WIBL: Workbench for Integrative Biological Learning

Victor Lesk[1], Jan Taubert[2], Chris Rawlings[2], Stuart Dunbar[3], Stephen Muggleton[1]

[1]Centre for Integrative Systems Biology, Imperial College, London, UK
[2]Biomathematics & Bioinformatics Department, Rothamsted Research, Harpenden, UK
[3]Syngenta, Jealott's hill International Research Centre, Bracknell, Berks, RG426EY, UK

Summary

The construction of integrated datasets from potentially hundreds of sources with bespoke formats, and their subsequent visualization and analysis, is a recurring challenge in systems biology. We present WIBL, a visualization and model development environment initially geared towards logic-based modelling of biological systems using integrated datasets. WIBL combines data integration, visualisation and modelling in a single portal-based workbench providing a comprehensive solution for interdisciplinary systems biology projects.

1 Introduction

Whole-system modelling in biology, which has become widespread in the high-throughput era, depends on integrating data and methods with diverse provenance: experimental and bioinformatic, public and private.

Current tools for data integration [1] and for the visualization of biological network data [2] differ in sophistication and emphasis. Ondex [3] for example has generalized analysis capabilities combined with data integration, whereas Arcadia [4] specializes in metabolic network visualisation, *etc*. Tools may also complement a particular theoretical basis for modelling, *e.g.* the Logic Programming/Machine learning used in our work, or other popular frameworks such as Bayesian networks or Flux-balance equations.

Our portal-based workbench, WIBL, is a feature-rich visualization and model development environment initially geared towards logic-based modelling of biological systems using integrated datasets. WIBL stands for "Workbench for Integrative Biological Learning" and is presented for the first time here. WIBL supports biological discovery by integrating the methodologies for the disciplines used in systems biology, namely theory formation, hypothesis testing, experimentation, analysis and theory revision. This is the oft-described virtuous circle for successful systems biology.

The diversity of tools and resources for the integration and analysis of experimental data can be overwhelming for biologists. WIBL adopts an intelligent workflows policy, *i.e.* providing unitary task chains as they become required by the project, rather than attempting to cater for arbitrary workflows as *e.g.* Taverna [5] does.

WIBL allows interaction with the data itself, access to a full model development and execution environment, and analysis of results. WIBL's philosophy is interdisciplinary, and it has been designed to provide diverse collaborators with a common, complete and scientifically literate representation of the project state, accommodating the different comfort levels and expectations of all parties. Written in Adobe Flash, the WIBL portal runs in a web browser, is quick to start up and requires no installation.

WIBL was initially developed to support a Campylobacter functional glycomics project within CISBIC, the BBSRC/EPSRC funded Centre for Integrative Systems Biology at Imperial College specializing in host-pathogen interactions. The modelling framework for the Campylobacter project was Muggleton and Bryant’s Progol implementation of the ILP machine learning technique [6] (described below). Progol was used to generate hypotheses for genotype-glycotype relationships in Campylobacter, based on directed experimentation conducted by the Wren group at The London School of Hygiene and Tropical medicine, and on information publicly available in KEGG [7] and BioCyc [8].

WIBL currently interacts directly with three external applications:

Progol (http://www.doc.ic.ac.uk/~shm/progol.html) is an implementation of ILP which combines Inverse Entailment with general-to-specific search through a refinement graph. Inverse Entailment is used with mode declarations to derive the most-specific clause within the mode language which entails a given example. Progol's search is efficient and has a provable guarantee of returning a solution having the maximum "compression" in the search-space. Progol deals with noisy data by using the compression measure to trade off the description of errors against the hypothesis description length. Progol allows arbitrary Prolog programs as background knowledge and arbitrary definite clauses as examples.

Arcadia (http://arcadiapathways.sourceforge.net/) is a lightweight standalone viewer for metabolic networks, with the layout improvement feature of representing a frequently occurring metabolite such as water or ATP as multiple disconnected copies. Arcadia supports existing standards such as SBML [9] and SBGN [10], and WIBL provides SBML export mediated by Ondex.

Ondex (http://www.ondex.org) is a data integration platform, which enables diverse biological data sets to be linked, integrated, transformed, visualised and analysed. Ondex’s flexible core data structures have the ability to integrate structured databases with unstructured sources such as biological sequence data and free text. WIBL invokes Ondex to translate raw data into Prolog syntax, and to export the project state to SBML. WIBL can also export into Ondex native XML format (OXL [11]), either for use in the standalone Ondex application or for viewing in the applet-based Ondex explorer.

WIBL has now been deployed to the tomato ripening and predictive toxicology projects in Syngenta’s Imperial University Innovation Centre (http://www3.imperial.ac.uk/syngenta-uic), and to a more theoretical project in the Muggleton group at Imperial College to develop an Inductive Logic Programme capable of efficiently filling multiple gaps in a logical model.

2 Design and Implementation

WIBL integrates several bioinformatics and machine learning tools running on the same server as depicted in Figure 1. In the first step background knowledge is assembled from experimental data and/or a variety of biological databases using the data integration functionality of Ondex (Figure 1(1)). The resulting data sets are exported into Prolog format. The Prolog files can be loaded into the flash-based WIBL front-end (Figure 1(2)) and fed into Progol for hypothesis generation using Logic Programming/Machine learning. Results from the machine learning step in Progol are again made available in Prolog (Figure 1(3)) and can be presented to the user as graphical annotations to the network visualisation in WIBL’s flash-based front-end. These networks can be converted into either OXL or SBML format using the integrated Ondex back-end functionality (Figure 1(4)). OXL and SBML can be loaded into the Ondex front-end and into Arcadia respectively for visual inspection and further analysis.

