. Interestingly, the UniProt and GOA-EBI appear to have a very similar composition, even though GOA-EBI was actually found to have 3281 more annotations. TAIR also has a very sizable proportion of ND annotation. This evidence code is used to indicate that the curators attempted but did not succeed in finding any meaningful annotations in this aspect of GO. According to the guidelines, it should only be used to support annotation to a root of one of the aspects of GO. As all of the terms in that aspect inherit its annotation, by extension it implies that no data was located for any of them. If these 9219 entries are excluded, TAIR provides 25524 meaningful annotations, which is comparable to other resources. Another thing to note is that the distribution of evidence types in TAIR appears to have a lot less computationally derived evidence types, but also the largest set of reviewed computationally derived annotations (ISS). However, even if the IEA and ISS annotations are combined, this number is still much less than those found in either UniProt or GOA-EBI. If the ND annotations are disregarded, it is also evident that TAIR actually has the most experimentally annotations (47.42% of all non-ND entries, versus 39.51% and 39.76% for UniProt and GOA-EBI respectively). Overall, it appears that TAIR has the best annotation with respect to evidence quality, but slightly lower coverage than the other two resources.

The comparison of redundancy and annotation specificity between the sources is shown in the upper panel of **Figure 4.10**, whereas the lower panel shows the comparison of evidence quality for each of the possible cases – i.e. more, less or the same specificity of annotation. The tiers on the lower panel compare the evidence codes according to their membership in higher level groups, where the quality relationship is assumed to be EXP > ISS > Curator/author statement > IEA. This comparison shows that TAIR has the largest proportion of the unique annotations, although if the ND annotations are excluded from this count, this number is reduced to 3481. This is still substantially higher than the next best – UniProt with 1229. Although UniProt and GOA-EBI have a very similar composition of evidence types, there appear to be some differences in the actual annotations made by the two resources – e.g. although GOA-EBI have more annotations overall, UniProt has more than twice the number of unique entries.

The lower panel of the **Figure 4.10** provides some further insight into the

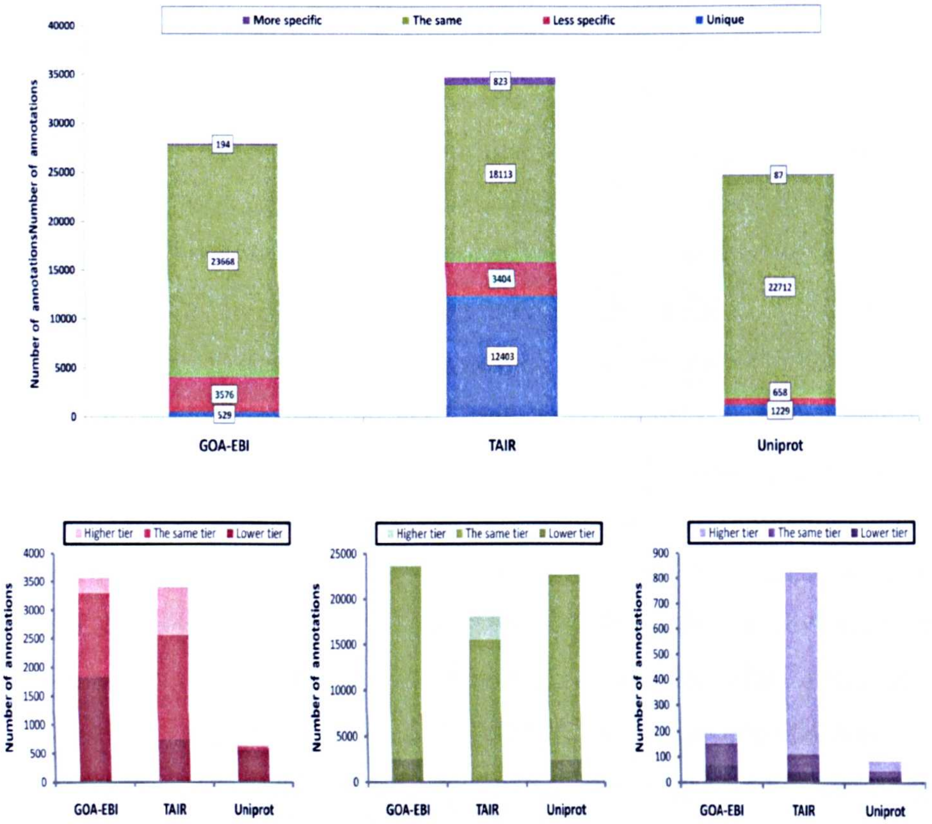


Figure 4.10 Comparison of the annotation specificity between different resources (upper panel) and evidence type confidence for the annotations in common (lower panel). The colouring is consistent between the two panels. The tiers refer to the quality of the evidence supporting the annotation. For example the "higher tier" means that another resource has a better evidence code to support an identical GO function assignment.

supporting evidence for all of these three categories. It appears that in the cases where TAIR had the more specific annotation, it was also the case that it was supported by the better quality type than the one found in either of the two sources. For the cases where the annotation term was matched exactly, the evidence source was also predominantly the same, possibly indicating data sharing between the resources. In the cases where a more general annotation is made, there also appears to be a much higher proportion of cases where the evidence type used for it was weaker – as indicated by a much higher proportion of 'lower tier' entries compare to the 'better' or 'the same' cases. In all three situations, TAIR appears to have the largest proportion of better quality entries, most likely due to the generally higher numbers of stronger evidence types present in the resource.

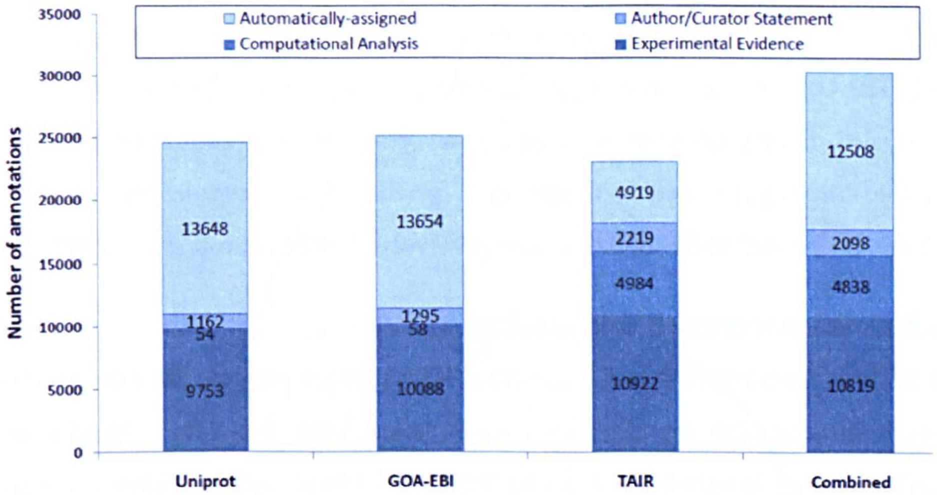


Figure 4.11 Comparison of the evidence types found in the datasets after the redundancy was removed. All sets were filtered by removing the more general GO annotations, if a descendant term was also used.

As only one, non-redundant set of annotations is required for subsequent analyses, after the data was integrated it was necessary to remove some of the annotations so that no entity was left annotated by both parent and child GO terms. It emerges from the above discussion; there are two possible strategies for doing this filtering. The first would be to maximise evidence quality and only keep the more confident annotations. The second would attempt to maximise the precision and retain the lowest-level and most informative terms. It was decided that the latter strategy was more appropriate for the purposes of this thesis. The rationale behind this choice was that the annotations to the higher level term are still retained due to the semantic relationship between them and as was demonstrated in **Figure 4.8**, there is a trend for more informative annotations to be associated with the better evidence types. **Figure 4.11** compares the contribution from the different evidence types in the final integrated non-redundant dataset with the three contributing resources. Note that for this comparison the internal redundancies within the individual datasets were resolved first and ND annotations were discarded. As expected, the integrated data set benefits from the unique annotations found in all of the sources. There is a very slight reduction in overall quality of evidence in all of the cases when the combined dataset is compared to TAIR. However, the number of IEA annotations is also smaller than in UniProt and GOA-EBI indicating that there was some kind of internal re-shuffling among the four evidence categories.

4.6 TRANSCRIPTION FACTOR ANNOTATION

4.6.1 Overview of relevant data sources

It order to use information about annotations of transcription regulator function in combination with coexpression networks to predict possible regulatory links (chapter 5) it is important to maximise the number of proteins annotated as potential transcriptional regulators. This was achieved by integrating three additional specialised resources for this type of annotation: AtTFDB (Davuluri *et al.*, 2003), DATF (Guo *et al.*, 2005) and PlnTFDB (Riano-Pachon *et al.*, 2007). The first two of these resources specialise in *Arabidopsis* only, whereas PlnTFDB contains data for other plant species. The AtTFDB resource is part of AGRIS family of resources and also provides data about transcription-factor-to-target-gene relationships and predicted *cis* regulatory sites via AtRegNet and AtCISDB databases respectively (Palaniswamy *et al.*, 2006).

4.6.2 Data integration methodology

All of these transcription factor information resources provide their data as one or more tabular files. In the cases where the exported data was presented as a set of multiple, interlinked files (AtTFDB and PlnTFDB) specialised parsers were written to import data from in to Ondex. As all of these resources use TAIR locus identifiers for their genes, this accession was used to merge the data with the ARA-REF dataset. From the combined set of all TAIR loci accessions, only 62 did not have any matches to ARA-REF and were removed using a data source-based filter. The evaluation of the information contributed by the different resources was simpler in this case because annotation as a transcription factor is used in its broadest sense and does not require hierarchical structure to model. Therefore, the only evaluation possible was a direct comparison of data content. From the perspective of gene ontology, the transcription factors are identified by the annotation term "regulation of gene expression" – or any of its descendants. The comparison to GO-BP also investigated the coverage of each subset by this overarching term and all of its descendants, and provided an additional check that the definitions of a "transcription factor protein" used by each of these resources still corresponded to the correct functional role.

In addition to the annotation data, all available experimentally confirmed TF-to-gene relationships from AtRegNet were also imported and retained in the dataset. The dataset of all of these annotations plus the content of the AtRegNet database in Ondex is referred to as TF-ALL in subsequent parts of this thesis.

4.6.3 Coverage analysis and statistics

At the time of writing, AtRegNet contained only 1451 transcription factor-target interactions for just 24 transcription factors. As such, this data only covered about 1% of all proteins annotated as "transcription factors" in all three databases. The shortage of curated data or experimentally verified data of this nature was evident from the very start of this project and was one of the main motivations for the development of the coexpression analysis pipeline (described in chapter 3). Interestingly, all of the resources have contributed

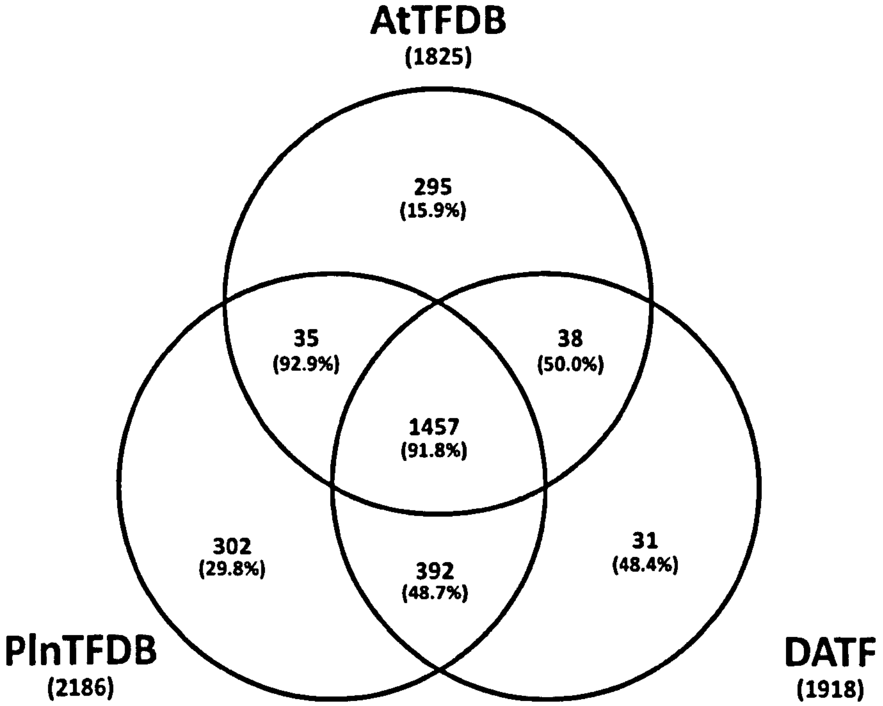


Figure 4.12. Comparison of annotation of *Arabidopsis* proteins as transcription factors by three databases. The percentage in brackets indicates which proportion of this number is also annotated to the "regulation of gene expression" GO term.

some unique annotations and had a considerable overlap with the corresponding GO categories (**Figure 4.12**). The smallest number of annotations to the corresponding GO category ("regulation of gene expression") was found to be for the "AtTFDB only" subset. However as AtTFDB uses the most conservative definition of a transcription factor, combined with the most complex analysis pipeline and manual curation of all entries, the most likely explanation for this observation is likely to be the higher sensitivity of their method that can detect the most transcription factors missed by all other approaches.

4.7 CELLULAR LOCALISATION ANNOTATION

4.7.1 Overview of relevant data sources

The quality of the interactomes is frequently evaluated (and improved) by looking at the proportions of proteins that are known to co-localise together (von Mering *et al.*, 2002) (Sprinzak *et al.*, 2003) (Geisler-Lee *et al.*, 2007). This because no interaction will be possible if they are never found together in the same place. Since proteins may be localised to more than one area of the cell, it is particularly important to assemble as complete as possible set of annotation in order to minimise the false negative assertions in the cases were localisation of one of the interaction partners to the compartment is not known. Protein localisation information is also available in the GOA format as a set of annotations from the Cellular Component (CC) aspect of the GO ontology and was acquired from the same three data provides (TAIR, GOA-EBI and UniProt). One of the shortcomings of the CC aspect of GO, however, is that the terms are arranged conceptually and the structure of the GO ontology is not designed to provide clear semantics for how different parts of the cell fit together and which components can have a common interface. Therefore, either additional information from other sources or a good understanding of cellular structure needs to be used in combination with the CC aspect of GO when using this information to determine which groups of proteins can come into contact with each other. The GO CC annotations have been used for the verification of the protein-protein interactions in *Arabidopsis* by (Geisler-Lee *et al.*, 2007) who compared the semantic similarities scores of interactors. However this study also used SUBA annotation as another method of evaluation and (Lin *et al.*, 2010) has also used SUBA for this validation.

SUBA offers a simpler classification of protein localisation than GO CC and uses only 13 different categories that correspond to major cellular compartments or structures. This resource is also backed by extensive curation from the literature, integrates information from other annotation sources and uses a number of computational annotation methods from a large selection of available approaches that predict protein cellular localisation. The higher granularity of localisation terms used in SUBA is more suitable for the PPI verification, as it provides a clear and unambiguous ways of defining co-localised groups. Both GOA-formatted annotations and SUBA annotations here were combined to construct the protein localisation dataset on Ondex.

4.7.2 Manual versus automatic term matching

As GO and SUBA cellular localisation definitions have different semantics, a manual identification of the corresponding terms was required. This list of pairings was then imported into Ondex as a set of equivalence relations between the terms of the two controlled vocabularies.

However, the possibility of recovering the same correspondence without the manual intervention was also investigated. For this purpose, a special filter was written that looked at the sets of genes that were annotated in both schemes, identified corresponding terms and filtered them based on a specified levels of coverage (relative all annotations by a particular SUBA term) and information content (relative to all annotations in the corresponding resource). In the case of GO, only the direct annotations and 'part of' descendants were used to compute this coverage level in order to minimise the effects of the hierarchical semantic dependence between the terms in GO

The most optimum result achieved is shown in the **Table 4.2**, with the manually selected GO term pairings highlighted in bold. This match was produced by only considering GO terms with the minimum coverage threshold of 40%, e.g. at least 40% of all genes annotated by a particular SUBA terms must also be annotated by that GO term in order for it to be included in the list of march candidates. It is possible to see that in all of the cases it was possible to recover a correct match, with the exception of "extracellular" category. Although there is a corresponding term in GO, which was identified manually, there were no annotations made to it in any of the three resources. Even though in some cases the matching was relatively straightforward, it is evident that in more complex cases like "cell plate" and "endosome" both the coverage and information content criteria were needed to find the correct association. Using this pairing of terms, a new association between all entities that had an experimental annotation to one of these GO terms (or its descendants) and the corresponding SUBA component. This new association was also linked to the original resource that had contributed it to allow further comparative analysis.

Table 4.2 Mapping of SUBA cellular components to the corresponding Cellular Component GO terms.

SUBA term	Annotations in SUBA	GO term identifier	GO term	Information content	Coverage	Annotated to both	Is most specific?
endoplasmic reticulum	323	**GO:0005783**	**ER**	**5.58**	**77.81%**	**249**	*
	323	GO:0016020	membrane	2.04	41.88%	134	*
nucleus	2413	**GO:0005634**	**nucleus**	**2.76**	**78.61%**	**1816**	*
plastid	2350	**GO:0009507**	**chloroplast**	**2.90**	**81.22%**	**1842**	*
plasma membrane	3311	**GO:0005904**	**plasma membrane**	**3.39**	**55.69%**	**1776**	*
vacuole	876	**GO:0005773**	**vacuole**	**5.25**	**46.87%**	**404**	*
	876	GO:0005904	plasma membrane	3.39	63.57%	548	*
cytosol	872	**GO:0005737**	**cytoplasm**	**1.71**	**57.96%**	**488**	*
mitochondrion	972	**GO:0005739**	**mitochondria**	**4.15**	**80.58%**	**751**	*
golgi	219	**GO:0005794**	**Golgi apparatus**	**5.89**	**77.42%**	**168**	*
	219	GO:0016020	membrane	2.04	57.14%	124	
	219	GO:0016021	integral to membrane	3.23	57.60%	125	*
peroxisome	297	**GO:0019818**	**peroxisome**	**7.00**	**52.48%**	**148**	*
cytoskeleton	101	GO:0005737	cytoplasm	1.71	42.27%	41	*
	101	**GO:0015630**	**microtubule cytoskeleton**	**7.16**	**60.82%**	**59**	*
cell plate	34	GO:0005856	cytoskeleton	6.59	56.25%	18	
	34	GO:0005737	cytoplasm	1.71	62.50%	20	
	34	GO:0005904	plasma membrane	3.39	53.13%	17	*
	34	**GO:0009524**	**phragmoplast**	**8.80**	**59.38%**	**19**	*
	34	GO:0015630	microtubule cytoskeleton	7.16	46.88%	15	*
endosome	11	GO:0005904	plasma membrane	3.39	63.64%	7	*
	11	**GO:0005768**	**endosome**	**8.01**	**45.45%**	**5**	*
	11	GO:0016021	integral to membrane	3.23	72.73%	8	*

4.7.3 Results and discussion

Figure 4.13 shows the sources of evidence reported by SUBA for the experimental fraction of the database. The reported sources include TAIR and UniProt, which were also integrated in this study as well as Gene Ontology Consortium annotation (AMIGO), which had previously been omitted from this study because it appears to defer to TAIR as a source of its *Arabidopsis* annotations. The inclusion of both sources by SUBA is likely to be an artefact from using older versions of these annotations, which were incorporated before AMIGO and TAIR were so closely linked. However, the largest fraction of annotations appears to come from the curation of mass-spectrometry and GFP experiments from the scientific literature, which constitute the largest part of all SUBA annotations. Most of the annotations are also only supported by just one evidence source.

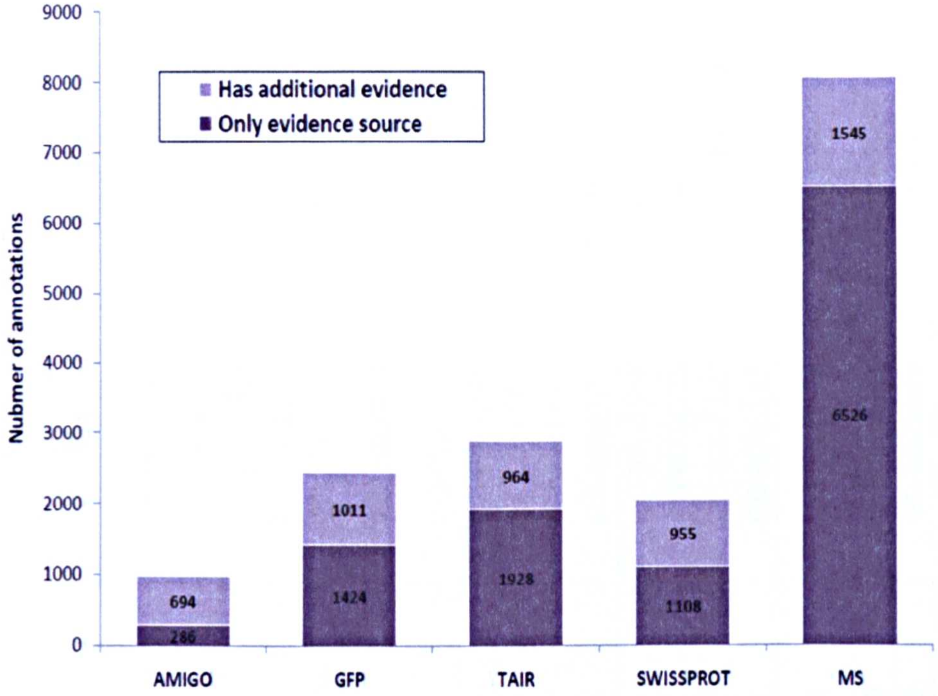


Figure 4.13 Provenance of the experimentally determined subset of the SUBA database.

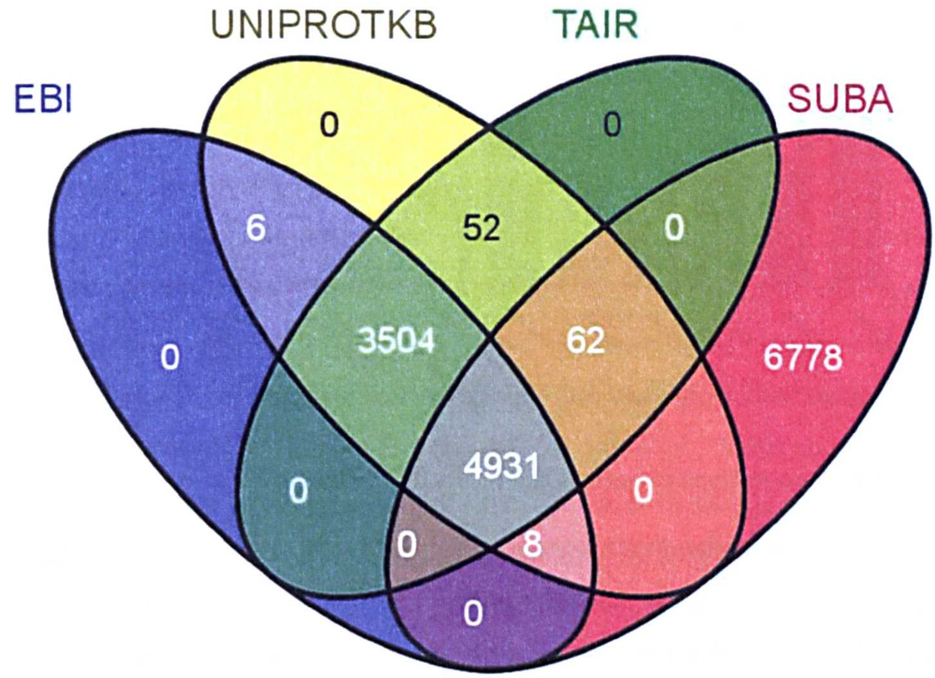


Figure 4.14 Comparison of the annotations to the 13 SUBA localisation terms between the four sources. Visualized using VENNY tool (Oliveros, 2007).

After the GO terms corresponding to the SUBA categories were identified, it was possible to compare the annotations by mapping all of the GO annotations onto the thirteen SUBA categories. The results of this comparison are presented in **Figure 4.14**, which was constructed by looking at the exact matches of protein-term pairs. SUBA appears to have 6778 unique annotations not found in any of the other resources. Surprisingly there are also very few differences between the other three resources, with neither of them having any unique annotations and the vast majority being found in the sets in common among all of them. However, as this representation does not use the original GO annotation to derive these statistics and excludes some of the annotations that are not covered by the 13 terms corresponding to SUBA categories, some of the differences between the resources may have been missed. **Figure 4.15**

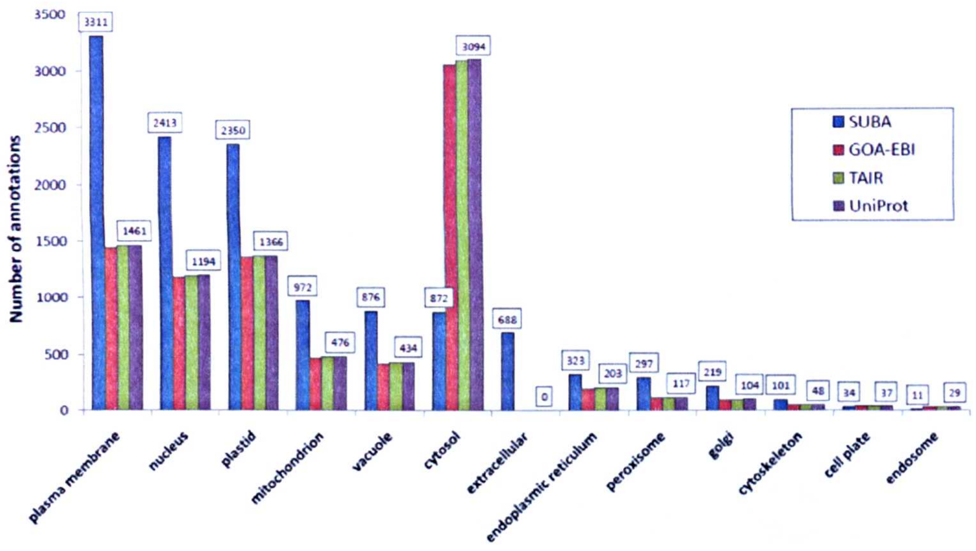


Figure 4.15 Annotations to each of the terms by the four data sources.

shows the counts of the annotations made to each of the thirteen terms. Again, it is possible to see a great deal of similarity between UniProt, GOA-EBI and TAIR, whereas SUBA appears to provide many more categories with the exception of "cytosol", "cell plate" and "endosome". The "cytosol" is the most striking, with SUBA having ~2000 less compared to other resources. As the combined number of annotations in these three cases is still much smaller than the 3560 unique annotations from **Figure 4.14**, the additional resources are also providing some unique annotations in the categories where SUBA has more annotations overall.

4.8 Discussion

The analysis of the different annotation sources have revealed that data integration is essential for assembling representative datasets with the best possible coverage. Although a number of initiatives are in place between data providers to exchange data with each other, there are still considerable differences evident in their content. In almost all of the cases investigated in this chapter each of the sources was found to provide at least some unique information of appropriate type. One notable exception was the use of Cellular Component aspect of GO (GO-CC) annotation set, which was found to be very similar between the three providers. However, considering that the SUBA database held many more Cellular Component annotations, the likely explanation is that the curation of this type of annotation by most resources is of a lower priority than that describing Biological Processes. Therefore, fewer differences develop between the updates that incorporate data curated by other sources of annotation.

The study in this chapter has looked at general data providers, which cover many species and data types (e.g. UniProt), species-specific data providers (e.g. TAIR) and data-type specific ones (e.g. IntAct or SUBA). This comparative analysis indicated that there is no clear relationship between the type/focus of a resource and its comprehensiveness for particular data type or species. This highlights the need for the continued monitoring and investigation of the emergent complexity of the biological data management such as the one published from this thesis (Lysenko *et al.*, 2009) and the rest of the work presented in this chapter, in order to both provide guidance for biological researchers and to improve the quality of information management in life-sciences.

Both the specialist and more general protein annotation resources were found to be important for the construction of the most comprehensive datasets possible, however it was also found there are considerable differences between the semantic models used by different data providers. The major, general data providers like EBI and UniProt now appear to be favouring the use of ontologies as a set of controlled terms to drive their annotation efforts. The smaller, often specialised data providers appear to prefer simpler and less expansive sets of annotation terms (SUBA) or forego such categorisation altogether (AtTFDB). However, although at first glance they appear to lack the resolution of annotation that comes from the use of ontologies, they often make up for it in coverage due to the use of more sophisticated, specialised annotation pipelines and curation teams. For example, the entire set of the GO transcription factor annotations from three major providers was subsumed and exceeded by combining the predictions form the three transcription factor databases. Likewise, SUBA resource contained records for ~6500 more experimentally determined subcellular localisations than were available from a combined set of all GO-CC annotations.

Another vital component that enables navigation among the wealth of data from different providers is the use of cross-references by different resources; where one data provider offers a set of accession numbers from another provider(s) which provide other information for the same entity (Draghici *et al.*, 2006) (Kohler *et al.*, 2003). However the standard accession numbers are often used inconsistently by different resources and despite the wide recognition of this issue by the bioinformatics research community (Pruitt *et al.*, 2005) (Draghici *et al.*, 2006) (Cote *et al.*, 2007), it was found that the problem of ambiguous protein identifier cross-references still remains. In the ARA-REF set there were 2.9% of entries that were found not to have a unique TAIR locus identifier after the TAIR protein-coding gene set was combined with that of UniProt. As mapping across the cross-references is often at the heart of many data integration pipelines, this situation is a source of an on-going concern.

Another topic investigated in this chapter was the management of provenance i.e. the sources of data and the evidence that supported it, provided by different data providers. Provenance is of particular importance when it is necessary to assemble datasets of high confidence entries or to make accurate comparisons between different resources (Zhao *et al.*, 2009). This is the case with the GO evidence types or IntAct experiment types, where provenance can often be employed to produce an estimate of reliability of the particular piece of information. Even in the cases where only the source of the data is retained, e.g. a reference to a paper, it is possible to assemble a higher quality dataset by only including the assertions supported by multiple independent pieces of evidence.

Ideally it is preferable to have knowledge about both the source of the evidence and the method used to obtain it. This makes it possible to identify the cases where an assertion was independently confirmed using the same method. Of all the data providers looked at in this chapter, the best provenance management was provided by IntAct, which used the PSI-MI XML format. This format allows an unambiguously defined set of accession numbers to be provided for each original publication, as well as several fields that capture controlled vocabulary terms for the experimental methods used. IntAct extended this structure by using an ontology of PPI detection methods, rather than just using the controlled vocabulary of method names, as is the case in TAIR and BioGrid. The GOA tab-delimited format endorsed by the GO consortium for distribution of GO annotation allows some information about the method to be captured (through the use of evidence codes) as well as a field for supplying a reference to the original source. However, as the latter is free-form, this information is very difficult to consistently extract computationally. Evidence codes are also too high-level, and do not allow for further dissection and post-processing. For example, it was already discussed in the context of the PPI data that different experimental methods appear to have different degree of confidence. However, in GOA format all of this complexity is concealed behind just one experimental evidence code of "IPI".

The complexity of the current biological methodology itself requires increasingly more sophisticated formats that capture as much metadata as possible about each documented fact (Quackenbush, 2004). This has led to the recent development of a number of relevant standards for exchanging biological data – most notably MIAME for microarray experiment description (Brazma *et al.*, 2001), PSI-MI for the protein-protein interaction data (Hermjakob *et al.*, 2004), BIOPAX for pathway data (Demir *et al.*, 2010) and SBML for biological models (Hucka *et al.*, 2003). However, these standards have not been universally adopted and smaller, less well-funded data resource providers cannot always afford the extra effort needing to capture these more extensive sets of metadata. However, their data still remains valuable and important because they often contribute specific, unique information of relevance – albeit, in a non-standard format(s).

As was highlighted by the GO-BP integration example, datasets of the same type may be different in three different ways – in terms of provenance, coverage and specificity of their annotation. Any data integration system that is to be of practical value therefore must not only provide the functional depth (in that the integrated representation captures the largest possible amount of information from the original source) but also breadth (i.e. support the largest possible number of formats and data sources). Additionally, the integration process itself needs to be tractable and reproducible (Oinn *et al.*, 2004). This is because experimental biological data is continuously updated and so it follows that the data integration process also needs to be re-run to keep the combined datasets consistent with the most up-to-date information. As was evident from the investigations described in this chapter, the Ondex data integration system can adequately manage most of these issues, allowing the integration and comparative analysis of different data sources, as well as the investigation and resolution of semantic heterogeneity between them. In the subsequent chapters, the integration methods and datasets presented here will be further utilized two different contexts, including the identification of functional modularity in gene expression networks (chapter 5) and interpretation of experimental data for prediction of candidate genes (in chapters 6).

