

# JWS ONLINE: A WEB-ACCESSIBLE MODEL DATABASE, SIMULATOR AND RESEARCH TOOL

**JACKY L. SNOEP**<sup>1,2,3\*</sup>, **CAREL VAN GEND**<sup>1</sup>, **RIANN CONRADIE**<sup>1</sup>,  
**FRANCO DU PREEZ**<sup>1</sup>, **GERALD PENKLER**<sup>1</sup> AND **COR STOOF**<sup>2</sup>

<sup>1</sup>Triple-J group for Molecular Cell Physiology, Department of Biochemistry, University of Stellenbosch, Private Bag X1, Matieland 7602, South Africa;

<sup>2</sup>Cellular BioInformatics, Vrije Universiteit, De Boelelaan 1087, NL-1081 HV Amsterdam, The Netherlands;

<sup>3</sup>Manchester Centre for Integrative Systems Biology, Manchester Interdisciplinary Biocentre, Manchester University, 131 Princess Street, Manchester M1 7ND, U.K.

**E-Mail:** [\\*jls@sun.ac.za](mailto:*jls@sun.ac.za)

*Received: 14<sup>th</sup> April 2008 / Published: 20<sup>th</sup> August 2008*

## ABSTRACT

In previous contributions to the ESCEC proceedings we focused on the functionality of JWS Online and we made a comparison between JWS Online and other model database initiatives. In the current chapter an update is given on new developments for JWS Online and we illustrate the functionality of JWS Online web services in workflows.

## INTRODUCTION

A number of Systems Biology tools have been made available in the JWS Online project [1]: 1) a database of curated kinetic models, 2) an easy to use, web-based simulator for those models and 3) a tool to help scientific journals with the reviewing of manuscripts that contain kinetic models. The project was initiated because there was a need for a model repository; most model descriptions in the literature are incomplete and for larger models it is not practical to re-code the model from a manuscript, even if a complete description were available. In 2000 we started building a repository for kinetic models of biological systems. The models are accessible via a web-based interface that enables the users to simulate the

models in a browser. Although the functionality of such a simulator is necessarily limited to queries that are not too computer intensive, it gives easy access for a first interaction to the model and, finally, it is a great tool for teaching.

For more elaborate or customer specific simulations the models can be downloaded in SBML and PySCeS format. The models can be accessed via three mirror sites: <http://jjj.biochem.sun.ac.za/> (Stellenbosch University, South Africa); <http://jjj.bio.vu.nl/> (Vrije Universiteit Amsterdam, the Netherlands) and <http://jjj.mib.ac.uk/> (Manchester University, UK).

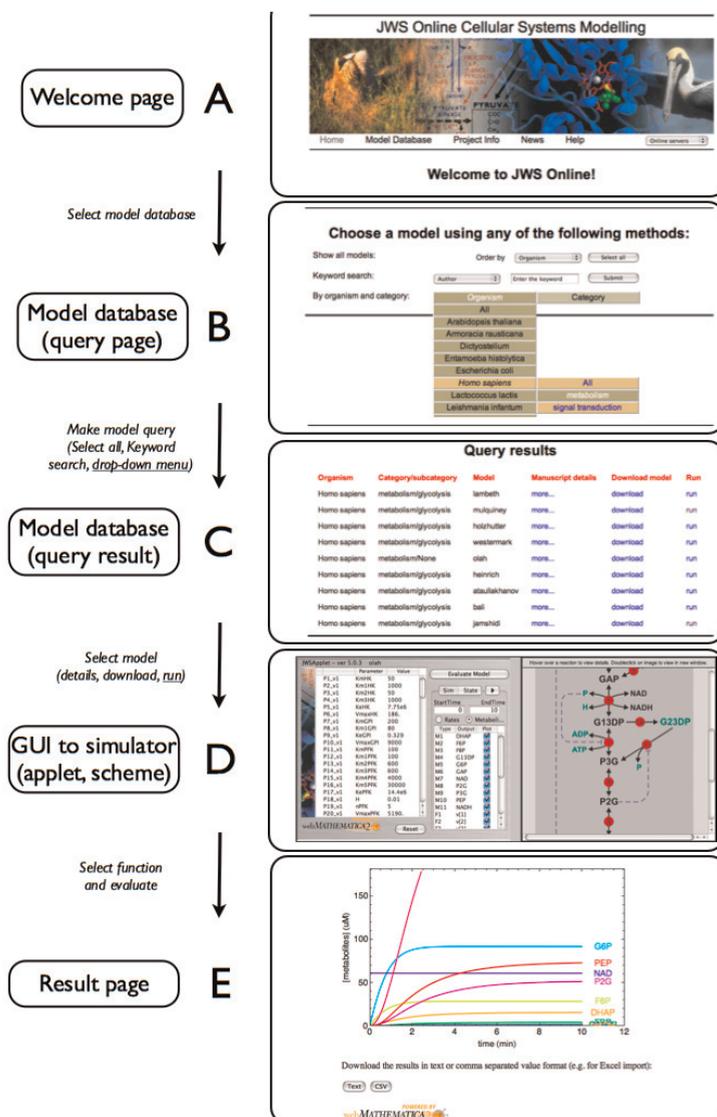
Currently (April 2008), 85 curated models can be accessed via the JWS Online model database. In 2005 we started collaborating with the Biomodels database, and we have made most of the Biomodels available via the JWS simulator (<http://jjj.biochem.sun.ac.za/biomodels> and <http://jjj.bio.vu.nl/biomodels>).