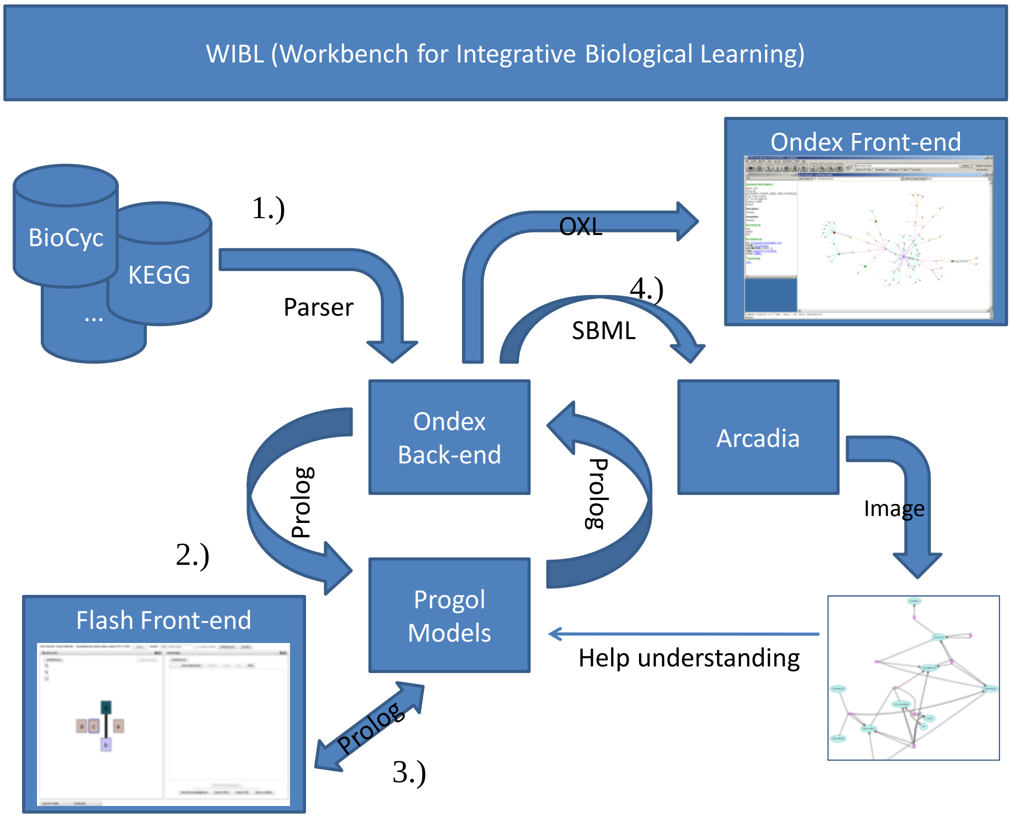
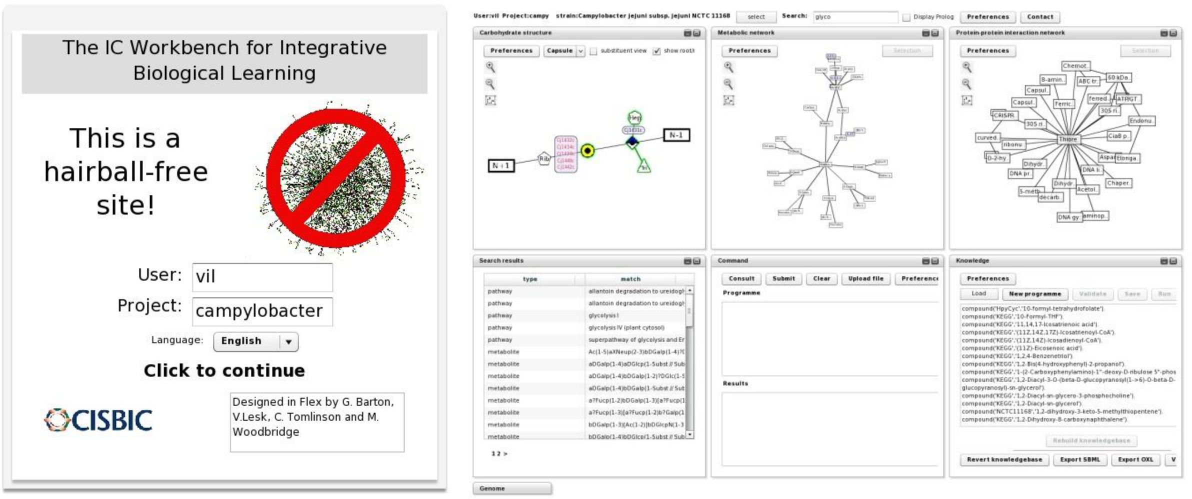


Figure 1 Interplay of the different components of WIBL: 1.) Assembly of background knowledge using Ondex back-end, 2.) Input of background knowledge as Prolog into Progol models, 3.) Rendering and modification of models using WIBLs flash-based front-end, 4.) Export of models into OXL or SBML and applicability to external tools like Arcadia pathways and Ondex front-end.

The WIBL front-end interface is flash-based, written in Adobe MXML and ActionScript and developed using the FlexBuilder application. After login, the user sees a screen divided into self-positioning frames, each of which manages a visualization or modelling task (see Figure 2). Above the frames is a toolbar with global information, preferences and a search box. The basic frames common to all projects using WIBL so far are the metabolic network, PPI network, search results browser, Prolog editor, and model execution frames. Most projects have additional bespoke frames, *e.g.* a glycan structure visualization frame and a gene locus browser frame for the Campylobacter glycomics project.