5 ASSESSING THE FUNCTIONAL COHERENCE OF MODULES FOUND IN MULTIPLE EVIDENCE NETWORKS

5.1 SUMMARY

Combining multiple evidence-types from different information sources has the potential to reveal new relationships in biological systems. The integrated information can be represented as a relationship network, and clustering the network can suggest possible functional modules. However, one of the challenges inherent to this process is the quantification of the functional coherence of modules in relationship networks. For this work, the functional coherence of modules was defined with respect to the Gene Ontology (GO) by considering two complementary aspects: (i) the fragmentation of the GO functional categories into the different modules and (ii) the most representative functions of the modules. These metrics were evaluated in a number of different relationship networks constructed from the data available for *Arabidopsis thaliana*. The types of data used for this analysis included protein-protein interaction, coexpression, co-occurrence of protein names in scientific literature abstracts and sequence similarity and a combined network with all four types of information. The analysis resulted in a number of novel observations about how functional annotation relate to the structure of different networks. Some of the metrics defined as part of this work were subsequently used as part of the applied application case presented in chapter 6.

The previous chapters have described how the Ondex system was extended to support more complex analyses and presented a range of new resources added to the network. This chapter consolidates this work by using this functionality to construct several different types of relationship networks for Arabidopsis proteins. A new set of methods was also implemented to quantify the functional coherence of the modules and gain better understanding of the effect of using multiple evidence-types. A novel metric (AIC-MICA) was also developed to explore the degree of trade-off between coverage and informativness of GO annotation for a given set of protein.

5.2 INTRODUCTION

The ever-increasing availability of high-volume proteomic, genomic and transcriptomics datasets has led to multiple studies aimed at the systems-level interpretation of this information using biological and relationship networks. Biological networks in this context are graphs where the nodes are molecules and edges indicate interactions between them (Alon, 2003, Aittokallio and Schwikowski, 2006). As explained in (Alon, 2003), in this type of network an allowance can be made for "suppression of detail", e.g. the intermediate components of some interactions may be omitted and instead represented by an edge. Most commonly this type of abstraction is used to represent gene regulation, where the DNA-protein interaction, transcription and translation are represented by just one edge between the regulator and its target protein. Relationship networks (Chen and Sharp, 2004) are a superset of biological networks, where there is no longer a restriction that an edge must represent actual real-life processes that link the two molecules, but instead may indicate a shared property, such as two proteins having the same type of protein domain or being mentioned in the same publication.

The types of data used for construction of such networks include, but are not limited to: sequence similarity (Weston *et al.*, 2004), shared sequence features (Lee *et al.*, 2010, Mostafavi and Morris, 2010), genetic interactions (Mostafavi and Morris, 2010, Bork *et al.*, 2004, Han *et al.*, 2004, Tong *et al.*, 2004, Gabow *et al.*, 2008), gene coexpression (Mostafavi and Morris, 2010, Lee *et al.*, 2010, Myers *et al.*, 2005, Mao *et al.*, 2009, Mentzen and Wurtele, 2008, Wei *et al.*, 2006), protein-protein interaction (Lee *et al.*, 2010, Mostafavi and Morris, 2010, Bork *et al.*, 2004, Dittrich *et al.*, 2008, Bu *et al.*, 2003, Myers *et al.*, 2005, Myers and Troyanskaya, 2007, Jensen *et al.*, 2008), domain interaction (Pandey *et al.*, 2010, Pandey *et al.*, 2008) and term co-occurrence in the scientific literature (Lee *et al.*, 2010, Chen and Sharp, 2004, Ponomarenko *et al.*, 2010, Myers *et al.*, 2005, Gabow *et al.*, 2008). These types of information can be analysed independently or integrated together in order to encompass a wider range of biological mechanisms, provide additional evidence of association between entities in the network and connect disjoint parts of the network. In these studies, different techniques have been developed for the analysis of relationship networks but they follow the same methodological pattern: partitioning the network into modules, identifying the graph-theoretic properties of the network and relating these to biological function. For the work described in this chapter, a similar approach was adopted and a set of metrics was devised for quantifying the functional coherence of the modules in order to explore the effect of using multiple evidence-types in an integrated relationship-network of *Arabidopsis thaliana* proteins.

Clustering approaches work by identifying densely interconnected areas within a network (Aittokallio and Schwikowski, 2006) and are commonly used to detect modular structure in graphs. In the context of biologically relevant networks, these groups are often referred to as functional modules (Bork *et al.*, 2004, Aittokallio and Schwikowski, 2006). Functional modules in biological networks are groups of molecules that are more linked to the other members of the group than to non-members and have similar function (Alon, 2003). The modular structure can be used to infer function of as yet unannotated proteins (Bu *et al.*, 2003), to discover previously unknown roles of proteins in diseases (Chuang *et al.*, 2007) as well as for better understanding the regulation and interrelationship between different elements of complex biological systems (Mao *et al.*, 2009). The function of a module is commonly identified from the annotation of its members with respect to the Gene Ontology (GO) (Ashburner *et al.*, 2000).

GO consists of three separate categories - Biological Process, Molecular Function and Cellular Component, where each category consists of a controlled vocabulary of terms structured as a directed acyclic graph with qualified edges describing the semantic relationship between these terms. Each protein can be annotated with multiple GO terms and inherits the annotation of the parent terms and this makes it challenging to quantify and analyse the functional similarity between GO annotations. This has stimulated a number of studies that have explored these problems in detail; in particular, the importance of the quantitative characterisation of GO-term specificity. One of the most well-known of these uses information content, (IC) as described by Lord (Lord *et al.*, 2003) and based on this metric, several pair-wise quantitative measurements were developed that take into account the structure and properties of the Gene Ontology (reviewed in (Pesquita *et al.*, 2008)). In a number of related but separate studies, metrics have been devised to measure the semantic consistency among the functional annotations for sets of proteins with the aim of identifying those which were significantly enriched. In this context, the enrichment can provide an indication of how over-represented particular function is in the module. Therefore, can be used pick out most important associations turned out by the analysis out of an often-extensive list of all annotations, many of which are just as likely to occur in that number by chance.

These studies, typically, did not take into account hierarchical structure of GO and although useful, these methods have a number of limitations. Zheng and Lu (2007) pointed out the problems of sensitivity suffering as a result of inconsistent annotation, failure to pick up on the importance of biologically meaningful links between functions and sensitivity to the relative size of the sets, which may lead to much greater importance being given to very rare annotations. Khatri and Draghici (2005) have also discussed the impacts of annotation completeness and correctness on this type of analysis and further identified inability to consider functions in an appropriate context as a limiting factor. Additionally, Khatri and Draghici (2005) have identified a number of implementation related issues of the current functional over-representation tools that impact their usefulness, in particular ease of installation, incompleteness of reference GO annotation datasets and the need to convert between different types of gene accessions. Another set of optional, but "nice-to-have" features suggested included the ability for the user to control the specificity of the terms considered by the analysis and visual presentation of the results. By implementing this type of analysis as part of an established, cross-platform data integration solution with advanced visualisation capabilities many of the above mentioned technical issues can be resolved with minimal effort.

To address the more fundamental shortcomings of the enrichment analysis approach, several extensions were proposed that combine some aspects of enrichment-based methods with adjustments for the relationship between the terms (Xu *et al.*, 2009, Alexa *et al.*, 2006, Richards *et al.*, 2010). At the same time, another set of measures were developed for the quantification of overall relatedness in a set of ontological annotations (Yu *et al.*, 2007, Wang *et al.*, 2007, Ruths *et al.*, 2009, Chagoyen *et al.*, 2008, Zheng and Lu, 2007). The insights that have emerged from these studies were used in this work in order to define a descriptive measure for comparing the functional annotation of protein sets. In particular, the approach presented in this chapter allows the functional annotation of a set of genes to be explored from the perspective of coverage and identifies a non-redundant set of terms that are most informative at that level by considering the ontological structure. The only limitation is that, for the time-being, the result is not supported by statistical validation and it is, therefore, left up to the user to decide whether the observed pattern is likely to be of relevance. However, as is elaborated in this chapter, this method allows effective quantification and comparison of the trade-off between the specificity and coverage of functional annotation in different networks. Additionally, classical enrichment analysis of GO annotation was also implemented as part of this work and both of these analyses were applied in a practical context in chapter 6.

In order to determine the biological relevance of a partitioning of a set of proteins, there are two important aspects that need to be taken into consideration. The first is that the set of GO terms, that best describes the common function of a representative proportion of proteins in the modules, can be found at any annotation specificity level. However, at the higher levels, which are close to the root of the Gene Ontology, the annotation will not be particularly informative. This leads to a trade-off between the specificity of annotation terms and the number of proteins in a module to which it applies. The needs of the particular application case may dictate which of these two components is more important, and metrics have been developed that allow the emphasis to be placed on one or the other (Joslyn *et al.*, 2004). Using the metric defined in this chapter (AIC-MICA) it was possible to explore these two properties in five different relationship networks. The second aspect to be considered is that the proteins with similar GO annotation can be fragmented, i.e. assigned to a number of different clusters by the clustering algorithm. Not only can the functionally similar group be spread across a number of clusters, but also may be more or less concentrated in the clusters where it is present.

To assess the functional coherence of modules from a relationship network, both of these aspects, namely the representative functions of modules and the fragmentation of functional categories, are relevant. In this chapter the potential of combined relationship networks to recover functional modules is investigated by considering four sources of information, protein-protein interaction (PPI), coexpression (COE), sequence similarity (SEQ) and co-occurrence of terms in the scientific literature (LIT). These were chosen because they are often used for inferring functional relationships among genes and proteins and are readily available from the application of high-throughput 'omics techniques. A large amount of coexpression data is available for *Arabidopsis* (see for example, (Obayashi *et al.*, 2009)). Measurements of sequence similarity can be obtained for all pairs of proteins (The Arabidopsis Genome Initiative, 2000) and co-occurrence of protein terms in abstracts can be extracted from the scientific literature (Hassani-Pak *et al.*, 2010).

The set of proteins used for evaluation was restricted to those for which protein-protein interaction information was available, because at the time of writing, this was the least abundant type of data available for *Arabidopsis*. This restriction means that a relatively small subset of *Arabidopsis* proteins was considered, but has the advantage that it leads to a more balanced distribution of evidence types from the four information sources among the relationships between proteins. This setting also allowed an evaluation of the extent that patterns and trends previously found in whole proteome-based networks still hold in situations where only a subset of the whole proteome is analysed. Another motivation was to evaluate the usefulness of these approaches for extracting the best possible information under conditions when data are scarce or incomplete.

5.2.1 Markov clustering algorithm

Some work described in chapters 5 and 6 of this thesis relied on the Markov clustering algorithm for graphs (MCL) (van Dongen, 2000) to partition the integrated networks into functional modules. One of the advantages of this clustering method is its scalability and performance, which means that it can be used to partition even very large networks. For example, it is used as part of the ortholog detection method OrthoMCL (Li *et al.*, 2003) to successfully partition networks of bidirectional BLAST hits for the sets of more than one million individual proteins. Aside from several very successful applications for clustering sequence homology networks (Li *et al.*, 2003, Enright *et al.*, 2002), other studies have also compared the performance of MCL versus other clustering methods in other biological contexts. One of such studies has looked at its ability to correctly detect the modularity in protein-protein interaction networks and has found that MCL outperformed all other approaches (Brohee and van Helden, 2006). Another study that looked at the partitioning of coexpression networks has reported that MCL tied for best performance with their own method (Mutwil *et al.*, 2010). Although no clustering approach can be the ultimate solution to unsupervised network partitioning problem, these reports of good performance of MCL combined with the high scalability of the algorithm suggest that it might be a good choice in a variety of biological network settings.

The MCL algorithm is based on the notion of random walks through the graph, which can be modelled by Markov chains. Such representation is realised by representing a graph as an adjacency matrix, with weights on edges representing a transition probability of a random walker traversing a particular edge between the two nodes. As each column of the matrix represents the edges of a particular nodes and weights representing the probabilities, the sum of all values in a column is always equal to 1.0. A Markov chain set-up allows modelling of the probabilities of a random walker traversing a particular edge after n steps. This set of probabilities is derived by successively multiplying the transition probability matrix by itself n times. As the Markov chain progresses through the steps, it is possible to observe that the more densely connected a region of a graph is, the more likely it is for the random walker to visit it. The MCL process exploits this property by emphasising it further by increasing the transition probabilities of links with a higher value, while at the same time reducing the transition probability of the weaker ones. This is done by raising every element of the matrix into a particular power I (termed "inflation parameter"). Subsequently, each column of the matrix is re-normalised to 1.0. The process of MCL clustering is realised by alternating two different steps:

the progress through the Markov chain (the "expansion" step) and the raising of the matrix into the power of I (the "inflation" step). The original work has demonstrated that the process simulated by this procedure converges on an equilibrium solution at a quadratic rate.

Once the algorithm converges on the solution, the resulting matrix can be interpreted to assign nodes to individual clusters. The clusters detected are the individual connected components, if the matrix is interpreted as an adjacency matrix. The granularity of the clustering can be controlled by an inflation parameter, with higher I generally leading to the recovery of a larger number of smaller clusters. It was demonstrated that the partitioning tends to be quite robust to the changes when used on sparse networks with a clearly defined modular structure, but the effect of I increases when applied to the more densely interconnected networks. Another study has also reported that there was a wide range of applicable values of I (1.5 to 3.0) where changes in inflation had little effect on optimality of clustering when MCL was used to partition the *Arabidopsis* coexpression network (Mao *et al.*, 2009). The strategies for optimising I for particular datasets vary greatly, from choosing the values that optimise a recovery of a particular property (Mentzen and Wurtele, 2008, Enright *et al.*, 2002) to empirical selection based on the visual correspondence between the graph layout and cluster assignment (Freeman *et al.*, 2007), when optimum partitioning cannot be established *a priori.*

5.3 METHODS

5.3.1 Overview

A protein-protein-interaction network was constructed based on experimentally established protein-protein-interaction data from the IntAct database (Aranda *et al.*, 2010) and combined with additional data, namely gene coexpression, sequence similarity and co-occurrence of protein names in the scientific literature. The same methodology for construction of an integrated network of PPI and gene coexpression data using Ondex was applied in (Lysenko *et al.*, 2009). The inherent modular structure of these networks was investigated and related to the underlying biological processes using the Gene Ontology (GO) (Ashburner *et al.*, 2000). Functional properties of these modules were quantified and compared using information content and semantic distance based measures.

5.3.2 Construction of the integrated relationship network

According to the formalism of network representation chosen for this work, nodes represented proteins and edges were added if there was at least one of the possible four evidence types linking these proteins: co-occurrence of protein names in PubMed abstracts, coexpression of genes that encode those proteins (where the magnitude of the Pearson correlation coefficient was greater than 0.6), sequence similarity (with E-value<0.001) or experimentally determined protein-protein interaction.

Protein-protein-interaction (PPI) data were imported from the IntAct database (PSI-MI XML format) into the Ondex system. After that, all entities that were not annotated with *Arabidopsis thaliana* NCBI taxonomy identifier and all interaction participants that were not proteins were removed. The interactions between multiple copies of the same protein were also discarded. All proteins that were not part of any interactions after this filtering were also removed from the set.

A coexpression network (COE) was constructed from *Arabidopsis* coexpression data from the ATTED-II database (Obayashi *et al.*, 2007). An edge was created in the coexpression network if the absolute value of Pearson's correlation coefficient of respective gene expression profiles was greater than 0.6.

For the literature-based co-occurrence analysis of protein names, 30,639 abstracts from PubMed were downloaded which contained the word "Arabidopsis". This set of publications together with *Arabidopsis* protein name information from UNIPROT was loaded into Ondex. The Ondex text-mining plug-in was used to create relations between proteins and publications and transform the output to a co-occurrence network, according to the method described in (Hassani-Pak *et al.*, 2010). An edge in the protein name co-occurrence network (LIT) indicates that there was at least one abstract that included a mention of both proteins.

Sequence similarity was determined by using TimeLogic® Tera-BLAST™

(Active Motif Inc., Carlsbad, CA) for all-against-all sequence-comparison of proteins in the interaction dataset, with E-value cut-off at 10^{-3} and a minimum percent sequence identity cut-off at 25%. One edge was created in the sequence-similarity network (SEQ) per pair of proteins with similar sequences.

5.3.3 Clustering the relationship networks

Natural groupings of the proteins was explored using the MCL clustering algorithm (van Dongen, 2000). This algorithm simulates flow in the network, and can be used to identify strongly connected groups of nodes. The implementation of the MCL (v10-148) algorithm (from http://www.micans.org/mcl/) was wrapped as a function and as a plug-in and made accessible from the Ondex data-integration platform. For this algorithm, the inflation coefficient (I) determines the granularity of the clusters. A value of I=2.8 was used for all of the clustering analysis described in this chapter. This value was chosen to get the best possible balance between the "useful" clusters produced by the algorithm. It was found that at lower thresholds ALL, COE and LIT networks had most of the nodes assigned to one large cluster because nodes in the core of the network were highly interconnected. At the higher values of I an increasingly large number of clusters of size 1 were produced. This value was chosen so that a partitioning of the dense core of the ALL, LIT and COE networks happened, but at the same time the number of clusters of size 1 was kept to a minimum. The partitioning of the SEQ and PPI sets appeared to be quite robust to the changes in I. The clustering was performed on an adjacency matrix relative to the edges of particular type. The analysis did not assign any weights to edges – e.g. any coexpression edge joining two nodes would result in a value of "1" in the adjacency matrix. Likewise, in the case of the combined network a presence of any of the evidence types would also result in a "1", regardless of how many different evidence types supported that edge.

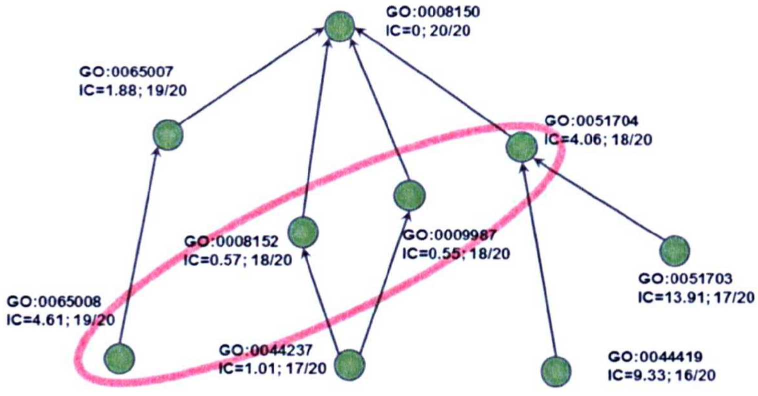
5.3.4 Gene Ontology annotation

To explore the functional groupings of proteins in the network, all available *Arabidopsis* GO annotations were combined from three sources: IntAct (Aranda *et al.*, 2010), GOA-EBI (Barrell *et al.*, 2009) and UNIPROT (UniProt Consortium, 2010). The Information Content (IC) (Shannon, 1997) of the annotations was calculated using the combined set of all GO annotations of the *Arabidopsis* proteome subset as identified in the UNIPROT database. All annotations to proteins not included in the proteome set were discarded prior to calculation of the IC.

5.3.5 Assessing the functional coherence of modules

The overall aim of this study was to assess the functional coherence of modules by exploring two aspects: (i) whether the clusters contain proteins that are generally similar in terms of their functions, as assigned by Gene Ontology terms, i.e. the most representative GO terms in a cluster; (ii) the way in which proteins with the same functional roles are distributed across different clusters, i.e. the fragmentation of the GO terms.

To study the first aspect of functional coherence, a measure was developed that quantifies the annotation similarity at various levels of coverage. Since the GO is described by a directed acyclic graph (DAG), one way of estimating the overall level of commonality of GO terms in a cluster is to find a set of representative common ancestor terms. Terms lower in the GO tree tend to have higher information content but also have a smaller number of descendants. The set of annotations that best summarise the commonality of proteins in the set should therefore be the most informative subset of all applicable ancestor terms. However, as the module identification process is not perfect, the set can contain some noise in the form of proteins that are not functionally related to the rest of the modules. Another possible scenario is that a module itself has complex structure and is composed of sub modules with different functions, which work together to realise some high level biological process. For example, regulatory processes often involve both transcription factors and signalling proteins. One way of accounting for these possibilities is to allow a certain number of outliers when identifying the set of most informative common ancestors. The IC of all terms in a set can be average to give Average Information Content of the Most Informative Common Ancestor set (AIC-MICA), which provides a measure of functional coherence for a set of proteins. The procedure for calculating AIC-MICA is explained schematically in **Figure 5.1**.



Most informative terms applicable to 90% of the cluster
AIC-MICA = (0.57+4.61+0.55+4.06)/4 =2.45

Figure 5.1 Example calculation of the average information content for cluster coverage level.

In order to study the second aspect of the functional coherence, two metrics were used to evaluate the fragmentation of the protein sets annotated with the same GO annotation terms compared to the groups to which they were assigned by the clustering algorithm. A term from the biological process category of GO is defined as t, a set of all proteins annotated to the term t as A_t and a set of clusters that contain at least one element of A_t as C_t. N_t denotes the number of fragments of t, as the cardinality of C_t, and p_k -- the proportion of the total number of proteins annotated with term t found in cluster k :

$$p_k = \frac{|k \cap A_t|}{|A_t|} \qquad (5.1)$$

with $k \in C_t$.

And the entropy (H_t) was defined as:

$$H_t = -\sum_{k,k \in C_t} p_k log(p_k) \qquad (5.2)$$

Similar to the number of fragments N_t, the entropy H_t gives a measure of the fragmentation of the term t across the clusters, but it also accounts for the distribution of the size of the fragments (see **Figure 5.2**). The average entropy obtained for each of the real networks was compared to a randomized control, where cluster labels were permuted 10000 times.

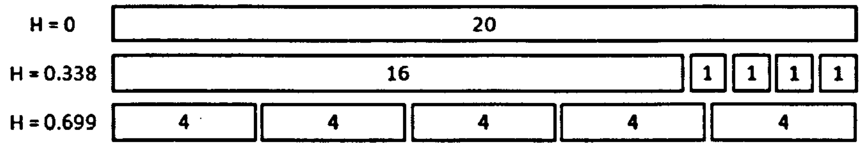


Figure 5.2 Schematic diagram showing how entropy provides a useful metric of fragmentation of a given GO term across clusters. If 20 proteins are associated with a given GO term and they all are in the same cluster then the entropy (H) is zero. If most (16) of the proteins are in one cluster and the remaining proteins are in separate clusters H=0.338. However, as the proteins get more evenly distributed across clusters the entropy increases.

In order to compare the number of fragments and the entropy of fragmentation to the sources of relationship data, both of them were ranked for each of the GO terms across all five networks. This was done by counting the number of times each of the data sources was assigned the best rank (i.e. the lowest value) and calculating a proportion with respect to the total number of GO categories. For the sake of brevity, the abbreviations BFRP (best fragment rank proportion) and BERP (best entropy rank proportion) were used when referring to these comparative measures.

5.3.6 Visualisation

The integration process was implemented as a set of workflows in the Ondex Integrator (Canevet, 2010). The resulting network was visualized and further analysed in an interactive manner using the Ondex user client by invoking visualisation and analysis functions from the command console. Both the Ondex Integrator tool and scripting environment for Ondex were developed to support the work in this thesis and more information about them may be found in chapter 2. For the analyses in this chapter, the Jython scripting interface was used in order to utilize methods from the NetworkX v0.99 graph-analysis library (Hagberg *et al.*, 2008b). Interactive visual exploration of the network used the visualization methods available in Ondex and exploited features which controlled settings such as the visibility, size/width and colour of nodes and the rendering of edges based on the numerical values of their attributes and/or group membership. The methods from NetworkX library were also used to calculate the graph properties in **Table 5.2**.

5.4 RESULTS

5.4.1 Network properties

Coexpression, protein-protein interaction, sequence similarity and name co-occurrence data were integrated using the Ondex system as described in 3.2.2. To better understand how each of the four sources contributed relationships in the Ondex graph, the number of edges and cases were they co-occurred was counted. These counts were further categorized to distinguish where an information source was the only source (exclusive), or where it may have also been supported by other edges, (inclusive). Additionally, key structural features of the networks were compared.

Table 5.1 Number of edges in the graph with evidence from the four information sources after applying a threshold on the relevant strength of the relationships (as defined in the Methods section). Exclusive combination means that only this exact combination of evidence types is present. Inclusive means that at least these evidence types are present, but others may be there as well.

COE	LIT	PPI	SEQ	Exclusive combinations		Inclusive combinations	
				N	%	N	%
✓	✓	✓	✓	9	0.04	9	0.04
✓	✓	✓		34	0.14	43	0.17
✓	✓		✓	83	0.33	92	0.37
✓	✓			84	0.33	210	0.83
✓		✓	✓	17	0.07	26	0.10
✓		✓		63	0.25	123	0.49
✓			✓	15	0.60	260	1.03
✓				9093	36.12	9534	37.88
	✓	✓	✓	123	0.49	132	0.52
	✓	✓		482	1.91	648	2.57
	✓		✓	692	2.75	907	3.60
	✓			4441	17.64	5948	23.63
		✓	✓	240	0.95	389	1.55
		✓		3459	13.74	4427	17.59
			✓	6201	24.63	7516	29.86

The contributions from the four information sources to the edges in the network are shown in **Table 4.2**.

There were 2355 proteins in the integrated network in total. The edges introduced into the network that came exclusively from each evidence source were: coexpression (COE) 36%; co-occurrence of protein names (LIT) 18%; protein interaction (PPI) 14% and sequence similarity (SEQ) 25%. The intersection of all evidence types was also very small (0.04%). This suggests that in this case each of the evidence sources tended to introduce new links into the combined network rather than reinforce the relationships already found in other sources.

The global properties for the relationship networks constructed from the four constituent information sources and the combined network (ALL) are shown in **Table** 5.2. As expected, the combined network had fewer connected components, since evidence from the other data sources connected previously unconnected nodes. The size of the largest component was also larger than that of any of the constituent networks. The diameters of the largest connected component of the SEQ, LIT and combined network (ALL) was of similar size (9, 9 and 10 respectively) and smaller than the COE and PPI networks (15 and 18 respectively), suggesting more cohesive or dense graphs. The increased density and the larger size of main connected component indicate that the ALL network is likely to be much harder to optimally partition using a clustering

Table 5.2 A comparison of graph theoretic properties for the different evidence types.

Evidence Network Type	Transitivity	Number of connected components	Size of the largest connected component	Diameter of the largest connected component
LIT	0.223	15	981	9
COE	0.580	24	991	15
PPI	0.070	100	1882	18
SEQ	0.746	268	241	9
ALL	0.406	9	2330	10

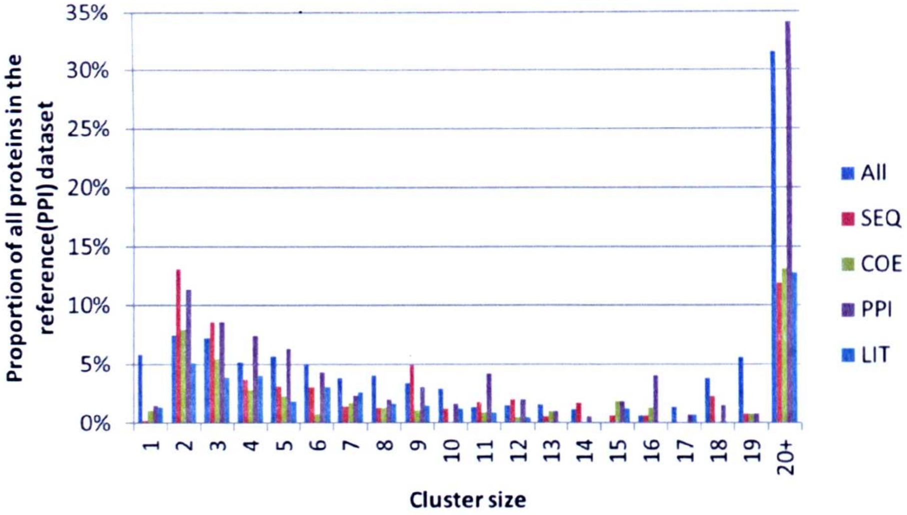


Figure 5.3 Cluster size distribution in five networks. The networks show are combined (ALL), protein-protein interaction (PPI), co-occurrence of protein names (LIT), sequence similarity (SEQ) and coexpression (COE).

approach.

The transitivity is a measure of clique-likeness of a graph. It was highest for the SEQ network (probably reflecting protein family structures) and the COE network, possibly reflecting shared transcriptional regulatory mechanisms.

Since the initial dataset was restricted to those proteins for which interaction data was available from *Arabidopsis* there were no unconnected proteins in the PPI and ALL networks. The number of orphan proteins (i.e. unconnected) for the SEQ, COE and LIT networks were 855, 1304 and 1343 respectively. The numbers of orphan proteins, however, depended on the score thresholds chosen, the values for which can be found in the section 5.3.

5.4.2 Network Clustering

The four single evidence networks and the combined network (ALL) were clustered according the protocol described in section 5.3.3. The distribution of cluster sizes is shown in **Figure 5.3**. The SEQ and PPI networks have a large number of clusters of size 2 and 3. The integrated network (ALL) and protein interaction network (PPI) contained the greatest number of larger clusters (size 20+). In the ALL network there were a large number of singletons (clusters of size 1). A total of 138 singletons accounted for 6.22% of all proteins in the network. This small proportion of singletons may be related to the cohesiveness of the ALL network, with tightly connected groupings leading to the exclusion of nodes by the MCL algorithm.

5.4.3 Coverage and specificity of the most representative function of modules

To explore the functional groupings of proteins in the network, *Arabidopsis* GO annotations from three sources: IntAct (Aranda *et al.*, 2010), GOA-EBI (Barrell *et al.*, 2009) and UniProt (UniProt Consortium, 2010) were combined with the relationship network. This was achieved by importing these data sources into Ondex and combining them with GO graph and relationship network using the accession-based mapping method (Taubert *et al.*, 2009) on GO term identifiers for the former and UniProt protein identifiers for the latter.