When we first recognized that kinetic models are very poorly described in the literature we also contacted a number of journals to point this out and we offered to help in the reviewing of manuscripts that contain kinetic models by making these models available on a secure site. We are now collaborating with four scientific journals: FEBS Journal, Microbiology, IET Systems Biology and Metabolomics for each of which separate web sites have been set-up to give reviewers access, via a secure site, to kinetic models described in submitted manuscripts.

JWS Online is actively used in a number of research projects, the Silicon Cell project (SiC) [2], the Yeast Systems Biology Network (YSBN) [3], and a number of new initiatives such as Systems Biology for Micro Organisms (SysMO) [4], and a seventh framework EU program on Systems Biology of eukaryotic unicellular organisms (UniCellSys) [5]. From the collaborations in these research projects it has become clear that a number of improvements needed to be made to the functionality of JWS Online to make it a good research tool, in addition to a service and educational tool. In this chapter we describe the functionality of the JWS Online simulator, with an emphasis on the newly added functions, but our focus is on a completely new functionality: web services. The importance of web services in research projects will be illustrated in an example workflow.

---

## JWS ONLINE: AN OVERVIEW



**Figure 1.** A typical example of a JWS Online simulation session.

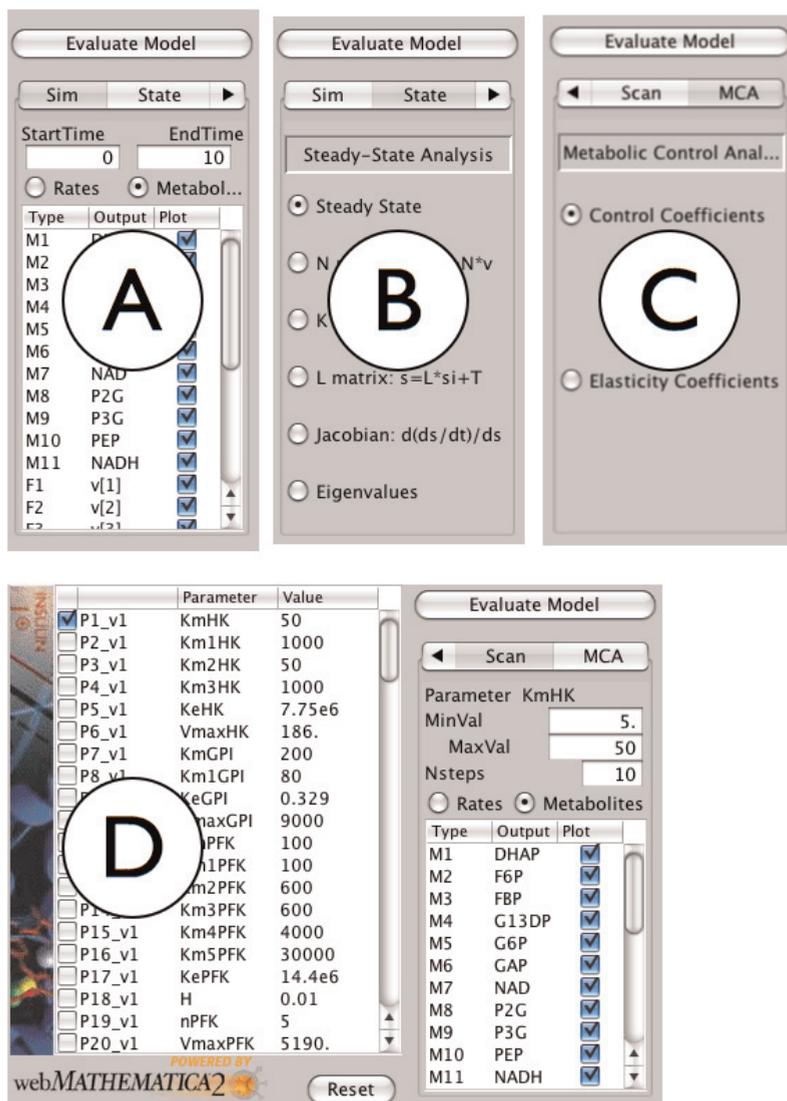
(A) the user enters JWS Online via the home page, (<http://jij.biochem.sun.ac.za>); (B) upon selecting the “model database” link, the database query page is loaded. The user selects a set of models via the query methods (drop down menu method is shown in the figure); (C) the result of the query (*Homo sapiens*, metabolism), is shown in the query result page (the user selects to run the olah model); (D) the olah model applet is loaded (a time simulation is evaluated); (E) the result page for the time simulation.