The IC Workbench for Integrative Biological Learning

This is a hairball-free site!

User: vil

Project: campylobacter

Language: English

Click to continue

Designed in Flex by G. Barton, V.Lesk, C. Tomlinson and M. Woodbridge

User:vil Project:campy strain:Campylobacter jejuni subsp. jejuni NCTC 11168

Search: glyco

□ Display Prolog

☐ substituent view ☑ show root/r

type	match
pathway	allantoin degradation to ureidogl
pathway	allantoin degradation to ureidogl
pathway	glycolysis I
pathway	glycolysis IV (plant cytosol)
pathway	superpathway of glycolysis and Er
metabolite	Ac(1-5)aXNeup(2-3)bDGalp(1-4)?(
metabolite	aDGalp(1-4)aDGlcp(1-Subst // Sub
metabolite	aDGalp(1-4)bDGalp(1-2)?DGlc(1-5
metabolite	aDGalp(1-4)bDGalp(1-Subst // Sub
metabolite	a?Fucp(1-2)bDGalp(1-3)[a?Fucp(1
metabolite	a?Fucp(1-3)[a?Fucp(1-2)b?Galp(1
metabolite	bDGalp(1-3)[Ac(1-2)]bDGlcpN(1-3
metabolite	bDGalp(1-4)bDGlcp(1-Subst // Sub

```
compound('HpyCyc','10-formyl-tetrahydrofolate').
compound('KEGG','10-Formyl-THF').
compound('KEGG','11,14,17-Icosatrienoic acid').
compound('KEGG','(11Z,14Z,17Z)-Icosatrienoyl-CoA').
compound('KEGG','(11Z,14Z)-Icosadienoyl-CoA').
compound('KEGG','(11Z)-Eicosenoic acid').
compound('KEGG','1,2,4-Benzenetriol').
compound('KEGG','1,2-Bis(4-hydroxyphenyl)-2-propanol').
compound('KEGG','1-(2-Carboxyphenylamino)-1''-deoxy-D-ribulose 5''-phos
compound('KEGG','1,2-Diacyl-3-O-(beta-D-glucopyranosyl(1->6)-O-beta-D-
glucopyranosyl)-sn-glycerol').
compound('KEGG','1,2-Diacyl-sn-glycero-3-phosphocholine').
compound('KEGG','1,2-Diacyl-sn-glycerol').
compound('NCTC11168','1,2-dihydroxy-3-keto-5-methylthiopentene').
compound('KEGG','1,2-Dihydroxy-8-carboxynaphthalene').
```

Figure 2 WIBL splash screen and dashboard

A given project is stored as a set of Prolog files on a central server. The network visualizer frames accept GraphML, which is served to them by a WIBL fast-cgi webservice component written in c++. This fast-cgi webservice also handles plaintext and regexp search requests. Model development and execution tasks are handled by bash cgi scripts which call Progol and return results to the portal user. OXL and SBML export requests are similarly forwarded to a server-side Ondex instance using short Ondex workflows generated and run by a WIBL webservice.

Networks are laid out using the Visualizer component developed by Kap IT (http://lab.kapit.fr/display/visualizer/Visualizer). WIBL's visualization conventions are designed to meet the expectations of biologists; KEGG chemical diagrams are used to represent metabolites where possible, and PDB [12] images represent proteins; glycan residues are drawn according to the Consortium for Functional Glycomics protocol.

The portal can be used to modify the project, either by direct Prolog text editing, by graphical editing in a network visualization frame, or by assimilating the result of a modelling experiment. In whichever case, a webservice applies the change to the appropriate underlying Prolog files on the server and invokes 'make' to reconstruct some server-based auxiliary files. Then the contents of all frames are refreshed to reflect the new project state.

3 Application cases

WIBL has so far been used to support four systems biology projects: on the glycomics of the *Campylobacter jejuni* pathogen, on the ripening process in tomato, on predictive toxicology, and on the ecological perturbation caused by genetically modified crops.

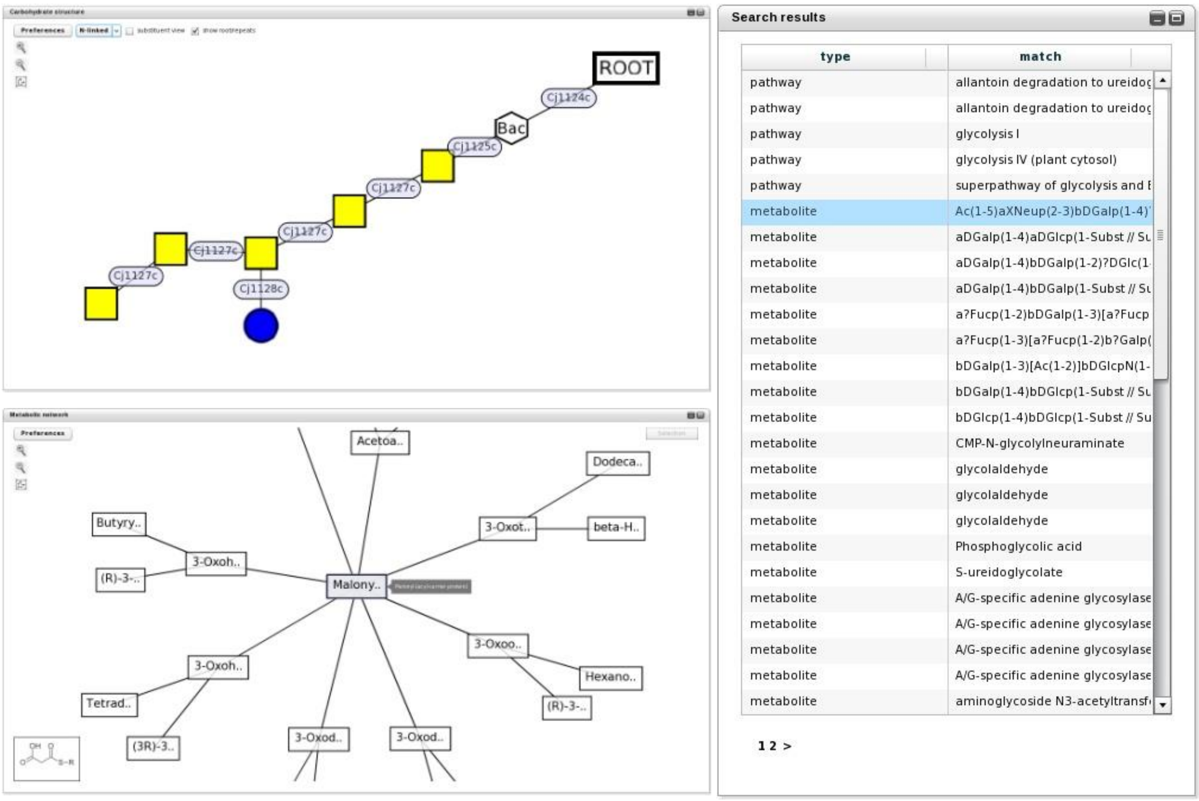
3.1 The glycomic basis for campylobacteriosis (with CISBIC)