The utility of clustering depends on being able to group together a large enough number of proteins, so as to facilitate exploration of the modular structure of the network without diluting the information content of the clusters to such an extent that the groupings do not capture biologically meaningful relationships. In particular, this is determined by (i) whether the clusters contain proteins that are generally similar in terms of their functions, as assigned by Gene Ontology (the most representative GO terms in a cluster) and (ii) the way in which proteins with the same functional roles are distributed

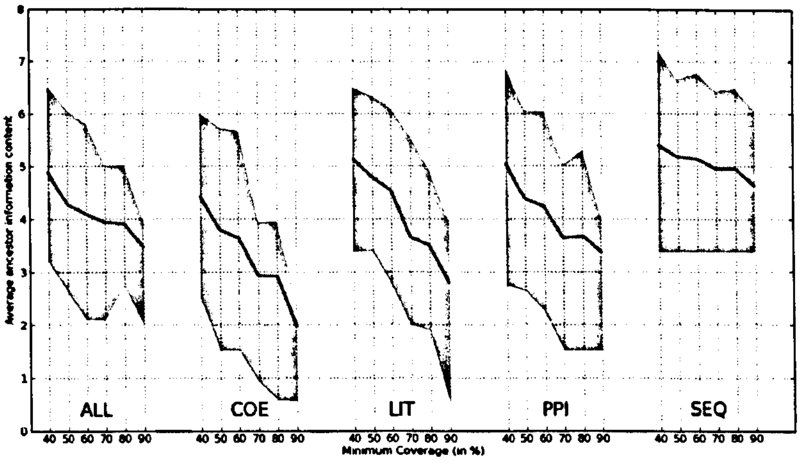
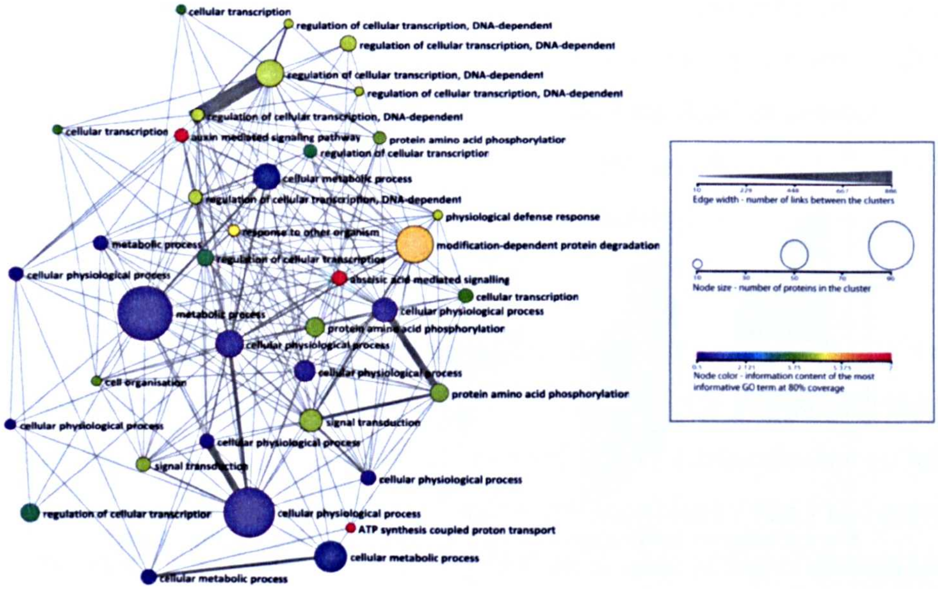


Figure 5.4 The Average Information Content of Most Informative Common Ancestor (AIC-MICA) across all clusters. AIC-MICA was calculated at 40-90% coverage levels. The solid line is the average IC and the shaded areas are 25 and 75 percent quartiles.

across different clusters (the fragmentation of GO terms)

The Average Information Content of the sets of these Most Informative Common Ancestor GO terms (AIC-MICA) was used to determine the coverage and the specificity of the most representative function of modules. If a cluster contained proteins that were of very diverse function, it would be expected that the GO categories corresponding to the most representative functions would not be very specific, i.e. the Most Informative Common Ancestor would be close to the root of the Ontology tree and thus would not represent a functionally meaningful grouping. As was explained earlier, the relationship network may not always reflect accurate functional relationships, and, therefore, there are likely to be some outliers present in the clusters. For this reason, rather than trying to identify a set of MICAs for all the proteins in the cluster, a sampling approach was used to find where term is applicable to at least a certain percentage of all proteins in a cluster. The analysis has been performed several times, with the minimum coverage parameter changed at 10% increments from 40% to 90%. This approach allowed simultaneous detection of functional similarities in more than one functional category and was more robust to outliers by design. The overall level of MICAs in the set



cellular transcription
regulation of cellular transcription, DNA-dependent
regulation of cellular transcription, DNA-dependent
regulation of cellular transcription, DNA-dependent
regulation of cellular transcription, DNA-dependent
regulation of cellular transcription, DNA-dependent
cellular transcription
auxin mediated signaling pathway
protein amino acid phosphorylatior
regulation of cellular transcriptior
cellular metabolic process
regulation of cellular transcription, DNA-dependent
physiological defense response
response to other organism
metabolic process
modification-dependent protein degradation
regulation of cellular transcriptior
cellular physiological process
abscisic acid mediated signalling
cellular transcription
metabolic process
cellular physiological process
protein amino acid phosphorylatior
cellular physiological process
cellular physiological process
cell organisation
protein amino acid phosphorylatior
cellular physiological process
signal transduction
cellular physiological process
signal transduction
cellular physiological process
regulation of cellular transcriptior
cellular physiological process
ATP synthesis coupled proton transport
cellular metabolic process
cellular metabolic process
10
229
448
667
886
Edge width - number of links between the clusters
10
30
50
70
90
Node size - number of proteins in the cluster
0.1
1.825
3.55
5.275
7
Node color - information content of the most informative GO term at 80% coverage

Figure 5.5 Modular structure of the combined network of all evidence types. The network nodes represent all clusters with 10 or more members. The width of the edges indicates the number of links between them. Clusters are annotated with the most informative GO term at 80% of the clusters proteins annotated with the GO term.

was calculated by averaging the IC of all the members of the set to produce the graph shown in **Figure 5.4**.

In **Figure 5.4** the AIC-MICA metric was plotted for the five relationship networks. As expected, the average information content of the representative GO terms decreases with the increase in cluster coverage. This implies that the common ancestor includes a greater proportion of proteins in the cluster. The average information content in the LIT network was similar to the ALL at lower coverage range (40%-50%), but declined very sharply and is second worst at the higher coverage level. This may be an indication that although useful associations can be found using term co-occurrence, these groupings tend to be less coherent at the whole-cluster level. Clusters in the COE network had the lowest information content of all coverage levels. The information content at coverage level of 90% was highest for the SEQ network followed by the ALL network. In the SEQ network, however, only 1496 proteins were assigned to clusters (of size greater than 1) whereas in the ALL network this figure was 2217. For proteins that cannot be assigned to a module, no inference can be made using the guilt-by-association principle. So, while for 5.9% of proteins, no new information could be gained from clustering the ALL network, whereas for the SEQ network this figure was 36.5%. Therefore, the ALL network had a much greater potential for suggesting biological context; supporting the hypothesis that the integration of multiple information sources can be useful when identifying functional modules.

5.4.4 Modules in the ALL relationship network and their most representative functions

Visual examination of complex network structure can be helpful for the identification of patterns. To facilitate the interactive analysis of functional annotation data, a method was developed to generate a meta-view of the modular structure as it is resolved by the clustering algorithm. In this view, each node represents a cluster and edges show the inter-links between them. This nodes and edges in this representation can be further annotated with additional properties, e.g. number of nodes in the cluster, degree of functional similarity, MICA, etc. **Figure 5.5** illustrates how this method of presenting data can be used to examine the modular structure found in the ALL network produced by application of the MCL algorithm. In this case, the clusters were annotated with the most informative of the representative GO terms at 80% level. It is possible to see that although the network was very densely interconnected, the clustering algorithm had performed reasonably well, with only a few cases where a very large number of links existed between separate clusters. One example of where the clustering was not optimal was where two clusters with the same annotation "regulation of cellular transcription, DNA-dependent" were linked together by more than 800 edges, but still were not joined together. In one of the cases, there were 6 of the 36 clusters with this same annotation where information content was in the middle of the range (4.0 – coloured green). Interestingly, this phenomenon was also seen in for clusters with other annotations relating to signalling and regulation of transcription. The two clusters with the most informative annotation were both related to hormone signalling (coloured red). There was also one large cluster annotated to "modification-dependant protein degradation", a similar cluster related to protein catabolism was also found in other studies that analysed PPI and coexpression networks (Bu *et al.*, 2003, Ulitsky and Shamir, 2007).

5.4.5 Fragmentation of functional categories

The other factor that needs to be taken into consideration when assessing the functional coherence of modules is fragmentation of functional categories. Fragmentation, and a loss of coherence arises because inevitably missing data and erroneous links will inevitably affect the performance of clustering algorithm, the correspondence of the current "perception" of how functional roles should be assigned to a group of proteins is also not guaranteed to perfectly correspond to the modules of a real biological system. Therefore, clustering can result in proteins with the same functional annotation being split across multiple clusters. This leads to the separation of this group of proteins into multiple fragments.

To assess the coherence of the clustering performed earlier, an analysis was undertaken to investigate how the Gene Ontology terms were distributed across the clusters. In **Table 5.3**, the Best Fragment Rank Proportion (BFRP) indicates that the GO terms are the least fragmented in the ALL network. This suggests that the combined network is better at grouping together identical GO terms, by

Table 5.3 The first two rows show the average entropy for the clustered networks and, for comparison, the average entropy for the networks with randomly permuted GO labels. The third row contains the decrease in entropy between the actual and randomly permuted networks. The fourth and fifth rows show the best fragment rank percentage and best entropy rank percentage statistics (defined in the Methods section). Note that percentages may not add up 100%, because when several networks performed equally well in the BERP/BFRP assessment, all of them were counted as "best" for that GO term.

	ALL	SEQ	COE	PPI	LIT
Average entropy (actual network)	2.72	2.96	3.30	2.86	2.96
Average entropy (randomly permuted network)	3.31	3.43	3.42	3.29	3.33
Relative decrease in entropy (compared to randomly permuted network)	17.8%	13.7%	3.5%	13.1%	11.1%
BFRP	49.43%	22.78%	3.55%	24.56%	28.43%
BERP	39.58%	16.64%	2.75%	18.58%	31.18%

comparison with the individual networks. To evaluate the level of fragmentation of functional categories, both the number of fragments and their size distribution need to be considered. The entropy of the fragmentation gives a measure of this size distribution. As can be seen in **Table 5.3**, the Best Entropy Rank Proportion (BERP) is also maximal for the ALL network, followed by the LIT network, indicating that overall the entropy with respect to GO categorisation was the lowest for these networks.

A lower entropy value implies more ordered data, both in terms of reduced fragmentation and prevalence of larger fragments. To provide a comparison with the level of entropy that could be expected by chance, "control" networks were generated by randomly permuting the cluster labels for all GO categories 10000 times. The complete results of this test are included in Appendix E – in all five cases none of the random networks were able to achieve comparable entropy value, indicating that this result is highly significant, with $p<0.00001$. To avoid the problems of small sample sizes, only those GO categories that were assigned to at least 10 proteins in the dataset were included. **Table 5.3** shows the average entropy values for each network. From this it can be seen that the ALL network has the lowest average entropy, again suggesting that it is better at grouping related proteins, the average entropy being 2.72 compared with 3.31 for the equivalent "control" network.

5.4.6 An example of fragmentation in the ALL relationship network

Figure 5.6 (A) shows all proteins (nodes) in the combined (ALL) network annotated to the high level GO term "response to hormone stimulus" and its more specialised categories (grey clusters). The average shortest path length (SPL) between all proteins with this annotation was 20% shorter compared to a control, where node labels were permuted 10000 times. The SPL reduction in distance for the child terms listed in **Figure 5.6** was even greater and ranged from 22-30%. It is interesting to note that the structure of the distribution of the proteins with these annotations echoes the hierarchy of the Gene Ontology, which was defined entirely independently by manual curation.

Figure 5.6 (B) shows the fragmentation of this cluster by visually separating all the MCL clusters across which this term is distributed. It is evident that the clustering was not able to group together all the nodes that were associated with the general process 'response to hormone stimulus". In this case, there were only two clusters which had more proteins in the cluster annotated with the same term (e.g. 'response to auxin stimulus' and 'response to abscisic acid stimulus'). However, even in the situations when the grouping is suboptimal, it is still useful to be able to determine and quantify how much the grouping differs from the one specified by annotations and structure of the Gene Ontology.

This analysis has shown that both the AIC-MICA, and BERP/BFRP types of metrics can be used to evaluate the impact made by choosing different clustering methods, data sources or GO aspect on the functional coherence of the modules. However, it is also evident that there may be more complex multi-level structural features present in the integrated functional networks, which may be difficult to detect using clustering approaches along. As was demonstrated by this example, interactive visual exploration of the network can be a useful tool for discovering such features, and provide useful insights for development of more rigorous computational approaches for better understanding of integrated networks.

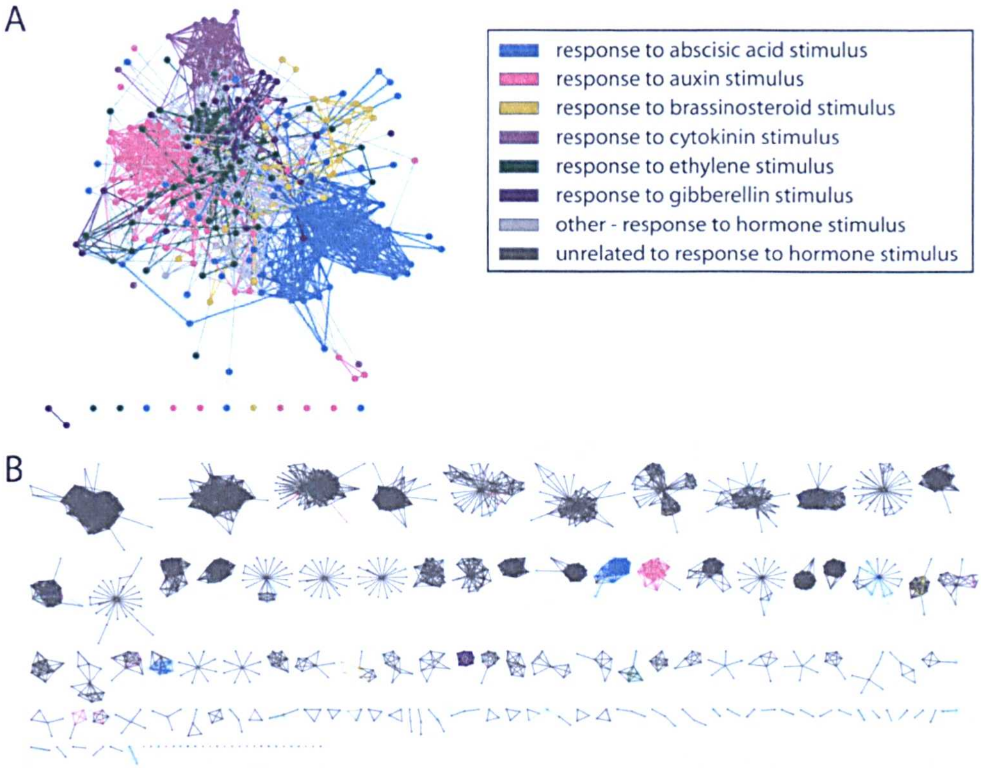

A
response to abscisic acid stimulus
response to auxin stimulus
response to brassinosteroid stimulus
response to cytokinin stimulus
response to ethylene stimulus
response to gibberellin stimulus
other - response to hormone stimulus
unrelated to response to hormone stimulus
B

Figure 5.6 A subnetwork from the combined (ALL) network of proteins annotated with the GO term "response to hormone stimulus". The diagram shows (A) the proteins annotated to this GO term and direct links between them and (B) the breakdown of this group of proteins into clusters. The colouring is consistent between the two panels. Proteins that are not annotated to this process are hidden on panel (A) and are coloured grey in panel (B). In (B) all clusters shown contain at least one member with "response to hormone stimulus" annotation and the only edges shown are the ones that link two members of the same cluster.

5.5 DISCUSSION

The aims of this research have been to explore the effect of using multiple sources of biological information about *Arabidopsis thaliana* proteins and, in particular, to assess whether integrating multiple evidence sources in a relationship network has potential benefits for applications such as detection of functionally coherent sets of proteins in relationship networks.

In this chapter the functional coherence of modules detected by clustering relationship networks was assessed from two different perspectives. The first considered the representative functions of the modules with respect to GO terms and the second was an analysis of the fragmentation of GO terms with respect to the proteins contained in the modules. The motivation behind the former approach was to investigate the trade-off between coverage and specificity of the representative function of modules. This was achieved by defining a novel metric (AIC-MICA). Additionally, two metrics describing the fragmentation of GO categories, namely BFRP and BERP, were introduced to evaluate how well the modular structure, recovered by the MCL algorithm, maps to GO Biological process terms. These metrics were then used to compare the usefulness of individual data sources and to test the effects of combining multiple sources on the coherence of these modules.

From the analysis of the trade-offs between coverage and specificity, the SEQ network was, as expected, best for recovering very specific functional association between proteins. This was evident from the high AIC-MICA values across all coverage levels. However, an important point to note is that it may not always be desirable to extract such close groupings, and a higher level GO categorisation may be helpful to provide a broader overview of biological functional class or to help dissect very large datasets. Compared to the other relationship networks, SEQ consisted of a large number of strongly connected components (results not shown), which resulted in the relatively high overall entropy with respect to the whole of the Gene Ontology. We also observed that the clusters recovered were only related to a small number of GO terms.

Another problem with sequence relations as the sole data source was that there was insufficient evidence to link most of the proteins in our reference set. By comparison with the SEQ network, it was possible to use the ALL network to assign a further 721 proteins to a cluster of size greater than one due to links that were contributed by other sources. Based on these findings, we conclude that, overall there is a clear benefit from the integration of additional data sources, although there is a small cost incurred because of a reduction in functional coherence. As the ALL network performed relatively well in terms of AIC-MICA (40-90), this dilution of annotation specificity does not appear to render it uninformative. In fact, the minimum information content value that was applicable at a 40% coverage level was 0.55 and was reached only for 5 clusters found in the ALL network. This value corresponds to the 'cellular physiological process' GO term, which is one of the direct descendants of the 'biological process' root term, and is therefore very general.

To support this work, several different visualisation strategies were developed that help to summarise complex integrated networks and identify high-level patterns in them. Using these visualisation methods, it was found that there was a hierarchically organised neighbourhood in the integrated network that was composed of the proteins annotated to the "response to hormone stimulus" GO term. This finding indicates there may be more complex and meaningful patterns than just the modules identified using clustering approaches.

Comparison of the graph theoretic properties of the four networks also appears to indicate that the addition of extra edges lead to the creation of a more compact network, with smaller diameter than the COE or PPI networks. Despite this, the transitivity has remained relatively low – indicating that the number of complete cliques was also small. These differences may be interpreted as an indication that, in the ALL network, potential modules are more difficult to recover and the results may be further improved using more robust clustering approaches, like spectral clustering methods (Ng *et al.*, 2002). Further investigation of the impact of increasing complexity of the network versus increasing levels of noise that arise from integration of additional data sources is necessary to confirm these trends.

The coexpression (COE) network performed the worst with respect to BFRP, BERP and AIC-MICA. At first glance, this result appears to contradict several earlier studies (Mao *et al.*, 2009, Mentzen and Wurtele, 2008), where many meaningful clusters were identified in the coexpression network. This discrepancy, however, is likely to be an artefact of the smaller subset of the proteome that was used in this study; a consequence of the decision to restrict the dataset to proteins with PPI information. In earlier reports, using large coexpression networks the patterns detected tended to be associated with much larger clusters containing more than a 1000 proteins (Mao *et al.*, 2009, Mentzen and Wurtele, 2008). This number is much larger than any of the clusters that were identified in any of the networks constructed in this study. This may be an indication that coexpression is a weaker source of evidence of functional similarity and more data are necessary in order to be able to make useful inferences from it.

In this study, the set of proteins in the network was restricted to those for which protein-protein-interaction information is available, as this is a currently limiting information source for *Arabidopsis*. Using a larger set of proteins would have meant that the contribution of the PPI data would have been highly unbalanced in relation to other available information. Although there are other species, in particular *Saccharomyces cerevisiae*, for which there is much more data available, it is also of importance to validate these types of approaches in more complex multicellular model organisms. This was particularly important within the context of this project, as *Arabidopsis thaliana* was the species of primary interest and therefore it was important to get some understanding about data available for it. Most importantly, this study illustrated that meaningful modules can be successfully identified by clustering the integrated relationship networks -- even in situations when limited data are available and only part of the complete proteome is considered.

5.6 CONCLUSIONS

Module detection in integrated biological and relationship networks is one of the most important tools for interpretation of complex biological datasets. As the amount of biological information continues to grow, it also becomes increasingly important to improve our understanding of inter-relationships within these data and, ultimately, their relationship to biological function. In this chapter these relationships were explored and quantified for several of the data types that are most commonly used for construction of such networks. It was found that for these datasets combining several types of evidence was beneficial with respect to the functional annotation of modules detected using MCL clustering algorithm, which on average more closely corresponded to the functional groupings in the Biological Process aspect of GO. Although the overall level of informativeness of cluster annotation was not as good as in the sequence similarity network, it was possible to link many more proteins using additional information sources. These findings indicate that there is benefit to the integration of additional information sources, as it allows more proteins to be assigned to functional modules with only a relatively small reduction in the module annotation precision. The overall outcomes of this study provide a number of insights into the relationship between integrated networks and protein function and may be of use for further refinement of related approaches that can better capture biologically relevant information from integrated datasets. A number of methods developed for the work described in this chapter were also used to assign function to clusters in coexpression (chapter 4) and protein-protein interaction (chapter 5) networks.

6 APPLICATION OF INTEGRATED NETWORKS FOR INTERPRETATION OF EXPERIMENTALLY DERIVED GENE LISTS

6.1 SUMMARY

Nitrogen uptake and metabolism in plants has an impact on a large number of genes and processes within the plant. Because of the complexity involved, better understanding of the regulatory complexity giving rise to these effects requires integration of multiple types of different data. To explore the mechanisms behind the regulation of nitrogen responses in *Arabidopsis*, several datasets already introduced earlier in this thesis were combined with a custom-generated coexpression network of nitrogen-related responses created using a coexpression analysis pipeline presented in chapter 3. This network was also combined with GO functional annotation and lists of differentially expressed genes from a study that looked at the differences between the wild-type *Arabidopsis* plant and a mutant that lacked an ability to target a low affinity nitrate transporter to the outer membrane. This chapter reports how the integrated datasets, analysis methods and visualisation tools developed as part of this work can be used for the interactive exploratory analysis aimed at greater understanding of the structure behind a particular list of candidate genes. The example also illustrates how his type of approach can be leveraged for the narrowing of the hypothesis space and the identification of potential candidate genes for further study.

6.2 INTRODUCTION

6.2.1 Nitrate uptake, assimilation and downstream responses in *Arabidopsis thaliana*

Nitrogen is an element with the periodic number 7 and atomic mass 15 (Moore and Gallagher, 1993), and it constitutes 78% of Earth's atmosphere (Lutgens and Tarbuck, 1986). It is also a mineral nutrient that is required by all plants in great quantities, as it is needed for the synthesis of essential cellular compounds like proteins and nucleic acids (Miller and Cramer, 2005). The most abundant form of nitrogen (N_2 gas in the atmosphere) can be directly utilised by legume plants in a symbiotic association with bacteria, but for most plants, nitrogen is taken up by the roots in the forms of nitrate (NO_3^-), ammonium (NH_4^+) and, less commonly, other organic forms (peptides, amino acids, urea). These compounds then form the overall pool of soil nitrogen available to plants (Marschner, 1986; Miller and Cramer, 2005). The concentration of available nitrogen in the soil can vary by several orders of magnitude, however only very extreme concentrations have negative impact on plant growth (Britto and Kronzucker, 2006). This resilience is the result of a well-coordinated system of responses on all levels of organisation – from cellular metabolism to plant physiology and development, which aim to maintain the nitrogen content of plant tissues at the optimum levels. Precise regulation of nitrogen uptake is essential to maintain this optimum. However, it is only one part of the complex regulatory processes involved. For better understanding of these processes they must be considered in a wider context, which includes some aspects of nitrogen assimilation, storage and translocation within the plant. This section aims to provide an overview about what is currently known about nitrogen uptake and related regulatory mechanisms in *Arabidopsis* in order to put the outcomes of the subsequent analysis into an appropriate biological context.

6.2.1.1 Transporters involved in primary uptake

Physiological studies of whole-root nitrate uptake have identified that there are two kinetically distinct nitrate uptake systems in *Arabidopsis* (Orsel *et al.*, 2006). The high-affinity transport system operates according to Michaelis-Menten saturable kinetics and is of primary importance at lower nitrate concentrations in the soil (<500μM (Orsel *et al.*, 2002)); whereas the low-affinity system is non-saturable and can allow effective uptake of nitrate at >1mM concentrations (Britto and Kronzucker, 2006). The high-affinity transport system has inducible (iHATS) and constitutively expressed (cHATS) groups of transporters. The expression of inducible transporters responds positively to the increase of nitrate availability. cHATS are expressed even when nitrate is not supplied to the plant (Miller and Cramer, 2005).

The HATS transporters in *Arabidopsis* roots were identified as AtNRT2.1 and AtNRT2.2 (Cerezo *et al.*, 2001, Okamoto *et al.*, 2003, Orsel *et al.*, 2002). A mutant lacking AtNRT2.1 and deficient in AtNRT2.2 was demonstrated to have limited HATS activity (Cerezo *et al.*, 2001). The AtNRT1.1 transporter was also shown to be capable of operating in both high-affinity and low-affinity modes, depending on its phosphorylation status (Liu *et al.*, 1999, Liu and Tsay, 2003) and it is believed to be a contributor to both cHATS and low-affinity transport system (LATS) activity (Crawford and Forde, 2002).

The plasma-membrane-targeting protein AtNAR2 was identified as an important interaction partner for AtNRT2.1 transporter (Orsel *et al.*, 2006). Comparison of the HATS activity of *atnar2.1* and *atnrt2.1* mutants has indicated that AtNAR2.1 was also required for the normal function of AtNRT2.2, as the *atnar2.1* plants appeared to be more impaired in nitrate uptake. Identification of *atnar2.1* mutant allows further exploration of the regulation of high affinity uptake system in *Arabidopsis* and the parts played by individual transporters.

Although several LATS transporters were also identified, further experiments have revealed that the known transporters do not account for all of the LATS activity (Miller *et al.*, 2007). Two *Arabidopsis* transporters known to take up nitrate in the low-affinity range are part of an NRT1 protein family with 53 different members, which are also involved in transport of amino acids and peptides (Williams and Miller, 2001). The first LATS transporter to be characterised in *Arabidopsis* was AtNRT1.1 (also known as CHL1) (Tsay *et al.*, 1993). However, the low-affinity transport in the AtNRT1.1-knockout was not affected when plants were supplied with nitrate as a sole nitrogen source and the reduction in LATS uptake only became evident when the plant was also supplied with ammonium (Touraine and Glass, 1997). Another transporter found to be important for low-affinity uptake was AtNRT1.2, but in this case the AtNRT1.2 antisense mutant was found to retain a disproportionately greater level of LATS activity compared to the reduction in the AtNRT1.2 expression (Huang *et al.*, 1999). One possible interpretation of these results is that there are more nitrate transporters that operate in the low-affinity range that have not yet been identified (Miller, 2010).

Nitrate homeostasis within the cell is maintained through the balance of uptake and removal processes and nitrate efflux from the root is an important component of this balancing equation. Under normal conditions, nitrate efflux

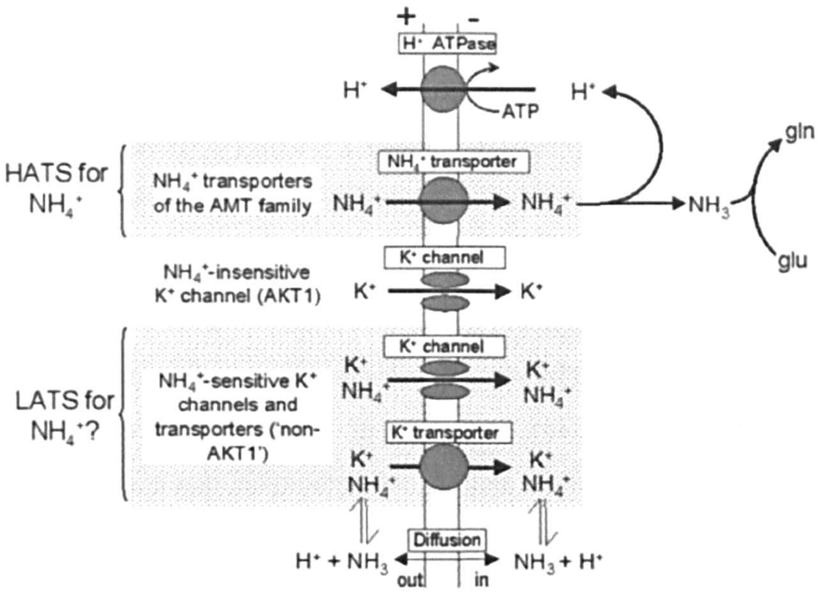


Figure 6.1 Ammonium uptake overview. Summary of possible ways ammonium can enter a plant cell. Image from (Crawford and Forde, 2002).

does not exceed nitrate uptake, although it can be quite substantial (Kronzucker *et al.*, 1999). At present, one nitrate efflux transporter has been identified in *Arabidopsis* – NAXT1, another member of the NRT1 family of transporters (Segonzac *et al.*, 2007).

It is believed that the ammonium ion rather than neutral ammonia is the predominant form taken up by higher plants, including *Arabidopsis* (von Wiren *et al.*, 2000). From the physiological point of view, ammonium uptake also has biphasic kinetics, with high- and low-affinity components. Two transporters believed to be of particular importance in the high-affinity range are AtAMT1.1 and AtAMT1.3 (Rawat *et al.*, 1999, Gazzarrini *et al.*, 1999).

At higher external concentrations, ammonium can enter the plant cell through a number of non-specific cation transporters (**Figure 6.1**). In particular, the potassium ion is very similar in size and charge to ammonium and it has also been demonstrated that non-specific cation transporters may transport ammonium (Howitt and Udvardi, 2000). In its non-charged form, as ammonia, it may also diffuse directly through the cell membrane or enter through aquaporins. Shelden, *et al.* (2001) have shown that the AtAMT1.2 transporter can function in both high-affinity and low-affinity modes (Crawford and Forde, 2002). As was already introduced in the Section 1.4.1.4, ammonium is also produced by the reduction of nitrate in the cell. This introduces a complicated trade-off into the control of uptake and assimilation process – on one hand, taking up ammonium directly is more energy-efficient because two reactions to convert nitrate can be skipped (Britto and Kronzucker, 2002). On the other hand, if allowed to accumulate, ammonium can become toxic to the cell (Britto and Kronzucker, 2002).

6.2.1.2 Nitrogen transport within the cell

The excess nitrate taken up by a plant cell is stored in the vacuole, from where it can be released if the external supply is interrupted. As nitrate is negatively charged, its movement into the vacuole is thermodynamically unfavourable, so the process requires active transport. This transport is mediated by an AtCLCa hydrogen/nitrate antiporter (De Angeli *et al.*, 2006, De Angeli *et al.*, 2009). The importance of this transporter for vacuolar storage of nitrate is supported by multiple pieces of evidence, reviewed in (Zifarelli and Pusch, 2010). In seeds, AtNRT2.7 is also important for the loading of NO_3^- into the vacuole (Chopin *et al.*, 2007). AtCLCa was also shown to localise to the tonoplast (De Angeli *et al.*, 2006), as well as three other members of the CLC family - AtCLCb, AtCLCc and AtCLCg (Lurin *et al.*, 1996). AtCLCc was identified to be important for the nitrate accumulation in QTL/mutation characterisation analysis (Harada *et al.*, 2004). The expression of the AtCLCa and AtCLCc is known to be regulated by nitrate, with the former one stimulated and the latter being repressed (Geelen *et al.*, 2000).

The process of ammonium storage is believed to be largely passive - it is predicted that at cytosolic pH some of the ammonium (~3%) will exist in the form of ammonia that can enter the vacuole by diffusing through the membrane directly or via aquaporins (Martinoia *et al.*, 2007). As the pH inside the vacuole is much lower, the ammonium will become protonated and this will prevent it from exiting the vacuole. This mechanism is known as "acid trapping". When the cytosolic ammonium is depleted, for example, due to the assimilation via glutamine synthetases (GS), a new chemical gradient for ammonia is established that favours its movement out of the vacuole.

6.2.1.3 Nitrogen translocation within the plant

There is some evidence which indicates that the cytoplasm of the plant cell is maintained in homeostasis with respect to both nitrate and ammonium (Martinoia *et al.*, 2007, Miller and Smith, 2008), although the concentrations can vary considerably, even between the adjacent tissues. Cytoplasmic homeostasis is maintained both through the regulation of uptake and assimilation processes and through extrusion back into the growth media or translocation to the xylem or vacuole.

Current research indicates that both nitrate and ammonium can be present in the xylem sap at relatively high concentrations. The concentrations of 10-37mM for nitrate and 0.4-5mM for ammonium were reported for various plants grown in conditions with sufficient supply of respective nitrogen source (Miller and Cramer, 2005). It was shown by Lin *et al.* (2008) that one of the transporters involved in xylem loading in the root is AtNRT1.5. However, the same study also found that this transporter does not completely account for all nitrate loaded into the xylem. AtNRT1.8 is involved in the reverse process - the removal of nitrate from the xylem sap into parenchyma cells (Demir *et al.*, 2010). Another member of the NRT1 family, AtNRT1.7 is involved in loading the nitrate into the phloem in the source leaves (Fan *et al.*, 2009). Some of the other members of the NRT1 family have been shown to be involved in the more tissue/organ-specific nitrate transport: AtNRT1.4 in leaf petiole (Chiu *et al.*, 2004), AtNRT1.3 in leaves (Okamoto *et al.*, 2003) and AtNRT1.6 in nitrate loading into seeds (Almagro *et al.*, 2008). The mechanism of ammonium entry into the xylem is presently unknown (Miller and Cramer, 2005).