The kinetic models of JWS Online are stored in a PostgreSQL database and the selection of a specific model and queries of the models can be made via menu selections (running Python scripts in the background). A typical sequence of actions leading to a simulation results are shown as a flow chart in Fig 1: on the welcome page of JWS Online (e.g. [6], Fig. 1A), a choice must be made between, “Model Database”, “Project Info”, “News”, and “Help”. In Fig. 1A, the “Model Database” option is selected and this leads to the Model database query page (Fig. 1B), where three query methods are available for model selections: select all, select via a key-word search (“author”, “title”, “journal” (referring to respectively the first author on the manuscript in which the model is described, the title and journal name of that manuscript), “organism”, “category” (e.g. metabolism), “subcategory” (e.g. glycolysis), “model type” (e.g. demonstration)), or select via a drop-down menu on the basis of “organism” and “category”. In Fig. 1B the drop down menu is used to select “*Homo sapiens*” as organism and “metabolism” as category, leading to Fig. 1C, the query result page. On the query result page links to “manuscript details” and “model downloads” (in SBML and PySCeS format) can be selected in addition to the “run” option, which leads to the user interface of the JWS Online simulator. This interface consists of an applet and a metabolic scheme, Fig. 1D. The applet is used to change parameter values, and to select the simulation query (see next section), and executes the evaluation, calculated by the Mathematica® [7] kernel on the JWS Online server. The result is shown as a pop-up window in the browser (Fig. 1E, for an example of a time simulation). Notice the “text” and “csv” buttons at the bottom of the result window. This is a new option that allows the user to save the simulation result in either of the two formats. The “csv” format can be directly loaded into spreadsheet programs such as Microsoft Excel®.

## JWS ONLINE: SIMULATOR FUNCTIONALITY

The original functionality of the JWS Online simulator consisted of: 1) time simulations (time integration of the models and options for plotting metabolites or rates; see Fig. 2A for the interface for this function); 2) steady state analyses, (steady state solution of the model, structural analyses; N matrix, K matrix and L matrix (see e.g. [8]), and analysis in terms of Jacobian matrix and eigenvalues (Fig. 2B and 3) metabolic control analysis, control and elasticity coefficients for the reactions in the model (Fig. 2C).

---



**Figure 2.** The control panels for the different simulation options.

(A) the time simulation panel, with options to set the time period, and select the metabolite or rates options; (B) the state panel, with options to calculate the steady state solution, and some structural and stability analysis functions; (C) the metabolic control analysis option with a selection of elasticity or control coefficients, (D) the scan options, for which the parameter table from which the scanning parameter must be selected, is also shown. For the scanning option the minimal and maximal value for the scanning parameter and the number of scanning points must be indicated in the control panel.