WIBL was first developed to support CISBIC's Campylobacter glycomics subproject. The aim of the glycomics subproject was to develop models that would allow prediction of the effect of pathogen genome changes on the glycosylations of surface molecules. These surface glycans are involved in triggering the innate immune response of the host, and are hypothesised to be the agents which, through molecular mimicry, cause rare but serious Campylobacter-associated neuropathies. The pathogen's purpose in expressing and regulating these glycans is to survive in contrasting environments, notably on unhygienically stored or prepared food and within the host gut.

The modelling background knowledge comprised metabolic, proteomic and genomic data from KEGG and BioCyc, as well as the sequences of the four main *C. jejuni* glycans (capsule polysaccharide, lipo-oligosaccharide, N-linked and O-linked) and their corresponding genetic loci. The model used observations of glycan sequence for single knockouts of all the glycan locus genes were used to learn the specific function of each gene in glycan synthesis.

WIBL's initial modules (see Figure 3) were a metabolic network visualizer, a knowledgebase search service, and a Prolog execution environment; these modules have been incorporated into almost all subsequent projects. WIBL also provides modules to display glycan structures, the protein interaction network and the relevant sections of the pathogen genome.

During the Campylobacter project development focused on WIBL's distributed back-end, including its seeding by the Ondex data integration framework, as well as its user management system and internationalization features.



Search results

type	match
pathway	allantoin degradation to ureidoc
pathway	allantoin degradation to ureidoc
pathway	glycolysis I
pathway	glycolysis IV (plant cytosol)
pathway	superpathway of glycolysis and E
metabolite	Ac(1-5)aXNeup(2-3)bDGalp(1-4)'
metabolite	aDGalp(1-4)aDGlcp(1-Subst // Su
metabolite	aDGalp(1-4)bDGalp(1-2)?DGlc(1-
metabolite	aDGalp(1-4)bDGalp(1-Subst // Su
metabolite	a?Fucp(1-2)bDGalp(1-3)[a?Fucp
metabolite	a?Fucp(1-3)[a?Fucp(1-2)b?Galp(
metabolite	bDGalp(1-3)[Ac(1-2)]bDGlcpN(1-
metabolite	bDGalp(1-4)bDGlcp(1-Subst // Su
metabolite	bDGlcp(1-4)bDGlcp(1-Subst // Su
metabolite	CMP-N-glycolylneuraminate
metabolite	glycolaldehyde
metabolite	glycolaldehyde
metabolite	glycolaldehyde
metabolite	Phosphoglycolic acid
metabolite	S-ureidoglycolate
metabolite	A/G-specific adenine glycosylase
metabolite	A/G-specific adenine glycosylase
metabolite	A/G-specific adenine glycosylase
metabolite	A/G-specific adenine glycosylase
metabolite	aminoglycoside N3-acetyltransf