6.2.1.4 *Nitrogen assimilation*

The primary nitrogen assimilation processes are mediated by four enzymes (Miller and Cramer, 2005). The first enzyme in nitrate assimilation is nitrate reductase (NR), which catalyses nitrate conversion to nitrite. In *Arabidopsis* this process is mediated by an NADH-dependant NR (Wilkinson and Crawford, 1993). NR expression levels show diurnal rhythms and are up-regulated by high intracellular nitrate. Post-transcriptionally, the enzyme can be reversibly inactivated by phosphorylation triggered by low cytoplasmic pH and anoxia (Campbell, 1999, Lillo *et al.*, 2004). Nitrite reductase (NiR) converts nitrite to ammonium. Its expression is positively regulated by nitrate and glucose, whereas ammonium has been shown to post-transcriptionally down-regulated it (Faure *et al.*, 1991, Crete *et al.*, 1997).

The pool of available ammonium is used for the synthesis of amino acids (**Figure 6.2**). This process commences with the two reactions that form a cycle: first ammonium is combined with L-glutamate to give glutamine, which is catalysed by the glutamine synthetases (GS). Then one amino group from the L-glutamine is transferred to α-ketoglutarate to form two molecules of L-glutamate (catalysed by the NADH-dependent glutamate synthases (NADH-GOGAT)). The reverse reaction, which produces the 2-oxoglutarate and ammonium from glutamate, is catalysed by the glutamate dehydrogenase (GDH) (Coruzzi, 2003).

Arabidopsis has four paralogues of GS with different kinetic properties and regulation mechanisms (Ishiyama *et al.*, 2004). Expression levels of all GS paralogues are induced by glucose and suppressed by glutamine and ammonium, apart from GS1.2, which is induced by ammonium (Oliveira and Coruzzi, 1999, Miflin and Habash, 2002, Ishiyama *et al.*, 2004).

Glutamine and glutamate are used for the synthesis of all other amino acids. The next ones in the chain are the aspartate, which is synthesised from the glutamate by the aspartate aminotransferase (AspAT) and asparagine, produced from aspartate by the asparagine synthase (ASP) (Coruzzi, 2003). The latter

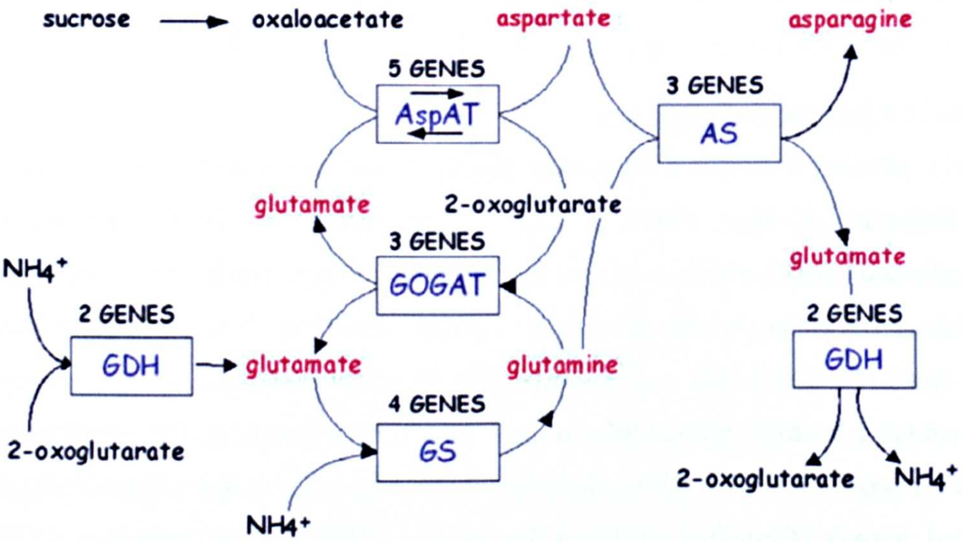


Figure 6.2 Key reactions, metabolites and enzymes of amino acid synthesis in Arabidopsis. From Coruzzi (2003).

reaction also results in the production of glutamate. It was suggested that amino acids may be important for the overall plant N status (Zhang *et al.*, 1999, Cooper and Clarkson, 1989), although as the inter-conversion between these amino acids is possible (Coruzzi, 2003), the exact identity of the ones that are crucial to this process is still unknown.

6.2.2 Regulatory effects

Nitrogen uptake is regulated to match the carbon status of the plant as well as its demand for nitrogen itself (Coruzzi and Zhou, 2001). It is believed that the N status of the whole plant plays an important part in this process via the xylem/phloem cycling of amino acids (Crawford and Forde, 2002). The intermediates of the nitrogen assimilation pathway (nitrite and ammonium) are toxic if allowed to accumulate, and one of the main ways of limiting their accumulation is to drive forward amino acid synthesis. This process requires carbon skeletons in the form of 2-oxoglutarate and reducing agents made during respiration, which, in turn, is reliant on the sugars produced by photosynthesis. This necessitates a close link between carbon and nitrogen metabolisms and the coupling of nitrogen uptake processes to the carbon status of the plant. As is summarised in **Figure 6.3**, nitrogen metabolites are believed to be important signals controlling this regulatory system (Jackson *et al.*, 2008).

An additional store of nitrate and ammonium is also maintained in the vacuole (Martinoia *et al.*, 2007). These reserves are believed to play an important role in maintaining the homeostatic concentration of these ions in the cytoplasm. The changes in homeostasis of nitrate have been suggested to play a role in the regulation of both nitrogen uptake and assimilation (Miller and Smith, 2008). High-affinity transporters for nitrate and ammonium are also subject to diurnal regulation, possibly induced by glucose (Glass *et al.*, 2002, Miller *et al.*, 2007).

Lejay *et al.* (1999) has shown that in the short-term, AtNRT2.1 expression is sensitive to the decrease of extracellular nitrate concentration (first 48 hours). This pattern of expression was explained as the combined result of two regulatory mechanisms: the repression by the products of downstream N assimilation and the stimulation by NO_3^- itself. As the levels of N metabolites fell during the first two days of starvation, the repression of AtNRT2.1 was lifted, but then the withdrawal of the stimulatory effects of nitrate became more significant (Lejay *et al.*, 1999). This interpretation is supported by the observation that the NR-deficient mutant of *Arabidopsis* also had a higher than wild-type level of AtNRT2.1 expression. AtNRT2.2 transcription is believed to be stimulated by high NO_3^- supply, followed by subsequent down-regulation, but in the case of AtNRT2.2, down regulation occurred much earlier (first 12 hours) (Lejay *et al.*, 1999).

The regulation of AtNRT1.1 was shown to be insensitive to the whole-tissue amount of N-metabolites (Lejay *et al.*, 1999). It was found that its expression level was greater in the NR-deficient mutants; however the exact nature of the regulatory mechanism responsible is still not clear. In the same study it was also demonstrated that both AtNRT1.1 and AtNRT2.1 transporters are up-regulated by sugars and light and that the low-affinity transport system is less sensitive to the stimulatory effects of sugars in the absence of light. Later AtNRT2.1 was found to have an important role in the regulation of nitrogen uptake by carbon status, as nitrate uptake in the AtNRT2.1-knockouts was no longer regulated by the carbon metabolites (Lejay *et al.*, 2003).

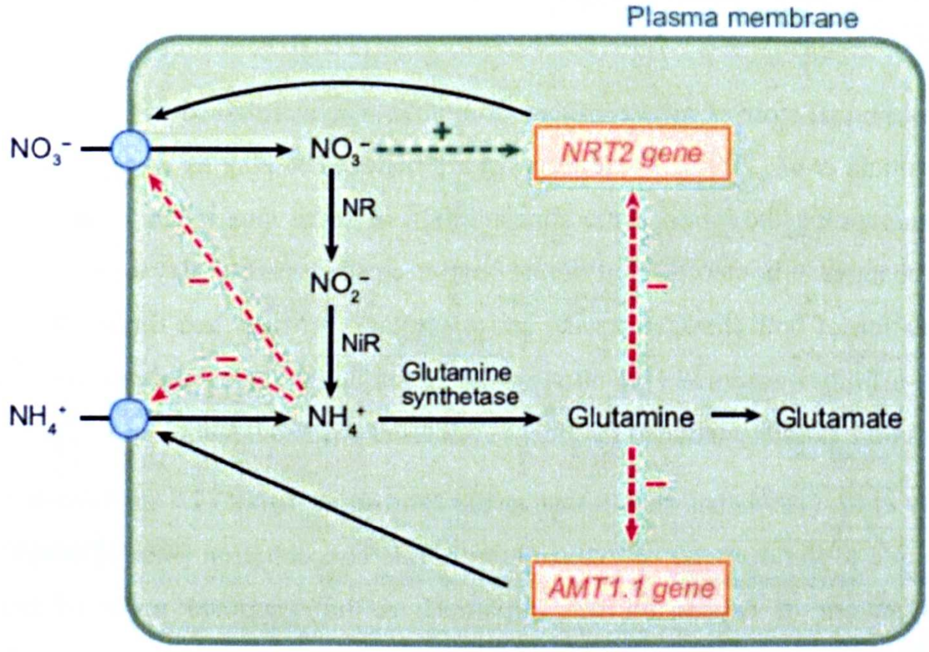


Figure 6.3 High-level overview of the effects of nitrogen metabolites on its uptake and assimilation processes. Image source: Jackson, et al (2008).

Exogenous application of NH_4^+ and amino acids was reported to result in down-regulation of AtNRT2.1 transcription; however it is difficult to establish the exact nature of the metabolite responsible for this effect. There is a considerable interchange among the different pools of nitrogen within the plant. In particular, amino acids can be rapidly inter-converted and ammonium can also be produced during normal metabolic processes (Miller *et al.*, 2007). All three ammonium transporters known to be important for the ammonium uptake in the roots (AtAMT1.1, AtAMT1.3 and AtAMT2.1) are up-regulated by sucrose supply and low nitrogen status (Yuan *et al.*, 2007).

As evident from the account above, there are now a number of high-level observations about the responses of the various components of the nitrogen uptake and assimilation system to different stimuli like extra- and intra-cellular ammonium, nitrate, amino acids, sucrose, light and carbon and nitrogen starvation. However, the identities of the system components of responsible for these effects at the gene and protein level still remains largely unknown. The overall picture about these important processes is still only available as a collection of largely disjoint pieces.

6.2.2.1 Sensors and regulatory pathways

This section reviews some of the key components of nitrogen-related regulatory processes that have been identified. Part of the problem is that to a large extent the plant nitrogen response and regulation systems have been found to have little homology with genes from other species(Vidal and Gutierrez, 2008). However some homology has been found in a few cases and using this information, some of the important components have been discovered by exploring these similarities (Moorhead and Smith, 2003, Lam *et al.*, 1998).

In particular, there are indications that several systems responsible for sensing nitrogen status in bacteria and animals have counterparts in *Arabidopsis*. It was proposed by Moorhead and Smith (2003) that plants may have a PII system for detecting glutamine concentration, similar to that found in bacteria – where the PII signalling protein is phosphorylated in response to changes in glutamine and 2-oxaglutarate concentration, triggering the downstream regulatory processes (Miller *et al.*, 2008). Previously, this protein was suggested to act as a nitrogen sensor in *Arabidopsis* (Hsieh *et al.*, 1998), proven to interact with N-acetyl glutamate kinase, an enzyme involved in arginine synthesis (Chen *et al.*, 2006b) and shown to be important for the control of argenine biosynthesis in a study by Ferrario-Mery *et al.* (2006). PII over-expression was linked to the reduced sensitivity to glutamine (Miller *et al.*, 2008). A recent paper suggests that NO_2 transport into the chloroplast is enhanced in PII-mutant plants (Ferrario-Mery *et al.*, 2008).

AtNRT1.1 has long been suspected to have a regulatory role – in particular, in the control of the root growth responses (Walch-Liu *et al.*, 2006, Walch-Liu and Forde, 2008) and the expression of nitrate-responsive genes (Wang *et al.*, 2009), including another nitrate transporter AtNRT2.1 (Munos *et al.*, 2004). In a Ho *et al.* (2009) study, the AtNRT1.1 was identified as the first transporter with a receptor functionality known in plants. From the insights in this and related works it became apparent that a change in the phosphorylation status of this transporter is important not only for the switch between low- and high-

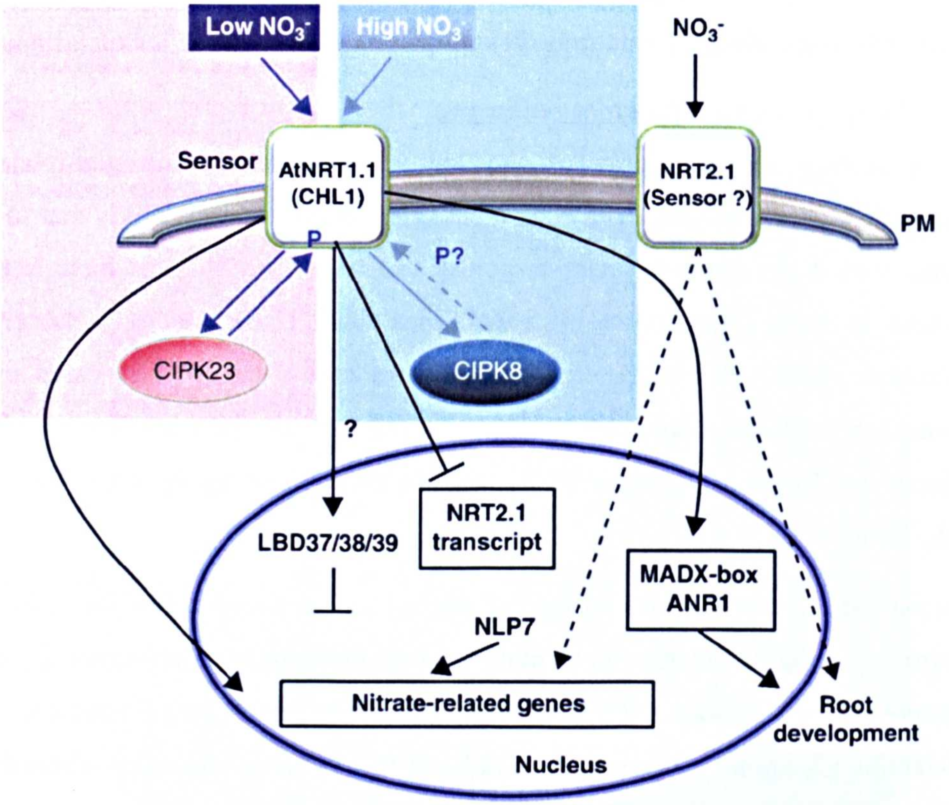


Figure 6.4 Consolidated overview of known nitrogen-related regulatory processes. Image from Ho and Tsay (2010).

affinity transport modes, but also for its role as a nitrate sensor. At the moment two of the kinases modulating this activity have been identified. One is the CIPK23, which is a negative regulator for the signalling pathways of the initial responses to high nitrate availability initiated by the AtNRT1.1 (Ho *et al.*, 2009). This kinase also mediates the switch between low- and high-affinity nitrate uptake modes (Ho *et al.*, 2009). Another one is the CIPK8, which is a positive regulator of the initial responses to low nitrate availability (Hu *et al.*, 2009). The identity of several transcription factors involved in these primary responses to nitrogen is also known – LBD27/38/39 (Rubin *et al.*, 2009) and NLP7 (Castaings *et al.*, 2009) are the negative and positive regulators of nitrate-related genes respectively. The genes known to be important for the initial response to changes in nitrate availability are reviewed in a recent publication by Ho and Tsay (2010), an overview from which is included here as **Figure 6.4**.

At present, only one gene has been identified as being involved in longer-term, adaptive responses to changes in nitrogen supply. This gene is NLA, which is a RING-type ubiquitin ligase (Peng *et al.*, 2007). Plants with a mutation in this gene did not initiate any adaptive responses found in wild-type plants, when exposed to nitrogen-limited conditions (Peng *et al.*, 2007).

The ANR1 transcription factor, a regulator of nitrate-responsive lateral root growth in *Arabidopsis* (Zhang and Forde, 1998). Experimental evidence from mutants suggests that this transcription factor and putative gene AXR4 both play a role in this pathway (Crawford and Forde, 2002, Walch-Liu *et al.*, 2006). Experiments with the *chl1-5* mutant, which has a defective AtNRT1.1 transporter, provided evidence that AtNRT1.1 itself is also important for nitrate-induced lateral root elongation and may be the origin of the signal transmitted by these other proteins (Walch-Liu and Forde, 2008, Walch-Liu *et al.*, 2006).

As reviewed in (Zhang and Forde, 2000), a combination of local and systemic nitrate supply and N status of the plant have been demonstrated to have profound effects on the development of root system architecture. High overall nitrate availability has an inhibitory effect on lateral root elongation (Signora *et al.*, 2001). This inhibition is abscisic acid-dependent and two transcription factors (ABI4 and ABI5) are important for this regulatory pathway (Signora *et al.*, 2001). This systemic effect appears to be different from the localised effects of nitrogen (described above), which stimulates lateral root growth. Another protein identified to play a role in root architecture development is ARF8, which is important for nitrate-controlled lateral root emergence (Gifford *et al.*, 2008).

Research by Remans *et al.* (2006) and Little *et al.* (2005) on the responses of the root system to nitrate availability indicates that AtNRT2.1 also plays a part in this process. The initiation of later root primordia in the *atnrt2.1* mutant is inhibited at low nitrate availability, but also shows reduced repression by high C:N ratio. Both of these responses are different from the wild-type and were shown to be independent of AtNRT2.1 function as a nitrate transporter. The *atnar2.1* mutant also has a distinct phenotype with respect to lateral root development manifested as an enhanced growth rate 4-5 days after the transfer

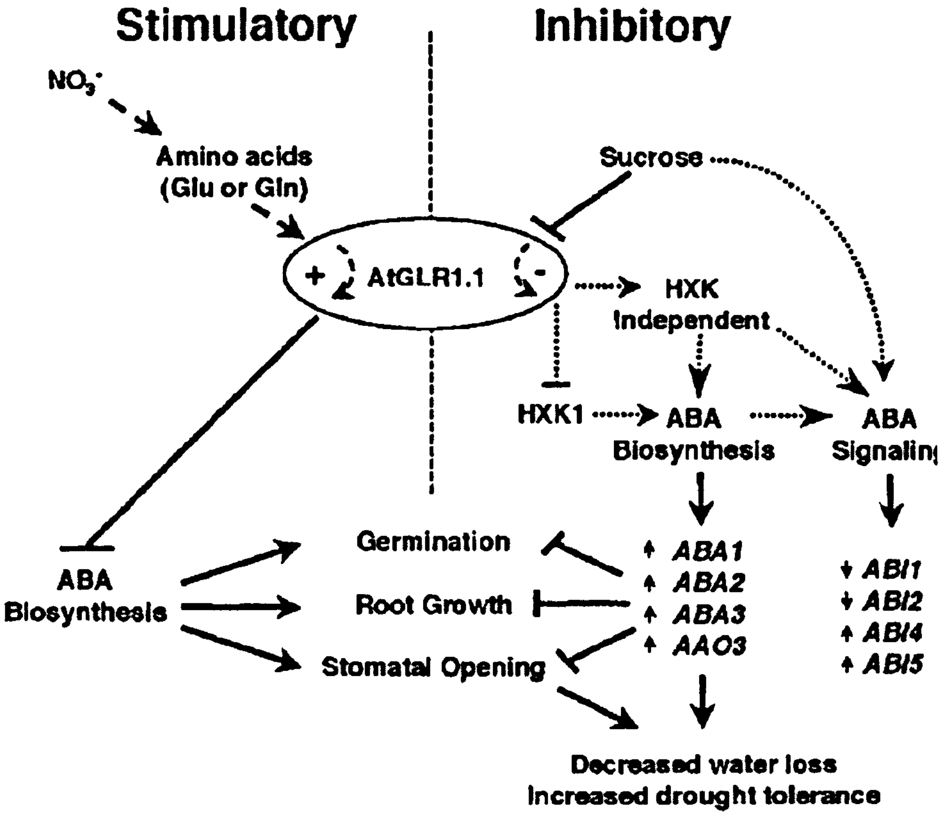


Figure 6.5 A putative model of the regulatory pathways involving AtGLR1.1 receptor. From Kang *et al.* (2004).

to high nitrate media and it was suggested by Orsel *et al.* (2006) that this may indicate that AtNAR2.1 has other functions, as yet unidentified.

Some of the genes important for the detection of C:N ratio and mediation of relation responses are also known. One such gene is AtGLR1.1, a member of a family of 20 putative glutamate receptors identified by homology (Lam *et al.*, 1998). In a study by Kang *et al.* (2004), it was shown to have a role in the mediation of the ABA biosynthesis during germination in response to N and C signals. In this study a model was produced that summarises these findings, which is shown in **Figure 6.5**. Another transcription factor implicated in the regulation of the C:N effects is DOF1 (Yanagisawa *et al.*, 2004). This gene is known to positively control the expression of several enzymes involved in the production of carbon skeletons used by the nitrate assimilation. Over-expression of the DOF1 gene led to an increased nitrogen use efficiency and higher nitrogen content in the tissues (Yanagisawa *et al.*, 2004). In a genetic characterisation study by Bi *et al.* (2005) (a member of a family of 30 GATA transcription factors) was identified to be nitrate-inducible. The 150-bp long *cis*-regulatory site adjacent to the AtNRT2.1 transporter was found to contain the possible binding motifs for both DOF1 and GNC (Girin *et al.*, 2007, Vidal and Gutierrez, 2008). However, experimental evidence is at present lacking to confirm this link. One other gene proposed as an important modulator of C:N responses is a putative methyltransferase OSU1/QUA2/TSD2 (Gao *et al.*, 2008). A mutation in this gene was found to cause plants to become more sensitive to the unbalanced C:N ratio and have higher expression of ASN1 enzyme (Gao *et al.*, 2008).

An organic nitrogen-responsive gene regulatory network consisting of CCA1, bZIP1 and GLK1 transcription factors was identified in a study by Gutierrez *et al.* (2008), in which a treatment was designed that allowed them to identify the groups of genes that respond to organic *versus* inorganic nitrogen metabolites. The same study also constructed a network model and produced some supporting evidence that these transcription factors are, in turn, regulating the enzymes important for the primary amino acid synthesis, namely ASN1, GDH1 and GLN1.3. CCA1 is also known to be part of the circadian clock circuit in *Arabidopsis*, and, in this way may also be important for the integration of the circadian effects and nitrogen metabolism (Gutierrez *et al.*, 2008).

6.3 Application case: AtNAR2.1 mutant study

As explained in the introduction section, high affinity transporters (HATs) mediate the uptake when the concentration of the substrate in the growth medium is low. One of the key components of nitrate HATs in *Arabidopsis* is AtNRT2.1. Although other transporters have been demonstrated to be able to take-up nitrate in the low affinity range, this protein is believed to be the main transporter for primary uptake in the *Arabidopsis* roots.

In the Orsel *et al.* (2006) study it was demonstrated that this transporter is part of a two-component system, where a protein-protein interaction with AtNAR2.1 protein is required to ensure that this transporter is correctly targeted to the outer cell membrane. In the same study it was also demonstrated that without the AtNAR2.1 protein the plants are unable to take up sufficient nitrate at the low affinity range, which results in growth retardation and dwarf phenotype (**Figure 6.6**). Besides being a key component of the nitrate uptake system, originally AtNAR2.1 was characterised as a wounding-response protein (Titarenko *et al.*, 1997). AtNRT2.1 was also hypothesised to be involved in signalling or nitrate sensing (Little *et al.*, 2005). As the mechanisms for both of these processes remain poorly understood, an expression study of the AtNAR2.1 mutant was chosen as an example for this chapter because of its relevance to this direction of on-going research.

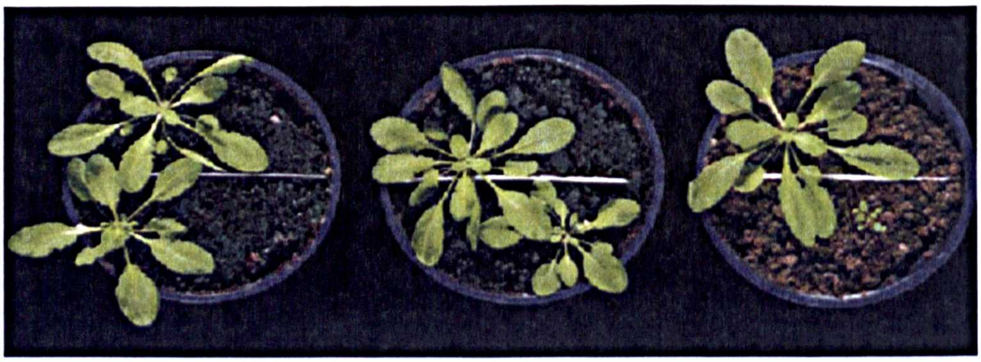


Figure 6.6 A mutant with a defective copy of the AtNAR2.1 protein (lower plants) and wild-type (upper plants) grown in soil with high medium and low nitrogen availability (right to left). Image from Orsel *et al.* (2006).

6.3.1 Experiment overview

The selected experiment looked at the differential expression in the shoot and root of the wild-type and an AtNAR2.1 mutant. In the mutant line the AtNAR2.1 protein copy is still expressed, but is defective with the effect of disrupting the targeting of the AtNRT2.1 transporter to the outer cell membrane. The pre-analysed results of this experiment where obtained from the CATdb database (Gagnot *et al.*, 2008), where it is available under the accession RA05-12_NAR2. The *nar2.1* mutant and wild-type plants were grown on hydroponic media with 6mM nitrate for 41 days. After that, they were transferred to the media containing either 0.2mM or 6mM nitrate. The plants were harvested after 24 hours and profiled using Complete *Arabidopsis* Transcriptome MicroArray (CATMA) technology (Crowe *et al.*, 2003). The shoot and root were profiled separately, with four biological and eight technical replicas in each case. The analysis done by the investigators produced four lists of differentially expressed genes (organ type x nitrate availability) where the expression was compared between the *nar2.1* and the wild-type plants subjected to the same treatment. These lists were loaded into the Ondex system, mapped onto the integrated network and served as a set of "guide genes" for further analysis.

6.3.2 Coexpression network construction

The coexpression network was constructed by selecting a subset of expression experiments where nitrogen regulation, assimilation or uptake systems were perturbed. The suitable experiments were selected by manually reviewing all of the expression profiling experiments from the NASC and ArrayExpress databases that used Affymetrix ATH1-121501 chip. The final set contained 220 slides from the 13 different experiments (**Table 4.2**). This set of slides was then processed using the coexpression network construction method described in chapter 6. For this analysis, the ambiguous probe sets were excluded which meant that the total number of genes in the dataset was reduced to 20440. The weighted version of the Pearson correlation was used and the cut-off threshold for the inclusion of edges was determined according to the method of Elo *et al.* (2007) and found to be 0.77. Application of this threshold resulted in a network of 11360 nodes and 882862 edges. This network was then imported into an

Table 6.1 Microarray experiments used for the construction of the nitrogen-relevant meta-coexpression network.

Array ID	Description	Total number of samples	Removed samples
NASC experiments			
NASCARRAYS-490	Cell-specific nitrogen responses in the *Arabidopsis* root	91	
NASCARRAYS-481	*Arabidopsis* treated with nitrite and nitrate	6	
NASCARRAYS-485	Systems approach identifies an organic nitrogen-responsive gene network that is regulated by the master clock control gene CCA1.	8	
NASCARRAYS-479	Treatment of *Arabidopsis* with low concn. of nitrate	8	
NASCARRAYS-480	WT vs. NR null mutant high nitrate concn. treatment	16	
NASCARRAYS-103	Identification of genes involved in nutritional regulation of root architecture	6	1
NASCARRAYS-46	Nutritional control of plant development: molecular analysis of the NO_3^- response pathway in *Arabidopsis* roots.	8	4
ArrayExpress experiments			
E-GEOD-20493	Transcriptional profiling of an Fd-GOGAT1/GLU1 mutant in *Arabidopsis thaliana*	16	
E-GEOD-18818	Transcription profiling of *Arabidopsis* over-expressers and mutants of TFs-gene LBD37 and LBD38 under different nitrogen regimes	18	
E-GEOD-9148	Transcription profiling of *Arabidopsis* 10 day old wild type and *chl1-5* plants exposed to 25 mM nitrate for 0h or 0.5h	12	
E-MEXP-828*	Transcription profiling of *Arabidopsis* roots from plants grown in nutrient solutions with various concentrations of nitrate and sucrose	34	10
E-MEXP-1771	Transcription profiling of *Arabidopsis* wild type seedlings grown with NH_4NO_3 or urea as nitrogen sources	4	
E-MEXP-1770	Transcription profiling of *Arabidopsis* wild type and choride channel d-1 mutant seedlings grown on various nitrogen sources	8	

*This set also contains samples that looked at the responses to carbon availability only, which were excluded

Ondex representation and integrated with the ARA-REF, GOA-BP and SUBA datasets using the TAIR accession identifiers.

6.3.3 Dataset analysis

The coexpression network was analysed in the Ondex front end and filtered by selecting only the portions of the network that corresponded to a particular set (or combination of sets) of differentially expressed genes. The subsequent analysis methods were then only applied to these selected subsets of nodes and edges. As the shoot and root sets for plants grown under high nitrogen had very small number of genes, only the two lists from the low-nitrogen treated plants were used for further analysis.

First, the combined functional enrichment of the root and shoot sets under the low nitrogen availability was conducted. The enrichment was calculated separately for each list relative to the total set of all Arabidopsis genes from the ARA-REF with GO biological process annotation. For each of the function categories found to be differentially expressed, a count of up- and down-regulated genes was made in order to interpret the type of the response. In order to understand the differences in the shoot and root responses, the produced two sets of GO terms were then compared with each other and commonalities were identified. The set membership of the enriched GO terms was then added as an attribute to the corresponding GO term node in order to enable visual inspection in the Ondex front end.

At the next step, the modular structure of these sub-networks was further explored through the application of the MCL clustering algorithm. For each clustering run, a sub-network of the whole coexpression network was selected

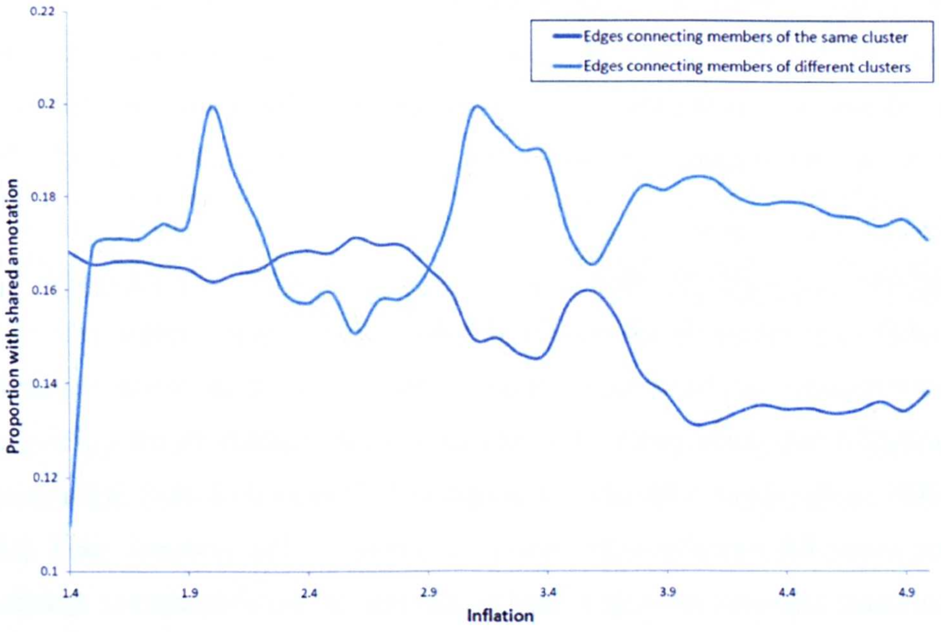


Figure 6.7 Optimization of the MCL inflation coefficient. The plot shows how the proportion of the edges that connect two genes that share at least one GO annotation changes with the inflation coefficient, as the edges are reassigned to either within- or across- cluster sets.

by only retaining a set of nodes corresponding to a set (or a combination of set) for particular gene list(s). This network was then interpreted as an adjacency matrix weighted by the absolute value of the Pearson correlation coefficient of the coexpression edges and passed to the MCL algorithm. The inflation coefficient (I) parameter of the MCL controls the granularity of clusters returned by the algorithm. For this application case, it was op timised to give the optimum distribution of the edges that connect functionally similar genes (**Figure 6.7**). As shown in this Figure, the optimum I range for this graph was found to be between 2.3-2.9. At the higher levels, only one large cluster and one-node large "orphan" clusters were produced with only a small numbers of edges around the periphery of the main cluster reassigned as the inflation was increased further. This accounts for the symmetric pattern observed in the graph after this point. The greatest positive difference between the within- and across- cluster links with shared functional was found to be 2.6, and this value of I was used for the analysis.