We have recently added a fourth functionality to vary a model parameter and plot the effect on the steady state values of the selected model variables. The interface to this functionality is shown in Fig. 2D, where in the parameter table on the left the scanning parameter can be selected by checking the box in front of it. In the example shown in Fig. 2D the parameter “P1\_v1, KmHk” is selected (by default the first parameter in the list is selected). The name of the scanning parameter is also indicated in the control panel of the scanning function, together with the options, “MinVal”, “MaxVal” and “Nsteps”, specifying respectively the lower and upper value for the parameter to be scanned and the number of scans that will be made. In addition the user needs to choose whether steady state “Metabolite” concentrations or “Rates” must be plotted by selecting the respective radio button (a further differentiation on which fluxes and metabolites must be plotted can be made, in the table by (de)selecting the variables of choice).

### **JWS ONLINE: A RESEARCH TOOL**

Up to now, most of the functionality that we discussed makes JWS Online a good service tool; a user can access models and make queries and download the models. A serious limitation in the functionality of JWS Online was the absence of a mechanism to save the simulation results. This has now been added (see JWS Online: An Overview), and the file that is saved includes the parameter values that were used for the simulation such that a complete record of the simulation is available to the user.

How can the functionality of JWS Online as a research tool further be improved?

We thought that the possibility to use simulation results in a workflow would be a significant improvement, certainly, if this could be done in an environment in which an analysis of the result can be made and used in a subsequent query. Web services are ideal to be used in such workflows and they will become an important tool for Systems Biology projects, because they make it possible to link databases and other information sources and to automate standard analyses methods. We here first introduce web services and show how they can be used (implemented in Taverna) to connect different database initiatives. Subsequently we show an example workflow in Mathematica® where JWS Online web services are used.

### **WEB SERVICES**

Web services are a means of providing remote access to program functionality over a network, commonly the World Wide Web. This enables a client to use the services of programs running on servers situated in geographically diverse locations. A simple and standardised interaction specification means that each service may be accessed using the same basic protocol. Clients send service requests as XML messages encoded according to the Simple Object Access Protocol (SOAP) standard. These are usually transmitted using the ubiquitous HTTP protocol, and so can be received by any web-service enabled web server.

---

The server processes the request and returns the results, also over HTTP, as SOAP encoded XML. It is usual for servers to make available a Web Service Description Language (WSDL) file describing the web services available on the server and details of the request and response messages used to access them.

Since no knowledge is required of the underlying implementation of a web service, users may concentrate their attention only on the content of these services. Similarly, groups or organisations which have developed a tool that may be useful to the community may easily make this available, without worrying about end-users needing specialised tools or knowledge of protocols to utilise this. Access to a number of databases relevant to systems biology has been provided in this way, such as the Kyoto Encyclopedia of Genes and Genomes (KEGG) [9], BRENDA [10], Biomodels [11], and SABIO-RK [12], a database containing information on biochemical reaction, reaction kinetics and parameters, and annotation of these detailing the experimental conditions under which they were measured.

As web services follow an open standard, anyone may write a client to access a particular service. Nevertheless, it is important that the SOAP encoded XML messages have the correct fields, are transmitted correctly, and that the response messages are correctly dissected. A number of platforms provide extensions which simplify the creation and transmission of these messages. Web browsers such as Firefox allow web service requests to be entered in the location bar, with the response displayed in the browser main window. Platforms such as Mathematica® similarly include a means of accessing web services, and additionally the results may be processed using Mathematica®'s powerful symbolic manipulation tools (see below).