1 2 >

Figure 3 Glycan structure, metabolic network and search result modules

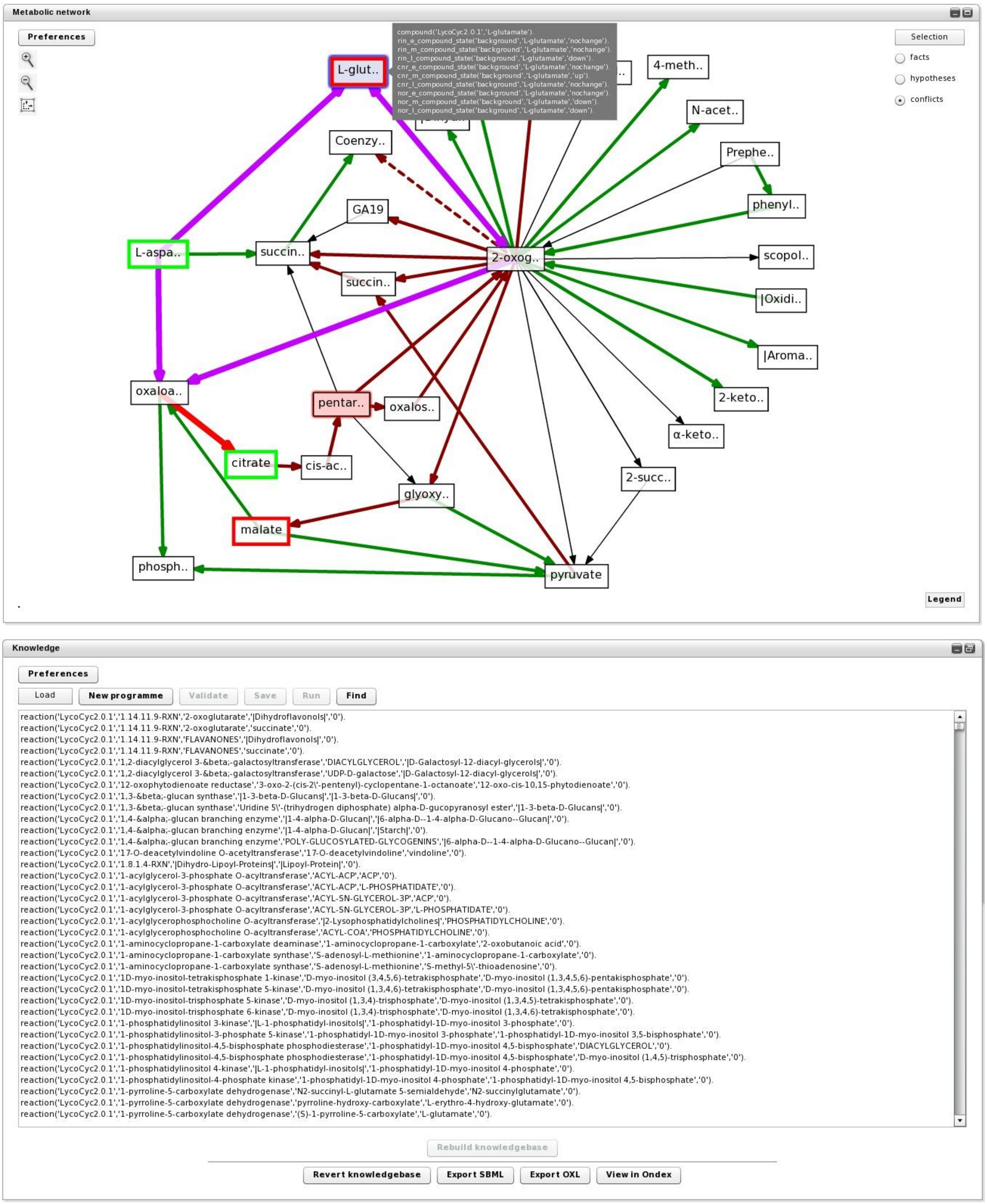
3.2 The ripening process in tomato, and predictive toxicology (in collaboration with Syngenta AG)

WIBL has been used in support of two projects run by Syngenta UIC (University Innovation Centre).

The first, the tomato ripening project has the goal of building a predictive model of tomato ripening and fruit quality (colour, texture, flavour) which would allow Syngenta to identify the main genetic control points in ripening. Syngenta hopes then to exploit this model through conventional breeding. The modelling background knowledge includes data from the databases LycoCyc [13] and KEGG [7] dealing with information on gene function and metabolism, and learns from gene expression and metabolomic data of genetic knockouts at four experimental time points.

The second project is in the area of predictive toxicology, and aims to build a predictive model of xenobiotic toxicity which will allow a more efficient screening of novel crop protection chemistries. The model uses KEGG data on metabolism to learn from observations in a pilot study of liver toxicity induced by phenobarbital in rat which is being used to develop a model of the differences between rodent and human toxicology.

The tomato and toxicology projects have seen the development of most of WIBL's inter-module communication features, as well as the introduction of an editor module allowing direct modifications to the project knowledgebase and models (see Figure 4). Knowledgebase export to Arcadia, OXL format and Ondex explorer was introduced for these projects.



```
compound('LycoCyc2.0.1','L-glutamate').
rin_e_compound_state('background','L-glutamate','nochange').
rin_m_compound_state('background','L-glutamate','nochange').
rin_l_compound_state('background','L-glutamate','down').
cnr_e_compound_state('background','L-glutamate','nochange').
cnr_m_compound_state('background','L-glutamate','up').
cnr_l_compound_state('background','L-glutamate','nochange').
nor_e_compound_state('background','L-glutamate','nochange').
nor_m_compound_state('background','L-glutamate','down').
nor_l_compound_state('background','L-glutamate','down').
```