After clustering, the GO function of the modules was explored using both the sets of representative MICAs and statistical enrichment (Fisher's exact test) approaches. This portion of the analysis was undertaken for the pooled set of genes from both root and shoot under the low nitrogen treatment under the hypothesis that, as largely similar or related set of functional responses was found in both organs, the combined set can serve to further highlight this commonality.

The transcription factor data was used to identify the likely regulators that might be controlling the expression with the modules and possible transcription factor-to-target relationships. In order to further explore the link between the AtNAR2.1 and the response to wounding, a list of genes from the Chini *et al.* (2007) study was also included in the network. This study looked at the genes that were differentially expressed in response to the jasmonic acid (JA) treatment. Jasmonic acid is a known initiator of the downstream systemic response to wounding (Titarenko *et al.*, 1997).

The analysis was also combined with the manual examination of the gene sets, underlying network and the relationship between the enriched functional terms and the structure of the Gene Ontology. In addition to the methods for

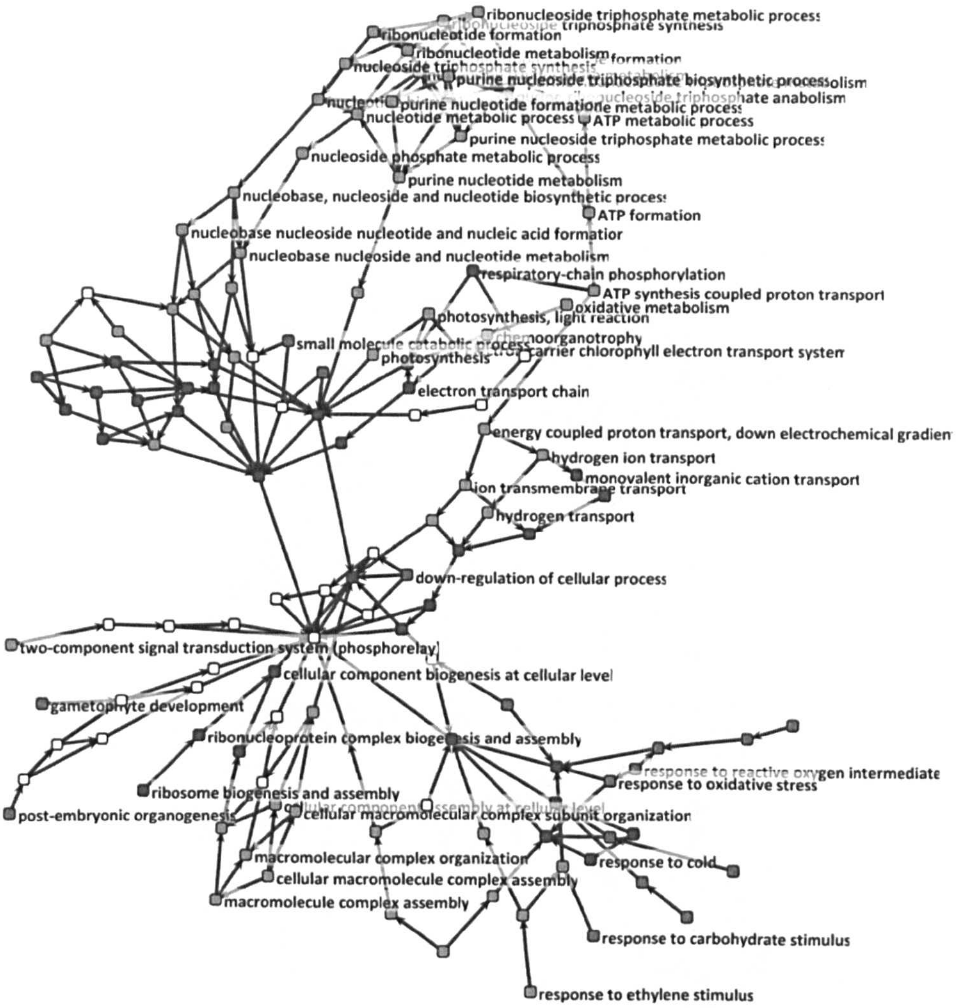

ribonucleoside triphosphate metabolic process
triphosphate synthesis
ribonucleotide formation
ribonucleotide metabolism
formation
nucleoside triphosphate synthesis
purine nucleoside triphosphate biosynthetic process
olism
hate anabolism
purine nucleotide formation
e metabolic process
nucleotide metabolic process
ATP metabolic process
purine nucleoside triphosphate metabolic process
nucleoside phosphate metabolic process
purine nucleotide metabolism
nucleobase, nucleoside and nucleotide biosynthetic process
ATP formation
nucleobase nucleoside nucleotide and nucleic acid formation
nucleobase nucleoside and nucleotide metabolism
respiratory-chain phosphorylation
ATP synthesis coupled proton transport
oxidative metabolism
photosynthesis, light reaction
small molecule catabolic process
chemoorganotrophy
carrier chlorophyll electron transport system
photosynthesis
electron transport chain
energy coupled proton transport, down electrochemical gradient
hydrogen ion transport
monovalent inorganic cation transport
ion transmembrane transport
hydrogen transport
down-regulation of cellular process
two-component signal transduction system (phosphorelay)
cellular component biogenesis at cellular level
gametophyte development
ribonucleoprotein complex biogenesis and assembly
response to reactive oxygen intermediate
response to oxidative stress
ribosome biogenesis and assembly
post-embryonic organogenesis
cellular macromolecular complex subunit organization
macromolecular complex organization
response to cold
cellular macromolecule complex assembly
macromolecule complex assembly
response to carbohydrate stimulus
response to ethylene stimulus

Figure 6.8 GO-centric representation showing the entirety of the significantly enriched biological processes in the shoot and root sets mapped onto a GO ontology DAG. Shoot processes are shown in green root – in red and those found in both organs – in blue. To reduce clutter, labels are only shown for the terms with information content greater than 6.25.

annotation and ontology-driven analysis, which were already introduced in the earlier chapters, new visualisation strategies in the Ondex front end were also implemented that complement them and allow the effective presentation of these often large statistical sets. As well as show-casing the developed analytical functionality of Ondex in an applied setting, the example presented in this chapter also aims to demonstrate these visual and interactive components of the system.

6.3.4 Results

In order to understand the underpinning set of biological differences between

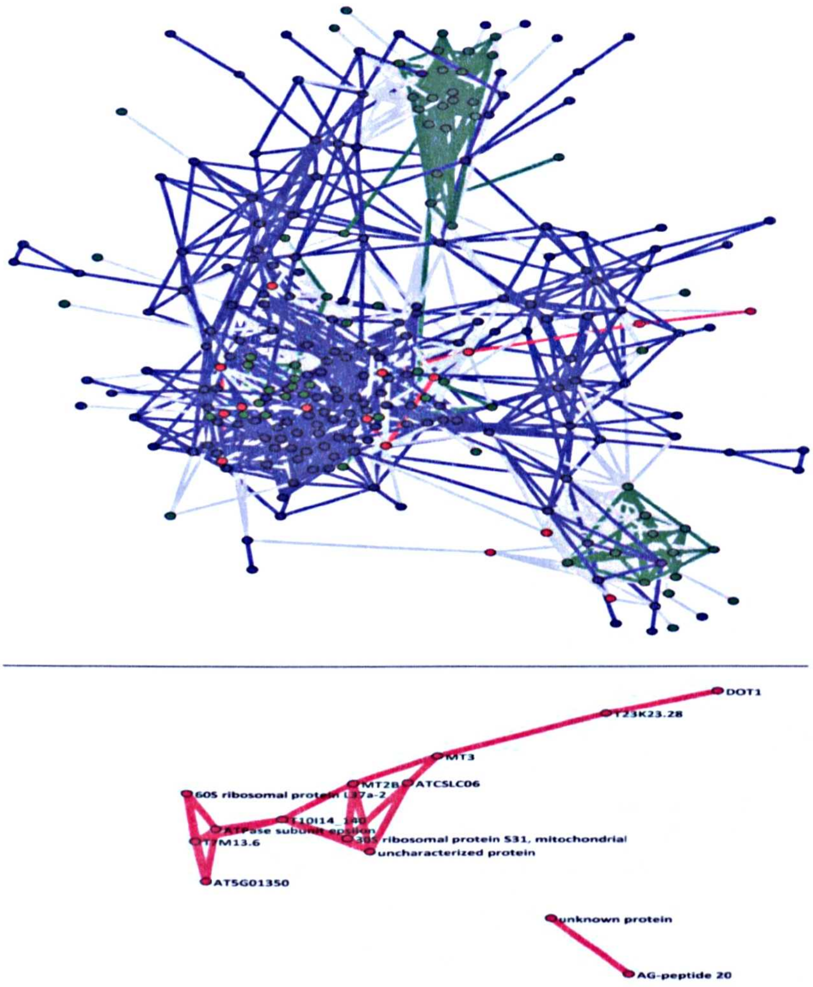


Figure 6.9 The main connected component of the coexpression network that was filtered to only contain the genes from the shoot and root sets from the low-nitrogen treatment (top). Root-specific nodes and connecting edges are shown in blue, shoot – in green and common to both – in pink. The lower panel shows the common subset only.

the mutant and a wild type the combined set was analysed for functional enrichment. The results of this analysis are presented in the **Figure 6.8**. A complete set of all enriched biological processes is also included in the Appendix. From this visualisation it is evident that, as expected the set includes a number of nitrogen metabolism-related terms. In particular, a lot of terms appear to be themed around nucleotide, purine and ribonucleotide metabolism (top of **Figure 6.8**). This image also illustrates a specialised view for presenting the GO annotation analysis results in Ondex front end developed as part of the work on this thesis. However, because of the constraint of the page margins, the image had to be condensed. As this lead to reduced readability, the same information is also provided as a supplementary table in the Appendix D of this thesis. This set of processes found to be significantly enriched by this analysis appears to be primarily restricted to the above ground part of the plant. A

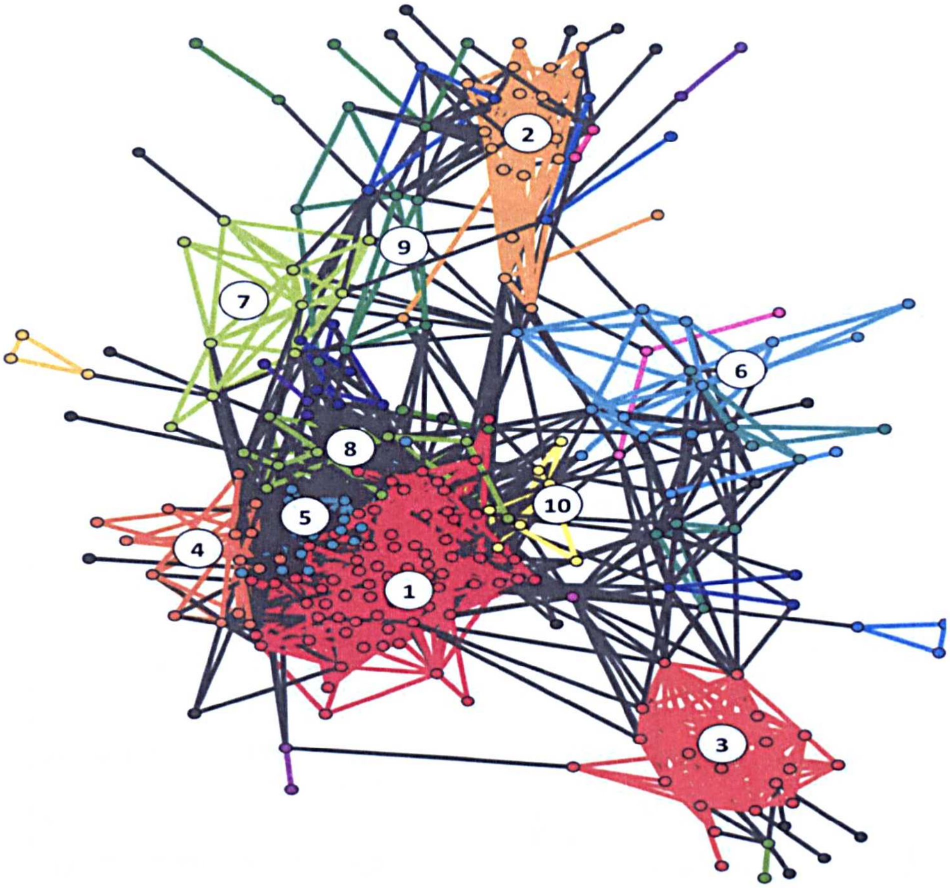


Figure 6.10 Modules in the main connected component as they have been resolved by the clustering algorithm. The numbers show the assigned identifiers by which they are referred to in the main text.

downstream, related process of "ribonucleoprotein complex biogenesis and assembly", is occurring in both of the organs.

Another set of processes specific to the shoot appears to be related to the generation of ATP/energy through combination photosynthesis ("photosynthesis", "electron transport chain") and respiration ("ATP formation", "respiratory-chain phosphorelation") processes. Another, indirect indicator that there may be an excess of carbon metabolites as a result of up-regulated photosynthetic activity in the mutant is the presence of the "response to carbohydrate stimulus" process in the root. There are also some evidence of the regulatory events – the "phosphorelay" and "response to ethylene stimulus" processes in the shoot and "down-regulation of cellular process" in the root.

Table 6.2 A selection of significantly enriched biological processes in clusters.

Cluster	Members	Go term	IC	Go term name
1	71	GO:0019538	2.92	protein metabolism
		GO:0043284	3.41	macromolecule biosynthesis
		GO:0006416	5.41	protein biosynthesis
		GO:0022613	7.01	ribonucleoprotein complex biogenesis and assembly
		GO:0042254	7.06	ribosome biogenesis and assembly
2	23	GO:0051869	2.52	physiological response to stimulus
		GO:0006950	3.23	response to stress
3	19	GO:0051869	2.52	physiological response to stimulus
		GO:0006950	3.23	response to stress
		GO:0051171	3.48	regulation of nitrogen metabolism
		GO:0009725	4.60	response to hormone stimulus
		GO:0023033	4.56	signaling pathway
		GO:0006355	4.45	regulation of transcription, DNA-dependent
		GO:0051252	4.44	regulation of RNA metabolism
		GO:0009873	7.09	ethylene mediated signaling pathway
		GO:0009743	6.84	response to carbohydrate stimulus
		GO:0009723	6.53	response to ethylene stimulus
5	14	GO:0006416	5.41	protein biosynthesis
6	13	GO:0006950	3.23	response to stress
7	10	GO:0006950	3.23	response to stress
		GO:0006807	2.93	nitrogen compound metabolic process

The "small molecule catabolic process" may be an indication of the nitrogen recycling in response to reduced supply, as the mutant plant would not be able to take up sufficient amounts of nitrate at the concentration used for the treatment. There are also a number of enriched terms in the "response to stress" branch of the ontology – e.g. "response to oxidative stress" and "response to cold".

The main component of the coexpression network linking the nodes in the two low nitrogen gene lists is shown in **Figure 6.9**. This set contains 260 out of 419 total genes in the pooled lists. Ten other genes located in very small connected components and 149 are not connected via the coexpression links (not included in the figure). The intersection between the shoot and root sets contains only 42 genes. This is in contrast to the much larger intersection of enriched processes common to both organs and may be an indication of a common regulatory control, which is then subject to the organ-specific modulation that leads to the same types of processes being realised by differing sets of proteins specific to those two organs. Interestingly, in the main connected component all but one of these genes are connected to at least one of the others via the coexpression edge. Of those, all but two are indirectly connected to all others in this way. This pattern may be an indication that the common mechanism of control is realised at the level of transcription. Therefore, this set of genes is particularly suitable for further study aiming to identify this putative common control mechanism and its link to the AtNAR2.1/AtNRT2.1 genes.

In the **Figure 6.10** it is shown how the main connected component was partitioned by the MCL algorithm. Including the other, smaller connected components 32 clusters have been identified in total. However, as evident from the image a large proportion are of size 1-2 and are of limited use for further interpretation of the network. All of the identified clusters were assigned numerical labels for further reference, in order of decreasing cluster size.

Ten of the largest clusters in **Figure 6.10** are identified by labels by which they will be referred to in this chapter. The **Table 6.4** shows the sets of representative sets of MICAs identified for the clusters, which are applicable to at least 40% of all members. The largest cluster (1) appears to have a lot of genes involved in "cellular biopolymer metabolic process". Further examination of its members appears to indicate that this activity is protein synthesis, as 42 of the proteins in this module were found to have annotation or name (from either TAIR or UniProt) that identifies them as components of the ribosome. This fact does not appear to be adequately represented by the GO annotation, where only 22 of these proteins were annotated to one of the protein synthesis or ribosome-related terms ("protein synthesis elongation", "protein synthesis" or "ribosome biogenesis"). This indicates that, even despite using the most complete GO set possible, the annotation sets for proteins are still incomplete. A similar situation was also observed in the case of cluster 5, which is located close to 1, is annotated by the "cellular biopolymer metabolic process" term and contains proteins where name or annotation indicates ribosome-related activity. Interestingly, despite the similar function and proximity within the network, cluster 5 appears to be predominantly shoots-specific, whereas 1 is mostly composed of root-localised proteins. This may be an indication that this separation may be biologically justified.

Clusters 2 and 3 appear to contain "response to stress"-annotated genes. Notably, cluster 3 also contains 8 genes annotated with "cellular nitrogen compound metabolism" term, indicating its immediate relevance to the nitrogen-related responses.

Table 6.3 All significantly enriched SUBA compartments in clusters.

Cluster	Number of members	SUBA compartment	Annotated genes
1	71	mitochondrion	13
		peroxisome	6
		cytosol	11
2	23	plastid	10
3	19	nucleus	6
5	14	vacuole	3
		cytosol	4
		nucleus	7
6	13	extracellular	3
		nucleus	4
7	10	plastid	5
8	9	golgi	3
		vacuole	4
10	7	peroxisome	2

The enrichment analysis of the clusters appears to be largely consistent with these observations. The **Table 6.2** provides an overview of the enriched processes – because the enrichment analysis is not limited by the low numbers of proteins, in this case the "protein metabolism" was identified as significantly over-represented in clusters 1 and 5. One of the largest groups of significantly over-represented gens in clusters 2, 3, 6 and 7 was found to be "response to stress" process. Additionally, Cluster 3 was enriched for "regulation of nitrogen metabolism" (8/19 genes) and cluster 7 – for "nitrogen compound metabolic process" (4/10) genes, which are of particular relevance to the topic of the experiment.

Table 6.4 All MICAs that apply to at least 40% of annotated cluster members.

Cluster	Cluster members	MICA	IC	Name	Coverage (relative to annotated genes)
1	71	GO:0034960	2.29	cellular biopolymer metabolic process	41.43%
		GO:0044238	1.67	primary metabolic process	45.71%
2	23	GO:0006950	3.23	response to stress	40.00%
		GO:0009987	1.28	cellular physiological process	40.00%
3	19	GO:0006950	3.23	response to stress	50.00%
		GO:0010033	4.22	response to organic substance	55.56%
		GO:0009719	4.49	response to endogenous stimulus	44.44%
		GO:0034641	2.99	cellular nitrogen compound metabolism	44.44%
		GO:0061019	3.54	regulation of cellular transcription	44.44%
		GO:0044238	1.67	primary metabolic process	44.44%
5	14	GO:0071840	3.72	cellular component organization or biogenesis	42.86%
		GO:0010467	3.38	gene expression	50.00%
		GO:0034960	2.29	cellular biopolymer metabolic process	57.14%
		GO:0044249	2.78	cellular biosynthetic process	42.86%
		GO:0044238	1.67	primary metabolic process	71.43%
6	13	GO:0008152	1.36	metabolic process	63.64%
7	10	GO:0044237	1.73	cellular metabolic process	60.00%

Cluster 3 had the largest count of enriched biological processes (51 terms). Among them, were the "ethylene mediated signalling pathway", "ethylene mediated signalling pathway", "response to carbohydrate stimulus", "regulation of RNA metabolism" and "regulation of transcription, DNA-dependent". The co-occurrence of these functions in the same cluster may be an indication that they are related. One possible hypothesis would be that the RNA synthesis (possibly contributing to the ribosome biogenesis process) is up-regulated in response to the ethylene and/or carbohydrate stimulus. As most of these functions are annotated to the same group of 7 proteins, which are appear to be transcription factors (both according to transcription factor databases and "regulation of transcription, DNA-dependent" GO annotation) the mechanism of this regulation appears to be on the level of transcription. The origin of the carbohydrate stimulus may be due to increased photosynthetic activity in shoot, which was noted earlier. Another set of annotations of note are "physiological defence response" and "response to chitin", as both of them are defence responses related to wounding – either due to predation or pathogen activity. Ethylene is also known to be one of the phytohormones important for wounding response in *Arabidopsis* (Titarenko *et al.*, 1997).

Additionally, the enrichment with respect to SUBA compartment categories was also looked at. The results are presented in **Table 6.3**. Cluster 1 appears to also include proteins localised to the mitochondrion and peroxisome. The former may be an indication that members of this module may be also important for the respiration-related processes, which were picked up in the combined set enrichment analysis, but were not observed in any of the modules. Clusters 2 and 7 contain a larger than expected number of plastid proteins, indicating its possible involvement in the photosynthesis-related activities. Cluster 2, which has 10 of plastid-localised proteins, also appears to be predominantly shoot-specific. That provides an additional indication of possible importance of this module for photosynthetic processes.

To further understand the regulation on the level of transcription that gave rise to the observed coexpression network, the transcription factors and the genes that they are directly coexpressed with were highlighted in the network (**Figure 6.11**). This was done on the basis of transcription factor annotation contributed by the SET. With the exception of bZIP47, all of the most highly connected transcription factors are found within larger clusters and tend to be coexpressed with the members of that cluster, indicating good correspondence between cluster assignment and likely co-regulated groups of genes. In the cluster 1, 39 out of 71 members are coexpressed by just three transcription factors. Out of them, 10 are with all three, and 22 are coexpressed with two. Surprisingly, the transcription factors themselves are not coexpressed with each other possibly

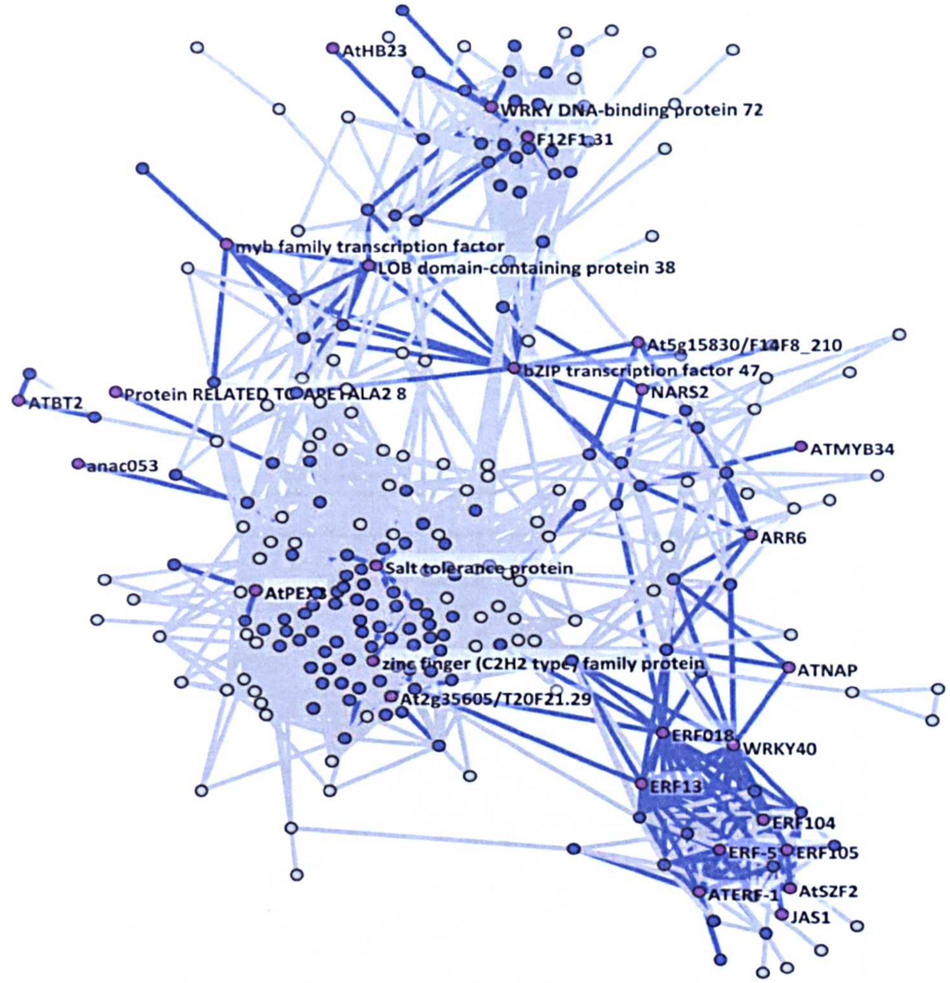

AtHB23
WRKY DNA-binding protein 72
F12F1.31
myb family transcription factor
LOB domain-containing protein 38
At5g15830/F14F8_210
bZIP transcription factor 47
ATBT2
Protein RELATED TO APETALA2 8
NARS2
ATMYB34
anac053
ARR6
Salt tolerance protein
AtPEX3
zinc finger (C2H2 type) family protein
ATNAP
At2g35605/T20F21.29
ERF018
WRKY40
ERF13
ERF104
ERF-5
ERF105
ATERF-1
AtSZF2
JAS1

Figure 6.11 Transcription factors (purple) and their targets (blue).

indicating that the mechanism that coordinates their activity in this experiment is not transcription-based. One of these transcription factors (AT2G35605) was found to be completely unannotated, so its association with this module via the coexpression links has provided some clues to its likely function.

A most notable feature highlighted by this view is that cluster 3 contains a large group of coexpressed transcription factors; all but two of them are Ethylene-Responsive Transcription Factors (ERFs). The remaining two transcription factors are AtSZF2 and WRKY40, both of which are believed to be particularly important for response to pathogens (Chen *et al.*, 2010, AbuQamar *et al.*, 2006). In the AbuQamar *et al.* (2006) study, AtSZF2 was also observed to be coexpressed with several of the ERF and WRKY transcription factors in response to *Botrytis* infection.

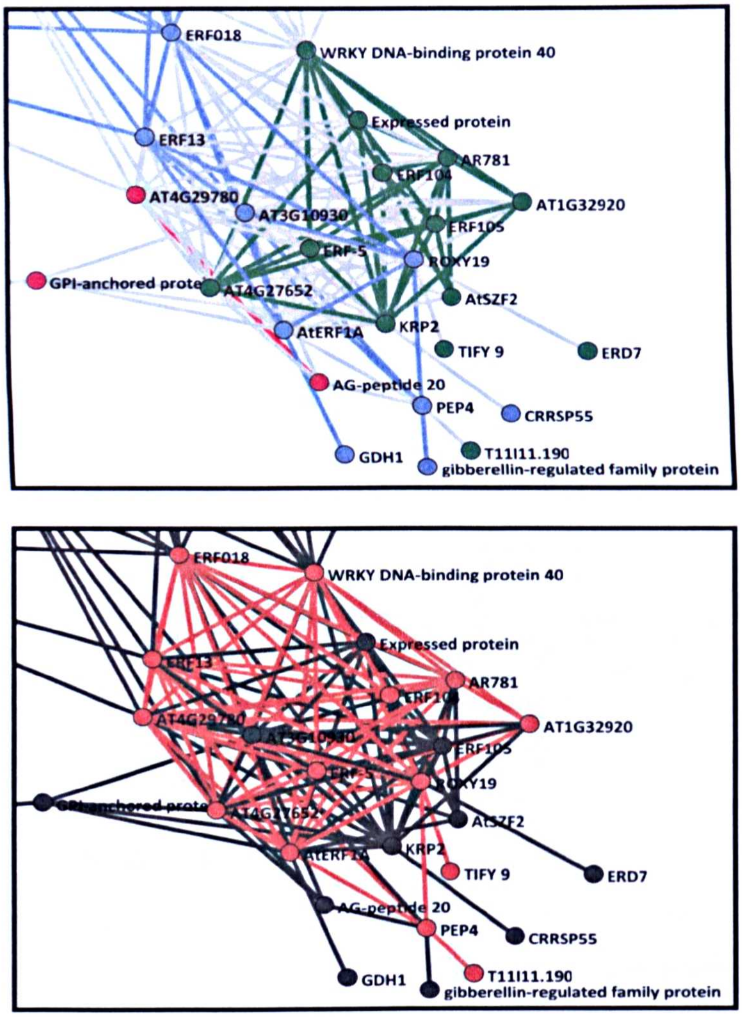


Figure 6.12 A labeled members of cluster 3 showing the root (blue), shoot (green) and both (red) gene list membership (top panel) and intersection of the pooled list of genes with the list of JA-responsive genes (in orange, bottom panel).

By overlaying the network with an additional list of genes known to respond to jasmonic acid (JA) treatment (**Figure 6.12**, top), it was possible to see that there is only one large, coexpressed group of genes in that intersection and that it almost directly corresponds to the cluster 3. Although none of the members of this cluster had any JA-related annotation, the only JA-related protein in the coexpression network (JAS1/TIFY9) was found to be directly connected to cluster 3 with its only edge in this set and was also part of the intersection of the AtNAR2.1 mutant and JA-responsive set, providing additional indirect evidence of the implication of this module in the wounding response and jasmonic acid.

Cluster 3 contains 11 shoot-specific proteins, 7 root-specific ones and 2 which are found in both organs (**Figure 6.12**, bottom). This effectively splits the cluster almost in half, again indicating that a similar biological process is realised by two largely distinct groups of genes. However, as AT4G29780 and AG-peptide 20 are found in both sets, it is possible to hypothesise that they may either be directly involved in the coordinated regulation of this model or are directly controlled by a regulatory mechanism found in both shoot and root. Of the two genes, only a completely uncharacterised AT4G29780 is coexpressed with all of the transcription factors, making it a good target for further research for better understanding of the .

6.3.5 Discussion

Studying the regulatory effects of nitrogen is often complicated not only by the sheer number and complexity of the regulatory pathways, but also by the context-specific nature of these responses. For example, in a meta-analysis study by Gutierrez *et al.* (2007a) of the 2021 genes differentially expressed in at least one of four N-system perturbation experiments, only 345 were found to be shared across all of them. The effects at tissue- and cell type-specific levels were also found to be very distinct (Gifford *et al.*, 2008). Previously the systems approach, which employs networks for interpretation of these complex heterogeneous datasets, was already successfully used in the study of nitrogen-dependent regulation (Gutierrez *et al.*, 2007b, Gutierrez *et al.*, 2008) and lead to the identification of the CCA1 as one of the important controllers of the nitrogen metabolism in addition to its involvement in the circadian clock. In this application case, a similar, network- and functional annotation driven approach was implemented based on the various datasets and components development of which was described in earlier chapters. The approach of constraining the coexpression data by applying an appropriately selected gene list as a filter was introduced as a way of providing a context-specific focus to an otherwise large and complex coexpression network.

By integrating the data from the NAR2.1 mutant with coexpression data it was possible to gain additional insight about the groups of genes that are involved in a particular biological functions affected in this mutant. In particular, a large group of highly interconnected nodes was dissected into clusters 1 and 5, which were found to be involved in protein synthesis in the root and shoot respectively. In addition to that, by identifying the transcription factors present in those two modules putative regulatory links were inferred. In module 5 the regulator appears to be the "salt tolerance protein", whereas in cluster 1 the two putative regulators were the "zinc finger (C2H2 type) family protein" and "At2g35605/T20F21.29". It is also notable that a number of transcription factors in this network were poorly annotated and, by considering the functional composition of the modules where they were found, it was possible to make inferences about their function. By combining the coexpression network with set-driven analysis that allows several gene lists to be explored in parallel, a group of transcription factors in forming a module shown on **Figure 6.12** was identified that are likely to be important for understanding a link between the NAR2.1/NRT2.1 and response to wounding previously reported in another study (Titarenko *et al.*, 1997). Further to that, a number of currently uncharacterised genes where also linked to that process by a combination of coexpression and clustering analysis.