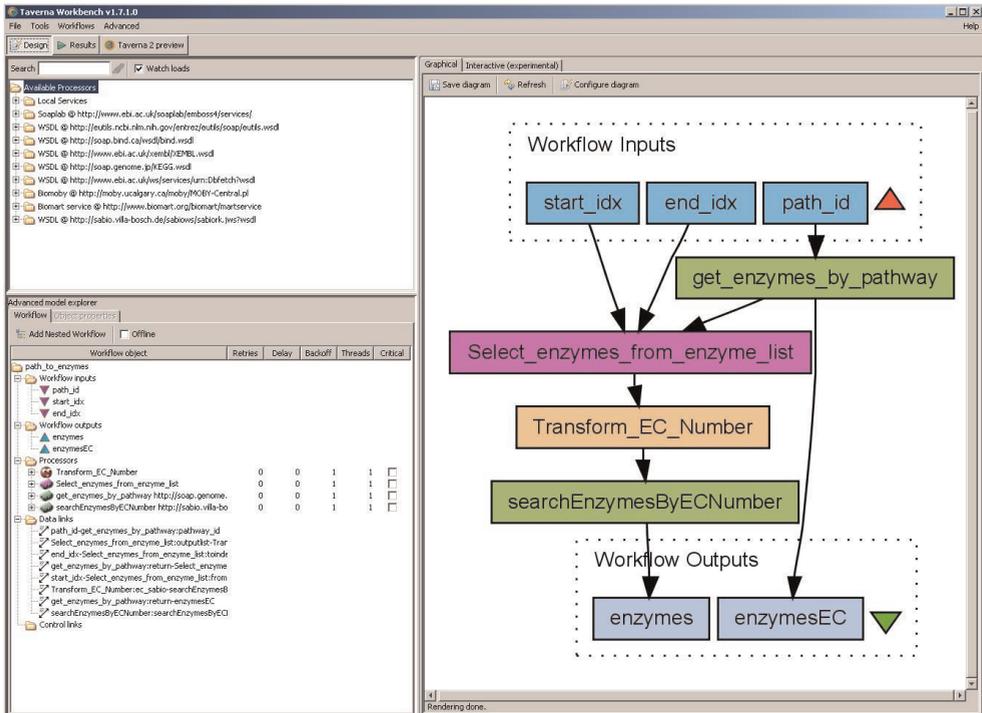
The software package Taverna [13, 14] workbench is a tool that allows the design and execution of workflows involving any number of web services. A graphical user interface allows the user to drag the required method of a service to the workflow, and link its inputs and outputs to those of other services. Taverna then allows the user to run the workflow, displaying the results graphically or saving them to a file. The scheduling, creation and transmission of the SOAP messages is handled automatically by Taverna, as is the reception and unpacking of the web service response.

## **TAVERNA WORKFLOW EXAMPLE**

Here, we describe a Taverna workflow that retrieves a list of EC numbers for the enzymes associated with a specified pathway from the KEGG web service, and then uses the SABIO-RK web service to retrieve the names of these enzymes. Although this is a trivial example, it illustrates the use of Taverna to retrieve information from multiple web services, as well as the use of Taverna's built in local transformation utilities and the Java bean shell.