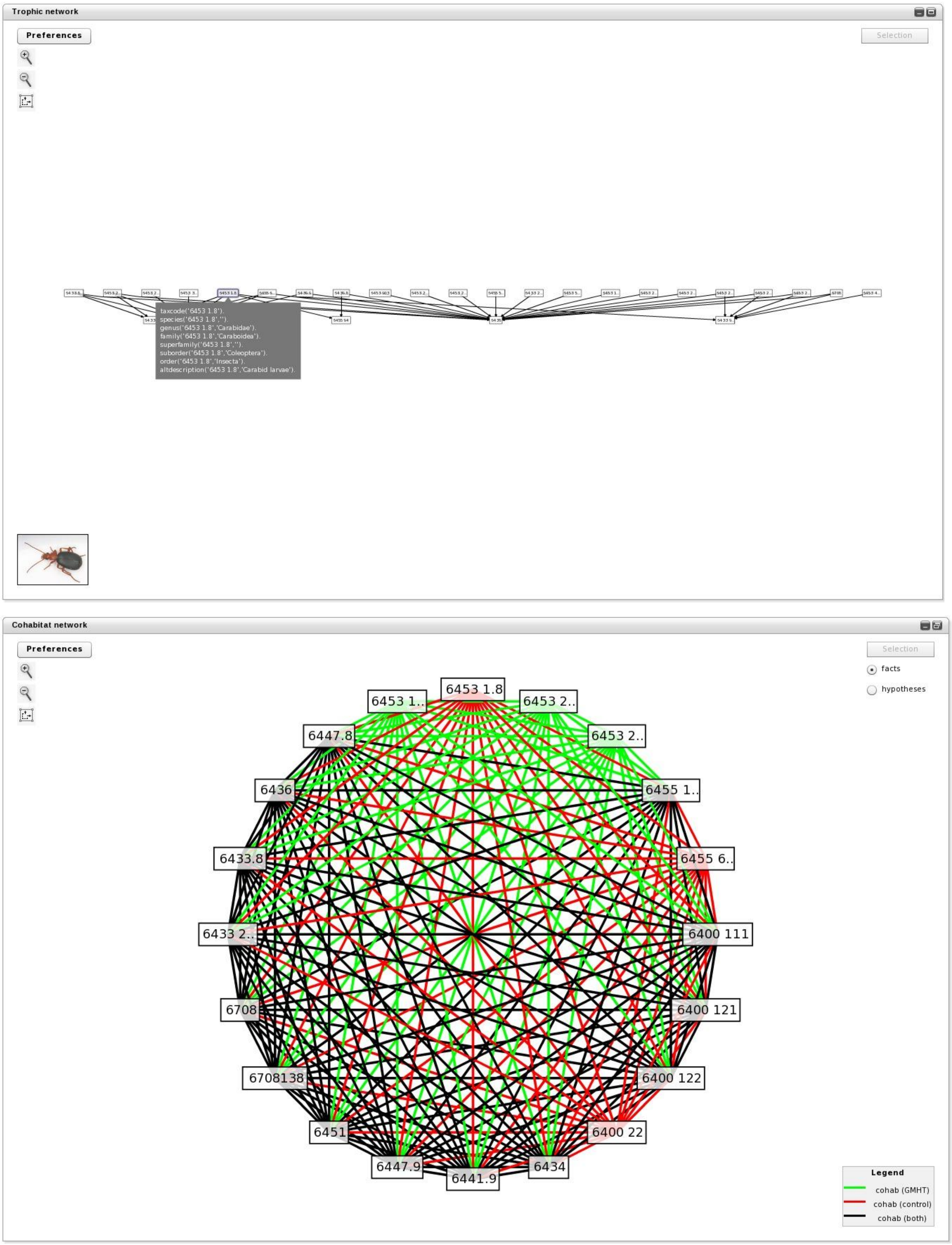
```
reaction('LycoCyc2.0.1','1.14.11.9-RXN','2-oxoglutarate','|Dihydroflavonols|','0').
reaction('LycoCyc2.0.1','1.14.11.9-RXN','2-oxoglutarate','succinate','0').
reaction('LycoCyc2.0.1','1.14.11.9-RXN','FLAVANONES','|Dihydroflavonols|','0').
reaction('LycoCyc2.0.1','1.14.11.9-RXN','FLAVANONES','succinate','0').
reaction('LycoCyc2.0.1','1,2-diacylglycerol 3-&beta;-galactosyltransferase','DIACYLGLYCEROL','|D-Galactosyl-12-diacyl-glycerols|','0').
reaction('LycoCyc2.0.1','1,2-diacylglycerol 3-&beta;-galactosyltransferase','UDP-D-galactose','|D-Galactosyl-12-diacyl-glycerols|','0').
reaction('LycoCyc2.0.1','12-oxophytodienoate reductase','3-oxo-2-(cis-2\'-pentenyl)-cyclopentane-1-octanoate','12-oxo-cis-10,15-phytodienoate','0').
reaction('LycoCyc2.0.1','1,3-&beta;-glucan synthase','|1-3-beta-D-Glucans|','|1-3-beta-D-Glucans|','0').
reaction('LycoCyc2.0.1','1,3-&beta;-glucan synthase','Uridine 5\'-(trihydrogen diphosphate) alpha-D-gucopyranosyl ester','|1-3-beta-D-Glucans|','0').
reaction('LycoCyc2.0.1','1,4-&alpha;-glucan branching enzyme','|1-4-alpha-D-Glucan|','|6-alpha-D--1-4-alpha-D-Glucano--Glucan|','0').
reaction('LycoCyc2.0.1','1,4-&alpha;-glucan branching enzyme','|1-4-alpha-D-Glucan|','|Starch|','0').
reaction('LycoCyc2.0.1','1,4-&alpha;-glucan branching enzyme','POLY-GLUCOSYLATED-GLYCOGENINS','|6-alpha-D--1-4-alpha-D-Glucano--Glucan|','0').
reaction('LycoCyc2.0.1','17-O-deacetylvindoline O-acetyltransferase','17-O-deacetylvindoline','vindoline','0').
reaction('LycoCyc2.0.1','1.8.1.4-RXN','|Dihydro-Lipoyl-Proteins|','|Lipoyl-Protein|','0').
reaction('LycoCyc2.0.1','1-acylglycerol-3-phosphate O-acyltransferase','ACYL-ACP','ACP','0').
reaction('LycoCyc2.0.1','1-acylglycerol-3-phosphate O-acyltransferase','ACYL-ACP','L-PHOSPHATIDATE','0').
reaction('LycoCyc2.0.1','1-acylglycerol-3-phosphate O-acyltransferase','ACYL-SN-GLYCEROL-3P','ACP','0').
reaction('LycoCyc2.0.1','1-acylglycerol-3-phosphate O-acyltransferase','ACYL-SN-GLYCEROL-3P','L-PHOSPHATIDATE','0').
reaction('LycoCyc2.0.1','1-acylglycerophosphocholine O-acyltransferase','|2-Lysophosphatidylcholines|','PHOSPHATIDYLCHOLINE','0').
reaction('LycoCyc2.0.1','1-acylglycerophosphocholine O-acyltransferase','ACYL-COA','PHOSPHATIDYLCHOLINE','0').
reaction('LycoCyc2.0.1','1-aminocyclopropane-1-carboxylate deaminase','1-aminocyclopropane-1-carboxylate','2-oxobutanoic acid','0').
reaction('LycoCyc2.0.1','1-aminocyclopropane-1-carboxylate synthase','S-adenosyl-L-methionine','1-aminocyclopropane-1-carboxylate','0').
reaction('LycoCyc2.0.1','1-aminocyclopropane-1-carboxylate synthase','S-adenosyl-L-methionine','S-methyl-5\'-thioadenosine','0').
reaction('LycoCyc2.0.1','1D-myo-inositol-tetrakisphosphate 1-kinase','D-myo-inositol (3,4,5,6)-tetrakisphosphate','D-myo-inositol (1,3,4,5,6)-pentakisphosphate','0').
reaction('LycoCyc2.0.1','1D-myo-inositol-tetrakisphosphate 5-kinase','D-myo-inositol (1,3,4,6)-tetrakisphosphate','D-myo-inositol (1,3,4,5,6)-pentakisphosphate','0').
reaction('LycoCyc2.0.1','1D-myo-inositol-trisphosphate 5-kinase','D-myo-inositol (1,3,4)-trisphosphate','D-myo-inositol (1,3,4,5)-tetrakisphosphate','0').
reaction('LycoCyc2.0.1','1D-myo-inositol-trisphosphate 6-kinase','D-myo-inositol (1,3,4)-trisphosphate','D-myo-inositol (1,3,4,6)-tetrakisphosphate','0').
reaction('LycoCyc2.0.1','1-phosphatidylinositol 3-kinase','|L-1-phosphatidyl-inositols|','1-phosphatidyl-1D-myo-inositol 3-phosphate','0').
reaction('LycoCyc2.0.1','1-phosphatidylinositol-3-phosphate 5-kinase','1-phosphatidyl-1D-myo-inositol 3-phosphate','1-phosphatidyl-1D-myo-inositol 3,5-bisphosphate','0').
reaction('LycoCyc2.0.1','1-phosphatidylinositol-4,5-bisphosphate phosphodiesterase','1-phosphatidyl-1D-myo-inositol 4,5-bisphosphate','DIACYLGLYCEROL','0').
reaction('LycoCyc2.0.1','1-phosphatidylinositol-4,5-bisphosphate phosphodiesterase','1-phosphatidyl-1D-myo-inositol 4,5-bisphosphate','D-myo-inositol (1,4,5)-trisphosphate','0').
reaction('LycoCyc2.0.1','1-phosphatidylinositol 4-kinase','|L-1-phosphatidyl-inositols|','1-phosphatidyl-1D-myo-inositol 4-phosphate','0').
reaction('LycoCyc2.0.1','1-phosphatidylinositol-4-phosphate kinase','1-phosphatidyl-1D-myo-inositol 4-phosphate','1-phosphatidyl-1D-myo-inositol 4,5-bisphosphate','0').
reaction('LycoCyc2.0.1','1-pyrroline-5-carboxylate dehydrogenase','N2-succinyl-L-glutamate 5-semialdehyde','N2-succinylglutamate','0').
reaction('LycoCyc2.0.1','1-pyrroline-5-carboxylate dehydrogenase','pyrroline-hydroxy-carboxylate','L-erythro-4-hydroxy-glutamate','0').
reaction('LycoCyc2.0.1','1-pyrroline-5-carboxylate dehydrogenase','(S)-1-pyrroline-5-carboxylate','L-glutamate','0').
```