Coexpression can provide important clues about the regulatory relationships between transcription factors and their target proteins. In the case of *Arabidopsis* the ability to extract these relationships using this method is particularly important – as at the moment very few such regulatory links are available from the databases, like AtRegNet (Palaniswamy *et al.*, 2006). This situation, however, also poses an additional problem in that although coexpression has been shown to be useful for extraction of regulatory links in other well-studied species, there is insufficient data to quantify the accuracy of these predictions. Even in the studies where such relationships were validated, some problems were encountered due to insufficiently representative negative control set size – e.g. the transcription factors that were demonstrated not to regulate particular targets. For this work, the assumption was made that coexpression between transcription factors and other types of proteins may infer direct regulatory relationships. However it is also recognised that there are likely to be a number of false positive associations that arose from this analysis, but there exact number cannot be ascertained exactly using currently available data.

Another limitation of the analysis is that the fact that the coexpression links were recovered from the set of related experiments does not necessarily imply that they were also active in the experiment from which the gene set have originated. As such, a link in a network should only be seen as a "best guess" about what coexpression effects may be of relevance to the gene set. Consequently, such associations should always be confirmed by more direct, experimental means if they are found to be particularly important. Some regulatory relationships may also be realised via PPI or protein-metabolite interactions, which are not captured in the coexpression network and proteins may also have currently unknown roles in regulation of transcription.

Coexpression networks also combine expression patterns from many different tissues and therefore some of the links may not actually be active in the same combination in reality. This problem was partially addressed in the work described here by only considering a subset of the network at a time, which consisted of the genes that have been shown to be expressed together at the same tissue. However, even this strategy may not produce the best possible result, as some of the tissues are actually composed of several transcriptomically distinct cell types – e.g. 'root' can be further decomposed into at least ten different types of cells (Dolan *et al.*, 1993). The results of the transcriptomic studies that used fluorescent sorting method (Bargmann and Birnbaum, 2010) to separate individual cell type may be an even more accurate way of addressing this problem, but at present very few expression data sets that use this technique are available. Data from experiments with better levels of spatial resolution are likely to become increasingly more common in the future because the adoption of next generation sequencing technologies enables very small amounts of RNA to be quantified (Hoen *et al.*, 2008), potentially opening up the possibility of a single-cell expression profiling.

Despite the challenges outlined above, the method described here provides a robust and flexible approach to coexpression network construction and incorporates a number of current methods for improving quality of the results produced. In this chapter it was also illustrated that this approach provides a functional way of mining the wealth of currently available microarray data and allows this type of data to be summarised and interactively explored. By applying additional level of filtering to the data – both at the stage when the subset of experiments for network construction is selected and by using gene lists to restrict the set further, may bring an additional advantage of detecting coexpression modules that only exist under specific conditions, however further work will be necessary to confirm that this is indeed the case. Another potential uses of this pipeline include target gene prioritisation and identification of putative transcription factor-target relationships.

7 CONCLUSION

The main motivation for the work in this thesis was to develop an integrated analysis framework for studying transcriptomics data from *Arabidopsis thaliana* using biological network analysis approaches. To realise this objective, it was necessary to extend and refine the Ondex system in two ways: (1) to develop software tools that would deliver the necessary analysis using flexible software architecture (2) to create an appropriate data model and a set of integration pipelines to populate it. Both these tasks are, however, interdependent to some extent, as the integration and analysis was defined in terms of the tools and components of the Ondex framework. The principles of generality and standardisation were rigorously applied on all levels of the development processes. As a result, the code that was contributed to the Ondex system can function both a self-contained application to deliver the analysis presented in the earlier chapters, but also can be decomposed into a set of independent functions and software modules. These simple units of organisation can then be re-used as the basis for subsequent developments or can be re-arranged to deliver different analysis pipelines.

Among the new functionality delivered was the addition of support for new types of data and resources previously unavailable in the Ondex data integration toolkit. In particular, this project realised the first introduction of protein-protein interaction and coexpression data to the system. A number of new supporting annotation resources were also added – among them, three transcription factor databases and cellular localisation resources. Extensions to the data analysis features in Ondex included a new interface for scripting environments which has opened up the option to use a wider range of third-party analysis routines within Ondex (in particular NetworkX and R/Bioconductor). Additionally, this project first introduced the use of clustering and graph analysis methods for mining Ondex networks and defined the required formalisms for the conversions of a typed knowledge networks into representations used by these methods. Ontology-driven analysis methods were also implemented in order to facilitate the process of relating the various network features identified by those approaches to the biological function.

These new developments to the Ondex system were used to address problems presented by different application cases from plant bioinformatics. These were presented in chapters 4-6. The first application case (chapter 4) developed a set of integrated gene function resources for plant biology to better understand the coverage and quality of information currently available in functional annotation and protein-protein interaction databases. Understanding how different approaches accumulate, manage and represent biological information impact on the bioinformatics analysis is of great importance for the development of better data integration strategies. The analysis presented in chapter 3 has quantified and compared the differences in the data managed by a number of key data providers of *Arabidopsis* data. The findings have indicated that there are clear benefits arising from the integration of multiple data sources, both in terms of improved confidence and quality of the data. The results of this work have now been published in (Lysenko *et al.*, 2009). The integrated datasets developed for this work were also used to support further analysis presented in chapter 6.

The integrated datasets developed constitute a valuable resource that can be used to drive further analysis. The continued relevance and currency of these datasets was assured by the collection of supporting workflows that can be used to update them with the newest data. As Ondex is a community-driven project with many developers and users, it is likely that, even if the data sources change, the corresponding parsers will be updated by the community to be able to cope with these changes. Many of the datasets developed as part of this work have now contributed to the research of others, and a number of them have now been made available to the wider community via the Ondex project website.

In chapter 5, four datasets of the commonly used evidence types used for gene function inference and annotation were constructed and integrated using the Ondex system. The fully assembled datasets included co-citation information from the scientific literature, protein-protein interaction, sequence similarity and coexpression components, as well as GO annotation, and TAIR and UniProt protein sequence and annotation data. The objective of this work was to gain a better understanding into the relationship between these information types and the impact of taking a union of all these data in the context of identifying and annotating of functional modules. To quantify these relationships, a number of novel strategies were developed that allowed the quantification of such features like entropy and fragmentation of the functionally similar protein sets and a trade-off between coverage and precision of GO annotation. At the moment of writing the work described in that chapter has been accepted for publication in the BMC Bioinformatics journal.

The final application case (chapter 6) aimed to demonstrate the immediate relevance of the integrated resources, tools and analysis methods developed in an applied setting. To that end, a microarray experiment was analysed by mapping the lists of differentially expressed genes reported by the authors and dissected these sets further, by considering their functional context and locations within a coexpression network. A coexpression network was specifically constructed from collected gene expression studies where various components of nitrogen uptake, assimilation or regulation had been explored, as they were likely to be of some relevance to the target experiment (studying the transcriptomic response to mutation in a nitrogen uptake pathway). This example illustrated that, as a result of the newly developed data integration and analysis capabilities, it was now possible to conduct a number of typical gene set analysis tasks entirely within the Ondex system. Additionally, a number of visualisation methods were developed and used to interactively mine and present the results. The analysis was successful in identifying a relevant functional module that may provide further clues about the involvement of the mutated protein in response to wounding. The integration of additional coexpression data has not only helped to relate a number of uncharacterised proteins to appropriate functional contexts, but to also deduce the possible transcription factor-target relationships and identify further structure within this list of genes.

A key objective of this thesis was to demonstrate the broad relevance of the methodological development efforts through their use in three different and mostly independent application cases. It was therefore inevitable that limited time could be dedicated to explore each of these research problems and while it was possible to publish work from the use cases presented in chapters 4 and 5, there is further follow-up work that could be considered. Another challenge faced during this work has been that data integration is a constantly changing and expanding area of bioinformatics. Currently established data resources are also constantly being updated both in terms of content and ways they capture model and share their data. New resources and experimental techniques are also constantly come into existence. On the other hand, new data exchange standards and frameworks are also continuously produced by the bioinformatics community in order to facilitate ease-of-use of these data and ensure its quality.

This means that any data integration tool, including Ondex, requires a constant investment of time and effort in order to update and incorporate new sources of information and analysis methods. Inevitably, this project was also greatly impacted by this need of continuous development and some of the outcome from this work has helped to address this problem. In particular, the development of the Integrator tool (chapter 2) greatly improved the way data integration workflows are created and managed in Ondex. The new workflow execution engine has also lifted the limitation on what data types can be passed between workflow components. The user interface allowed more complex configuration options for the plug-ins to be managed and validated, compared to what was possible prior to this project. Another important achievement was the incorporation of the ability to easily recover workflows with outdated configuration parameters, which used to be a particularly prominent and frequent problem under the previous system. These developments helped to increase both the power of the Ondex system and its relevance to users in the wider research community. As workflows are now considered to be a very important paradigm for organisation, sharing and realisation of complex bioinformatics pipelines (Goble *et al.*, 2010), a high-quality solution for managing workflows is of particular importance to the users of the system.

Many of the individual components of the analysis toolkit implemented during this project are also available as part of other tools. Some examples include network visualisation (Shannon *et al.*, 2003, Enright and Ouzounis, 2001, Breitkreutz *et al.*, 2003), GO enrichment analysis (Shah and Fedoroff, 2004, Zheng and Wang, 2008, Bauer *et al.*, 2008) and coexpression analysis (Manfield *et al.*, 2006, Ernst and Bar-Joseph, 2006, Mostafavi *et al.*, 2008). However, in Ondex these methods are made immediately inter-operable by working off a unified and consistent data model. This not only assures the ease of pulling diverse assortments of data through a series of different analysis methods, but also increases the overall power and flexibility by enabling more ways to present and interrogate the data. An example of this is evident in the way it was possible to visualise data for chapter 6, where both the Gene Ontology graph and the network itself were used to present various aspects of the functional annotation. This is achieved by building a system from a generic set of specialised components that can be combined to enable more complex analysis. This philosophy makes the Ondex system more flexible than many other, small-scale bioinformatics analysis tools that may offer some of the similar functionality, but are ultimately highly restricted to their original purposes.

As high-throughput techniques are now increasingly rising in prominence in biology, larger and more complex datasets are becoming available and require analysing in the context of the pre-existing data and knowledge in Bioinformatics and genomics databases. The Ondex software platform is one possible way to effectively manage and analyse these complex and heterogeneous data. Although a number of alternative solutions are also available, Ondex is the only solution that specialise in catering to the data integration needs of plant biology community. This work has not only addressed a number of short-comings of the system, but has also contributed new data sources and types of analysis that will ensure that it remains highly relevant and useful to the plant bioinformatics researchers.

8 REFERENCES

ABUQAMAR, S., CHEN, X., DHAWAN, R., BLUHM, B., SALMERON, J., LAM, S., DIETRICH, R. A. & MENGISTE, T. 2006. Expression profiling and mutant analysis reveals complex regulatory networks involved in Arabidopsis response to Botrytis infection. *The Plant journal : for cell and molecular biology,* 48, 28-44.

ADAR, E. GUESS: a language and interface for graph exploration. Proceedings of the SIGCHI conference on Human Factors in computing systems, 2006 Montréal, Québec, Canada ACM New York, NY, USA.

AITTOKALLIO, T. & SCHWIKOWSKI, B. 2006. Graph-based methods for analysing networks in cell biology. *Briefings in Bioinformatics,* 7, 243-55.

ALEXA, A., RAHNENFUHRER, J. & LENGAUER, T. 2006. Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics,* 22, 1600-7.

ALMAGRO, A., LIN, S. H. & TSAY, Y. F. 2008. Characterization of the Arabidopsis nitrate transporter NRT1.6 reveals a role of nitrate in early embryo development. *Plant Cell,* 20, 3289-99.

ALON, U. 2003. Biological networks: the tinkerer as an engineer. *Science,* 301, 1866-7.

APWEILER, R., BAIROCH, A., WU, C. H., BARKER, W. C., BOECKMANN, B., FERRO, S., GASTEIGER, E., HUANG, H., LOPEZ, R., MAGRANE, M., MARTIN, M. J., NATALE, D. A., O'DONOVAN, C., REDASCHI, N. & YEH, L. S. 2004. UniProt: the Universal Protein knowledgebase. *Nucleic Acids Res,* 32, D115-9.

ARANDA, B., ACHUTHAN, P., ALAM-FARUQUE, Y., ARMEAN, I., BRIDGE, A., DEROW, C., FEUERMANN, M., GHANBARIAN, A. T., KERRIEN, S., KHADAKE, J., KERSSEMAKERS, J., LEROY, C., MENDEN, M., MICHAUT, M., MONTECCHI-PALAZZI, L., NEUHAUSER, S. N., ORCHARD, S., PERREAU, V., ROECHERT, B., VAN EIJK, K. & HERMJAKOB, H. 2010. The IntAct molecular interaction database in 2010. *Nucleic Acids Research,* 38, D525-31.

ASHBURNER, M., BALL, C. A., BLAKE, J. A., BOTSTEIN, D., BUTLER, H., CHERRY, J. M., DAVIS, A. P., DOLINSKI, K., DWIGHT, S. S., EPPIG, J. T., HARRIS, M. A., HILL, D. P., ISSEL-TARVER, L., KASARSKIS, A., LEWIS, S., MATESE, J. C., RICHARDSON, J. E., RINGWALD, M., RUBIN, G. M. & SHERLOCK, G. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet,* 25, 25-9.

AYTON, G. S., NOID, W. G. & VOTH, G. A. 2007. Multiscale modeling of biomolecular systems: in serial and in parallel. *Curr Opin Struct Biol,* 17, 192-8.

BADER, G. D., BETEL, D. & HOGUE, C. W. 2003. BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Res,* 31, 248-50.

BADER, G. D. & HOGUE, C. W. 2002. Analyzing yeast protein-protein interaction data obtained from different sources. *Nat Biotechnol,* 20, 991-7.

BALASUBRAMANIYAN, R., HULLERMEIER, E., WESKAMP, N. & KAMPER, J. 2005. Clustering of gene expression data using a local shape-based similarity measure. *Bioinformatics,* 21, 1069-77.

BANSAL, M., BELCASTRO, V., AMBESI-IMPIOMBATO, A. & DI BERNARDO, D. 2007. How to infer gene networks from expression profiles. *Mol Syst Biol,* 3, 78.

BARGMANN, B. O. & BIRNBAUM, K. D. 2010. Fluorescence activated cell sorting of plant protoplasts. *J Vis Exp.*

BARRELL, D., DIMMER, E., HUNTLEY, R. P., BINNS, D., O'DONOVAN, C. & APWEILER, R. 2009a. The GOA database in 2009--an integrated Gene Ontology Annotation resource. *Nucleic Acids Research,* 37, D396-403.

BARRETT, T., SUZEK, T. O., TROUP, D. B., WILHITE, S. E., NGAU, W. C., LEDOUX, P., RUDNEV, D., LASH, A. E., FUJIBUCHI, W. & EDGAR, R. 2005. NCBI GEO: mining millions of expression profiles--database and tools. *Nucleic Acids Res,* 33, D562-6.

BASTIAN, M., HEYMANN, S. & JACOMY, M. 2009. *Gephi: An Open Source Software for Exploring and Manipulating Networks.*

BAUER, S., GROSSMANN, S., VINGRON, M. & ROBINSON, P. N. 2008. Ontologizer 2.0—a multifunctional tool for GO term enrichment analysis and data exploration. *Bioinformatics,* 24, 1650-1651.

BAXEVANIS, A. D. 2000. The Molecular Biology Database Collection: an online compilation of relevant database resources. *Nucleic Acids Research,* 28, 1-7.

BHARDWAJ, N. & LU, H. 2005. Correlation between gene expression profiles and protein-protein interactions within and across genomes. *Bioinformatics,* 21, 2730-8.

BI, Y. M., ZHANG, Y., SIGNORELLI, T., ZHAO, R., ZHU, T. & ROTHSTEIN, S. 2005. Genetic analysis of Arabidopsis GATA transcription factor gene family reveals a nitrate-inducible member important for chlorophyll synthesis and glucose sensitivity. *Plant J,* 44, 680-92.

BODENREIDER, O. 2008. Ontologies and Data Integration in Biomedicine: Success Stories and Challenging Issues. *In:* BAIROCH, A., COHEN-BOULAKIA, S. & FROIDEVAUX, C. (eds.) *Data Integration in the Life Sciences.* Springer Berlin / Heidelberg.

BORK, P., JENSEN, L. J., VON MERING, C., RAMANI, A. K., LEE, I. & MARCOTTE, E. M. 2004. Protein interaction networks from yeast to human. *Current Opinions in Structural Biololgy,* 14, 292-9.

BOULESTEIX, A. L. 2010. Over-optimism in bioinformatics research. *Bioinformatics,* 26, 437-9.

BRADFORD, J. R., NEEDHAM, C. J., TEDDER, P., CARE, M. A., BULPITT, A. J. & WESTHEAD, D. R. 2010. GO-At: in silico prediction of gene function in Arabidopsis thaliana by combining heterogeneous data. *The Plant journal : for cell and molecular biology,* 61, 713-21.

BRAZMA, A., HINGAMP, P., QUACKENBUSH, J., SHERLOCK, G., SPELLMAN, P., STOECKERT, C., AACH, J., ANSORGE, W., BALL, C. A., CAUSTON, H. C., GAASTERLAND, T., GLENISSON, P., HOLSTEGE, F. C., KIM, I. F., MARKOWITZ, V., MATESE, J. C., PARKINSON, H., ROBINSON, A., SARKANS, U., SCHULZE-KREMER, S., STEWART, J., TAYLOR, R., VILO, J. & VINGRON,

M. 2001. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat Genet,* 29, 365-71.

BRAZMA, A., PARKINSON, H., SARKANS, U., SHOJATALAB, M., VILO, J., ABEYGUNAWARDENA, N., HOLLOWAY, E., KAPUSHESKY, M., KEMMEREN, P., LARA, G. G., OEZCIMEN, A., ROCCA-SERRA, P. & SANSONE, S. A. 2003. ArrayExpress--a public repository for microarray gene expression data at the EBI. *Nucleic Acids Res,* 31, 68-71.

BREITKREUTZ, B. J., STARK, C., REGULY, T., BOUCHER, L., BREITKREUTZ, A., LIVSTONE, M., OUGHTRED, R., LACKNER, D. H., BAHLER, J., WOOD, V., DOLINSKI, K. & TYERS, M. 2008. The BioGRID Interaction Database: 2008 update. *Nucleic Acids Res,* 36, D637-40.

BREITKREUTZ, B. J., STARK, C. & TYERS, M. 2003. Osprey: a network visualization system. *Genome biology,* 4, R22.

BRITTO, D. T. & KRONZUCKER, H. J. 2002. NH4+ toxicity in higher plants: a critical review. *Journal of Plant Physiology,* 159, 567-584.

BRITTO, D. T. & KRONZUCKER, H. J. 2006. Plant Nitrogen Transport and Its Regulation in Changing Soil Environments. *Journal of Crop Improvement,* 15, 1-23.

BROHEE, S. & VAN HELDEN, J. 2006. Evaluation of clustering algorithms for protein-protein interaction networks. *BMC bioinformatics,* 7, 488.

BU, D., ZHAO, Y., CAI, L., XUE, H., ZHU, X., LU, H., ZHANG, J., SUN, S., LING, L., ZHANG, N., LI, G. & CHEN, R. 2003. Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research,* 31, 2443-50.

BUTENKO, S., CHAOVALITWONGSE, W. A. & PARDALOS, P. M. 2009. *Clustering challenges in biological networks,* New Jersey ; London, World Scientific.

BUTTE, A. J. & KOHANE, I. S. 2000. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pac Symp Biocomput,* 418-29.

CAMPBELL, W. H. 1999. Nitrate Reductase Structure, Function and Regulation: Bridging the Gap between Biochemistry and Physiology. *Annu Rev Plant Physiol Plant Mol Biol,* 50, 277-303.

CANEVET, C. 2010. Ondex tutorial and user guide.

CASTAINGS, L., CAMARGO, A., POCHOLLE, D., GAUDON, V., TEXIER, Y., BOUTET-MERCEY, S., TACONNAT, L., RENOU, J. P., DANIEL-VEDELE, F., FERNANDEZ, E., MEYER, C. & KRAPP, A. 2009. The nodule inception-like protein 7 modulates nitrate sensing and metabolism in Arabidopsis. *Plant J,* 57, 426-35.

CEREZO, M., TILLARD, P., FILLEUR, S., MUNOS, S., DANIEL-VEDELE, F. & GOJON, A. 2001. Major alterations of the regulation of root NO(3)(-) uptake are associated with the mutation of Nrt2.1 and Nrt2.2 genes in Arabidopsis. *Plant Physiol,* 127, 262-71.

CHAGOYEN, M., CARAZO, J. M. & PASCUAL-MONTANO, A. 2008. Assessment of protein set coherence using functional annotations. *BMC Bioinformatics,* 9, 444.

CHEN, H., LAI, Z., SHI, J., XIAO, Y., CHEN, Z. & XU, X. 2010. Roles of arabidopsis WRKY18, WRKY40 and WRKY60 transcription factors in plant responses to abscisic acid and abiotic stress. *BMC plant biology,* 10, 281.

CHEN, H. & SHARP, B. M. 2004. Content-rich biological network constructed by mining PubMed abstracts. *BMC Bioinformatics,* 5, 147.

CHEN, X., CHEN, M. & NING, K. 2006a. BNArray: an R package for constructing gene regulatory networks from microarray data by using Bayesian network. *Bioinformatics,* 22, 2952-4.

CHEN, Y. M., FERRAR, T. S., LOHMEIER-VOGEL, E. M., MORRICE, N., MIZUNO, Y., BERENGER, B., NG, K. K., MUENCH, D. G. & MOORHEAD, G. B. 2006b. The PII signal transduction protein of Arabidopsis thaliana forms an arginine-regulated complex with plastid N-acetyl glutamate kinase. *J Biol Chem,* 281, 5726-33.

CHEREPINSKY, V., FENG, J., REJALI, M. & MISHRA, B. 2003. Shrinkage-based similarity metric for cluster analysis of microarray data. *Proc Natl Acad Sci U S A,* 100, 9668-73.

CHINI, A., FONSECA, S., FERNANDEZ, G., ADIE, B., CHICO, J. M., LORENZO, O., GARCIA-CASADO, G., LOPEZ-VIDRIERO, I., LOZANO, F. M., PONCE, M. R., MICOL, J. L. & SOLANO, R. 2007. The JAZ family of repressors is the missing link in jasmonate signalling. *Nature,* 448, 666-71.

CHIU, C. C., LIN, C. S., HSIA, A. P., SU, R. C., LIN, H. L. & TSAY, Y. F. 2004. Mutation of a nitrate transporter, AtNRT1:4, results in a reduced petiole nitrate content and altered leaf development. *Plant Cell Physiol,* 45, 1139-48.

CHOPIN, F., ORSEL, M., DORBE, M. F., CHARDON, F., TRUONG, H. N., MILLER, A. J., KRAPP, A. & DANIEL-VEDELE, F. 2007. The Arabidopsis ATNRT2.7 nitrate transporter controls nitrate content in seeds. *Plant Cell,* 19, 1590-602.

CHUANG, H. Y., LEE, E., LIU, Y. T., LEE, D. & IDEKER, T. 2007. Network-based classification of breast cancer metastasis. *Molecular Systems Biology,* 3, 140.

COCHRANE, G. R. & GALPERIN, M. Y. 2010. The 2010 Nucleic Acids Research Database Issue and online Database Collection: a community of data resources. *Nucleic Acids Research,* 38, D1-D4.

COCK, P. J., ANTAO, T., CHANG, J. T., CHAPMAN, B. A., COX, C. J., DALKE, A., FRIEDBERG, I., HAMELRYCK, T., KAUFF, F., WILCZYNSKI, B. & DE HOON, M. J. 2009. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics,* 25, 1422-3.

CONESA, A. & GOTZ, S. 2008. Blast2GO: A comprehensive suite for functional analysis in plant genomics. *International journal of plant genomics,* 2008, 619832.

CONESA, A., GOTZ, S., GARCIA-GOMEZ, J. M., TEROL, J., TALON, M. & ROBLES, M. 2005. Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics,* 21, 3674-6.

COOPER, H. D. & CLARKSON, D. T. 1989. Cycling of Amino-Nitrogen and other Nutrients between Shoots and Roots in Cereals—A Possible Mechanism Integrating Shoot and Root in the Regulation of Nutrient Uptake. *J Exp Bot,* 40, 753-762.

CORUZZI, G. M. 2003. Primary N-assimilation into Amino Acids in Arabidopsis. *In:* SOMERVILLE, C. & MEYEROWITZ, E. (eds.) *The Arabidopsis Book.* Rockville: American Society of Plant Physiologists.

CORUZZI, G. M. & ZHOU, L. 2001. Carbon and nitrogen sensing and signaling in plants: emerging 'matrix effects'. *Curr Opin Plant Biol,* 4, 247-53.

COTE, R. G., JONES, P., MARTENS, L., KERRIEN, S., REISINGER, F., LIN, Q., LEINONEN, R., APWEILER, R. & HERMJAKOB, H. 2007. The Protein Identifier Cross-Referencing (PICR) service: reconciling protein identifiers across multiple source databases. *BMC Bioinformatics,* 8, 401.

CRAIGON, D. J., JAMES, N., OKYERE, J., HIGGINS, J., JOTHAM, J. & MAY, S. 2004. NASCArrays: a repository for microarray data generated by NASC's transcriptomics service. *Nucleic Acids Res,* 32, D575-7.

CRAWFORD, N. M. & FORDE, B. G. 2002. Molecular and developmental biology of inorganic nitrogen nutrition. *In:* SOMERVILLE, C. & MEYEROWITZ, E. (eds.) *The Arabidopsis Book.* Rockville: American Society of Plant Physiologists.

CRETE, P., CABOCHE, M. & MEYER, C. 1997. Nitrite reductase expression is regulated at the post-transcriptional level by the nitrogen source in Nicotiana plumbaginifolia and Arabidopsis thaliana. *Plant J,* 11, 625-34.

CROWE, M. L., SERIZET, C., THAREAU, V., AUBOURG, S., ROUZE, P., HILSON, P., BEYNON, J., WEISBEEK, P., VAN HUMMELEN, P., REYMOND, P., PAZ-ARES, J., NIETFELD, W. & TRICK, M. 2003. CATMA: a complete Arabidopsis GST database. *Nucleic Acids Res,* 31, 156-8.

CURCIN, V. & GHANEM, M. Scientific workflow systems - can one size fit all? Biomedical Engineering Conference, 2008. CIBEC 2008. Cairo International, 2008. 1-9.

DAIGLE, B. J., JR. & ALTMAN, R. B. 2008. M-BISON: microarray-based integration of data sources using networks. *BMC Bioinformatics,* 9, 214.

DAUB, C. O., STEUER, R., SELBIG, J. & KLOSKA, S. 2004. Estimating mutual information using B-spline functions--an improved similarity measure for analysing gene expression data. *BMC Bioinformatics,* 5, 118.

DAVIDOV, E., HOLLAND, J., MARPLE, E. & NAYLOR, S. 2003. Advancing drug discovery through systems biology. *Drug Discov Today,* 8, 175-83.

DAVULURI, R. V., SUN, H., PALANISWAMY, S. K., MATTHEWS, N., MOLINA, C., KURTZ, M. & GROTEWOLD, E. 2003. AGRIS: Arabidopsis gene regulatory information server, an information resource of Arabidopsis cis-regulatory elements and transcription factors. *BMC Bioinformatics,* 4, 25.

DE ANGELI, A., MONACHELLO, D., EPHRITIKHINE, G., FRACHISSE, J. M., THOMINE, S., GAMBALE, F. & BARBIER-BRYGOO, H. 2006. The nitrate/proton antiporter AtCLCa mediates nitrate accumulation in plant vacuoles. *Nature,* 442, 939-42.

DE ANGELI, A., MONACHELLO, D., EPHRITIKHINE, G., FRACHISSE, J. M., THOMINE, S., GAMBALE, F. & BARBIER-BRYGOO, H. 2009. Review. CLC-mediated anion transport in plant cells. *Philos Trans R Soc Lond B Biol Sci,* 364, 195-201.

DE JONG, H., GEISELMANN, J., HERNANDEZ, C. & PAGE, M. 2003. Genetic Network Analyzer: qualitative simulation of genetic regulatory networks. *Bioinformatics,* 19, 336-44.

DEANE, C. M., SALWINSKI, L., XENARIOS, I. & EISENBERG, D. 2002. Protein interactions: two methods for assessment of the reliability of high throughput observations. *Mol Cell Proteomics,* 1, 349--356.

DEMIR, E., CARY, M. P., PALEY, S., FUKUDA, K., LEMER, C., VASTRIK, I., WU, G., D'EUSTACHIO, P., SCHAEFER, C., LUCIANO, J., SCHACHERER, F., MARTINEZ-FLORES, I., HU, Z., JIMENEZ-JACINTO, V., JOSHI-TOPE, G., KANDASAMY, K., LOPEZ-FUENTES, A. C., MI, H., PICHLER, E., RODCHENKOV, I., SPLENDIANI, A., TKACHEV, S., ZUCKER, J., GOPINATH, G., RAJASIMHA, H., RAMAKRISHNAN, R., SHAH, I., SYED, M., ANWAR, N., BABUR, O., BLINOV, M., BRAUNER, E., CORWIN, D., DONALDSON, S., GIBBONS, F., GOLDBERG, R., HORNBECK, P., LUNA, A., MURRAY-RUST, P., NEUMANN, E., REUBENACKER, O., SAMWALD, M., VAN IERSEL, M., WIMALARATNE, S., ALLEN, K., BRAUN, B., WHIRL-CARRILLO, M., CHEUNG, K. H., DAHLQUIST, K., FINNEY, A., GILLESPIE, M., GLASS, E., GONG, L., HAW, R., HONIG, M., HUBAUT, O., KANE, D., KRUPA, S., KUTMON, M., LEONARD, J., MARKS, D., MERBERG, D., PETRI, V., PICO, A., RAVENSCROFT, D., REN, L., SHAH, N., SUNSHINE, M., TANG, R., WHALEY, R., LETOVKSY, S., BUETOW, K. H., RZHETSKY, A., SCHACHTER, V., SOBRAL, B. S., DOGRUSOZ, U., MCWEENEY, S., ALADJEM, M., BIRNEY, E., COLLADO-VIDES, J., GOTO, S., HUCKA, M., LE NOVERE, N., MALTSEV, N., PANDEY, A., THOMAS, P., WINGENDER, E., KARP, P. D., SANDER, C. & BADER, G. D. 2010. The BioPAX community standard for pathway data sharing. *Nat Biotechnol,* 28, 935-42.

DENG, M., SUN, F. & CHEN, T. 2003. Assessment of the reliability of protein-protein interactions and protein function prediction. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing,* 140-51.

DEONIER, R. C., TAVARÉ, S. & WATERMAN, M. S. 2005. *Computational genome analysis : an introduction,* New York, N.Y., Springer.

DERISI, J. L., IYER, V. R. & BROWN, P. O. 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science,* 278, 680-6.

DI BERNARDO, D., THOMPSON, M. J., GARDNER, T. S., CHOBOT, S. E., EASTWOOD, E. L., WOJTOVICH, A. P., ELLIOTT, S. J., SCHAUS, S. E. & COLLINS, J. J. 2005. Chemogenomic profiling on a genomewide scale using reverse-engineered gene networks. *Nature Biotechnology,* 23, 377-383.

DIESTEL, R. 2005. *Graph theory,* Berlin ; [London], Springer.

DITTRICH, M. T., KLAU, G. W., ROSENWALD, A., DANDEKAR, T. &

MULLER, T. 2008. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics,* 24, i223-31.

DOLAN, L., JANMAAT, K., WILLEMSEN, V., LINSTEAD, P., POETHIG, S., ROBERTS, K. & SCHERES, B. 1993. Cellular organisation of the Arabidopsis thaliana root. *Development,* 119, 71-84.

DRAGHICI, S., SELLAMUTHU, S. & KHATRI, P. 2006. Babel's tower revisited: a universal resource for cross-referencing across annotation databases. *Bioinformatics,* 22, 2934-9.