Figure 4 Top: Visualization of enzyme hypotheses; Bottom: Editor module with export features to SBML and OXL

3.3 Effect of GM crops on the local ecosystem (with Rothamsted Research and Syngenta AG)

The most recent project, also run within the Syngenta UIC, relates to ecosystem modelling. This aim of this project is to model how the ecosystem of a field is perturbed when the field is sown with genetically modified crops. The modelling background knowledge comprises observations of trophic networks and organism co-localisation in GM-sown and control fields taken from the Defra-funded farm scale evaluations of genetically modified herbicide-tolerant crops [14, 15].

Two new WIBL modules have been developed to support these types of information: one supports browsing of food-web networks, and the other visualizes organism co-localization information (see Figure 5).



```
taxcode('6453 1.8').
species('6453 1.8','').
genus('6453 1.8','Carabidae').
family('6453 1.8','Caraboidea').
superfamily('6453 1.8','').
suborder('6453 1.8','Coleoptera').
order('6453 1.8','Insecta').
altdescription('6453 1.8','Carabid larvae').
```

Figure 5 Co-habitat and food web visualization modules

Each project has successively expanded WIBL's repertoire of functional modules through its specific requirements, and grown WIBL's user base in both industry and academia. Some WIBL modules like the Prolog Editor are common to all projects, whereas other functionality is project specific, e.g. the glycan structure visualisation for the campylobacter glycomics project or the co-habitation network for ecosystem modelling project.

4 Discussion

WIBL’s main design principle is interdisciplinarity. It combines data integration, visualisation and modelling (initially machine learning) into a web portal suitable for interdisciplinary biological problems in general, and systems biology in particular. WIBL’s intuitive interface and visualization protocols are designed to give modellers access to a biologically relevant perspective on a project, and to help biologists follow the steps that modellers are taking. WIBL enables any user to modify any aspects of the background knowledge, non-ground rules and produced hypotheses.

While WIBL’s role as a collaborative workbench demands that it integrate some external tools, it also has significant native features. It provides the first dedicated graphical user interface for machine learning development, as well as a versatile engine for generating GraphML from a Prolog fact file. Nonetheless WIBL seamlessly outsources functions to external applications where appropriate, specifically data integration tasks (to Ondex) and machine learning experiments (to Progol). WIBL’s compatibility with SBML allows the analysis of learned results and integrated models using the Arcadia pathway visualizer, as well as over 200 other SBML compatible tools [see http://sbml.org/SBML_Software_Guide].

The four application cases presented highlight the challenge of providing the user of the system with the most intuitive representation of the respective models. Whereas some general functionality like simple text editors or search functionality can be reused across several projects, other visualisation methods have to be developed for each separate project. WIBL’s modular approach facilitates feature reuse and adaptation, minimizing development time for each new application case.

5 Availability

An interactive demonstration of WIBL is located at http://www.sbg.bio.ic.ac.uk/wibl/, with all source code available on request from the corresponding author.

Arcadia is an Open Source project at http://arcadiapathways.sourceforge.net. Ondex is also developed as an Open Source project and can be found at http://www.ondex.org. Progol is available free of charge for academic research and teaching at http://www.doc.ic.ac.uk/~shm/progol.html. The flash visualisation component of WIBL uses proprietary libraries and the Adobe Flex Builder which require a license for non-academic use. Acknowledgements

CISBIC is jointly funded by the BBSRC and EPSRC. Rothamsted Research is supported by the BBSRC. The work on Ondex has been funded by BBSRC Grants BBS/B/13640 & BB/F006039/1.

References

[1] C. Goble and R. Stevens, "State of the nation in data integration for bioinformatics," *J Biomed Inform*, vol. 41, pp. 687-93, Oct 2008.

[2] G. A. Pavlopoulos, A. L. Wegener, and R. Schneider, "A survey of visualization tools for biological network analysis," *BioData Min*, vol. 1, p. 12, 2008.

[3] J. Köhler, J. Baumbach, J. Taubert, M. Specht, A. Skusa, A. Ruegg, C. Rawlings, P. Verrier, and S. Philippi, "Graph-based analysis and visualization of experimental results with ONDEX," *Bioinformatics,* vol. 22, pp. 1383-90, Jun 1 2006.

[4] A. C. Villeger, S. R. Pettifer, and D. B. Kell, "Arcadia: a visualization tool for metabolic pathways," *Bioinformatics,* vol. 26, pp. 1470-1, Jun 1 2010.

[5] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock*, et al.*, "Taverna: a tool for the composition and enactment of bioinformatics workflows," *Bioinformatics,* vol. 20, pp. 3045-54, Nov 22 2004.

[6] S. Muggleton, "Learning from positive data," *Inductive Logic Programming,* vol. 1314, pp. 358-376, 1997.

[7] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa, "KEGG: Kyoto Encyclopedia of Genes and Genomes," *Nucleic Acids Res,* vol. 27, pp. 29-34, Jan 1 1999.

[8] P. D. Karp, C. A. Ouzounis, C. Moore-Kochlacs, L. Goldovsky, P. Kaipa, D. Ahren, S. Tsoka, N. Darzentas, V. Kunin*, et al.*, "Expansion of the BioCyc collection of pathway/genome databases to 160 genomes," *Nucleic Acids Res,* vol. 33, pp. 6083-9, 2005.

[9] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, A. P. Arkin, B. J. Bornstein, D. Bray*, et al.*, "The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models," *Bioinformatics,* vol. 19, pp. 524-31, Mar 1 2003.

[10] T. Czauderna, C. Klukas, and F. Schreiber, "Editing, validating and translating of SBGN maps," *Bioinformatics,* vol. 26, pp. 2340-1, Sep 15 2010.

[11] J. Taubert, K. P. Sieren, M. Hindle, B. Hoekman, R. Winnenburg, S. Philippi, C. Rawlings, and J. Köhler, "The OXL format for the exchange of integrated datasets," *Journal of Integrative Bioinformatics,* vol. 4, 2007.

[12] J. L. Sussman, D. Lin, J. Jiang, N. O. Manning, J. Prilusky, O. Ritter, and E. E. Abola, "Protein Data Bank (PDB): database of three-dimensional structural information of biological macromolecules," *Acta Crystallogr D Biol Crystallogr,* vol. 54, pp. 1078-84, Nov 1 1998.

[13] Boyce Thompson Institute for Plant Research. (2009). *The tomato pathway genome database LycoCyc* Available: http://solcyc.solgenomics.net/LYCO/

[14] L. G. Firbank, M. S. Heard, I. P. Woiwod, C. Hawes, A. J. Haughton, G. T. Champion, R. J. Scott, M. O. Hill, A. M. Dewar*, et al.*, "An Introduction to the Farm-Scale Evaluations of Genetically Modified Herbicide-Tolerant Crops," *Journal of Applied Ecology,* vol. 40, pp. 2-16, 2003.

[15] C. Hawes, A. J. Haughton, J. L. Osborne, D. B. Roy, S. J. Clark, J. N. Perry, P. Rothery, D. A. Bohan, D. R. Brooks*, et al.*, "Responses of plants and invertebrate trophic groups to contrasting herbicide regimes in the Farm Scale Evaluations of genetically modified herbicide-tolerant crops," *Philos Trans R Soc Lond B Biol Sci,* vol. 358, pp. 1899-913, Nov 29 2003.