EDWARDS, D. & BATLEY, J. 2004. Plant bioinformatics: from genome to phenome. *Trends in Biotechnology,* 22, 232-237.

EISEN, M. B., SPELLMAN, P. T., BROWN, P. O. & BOTSTEIN, D. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A,* 95, 14863-8.

ELO, L. L., JARVENPAA, H., ORESIC, M., LAHESMAA, R. & AITTOKALLIO, T. 2007. Systematic construction of gene coexpression networks with applications to human T helper cell differentiation process. *Bioinformatics,* 23, 2096-103.

ENRIGHT, A. J. & OUZOUNIS, C. A. 2001. BioLayout—an automatic graph layout algorithm for similarity visualization. *Bioinformatics,* 17, 853-854.

ENRIGHT, A. J., VAN DONGEN, S. & OUZOUNIS, C. A. 2002. An efficient algorithm for large-scale detection of protein families. *Nucleic acids research,* 30, 1575-84.

ERNST, J. & BAR-JOSEPH, Z. 2006. STEM: a tool for the analysis of short time series gene expression data. *BMC bioinformatics,* 7, 191.

EUBEL, H., JANSCH, L. & BRAUN, H. P. 2003. New insights into the respiratory chain of plant mitochondria. Supercomplexes and a unique composition of complex II. *Plant Physiol,* 133, 274-86.

FAN, S. C., LIN, C. S., HSU, P. K., LIN, S. H. & TSAY, Y. F. 2009. The Arabidopsis nitrate transporter NRT1.7, expressed in phloem, is responsible for source-to-sink remobilization of nitrate. *Plant Cell,* 21, 2750-61.

FAURE, J.-D., VINCENTZ, M., KRONENBERGER, J. & CABOCHE, M. 1991. Co-regulated expression of nitrate and nitrite reductases. *The Plant Journal,* 1, 107-113.

FERRARIO-MERY, S., BESIN, E., PICHON, O., MEYER, C. & HODGES, M. 2006. The regulatory PII protein controls arginine biosynthesis in Arabidopsis. *FEBS Lett,* 580, 2015-20.

FERRARIO-MERY, S., MEYER, C. & HODGES, M. 2008. Chloroplast nitrite uptake is enhanced in Arabidopsis PII mutants. *FEBS Lett,* 582, 1061-6.

FERRIER, T., MATUS, J. T., JIN, J. & RIECHMANN, J. L. 2010. Arabidopsis paves the way: genomic and network analyses in crops. *Curr Opin Biotechnol.*

FIELDS, S. & SONG, O. 1989. A novel genetic system to detect protein-protein interactions. *Nature,* 340, 245-6.

FLICEK, P., AMODE, M. R., BARRELL, D., BEAL, K., BRENT, S., CARVALHO-SILVA, D., CLAPHAM, P., COATES, G., FAIRLEY, S., FITZGERALD, S., GIL, L., GORDON, L., HENDRIX, M., HOURLIER, T., JOHNSON, N., KAHARI, A. K., KEEFE, D., KEENAN, S., KINSELLA, R., KOMOROWSKA, M., KOSCIELNY, G., KULESHA, E., LARSSON, P., LONGDEN, I., MCLAREN, W., MUFFATO, M., OVERDUIN, B., PIGNATELLI, M., PRITCHARD, B., RIAT, H. S., RITCHIE, G. R., RUFFIER, M., SCHUSTER, M., SOBRAL, D., TANG, Y. A., TAYLOR, K., TREVANION, S., VANDROVCOVA, J., WHITE, S., WILSON, M., WILDER, S. P., AKEN, B. L., BIRNEY, E., CUNNINGHAM, F., DUNHAM, I., DURBIN, R., FERNANDEZ-SUAREZ, X. M., HARROW, J., HERRERO, J., HUBBARD, T. J., PARKER, A., PROCTOR, G., SPUDICH, G., VOGEL, J., YATES, A., ZADISSA, A. & SEARLE, S. M. 2012. Ensembl 2012. *Nucleic acids research,* 40, D84-90.

FREEMAN, T. C., GOLDOVSKY, L., BROSCH, M., VAN DONGEN, S., MAZIERE, P., GROCOCK, R. J., FREILICH, S., THORNTON, J. & ENRIGHT, A. J. 2007. Construction, visualisation, and clustering of transcription networks from microarray expression data. *PLoS computational biology,* 3, 2032-42.

FRIEDBERG, I. 2006. Automated protein function prediction--the genomic challenge. *Briefings in bioinformatics,* 7, 225-42.

GABOW, A. P., LEACH, S. M., BAUMGARTNER, W. A., HUNTER, L. E. & GOLDBERG, D. S. 2008. Improving protein function prediction methods with integrated literature data. *BMC Bioinformatics,* 9, 198.

GAGNOT, S., TAMBY, J. P., MARTIN-MAGNIETTE, M. L., BITTON, F., TACONNAT, L., BALZERGUE, S., AUBOURG, S., RENOU, J. P., LECHARNY, A. & BRUNAUD, V. 2008. CATdb: a public access to Arabidopsis transcriptome data from the URGV-CATMA platform. *Nucleic Acids Res,* 36, D986-90.

GALPERIN, M. Y. 2006. The Molecular Biology Database Collection: 2006 update. *Nucleic Acids Research,* 34, D3-D5.

GAO, P., XIN, Z. & ZHENG, Z. L. 2008. The OSU1/QUA2/TSD2-encoded putative methyltransferase is a critical modulator of carbon and nitrogen nutrient balance response in Arabidopsis. *PLoS One,* 3, e1387.

GARDNER, T. S., DI BERNARDO, D., LORENZ, D. & COLLINS, J. J. 2003. Inferring genetic networks and identifying compound mode of action via expression profiling. *Science,* 301, 102-105.

GAUTIER, L., COPE, L., BOLSTAD, B. M. & IRIZARRY, R. A. 2004. affy--analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics,* 20, 307-15.

GAZZARRINI, S., LEJAY, L., GOJON, A., NINNEMANN, O., FROMMER, W. B. & VON WIREN, N. 1999. Three functional transporters for constitutive, diurnally regulated, and starvation-induced uptake of ammonium into Arabidopsis roots. *Plant Cell,* 11, 937-48.

GE, H., LIU, Z., CHURCH, G. M. & VIDAL, M. 2001. Correlation between transcriptome and interactome mapping data from Saccharomyces cerevisiae. *Nat Genet,* 29, 482-6.

GE, H., WALHOUT, A. J. & VIDAL, M. 2003. Integrating 'omic' information: a bridge between genomics and systems biology. *Trends Genet,* 19, 551-60.

GEELEN, D., LURIN, C., BOUCHEZ, D., FRACHISSE, J. M., LELIEVRE, F., COURTIAL, B., BARBIER-BRYGOO, H. & MAUREL, C. 2000.

Disruption of putative anion channel gene AtCLC-a in Arabidopsis suggests a role in the regulation of nitrate content. *Plant J,* 21, 259-67.

GEISLER-LEE, J., O'TOOLE, N., AMMAR, R., PROVART, N. J., MILLAR, A. H. & GEISLER, M. 2007. A predicted interactome for Arabidopsis. *Plant Physiol,* 145, 317-29.

GENTLEMAN, R. C., CAREY, V. J., BATES, D. M., BOLSTAD, B., DETTLING, M., DUDOIT, S., ELLIS, B., GAUTIER, L., GE, Y., GENTRY, J., HORNIK, K., HOTHORN, T., HUBER, W., IACUS, S., IRIZARRY, R., LEISCH, F., LI, C., MAECHLER, M., ROSSINI, A. J., SAWITZKI, G., SMITH, C., SMYTH, G., TIERNEY, L., YANG, J. Y. & ZHANG, J. 2004a. Bioconductor: open software development for computational biology and bioinformatics. *Genome biology,* 5, R80.

GIARDINE, B., RIEMER, C., HARDISON, R. C., BURHANS, R., ELNITSKI, L., SHAH, P., ZHANG, Y., BLANKENBERG, D., ALBERT, I., TAYLOR, J., MILLER, W., KENT, W. J. & NEKRUTENKO, A. 2005. Galaxy: a platform for interactive large-scale genome analysis. *Genome research,* 15, 1451-5.

GIFFORD, M. L., DEAN, A., GUTIERREZ, R. A., CORUZZI, G. M. & BIRNBAUM, K. D. 2008. Cell-specific nitrogen responses mediate developmental plasticity. *Proc Natl Acad Sci U S A,* 105, 803-8.

GIRIN, T., LEJAY, L., WIRTH, J., WIDIEZ, T., PALENCHAR, P. M., NAZOA, P., TOURAINE, B., GOJON, A. & LEPETIT, M. 2007. Identification of a 150 bp cis-acting element of the AtNRT2.1 promoter involved in the regulation of gene expression by the N and C status of the plant. *Plant Cell Environ,* 30, 1366-80.

GLASS, A. D., BRITTO, D. T., KAISER, B. N., KINGHORN, J. R., KRONZUCKER, H. J., KUMAR, A., OKAMOTO, M., RAWAT, S., SIDDIQI, M. Y., UNKLES, S. E. & VIDMAR, J. J. 2002. The regulation of nitrate and ammonium transport systems in plants. *J Exp Bot,* 53, 855-64.

GO. 2004. The OBO Flat File Format Specification Version 1.2. Available: http://www.geneontology.org/GO.format.obo-1_2.shtml.

GOBLE, C. & STEVENS, R. 2008. State of the nation in data integration for bioinformatics. *J Biomed Inform,* 41, 687-93.

GOBLE, C. A., BHAGAT, J., ALEKSEJEVS, S., CRUICKSHANK, D., MICHAELIDES, D., NEWMAN, D., BORKUM, M., BECHHOFER, S., ROOS, M., LI, P. & DE ROURE, D. 2010. myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic acids research,* 38, W677-82.

GOFFARD, N., GARCIA, V., IRAGNE, F., GROPPI, A. & DE DARUVAR, A. 2003. IPPRED: server for proteins interactions inference. *Bioinformatics,* 19, 903-4.

GOODSTEIN, D. M., SHU, S., HOWSON, R., NEUPANE, R., HAYES, R. D., FAZO, J., MITROS, T., DIRKS, W., HELLSTEN, U., PUTNAM, N. & ROKHSAR, D. S. 2012. Phytozome: a comparative platform for green plant genomics. *Nucleic acids research,* 40, D1178-86.

GRUBER, T. R. 1993. A translation approach to portable ontology specifications. *Knowl. Acquis.,* 5, 199-220.

GUEGUEN, L. 2005. Sarment: Python modules for HMM analysis and partitioning of sequences. *Bioinformatics,* 21, 3427-8.

GUO, A., HE, K., LIU, D., BAI, S., GU, X., WEI, L. & LUO, J. 2005. DATF: a database of Arabidopsis transcription factors. *Bioinformatics,* 21, 2568-9.

GUPTA, A., MARANAS, C. D. & ALBERT, R. 2006. Elucidation of directionality for co-expressed genes: predicting intra-operon termination sites. *Bioinformatics,* 22, 209-14.

GUTIERREZ, R. A., GIFFORD, M. L., POULTNEY, C., WANG, R., SHASHA, D. E., CORUZZI, G. M. & CRAWFORD, N. M. 2007a. Insights into the genomic nitrate response using genetics and the Sungear Software System. *J Exp Bot,* 58, 2359-67.

GUTIERREZ, R. A., LEJAY, L. V., DEAN, A., CHIAROMONTE, F., SHASHA, D. E. & CORUZZI, G. M. 2007b. Qualitative network models and genome-wide expression data define carbon/nitrogen-responsive molecular machines in Arabidopsis. *Genome Biol,* 8, R7.

GUTIERREZ, R. A., STOKES, T. L., THUM, K., XU, X., OBERTELLO, M., KATARI, M. S., TANURDZIC, M., DEAN, A., NERO, D. C., MCCLUNG, C. R. & CORUZZI, G. M. 2008. Systems approach identifies an organic nitrogen-responsive gene network that is regulated by the master clock control gene CCA1. *Proc Natl Acad Sci U S A,* 105, 4939-44.

GYGI, S. P., ROCHON, Y., FRANZA, B. R. & AEBERSOLD, R. 1999. Correlation between protein and mRNA abundance in yeast. *Mol Cell Biol,* 19, 1720-30.

HAGBERG, A., SWART, P. & S CHULT, D. 2008a. *Exploring network structure, dynamics, and function using networkx.*

HAGBERG, A. A., SCHULT, D. A. & SWART, P. J. 2008b. Exploring network structure, dynamics, and function using NetworkX. *Proceedings of the 7th Python in Science Conference (SciPy2008),* 11-15.

HAMDI-CHERIF, A. 2010. Intelligent Control and Biological Regulation For Bioinformatics. *INTERNATIONAL JOURNAL OF MATHEMATICAL MODELS AND METHODS IN APPLIED SCIENCES,* 4, 93-104.

HAN, J. D., BERTIN, N., HAO, T., GOLDBERG, D. S., BERRIZ, G. F., ZHANG, L. V., DUPUY, D., WALHOUT, A. J., CUSICK, M. E., ROTH, F. P. & VIDAL, M. 2004. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature,* 430, 88-93.

HARADA, H., KUROMORI, T., HIRAYAMA, T., SHINOZAKI, K. & LEIGH, R. A. 2004. Quantitative trait loci analysis of nitrate storage in Arabidopsis leading to an investigation of the contribution of the anion channel gene, AtCLC-c, to variation in nitrate levels. *J Exp Bot,* 55, 2005-14.

HASSANI-PAK, K., LEGAIE, R., CANEVET, C., VAN DEN BERG, H. A., MOORE, J. D. & RAWLINGS, C. J. 2010. Enhancing data integration with text analysis to find proteins implicated in plant stress response. *Journal of Integrative Bioinformatics,* 7.

HEAZLEWOOD, J. L., VERBOOM, R. E., TONTI-FILIPPINI, J., SMALL, I. & MILLAR, A. H. 2007. SUBA: the Arabidopsis Subcellular Database. *Nucleic Acids Res,* 35, D213-8.

HERMJAKOB, H., MONTECCHI-PALAZZI, L., BADER, G., WOJCIK, J., SALWINSKI, L., CEOL, A., MOORE, S., ORCHARD, S., SARKANS, U., VON MERING, C., ROECHERT, B., POUX, S., JUNG, E., MERSCH, H., KERSEY, P., LAPPE, M., LI, Y., ZENG, R., RANA, D., NIKOLSKI, M., HUSI, H., BRUN, C., SHANKER, K., GRANT, S. G., SANDER, C., BORK, P., ZHU, W., PANDEY, A., BRAZMA, A., JACQ, B., VIDAL, M., SHERMAN, D., LEGRAIN, P., CESARENI, G., XENARIOS, I., EISENBERG, D., STEIPE, B., HOGUE, C. & APWEILER, R. 2004. The HUPO PSI's molecular interaction format--a community standard for the representation of protein interaction data. *Nat Biotechnol,* 22, 177-83.

HERNANDEZ, T. & KAMBHAMPATI, S. 2004. Integration of biological sources: current systems and challenges ahead. *SIGMOD Rec.,* 33, 51-60.

HO, C. H., LIN, S. H., HU, H. C. & TSAY, Y. F. 2009. CHL1 functions as a nitrate sensor in plants. *Cell,* 138, 1184-94.

HO, C. H. & TSAY, Y. F. 2010. Nitrate, ammonium, and potassium sensing and signaling. *Curr Opin Plant Biol,* 13, 604-10.

HOEN, P. A., ARIYUREK, Y., THYGESEN, H. H., VREUGDENHIL, E., VOSSEN, R. H., DE MENEZES, R. X., BOER, J. M., VAN OMMEN, G. J. & DEN DUNNEN, J. T. 2008. Deep sequencing-based expression analysis shows major advances in robustness, resolution and inter-lab portability over five microarray platforms. *Nucleic Acids Res,* 36, e141.

HOWITT, S. M. & UDVARDI, M. K. 2000. Structure, function and regulation of ammonium transporters in plants. *Biochim Biophys Acta,* 1465, 152-70.

HSIEH, M. H., LAM, H. M., VAN DE LOO, F. J. & CORUZZI, G. 1998. A PII-like protein in Arabidopsis: putative role in nitrogen sensing. *Proc Natl Acad Sci U S A,* 95, 13965-70.

HU, H. C., WANG, Y. Y. & TSAY, Y. F. 2009. AtCIPK8, a CBL-interacting protein kinase, regulates the low-affinity phase of the primary nitrate response. *Plant J,* 57, 264-78.

HUANG, N. C., LIU, K. H., LO, H. J. & TSAY, Y. F. 1999. Cloning and functional characterization of an Arabidopsis nitrate transporter gene that encodes a constitutive component of low-affinity uptake. *Plant Cell,* 11, 1381-92.

HUANG, T. W., TIEN, A. C., HUANG, W. S., LEE, Y. C., PENG, C. L., TSENG, H. H., KAO, C. Y. & HUANG, C. Y. 2004. POINT: a database for the prediction of protein-protein interactions based on the orthologous interactome. *Bioinformatics,* 20, 3273-6.

HUCKA, M., FINNEY, A., SAURO, H. M., BOLOURI, H., DOYLE, J. C., KITANO, H., ARKIN, A. P., BORNSTEIN, B. J., BRAY, D., CORNISH-BOWDEN, A., CUELLAR, A. A., DRONOV, S., GILLES, E. D., GINKEL, M., GOR, V., GORYANIN, II, HEDLEY, W. J., HODGMAN, T. C., HOFMEYR, J. H., HUNTER, P. J., JUTY, N. S., KASBERGER, J. L., KREMLING, A., KUMMER, U., LE NOVERE, N., LOEW, L. M., LUCIO, D., MENDES, P., MINCH, E., MJOLSNESS, E. D., NAKAYAMA, Y., NELSON, M. R., NIELSEN, P. F., SAKURADA, T., SCHAFF, J. C., SHAPIRO, B. E., SHIMIZU, T. S., SPENCE, H. D., STELLING, J., TAKAHASHI, K., TOMITA, M., WAGNER, J. & WANG, J. 2003. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics,* 19, 524-31.

HULL, D., WOLSTENCROFT, K., STEVENS, R., GOBLE, C., POCOCK, M. R., LI, P. & OINN, T. 2006. Taverna: a tool for building and running workflows of services. *Nucleic acids research,* 34, W729-32.

HWANG, D., SMITH, J. J., LESLIE, D. M., WESTON, A. D., RUST, A. G., RAMSEY, S., DE ATAURI, P., SIEGEL, A. F., BOLOURI, H., AITCHISON, J. D. & HOOD, L. 2005. A data integration methodology for systems biology: experimental verification. *Proc Natl Acad Sci U S A,* 102, 17302-7.

IDEKER, T., OZIER, O., SCHWIKOWSKI, B. & SIEGEL, A. F. 2002. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics,* 18 Suppl 1, S233-40.

IDEKER, T., THORSSON, V., RANISH, J. A., CHRISTMAS, R., BUHLER, J., ENG, J. K., BUMGARNER, R., GOODLETT, D. R., AEBERSOLD, R. & HOOD, L. 2001. Integrated Genomic and Proteomic Analyses of a Systematically Perturbed Metabolic Network. *Science,* 292, 929-934.

IRIZARRY, R. A., BOLSTAD, B. M., COLLIN, F., COPE, L. M., HOBBS, B. & SPEED, T. P. 2003. Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res,* 31, e15.

ISHIYAMA, K., INOUE, E., WATANABE-TAKAHASHI, A., OBARA, M., YAMAYA, T. & TAKAHASHI, H. 2004. Kinetic properties and ammonium-dependent regulation of cytosolic isoenzymes of glutamine synthetase in Arabidopsis. *J Biol Chem,* 279, 16598-605.

JACKSON, L. E., BURGER, M. & CAVAGNARO, T. R. 2008. Roots, nitrogen transformations, and ecosystem services. *Annu Rev Plant Biol,* 59, 341-63.

JANSEN, R., GREENBAUM, D. & GERSTEIN, M. 2002. Relating whole-genome expression data with protein-protein interactions. *Genome Res,* 12, 37-46.

JELIZAROW, M., GUILLEMOT, V., TENENHAUS, A., STRIMMER, K. & BOULESTEIX, A. L. 2010. Over-optimism in bioinformatics: an illustration. *Bioinformatics,* 26, 1990-8.

JEN, C. H., MANFIELD, I. W., MICHALOPOULOS, I., PINNEY, J. W., WILLATS, W. G., GILMARTIN, P. M. & WESTHEAD, D. R. 2006. The Arabidopsis co-expression tool (ACT): a WWW-based tool and database for microarray-based gene expression analysis. *Plant J,* 46, 336-48.

JENSEN, L. J. & BORK, P. 2010. Ontologies in quantitative biology: a basis for comparison, integration, and discovery. *PLoS Biol,* 8, e1000374.

JENSEN, L. J., KUHN, M., STARK, M., CHAFFRON, S., CREEVEY, C., MULLER, J., DOERKS, T., JULIEN, P., ROTH, A., SIMONOVIC, M., BORK, P. & VON MERING, C. 2008. STRING 8-a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Research.*

JONES, C. E., BROWN, A. L. & BAUMANN, U. 2007. Estimating the annotation error rate of curated GO database sequence annotations. *BMC Bioinformatics,* 8, 170.

JOSLYN, C. A., MNISZEWSKI, S. M., FULMER, A. & HEATON, G. 2004.

The gene ontology categorizer. *Bioinformatics,* 20 Suppl 1, i169-77.

JOYCE, A. R. & PALSSON, B. O. 2006. The model organism as a system: integrating 'omics' data sets. *Nat Rev Mol Cell Biol,* 7, 198-210.

JUNKER, B. H. & SCHREIBER, F. 2008. *Analysis of biological networks,* Hoboken, N.J., Wiley ; Chichester : John Wiley [distributor].

KANG, J., MEHTA, S. & TURANO, F. J. 2004. The putative glutamate receptor 1.1 (AtGLR1.1) in Arabidopsis thaliana regulates abscisic acid biosynthesis and signaling to control development and water loss. *Plant Cell Physiol,* 45, 1380-9.

KAPUSHESKY, M., EMAM, I., HOLLOWAY, E., KURNOSOV, P., ZORIN, A., MALONE, J., RUSTICI, G., WILLIAMS, E., PARKINSON, H. & BRAZMA, A. 2010. Gene expression atlas at the European bioinformatics institute. *Nucleic acids research,* 38, D690-8.

KATARI, M. S., NOWICKI, S. D., ACEITUNO, F. F., NERO, D., KELFER, J., THOMPSON, L. P., CABELLO, J. M., DAVIDSON, R. S., GOLDBERG, A. P., SHASHA, D. E., CORUZZI, G. M. & GUTIERREZ, R. A. 2010. VirtualPlant: A Software Platform to Support Systems Biology Research. *Plant Physiol.,* 152, 500-515.

KAWASAKI, E. S. 2006. The end of the microarray Tower of Babel: will universal standards lead the way? *J Biomol Tech,* 17, 200-6.

KELL, D. B. & KNOWLES, J. D. 2006. The role of modeling in systems biology. *In:* SZALLASI, Z., STELLING, J. & PERIWAL, V. (eds.) *System modeling in cell biology : from concepts to nuts and bolts.* Cambridge, Mass. ; London: MIT Press.

KEMMEREN, P., VAN BERKUM, N. L., VILO, J., BIJMA, T., DONDERS, R., BRAZMA, A. & HOLSTEGE, F. C. 2002. Protein interaction verification and functional annotation by integrated analysis of genome-scale data. *Mol Cell,* 9, 1133-43.

KERRIEN, S., ALAM-FARUQUE, Y., ARANDA, B., BANCARZ, I., BRIDGE, A., DEROW, C., DIMMER, E., FEUERMANN, M., FRIEDRICHSEN, A., HUNTLEY, R., KOHLER, C., KHADAKE, J., LEROY, C., LIBAN, A., LIEFTINK, C., MONTECCHI-PALAZZI, L., ORCHARD, S., RISSE, J., ROBBE, K., ROECHERT, B., THORNEYCROFT, D., ZHANG, Y., APWEILER, R. & HERMJAKOB, H. 2007. IntAct--open source resource for molecular interaction data. *Nucleic Acids Res,* 35, D561--D565.

KHATRI, P. & DRAGHICI, S. 2005. Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics,* 21, 3587-95.

KIM, K., ZHANG, S., JIANG, K., CAI, L., LEE, I. B., FELDMAN, L. J. & HUANG, H. 2007. Measuring similarities between gene expression profiles through new data transformations. *BMC Bioinformatics,* 8, 29.

KITANO, H. 2000. Perspectives on systems biology. *New Gen. Comput.,* 18, 199-216.

KITANO, H. 2002. Systems biology: a brief overview. *Science,* 295, 1662-4.

KOEHLER, J., RAWLINGS, C., VERRIER, P., MITCHELL, R., SKUSA, A., RUEGG, A. & PHILIPPI, S. 2005. Linking experimental results, biological networks and sequence analysis methods using Ontologies and Generalised Data Structures. *In Silico Biol,* 5, 33-44.

KOHANE, I. S., KHO, A. T. & BUTTE, A. J. 2003. *Microarrays for an* *integrative genomics,* Cambridge, Mass. ; London, MIT Press.

KÖHLER, J. 2004. Integration of life science databases. *Drug Discovery Today: BIOSILICO,* 2, 61-69.

KOHLER, J., BAUMBACH, J., TAUBERT, J., SPECHT, M., SKUSA, A., RUEGG, A., RAWLINGS, C., VERRIER, P. & PHILIPPI, S. 2006. Graph-based analysis and visualization of experimental results with ONDEX. *Bioinformatics,* 22, 1383-90.

KOHLER, J., PHILIPPI, S. & LANGE, M. 2003. SEMEDA: ontology based semantic integration of biological databases. *Bioinformatics,* 19, 2420-7.

KORENBERG, M. J. 2007. *Microarray data analysis : methods and applications,* Totowa, N.J., Humana Press.

KOURMPETIS, Y. A., VAN DIJK, A. D., VAN HAM, R. C. & TER BRAAK, C. J. 2011. Genome-wide computational function prediction of Arabidopsis proteins by integration of multiple data sources. *Plant physiology,* 155, 271-81.

KRONZUCKER, H. J., SIDDIQI, M. Y., GLASS, A. D. & KIRK, G. J. 1999. Nitrate-ammonium synergism in rice. A subcellular flux analysis. *Plant Physiol,* 119, 1041-6.

LACROIX, Z. & CRITCHLOW, T. 2003. *Bioinformatics : managing scientific data,* San Francisco, Calif., Morgan Kaufmann ; Oxford : Elsevier Science.

LAM, H.-M., CHIU, J., HSIEH, M.-H., MEISEL, L., OLIVEIRA, I. C., SHIN, M. & CORUZZI, G. 1998. Glutamate-receptor genes in plants. *Nature,* 396, 125-126.

LASSILA, O., SWICK, R. R. & CONSORTIUM, W. W. A. W. 1998. Resource Description Framework (RDF) Model and Syntax Specification.

LEE, H. K., HSU, A. K., SAJDAK, J., QIN, J. & PAVLIDIS, P. 2004a. Coexpression analysis of human genes across many microarray data sets. *Genome Res,* 14, 1085-94.

LEE, I., AMBARU, B., THAKKAR, P., MARCOTTE, E. M. & RHEE, S. Y. 2010. Rational association of genes with traits using a genome-scale gene network for Arabidopsis thaliana. *Nature biotechnology,* 28, 149-56.

LEE, I., DATE, S. V., ADAI, A. T. & MARCOTTE, E. M. 2004b. A probabilistic functional network of yeast genes. *Science,* 306, 1555-8.

LEE, S. Y., LEE, D. Y. & KIM, T. Y. 2005. Systems biotechnology for strain improvement. *Trends Biotechnol,* 23, 349-58.

LEJAY, L., GANSEL, X., CEREZO, M., TILLARD, P., MULLER, C., KRAPP, A., VON WIREN, N., DANIEL-VEDELE, F. & GOJON, A. 2003. Regulation of root ion transporters by photosynthesis: functional importance and relation with hexokinase. *Plant Cell,* 15, 2218-32.

LEJAY, L., TILLARD, P., LEPETIT, M., OLIVE, F., FILLEUR, S., DANIEL-VEDELE, F. & GOJON, A. 1999. Molecular and functional regulation of two NO3- uptake systems by N- and C-status of Arabidopsis plants. *Plant J,* 18, 509-19.

LI, G. G. & WANG, Z. Z. 2009. Evaluation of similarity measures for gene expression data and their correspondent combined measures. *Interdiscip Sci,* 1, 72-80.

LI, J., LI, X., SU, H., CHEN, H. & GALBRAITH, D. W. 2006. A framework of integrating gene relations from heterogeneous data sources: an experiment on Arabidopsis thaliana. *Bioinformatics,* 22, 2037-43.

LI, L., STOECKERT, C. J., JR. & ROOS, D. S. 2003. OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome research,* 13, 2178-89.

LILLO, C., MEYER, C., LEA, U. S., PROVAN, F. & OLTEDAL, S. 2004. Mechanism and importance of post-translational regulation of nitrate reductase. *J Exp Bot,* 55, 1275-82.

LIN, M., SHEN, X. & CHEN, X. 2010. PAIR: the predicted Arabidopsis interactome resource. *Nucleic acids research.*

LIN, S. H., KUO, H. F., CANIVENC, G., LIN, C. S., LEPETIT, M., HSU, P. K., TILLARD, P., LIN, H. L., WANG, Y. Y., TSAI, C. B., GOJON, A. & TSAY, Y. F. 2008. Mutation of the Arabidopsis NRT1.5 nitrate transporter causes defective root-to-shoot nitrate transport. *Plant Cell,* 20, 2514-28.

LITTLE, D. Y., RAO, H., OLIVA, S., DANIEL-VEDELE, F., KRAPP, A. & MALAMY, J. E. 2005. The putative high-affinity nitrate transporter NRT2.1 represses lateral root initiation in response to nutritional cues. *Proc Natl Acad Sci U S A,* 102, 13693-8.

LIU, G., LORAINE, A. E., SHIGETA, R., CLINE, M., CHENG, J., VALMEEKAM, V., SUN, S., KULP, D. & SIANI-ROSE, M. A. 2003. NetAffx: Affymetrix probesets and annotations. *Nucleic acids research,* 31, 82-6.

LIU, K. H., HUANG, C. Y. & TSAY, Y. F. 1999. CHL1 is a dual-affinity nitrate transporter of Arabidopsis involved in multiple phases of nitrate uptake. *Plant Cell,* 11, 865-74.

LIU, K. H. & TSAY, Y. F. 2003. Switching between the two action modes of the dual-affinity nitrate transporter CHL1 by phosphorylation. *Embo J,* 22, 1005-13.

LORD, P. W., STEVENS, R. D., BRASS, A. & GOBLE, C. A. 2003. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics,* 19, 1275-83.

LOUIE, B., MORK, P., MARTIN-SANCHEZ, F., HALEVY, A. & TARCZY-HORNOCH, P. 2007. Data integration and genomic medicine. *J Biomed Inform,* 40, 5-16.

LU, L. J., XIA, Y., PACCANARO, A., YU, H. & GERSTEIN, M. 2005. Assessing the limits of genomic data integration for predicting protein networks. *Genome Res,* 15, 945-53.

LURIN, C., GEELEN, D., BARBIER-BRYGOO, H., GUERN, J. & MAUREL, C. 1996. Cloning and functional expression of a plant voltage-dependent chloride channel. *Plant Cell,* 8, 701-11.

LYSENKO, A., HINDLE, M. M., TAUBERT, J., SAQI, M. & RAWLINGS, C. J. 2009a. Data integration for plant genomics--exemplars from the integration of Arabidopsis thaliana databases. *Brief Bioinform,* 10, 676-93.

MANFIELD, I. W., JEN, C. H., PINNEY, J. W., MICHALOPOULOS, I., BRADFORD, J. R., GILMARTIN, P. M. & WESTHEAD, D. R. 2006a. Arabidopsis Co-expression Tool (ACT): web server tools for microarray-based gene expression analysis. *Nucleic acids research,* 34,

MAO, L., VAN HEMERT, J. L., DASH, S. & DICKERSON, J. A. 2009. Arabidopsis gene co-expression network and its functional modules. *BMC bioinformatics,* 10, 346.

MARCOTTE, E. M., PELLEGRINI, M., THOMPSON, M. J., YEATES, T. O. & EISENBERG, D. 1999. A combined algorithm for genome-wide prediction of protein function. *Nature,* 402, 83-6.

MARGOLIN, A. A., NEMENMAN, I., BASSO, K., WIGGINS, C., STOLOVITZKY, G., DALLA FAVERA, R. & CALIFANO, A. 2006. ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics,* 7 Suppl 1, S7.

MARKOWETZ, F. & SPANG, R. 2007. Inferring cellular networks--a review. *BMC Bioinformatics,* 8 Suppl 6, S5.

MARTINOIA, E., MAESHIMA, M. & NEUHAUS, H. E. 2007. Vacuolar transporters and their essential role in plant metabolism. *J Exp Bot,* 58, 83-102.

MCBRIDE, B. 2001. Jena: Implementing the RDF Model and Syntax Specification. *Semantic Web Workshop 2001.*

MEINKE, D. W., CHERRY, J. M., DEAN, C., ROUNSLEY, S. D. & KOORNNEEF, M. 1998. Arabidopsis thaliana: a model plant for genome analysis. *Science,* 282, 662, 679-82.

MENTZEN, W. I. & WURTELE, E. S. 2008a. Regulon organization of Arabidopsis. *BMC Plant Biology,* 8, 99.

MESAROVIC, M. D. E. 1968. *Systems Theory and Biology,* Berlin, Springer-Verlag.

METZKER, M. L. 2010. Sequencing technologies - the next generation. *Nature reviews. Genetics,* 11, 31-46.

MIFLIN, B. J. & HABASH, D. Z. 2002. The role of glutamine synthetase and glutamate dehydrogenase in nitrogen assimilation and possibilities for improvement in the nitrogen utilization of crops. *J Exp Bot,* 53, 979-87.

MILLER, A. & CRAMER, M. 2005. Root Nitrogen Acquisition and Assimilation. *Plant and Soil,* 274, 1-36.

MILLER, A. J. 2010. *Plant Nitrogen Nutrition and Transport,* John Wiley & Sons, Ltd.

MILLER, A. J., FAN, X., ORSEL, M., SMITH, S. J. & WELLS, D. M. 2007. Nitrate transport and signalling. *J. Exp. Bot.*, erm066.

MILLER, A. J., FAN, X., SHEN, Q. & SMITH, S. J. 2008. Amino acids and nitrate as signals for the regulation of nitrogen acquisition. *J Exp Bot,* 59, 111-9.

MILLER, A. J. & SMITH, S. J. 2008. Cytosolic nitrate ion homeostasis: could it have a role in sensing nitrogen status? *Ann Bot (Lond),* 101, 485-9.

MOCHIDA, K. & SHINOZAKI, K. 2010. Genomics and bioinformatics resources for crop improvement. *Plant Cell Physiol,* 51, 497-523.

MOORHEAD, G. B. & SMITH, C. S. 2003. Interpreting the plastid carbon, nitrogen, and energy status. A role for PII? *Plant Physiol,* 133, 492-8.

MOSTAFAVI, S. & MORRIS, Q. 2010. Fast integration of heterogeneous data sources for predicting gene function with limited annotation. *Bioinformatics,* 26, 1759-65.

MOSTAFAVI, S., RAY, D., WARDE-FARLEY, D., GROUIOS, C. &

MORRIS, Q. 2008. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome biology,* 9 Suppl 1, S4.

MUNOS, S., CAZETTES, C., FIZAMES, C., GAYMARD, F., TILLARD, P., LEPETIT, M., LEJAY, L. & GOJON, A. 2004. Transcript profiling in the chl1-5 mutant of Arabidopsis reveals a role of the nitrate transporter NRT1.1 in the regulation of another nitrate transporter, NRT2.1. *Plant Cell,* 16, 2433-47.

MUTWIL, M., USADEL, B., SCHUTTE, M., LORAINE, A., EBENHOH, O. & PERSSON, S. 2010. Assembly of an interactive correlation network for the Arabidopsis genome using a novel heuristic clustering algorithm. *Plant physiology,* 152, 29-43.

MYERS, C. L., ROBSON, D., WIBLE, A., HIBBS, M. A., CHIRIAC, C., THEESFELD, C. L., DOLINSKI, K. & TROYANSKAYA, O. G. 2005. Discovery of biological networks from diverse functional genomic data. *Genome Biology,* 6, R114.

MYERS, C. L. & TROYANSKAYA, O. G. 2007. Context-sensitive data integration and prediction of biological networks. *Bioinformatics,* 23, 2322-2330.

NG, A. Y., JORDAN, M. I. & WEISS, Y. 2002. On spectral clustering: analysis and an algorithm. *Neural Information Processing Systems,* 14, 849-856.

NGUYEN, V. A. & LIO, P. 2009. Measuring similarity between gene expression profiles: a Bayesian approach. *BMC Genomics,* 10 Suppl 3, S14.

OBAYASHI, T., HAYASHI, S., SAEKI, M., OHTA, H. & KINOSHITA, K. 2009. ATTED-II provides coexpressed gene networks for Arabidopsis. *Nucleic Acids Res,* 37, D987--D991.

OBAYASHI, T. & KINOSHITA, K. 2009. Rank of correlation coefficient as a comparable measure for biological significance of gene coexpression. *DNA Res,* 16, 249-60.

OBAYASHI, T. & KINOSHITA, K. 2011. COXPRESdb: a database to compare gene coexpression in seven model animals. *Nucleic Acids Res,* 39, D1016-22.

OBAYASHI, T., KINOSHITA, K., NAKAI, K., SHIBAOKA, M., HAYASHI, S., SAEKI, M., SHIBATA, D., SAITO, K. & OHTA, H. 2007a. ATTED-II: a database of co-expressed genes and cis elements for identifying co-regulated gene groups in Arabidopsis. *Nucleic Acids Res,* 35, D863-9.

OINN, T., ADDIS, M., FERRIS, J., MARVIN, D., SENGER, M., GREENWOOD, M., CARVER, T., GLOVER, K., POCOCK, M. R., WIPAT, A. & LI, P. 2004. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics,* 20, 3045-54.

OKAMOTO, M., VIDMAR, J. J. & GLASS, A. D. 2003. Regulation of NRT1 and NRT2 gene families of Arabidopsis thaliana: responses to nitrate provision. *Plant Cell Physiol,* 44, 304-17.

OLIVEIRA, I. C. & CORUZZI, G. M. 1999. Carbon and amino acids reciprocally modulate the expression of glutamine synthetase in Arabidopsis. *Plant Physiol,* 121, 301-10.

OLIVIER, B. G., ROHWER, J. M. & HOFMEYR, J. H. 2005. Modelling cellular systems with PySCeS. *Bioinformatics,* 21, 560-1.

OLSON, M. A., BOSTIC, K. & SELTZER, M. 1999. Berkeley DB. *Proceedings of the annual conference on USENIX Annual Technical Conference.* Monterey, California: USENIX Association.

ORSEL, M., CHOPIN, F., LELEU, O., SMITH, S. J., KRAPP, A., DANIEL-VEDELE, F. & MILLER, A. J. 2006. Characterization of a two-component high-affinity nitrate uptake system in Arabidopsis. Physiology and protein-protein interaction. *Plant Physiol,* 142, 1304-17.

ORSEL, M., KRAPP, A. & DANIEL-VEDELE, F. 2002. Analysis of the NRT2 nitrate transporter family in Arabidopsis. Structure and gene expression. *Plant Physiol,* 129, 886-96.

PALANISWAMY, S. K., JAMES, S., SUN, H., LAMB, R. S., DAVULURI, R. V. & GROTEWOLD, E. 2006. AGRIS and AtRegNet. a platform to link cis-regulatory elements and transcription factors into regulatory networks. *Plant Physiol,* 140, 818-29.

PANDEY, J., KOYUTURK, M. & GRAMA, A. 2010. Functional characterization and topological modularity of molecular interaction networks. *BMC Bioinformatics,* 11 Suppl 1, S35.

PANDEY, J., KOYUTURK, M., SUBRAMANIAM, S. & GRAMA, A. 2008. Functional coherence in domain interaction networks. *Bioinformatics,* 24, i28-34.

PEASE, A. C., SOLAS, D., SULLIVAN, E. J., CRONIN, M. T., HOLMES, C. P. & FODOR, S. P. 1994. Light-generated oligonucleotide arrays for rapid DNA sequence analysis. *Proc Natl Acad Sci U S A,* 91, 5022-6.

PENG, M., HANNAM, C., GU, H., BI, Y. M. & ROTHSTEIN, S. J. 2007. A mutation in NLA, which encodes a RING-type ubiquitin ligase, disrupts the adaptability of Arabidopsis to nitrogen limitation. *Plant J,* 50, 320-37.

PESCH, R., LYSENKO, A., HINDLE, M., HASSANI-PAK, K., THIELE, R., RAWLINGS, C., KOHLER, J. & TAUBERT, J. 2008. Graph-based sequence annotation using a data integration approach. *J Integr Bioinform,* 5.

PESQUITA, C., FARIA, D., BASTOS, H., FERREIRA, A. E., FALCAO, A. O. & COUTO, F. M. 2008. Metrics for GO based protein semantic similarity: a systematic evaluation. *BMC Bioinformatics,* 9 Suppl 5, S4.

PETERSEN, D., CHANDRAMOULI, G. V., GEOGHEGAN, J., HILBURN, J., PAARLBERG, J., KIM, C. H., MUNROE, D., GANGI, L., HAN, J., PURI, R., STAUDT, L., WEINSTEIN, J., BARRETT, J. C., GREEN, J. & KAWASAKI, E. S. 2005. Three microarray platforms: an analysis of their concordance in profiling gene expression. *BMC Genomics,* 6, 63.

PETERSON, P. 2009. F2PY: a tool for connecting Fortran and Python programs *International Journal of Computational Science and Engineering,* 4, 296-305.

PHIZICKY, E. M. & FIELDS, S. 1995. Protein-protein interactions: methods for detection and analysis. *Microbiol Rev,* 59, 94-123.

PIPER, M. D., DARAN-LAPUJADE, P., BRO, C., REGENBERG, B., KNUDSEN, S., NIELSEN, J. & PRONK, J. T. 2002. Reproducibility of oligonucleotide microarray transcriptome analyses. An interlaboratory comparison using chemostat cultures of Saccharomyces cerevisiae. *J Biol Chem,* 277, 37001-8.

PONOMARENKO, E. A., LISITSA, A. V., IL'GISONIS, E. V. & ARCHAKOV, A. I. 2010. [Construction of protein semantic networks using PubMed/MEDLINE]. *Molekuliarnaia biologiia,* 44, 152-61.

PRASAD, A. R. D. & PATEL, D. Lucene Search Engine - An Overview DRTC-HP International Workshop on Building Libraries using DSpace, 2005.

PRITCHARD, L., WHITE, J. A., BIRCH, P. R. & TOTH, I. K. 2006. GenomeDiagram: a python package for the visualization of large-scale genomic data. *Bioinformatics,* 22, 616-7.

PRUD'HOMMEAUX, E. & SEABORNE, A. 2006. SPARQL query language for RDF. *W3C working draft.* W3C.

PRUITT, K. D., TATUSOVA, T. & MAGLOTT, D. R. 2005. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res,* 33, D501-4.

QUACKENBUSH, J. 2004. Data standards for 'omic' science. *Nat Biotechnol,* 22, 613-4.

RAMU, C. 2001. SIR: a simple indexing and retrieval system for biological flat file databases. *Bioinformatics,* 17, 756-8.

RAWAT, S. R., SILIM, S. N., KRONZUCKER, H. J., SIDDIQI, M. Y. & GLASS, A. D. 1999. AtAMT1 gene expression and NH4+ uptake in roots of Arabidopsis thaliana: evidence for regulation by root glutamine levels. *Plant J,* 19, 143-52.

RE, M. & VALENTINI, G. 2010. Integration of heterogeneous data sources for gene function prediction using decision templates and ensembles of learning machines. *Neurocomputing,* 73, 1533-1537.

REDMAN, J. C., HAAS, B. J., TANIMOTO, G. & TOWN, C. D. 2004. Development and evaluation of an Arabidopsis whole genome Affymetrix probe array. *Plant J,* 38, 545-61.

REIMERS, M. 2010. Making informed choices about microarray data analysis. *PLoS Comput Biol,* 6, e1000786.

REISS, D. J., AVILA-CAMPILLO, I., THORSSON, V., SCHWIKOWSKI, B. & GALITSKI, T. 2005. Tools enabling the elucidation of molecular pathways active in human disease: application to Hepatitis C virus infection. *BMC bioinformatics,* 6, 154.

REMANS, T., NACRY, P., PERVENT, M., GIRIN, T., TILLARD, P., LEPETIT, M. & GOJON, A. 2006. A central role for the nitrate transporter NRT2.1 in the integrated morphological and physiological responses of the root system to nitrogen limitation in Arabidopsis. *Plant Physiol,* 140, 909-21.

RESNIK, P. 1999. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research,* 11, 95-130.

REVERTER, A. & CHAN, E. K. 2008. Combining partial correlation and an information theory approach to the reversed engineering of gene co-expression networks. *Bioinformatics,* 24, 2491-7.

RHEE, S. Y., BEAVIS, W., BERARDINI, T. Z., CHEN, G., DIXON, D., DOYLE, A., GARCIA-HERNANDEZ, M., HUALA, E., LANDER, G., MONTOYA, M., MILLER, N., MUELLER, L. A., MUNDODI, S., REISER, L., TACKLIND, J., WEEMS, D. C., WU, Y., XU, I., YOO, D., YOON, J. & ZHANG, P. 2003. The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. *Nucleic Acids Res,* 31, 224-8.

RIANO-PACHON, D. M., RUZICIC, S., DREYER, I. & MUELLER-ROEBER, B. 2007. PlnTFDB: an integrative plant transcription factor database. *BMC Bioinformatics,* 8, 42.

RICHARDS, A. J., MULLER, B., SHOTWELL, M., COWART, L. A., ROHRER, B. & LU, X. 2010. Assessing the functional coherence of gene sets with metrics based on the Gene Ontology graph. *Bioinformatics,* 26, i79-87.

ROCKETT, J. C. & HELLMANN, G. M. 2004. Confirming microarray data--is it really necessary? *Genomics,* 83, 541-9.

RUAN, J., DEAN, A. K. & ZHANG, W. 2010. A general co-expression network-based approach to gene expression analysis: comparison and applications. *BMC Syst Biol,* 4, 8.

RUBIN, G., TOHGE, T., MATSUDA, F., SAITO, K. & SCHEIBLE, W. R. 2009. Members of the LBD family of transcription factors repress anthocyanin synthesis and affect additional nitrogen responses in Arabidopsis. *Plant Cell,* 21, 3567-84.

RUTHS, T., RUTHS, D. & NAKHLEH, L. 2009. GS2: an efficiently computable measure of GO-based similarity of gene sets. *Bioinformatics,* 25, 1178-84.

SCHENA, M., SHALON, D., DAVIS, R. W. & BROWN, P. O. 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science,* 270, 467-70.

SCHOOF, H. & KARLOWSKI, W. M. 2003. Comparison of rice and Arabidopsis annotation. *Current opinion in plant biology,* 6, 106-12.

SCHULZE-KREMER, S. 2002. Ontologies for molecular biology and bioinformatics. *In Silico Biol,* 2, 179-193.

SEGONZAC, C., BOYER, J. C., IPOTESI, E., SZPONARSKI, W., TILLARD, P., TOURAINE, B., SOMMERER, N., ROSSIGNOL, M. & GIBRAT, R. 2007. Nitrate efflux at the root plasma membrane: identification of an Arabidopsis excretion transporter. *Plant Cell,* 19, 3760-77.

SHAH, N. H. & FEDOROFF, N. V. 2004. CLENCH: a program for calculating Cluster ENriCHment using the Gene Ontology. *Bioinformatics,* 20, 1196-1197.

SHANNON, C. E. 1997. The mathematical theory of communication. 1963. *MD Computing,* 14, 306-17.

SHANNON, P., MARKIEL, A., OZIER, O., BALIGA, N. S., WANG, J. T., RAMAGE, D., AMIN, N., SCHWIKOWSKI, B. & IDEKER, T. 2003a. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome research,* 13, 2498-504.

SHELDEN, M. C., DONG, B., DE BRUXELLES, G. L., TREVASKIS, B., WHELAN, J., RYAN, P. R., HOWITT, S. M. & UDVARDI, M. K. 2001. Arabidopsis ammonium transporters, AtAMT1;1 and AtAMT1;2, have different biochemical properties and functional roles. *Plant and Soil,* 231, 151-160.

SHI, Z., DEROW, C. K. & ZHANG, B. 2010. Co-expression module analysis reveals biological processes, genomic gain, and regulatory mechanisms associated with breast cancer progression. *BMC Syst Biol,* 4, 74.

SHINOZAKI, K. & SAKAKIBARA, H. 2009. Omics and bioinformatics: an essential toolbox for systems analyses of plant functions beyond 2010. *Plant Cell Physiol,* 50, 1177-80.

SHIPPY, R., SENDERA, T. J., LOCKNER, R., PALANIAPPAN, C., KAYSSER-KRANICH, T., WATTS, G. & ALSOBROOK, J. 2004. Performance evaluation of commercial short-oligonucleotide microarrays and the impact of noise in making cross-platform correlations. *BMC Genomics,* 5, 61.

SIGNORA, L., DE SMET, I., FOYER, C. H. & ZHANG, H. 2001. ABA plays a central role in mediating the regulatory effects of nitrate on root branching in Arabidopsis. *Plant J,* 28, 655-62.

SLEPCHENKO, B. M., SCHAFF, J. C., MACARA, I. & LOEW, L. M. 2003. Quantitative cell biology with the Virtual Cell. *Trends in Cell Biology,* 13, 570-576.

SMOOT, M. E., ONO, K., RUSCHEINSKI, J., WANG, P. L. & IDEKER, T. 2011. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics,* 27, 431-2.

SORANI, M. D., ORTMANN, W. A., BIERWAGEN, E. P. & BEHRENS, T. W. 2010. Clinical and biological data integration for biomarker discovery. *Drug Discov Today,* 15, 741-8.

SOUTHERN, J., PITT-FRANCIS, J., WHITELEY, J., STOKELEY, D., KOBASHI, H., NOBES, R., KADOOKA, Y. & GAVAGHAN, D. 2008. Multi-scale computational modelling in biology and physiology. *Prog Biophys Mol Biol,* 96, 60-89.

SPRINZAK, E., SATTATH, S. & MARGALIT, H. 2003. How reliable are experimental protein-protein interaction data? *J Mol Biol,* 327, 919-23.

STEUER, R., KURTHS, J., DAUB, C. O., WEISE, J. & SELBIG, J. 2002. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics,* 18 Suppl 2, S231-40.

STUART, J. M., SEGAL, E., KOLLER, D. & KIM, S. K. 2003. A gene-coexpression network for global discovery of conserved genetic modules. *Science,* 302, 249-55.

SUJANSKY, W. 2001. Heterogeneous database integration in biomedicine. *J Biomed Inform,* 34, 285-98.

SWARBRECK, D., WILKS, C., LAMESCH, P., BERARDINI, T. Z., GARCIA-HERNANDEZ, M., FOERSTER, H., LI, D., MEYER, T., MULLER, R., PLOETZ, L., RADENBAUGH, A., SINGH, S., SWING, V., TISSIER, C., ZHANG, P. & HUALA, E. 2008. The Arabidopsis Information Resource (TAIR): gene structure and function annotation. *Nucleic Acids Res,* 36, D1009--D1014.

SWARUP, R., KRAMER, E. M., PERRY, P., KNOX, K., LEYSER, H. M., HASELOFF, J., BEEMSTER, G. T., BHALERAO, R. & BENNETT, M. J. 2005. Root gravitropism requires lateral root cap and epidermal cells for transport and response to a mobile auxin signal. *Nat Cell Biol,* 7, 1057-65.

TAUBERT, J., HINDLE, M., LYSENKO, A., WEILE, J., K\, J., \#246, HLER & RAWLINGS, C. J. 2009. Linking Life Sciences Data Using Graph-

Based Mapping. *Proceedings of the 6th International Workshop on Data Integration in the Life Sciences.* Manchester, UK: Springer-Verlag.

TAUBERT, J., SIEREN, K. P., HINDLE, M., HOEKMAN, B., WINNENBURG, R., PHILIPPI, S., RAWLINGS, C. & KÖHLER, J. 2007. The OXL format for the exchange of integrated datasets. *Journal of Integrative Bioinformatics,* 4.

R DEVELOPMENT CORE TEAM, 2008. R: A Language and Environment for Statistical Computing *Vienna Austria R Foundation for Statistical Computing,* 1, ISBN 3-900051-07-0.

THE ARABIDOPSIS GENOME INITIATIVE 2000. Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. *Nature,* 408, 796-815.

TITARENKO, E., ROJO, E., LEON, J. & SANCHEZ-SERRANO, J. J. 1997. Jasmonic acid-dependent and -independent signaling pathways control wound-induced gene activation in Arabidopsis thaliana. *Plant Physiol,* 115, 817-26.

TONG, A. H., LESAGE, G., BADER, G. D., DING, H., XU, H., XIN, X., YOUNG, J., BERRIZ, G. F., BROST, R. L., CHANG, M., CHEN, Y., CHENG, X., CHUA, G., FRIESEN, H., GOLDBERG, D. S., HAYNES, J., HUMPHRIES, C., HE, G., HUSSEIN, S., KE, L., KROGAN, N., LI, Z., LEVINSON, J. N., LU, H., MENARD, P., MUNYANA, C., PARSONS, A. B., RYAN, O., TONIKIAN, R., ROBERTS, T., SDICU, A. M., SHAPIRO, J., SHEIKH, B., SUTER, B., WONG, S. L., ZHANG, L. V., ZHU, H., BURD, C. G., MUNRO, S., SANDER, C., RINE, J., GREENBLATT, J., PETER, M., BRETSCHER, A., BELL, G., ROTH, F. P., BROWN, G. W., ANDREWS, B., BUSSEY, H. & BOONE, C. 2004. Global mapping of the yeast genetic interaction network. *Science,* 303, 808-13.

TOURAINE, B. & GLASS, A. D. 1997. NO3- and ClO3- fluxes in the chl1-5 mutant of Arabidopsis thaliana. Does the CHL1-5 gene encode a low-affinity NO3- transporter? *Plant Physiol,* 114, 137-44.

TROYANSKAYA, O. G., DOLINSKI, K., OWEN, A. B., ALTMAN, R. B. & BOTSTEIN, D. 2003. A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae). *Proc Natl Acad Sci U S A,* 100, 8348-53.

TSAY, Y. F., SCHROEDER, J. I., FELDMANN, K. A. & CRAWFORD, N. M. 1993. The herbicide sensitivity gene CHL1 of Arabidopsis encodes a nitrate-inducible nitrate transporter. *Cell,* 72, 705-13.

ULITSKY, I. & SHAMIR, R. 2007. Identification of functional modules using network topology and high-throughput data. *BMC Systems Biology,* 1, 8.

UNIPROT CONSORTIUM 2010. The Universal Protein Resource (UniProt) in 2010. *Nucleic Acids Research,* 38, D142-8.

VAN BEL, M., PROOST, S., WISCHNITZKI, E., MOVAHEDI, S., SCHEERLINCK, C., VAN DE PEER, Y. & VANDEPOELE, K. 2012. Dissecting plant genomes with the PLAZA comparative genomics platform. *Plant physiology,* 158, 590-600.

VAN DONGEN, S. 2000. A cluster algorithm for graphs. National Research Institute for Mathematics and Computer Science in the http://www.cwi.nl/ftp/CWIreports/INS/INS-R9814.ps.gz

VANHOLME, R., VAN ACKER, R. & BOERJAN, W. 2010. Potential of Arabidopsis systems biology to advance the biofuel field. *Trends Biotechnol,* 28, 543-7.

VIDAL, E. A. & GUTIERREZ, R. A. 2008. A systems view of nitrogen nutrient and metabolite responses in Arabidopsis. *Curr Opin Plant Biol,* 11, 521-9.

VILELLA, A. J., SEVERIN, J., URETA-VIDAL, A., HENG, L., DURBIN, R. & BIRNEY, E. 2009. EnsemblCompara GeneTrees: Complete, duplication-aware phylogenetic trees in vertebrates. *Genome research,* 19, 327-35.

VON MERING, C., HUYNEN, M., JAEGGI, D., SCHMIDT, S., BORK, P. & SNEL, B. 2003. STRING: a database of predicted functional associations between proteins. *Nucleic Acids Res,* 31, 258-61.

VON MERING, C., JENSEN, L. J., SNEL, B., HOOPER, S. D., KRUPP, M., FOGLIERINI, M., JOUFFRE, N., HUYNEN, M. A. & BORK, P. 2005. STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Res,* 33, D433-7.

VON MERING, C., KRAUSE, R., SNEL, B., CORNELL, M., OLIVER, S. G., FIELDS, S. & BORK, P. 2002. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature,* 417, 399-403.

VON WIREN, N., GAZZARRINI, S., GOJON, A. & FROMMER, W. B. 2000. The molecular physiology of ammonium uptake and retrieval. *Curr Opin Plant Biol,* 3, 254-61.

WABNIK, K., HVIDSTEN, T. R., KEDZIERSKA, A., VAN LEENE, J., DE JAEGER, G., BEEMSTER, G. T., KOMOROWSKI, J. & KUIPER, M. T. 2009. Gene expression trends and protein features effectively complement each other in gene function prediction. *Bioinformatics,* 25, 322-30.

WALCH-LIU, P. & FORDE, B. G. 2008. Nitrate signalling mediated by the NRT1.1 nitrate transporter antagonises L-glutamate-induced changes in root architecture. *Plant J,* 54, 820-8.

WALCH-LIU, P., IVANOV, II, FILLEUR, S., GAN, Y., REMANS, T. & FORDE, B. G. 2006. Nitrogen regulation of root branching. *Ann Bot (Lond),* 97, 875-81.

WANG, J. Z., DU, Z., PAYATTAKOOL, R., YU, P. S. & CHEN, C. F. 2007. A new method to measure the semantic similarity of GO terms. *Bioinformatics,* 23, 1274-81.

WANG, R., XING, X., WANG, Y., TRAN, A. & CRAWFORD, N. M. 2009. A genetic screen for nitrate regulatory mutants captures the nitrate transporter gene NRT1.1. *Plant Physiol,* 151, 472-8.

WATSON-HAIGH, N. S., KADARMIDEEN, H. N. & REVERTER, A. 2010. PCIT: an R package for weighted gene co-expression networks based on partial correlation and information theory approaches. *Bioinformatics,* 26, 411-3.

WATTS, D. J. & STROGATZ, S. H. 1998. Collective dynamics of 'small-world' networks. *Nature,* 393, 440-2.

WEI, H., PERSSON, S., MEHTA, T., SRINIVASASAINAGENDRA, V., CHEN, L., PAGE, G. P., SOMERVILLE, C. & LORAINE, A. 2006.

Transcriptional coordination of the metabolic network in Arabidopsis. *Plant Physiology,* 142, 762-74.

WEIGEL, D. & MOTT, R. 2009. The 1001 genomes project for Arabidopsis thaliana. *Genome Biol,* 10, 107.

WEILE, J., POCOCK, M., COCKELL, S. J., LORD, P., DEWAR, J. M., HOLSTEIN, E. M., WILKINSON, D., LYDALL, D., HALLINAN, J. & WIPAT, A. 2011. Customizable views on semantically integrated networks for systems biology. *Bioinformatics,* 27, 1299-306.

WESTON, J., ELISSEEFF, A., ZHOU, D., LESLIE, C. S. & NOBLE, W. S. 2004. Protein ranking: from local to global structure in the protein similarity network. *Proceedings of the National Academy of Sciences, USA,* 101, 6559-63.

WHITE, S., MADADHAIN, J., FISHER, D. & BOEY, Y. B. 2004. *Jung - Java universal network/graph framework* [Online]. Available: http://jung.sourceforge.net/index.html

WILCZYNSKI, B. & DOJER, N. 2009. BNFinder: exact and efficient method for learning Bayesian networks. *Bioinformatics,* 25, 286-7.

WILKINSON, J. Q. & CRAWFORD, N. M. 1993. Identification and characterization of a chlorate-resistant mutant of Arabidopsis thaliana with mutations in both nitrate reductase structural genes NIA1 and NIA2. *Molecular and General Genetics MGG,* 239, 289-297.

WILLIAMS, L. & MILLER, A. 2001. Transporters Responsible for the Uptake and Partitioning of Nitrogenous Solutes. *Annu Rev Plant Physiol Plant Mol Biol,* 52, 659-688.

WOLFE, C. J., KOHANE, I. S. & BUTTE, A. J. 2005. Systematic survey reveals general applicability of "guilt-by-association" within gene coexpression networks. *BMC Bioinformatics,* 6, 227.

WU, C. T. 2006. *An introduction to object-oriented programming with Java,* Boston, McGraw-Hill Higher Education.

WU, L. F., HUGHES, T. R., DAVIERWALA, A. P., ROBINSON, M. D., STOUGHTON, R. & ALTSCHULER, S. J. 2002. Large-scale prediction of Saccharomyces cerevisiae gene function using overlapping transcriptional clusters. *Nat Genet,* 31, 255-65.

XU, T., GU, J., ZHOU, Y. & DU, L. 2009. Improving detection of differentially expressed gene sets by applying cluster enrichment analysis to Gene Ontology. *BMC Bioinformatics,* 10, 240.

YANAGISAWA, S., AKIYAMA, A., KISAKA, H., UCHIMIYA, H. & MIWA, T. 2004. Metabolic engineering with Dofl transcription factor in plants: Improved nitrogen assimilation and growth under low-nitrogen conditions. *Proc Natl Acad Sci U S A,* 101, 7833-8.

YONA, G., DIRKS, W., RAHMAN, S. & LIN, D. M. 2006. Effective similarity measures for expression profiles. *Bioinformatics,* 22, 1616-22.

YU, H., JANSEN, R., STOLOVITZKY, G. & GERSTEIN, M. 2007. Total ancestry measure: quantifying the similarity in tree-like classification, with genomic applications. *Bioinformatics,* 23, 2163-73.

YU, J., SMITH, V. A., WANG, P. P., HARTEMINK, A. J. & JARVIS, E. D. 2004. Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics,* 20, 3594-603.

YUAN, L., LOQUE, D., KOJIMA, S., RAUCH, S., ISHIYAMA, K., INOUE, E., TAKAHASHI, H. & VON WIREN, N. 2007. The organization of high-affinity ammonium uptake in Arabidopsis roots depends on the spatial arrangement and biochemical properties of AMT1-type transporters. *Plant Cell,* 19, 2636-52.

ZHANG, B. & HORVATH, S. 2005. A general framework for weighted gene co-expression network analysis. *Stat Appl Genet Mol Biol,* 4, Article17.

ZHANG, H. & FORDE, B. G. 1998. An Arabidopsis MADS box gene that controls nutrient-induced changes in root architecture. *Science,* 279, 407-9.

ZHANG, H. & FORDE, B. G. 2000. Regulation of Arabidopsis root development by nitrate availability. *J Exp Bot,* 51, 51-9.

ZHANG, H., JENNINGS, A., BARLOW, P. W. & FORDE, B. G. 1999. Dual pathways for regulation of root branching by nitrate. *Proc Natl Acad Sci U S A,* 96, 6529-34.

ZHAO, J., MILES, A., KLYNE, G. & SHOTTON, D. 2009. Linked data and provenance in biological data webs. *Brief Bioinform,* 10, 139-52.

ZHENG, B. & LU, X. 2007. Novel metrics for evaluating the functional coherence of protein groups via protein semantic network. *Genome Biology,* 8, R153.

ZHENG, Q. & WANG, X. J. 2008. GOEAST: a web-based software toolkit for Gene Ontology enrichment analysis. *Nucleic acids research,* 36, W358-63.

ZHENG, X., LIU, T., YANG, Z. & WANG, J. 2010. Large cliques in Arabidopsis gene coexpression network and motif discovery. *J Plant Physiol.*

ZHOU, Q. & LIU, J. S. 2008. Extracting sequence features to predict protein-DNA interactions: a comparative study. *Nucleic Acids Res,* 36, 4137-48.

ZHOU, X., KAO, M. C. & WONG, W. H. 2002. Transitive functional annotation by shortest-path analysis of gene expression data. *Proc Natl Acad Sci U S A,* 99, 12783-8.

ZIFARELLI, G. & PUSCH, M. 2010. CLC transport proteins in plants. *FEBS Lett,* 584, 2122-7.