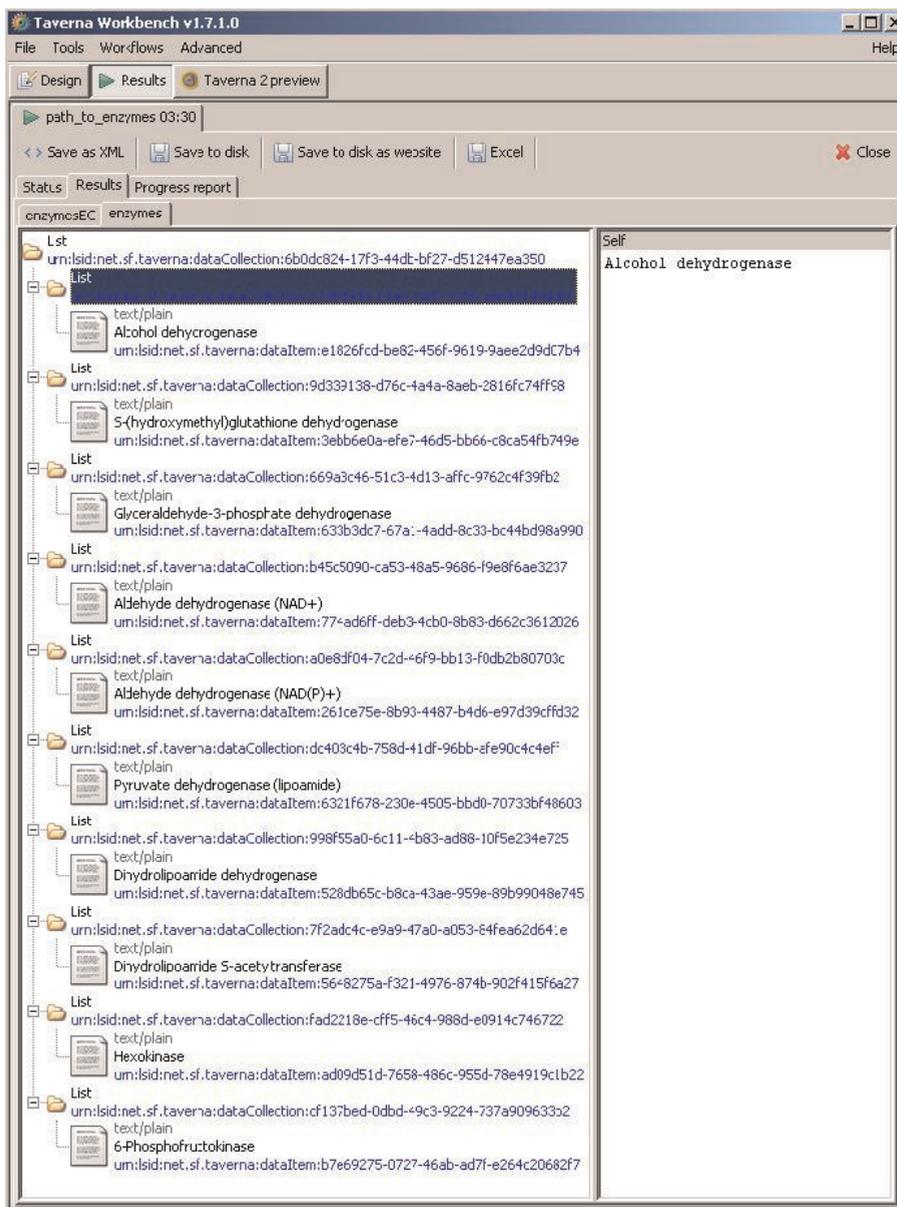
---

The pathway takes as initial input the KEGG identifier for a particular pathway (we have used the glycolysis pathway of *Saccharomyces cerevisiae*), and queries the `get_enzymes_by_pathway` method of the KEGG web service for the Enzyme Classification (EC) numbers of all enzymes associated with this pathway (Fig. 3A). We then use Taverna's list selection local service (here labelled `Select_enzymes_from_enzyme_list`) to select a subset of the list. This takes, in addition to the list of enzyme EC numbers, a start and end index, defining the subset.



**Figure 3A.** Screenshots of a Taverna workflow example.

The Taverna workbench interface, with on the left the available processors and the workflow explorer, and on the right the graphical representation of the workflow.



**Figure 3B.** Screenshots of a Taverna workflow example.

The Taverna results output for the query. See the text for more detail on the workflow.

We will then request from the SABIO-RK web service the names of all of these enzymes, but since the EC number format output by KEGG is somewhat different from the format used by SABIO-RK, we use the Java beanshell supplied with Taverna to transform these

(labelled `Transform_EC_Number`). Having done this the `searchEnzymesByEC-Number` method of the `SABIO_RK` web service returns the list of enzyme names (labelled `enzymes` in the workflow diagram). A feature of Taverna is that the output of a particular processor may be directed to multiple sinks; in this case the `get_enzymes_by_pathway` processor also sends its output to the `enzymesEC` output, so that we may view the results of this step.

When the pathway is started, a window appears allowing the user to input the various parameters required, and on completion the results (with separate tabs for separate outputs) as well as the workflow status and progress report are displayed in the results pane of Taverna (Fig. 3B).

## JWS ONLINE WEB SERVICES

JWS Online now offers a web service interface to the underlying functionality. This allows users to include JWS Online in workflows. The JWS Online web service is implemented in Java using Apache Axis which enables rapid development of functionality and automates much of the otherwise tedious coding. Each web service function is implemented as a Java method which connects to the JWS Online database through the JDBC Postgresql connector, or the Mathematica® kernel through JLink. The latter means that the existing JWS Online functionality, encoded in Mathematica® model files, can easily be accessed by the web service, and also makes the provision of new functionality straightforward. Functions exist which allow the database to be queried for a list of all models, or only models of a particular type or for specific organisms.

## MATHEMATICA® WORKFLOW EXAMPLE

The following workflow will analyse the stoichiometric matrix of all the models in the JWS Online database and calculates the number of reactions that each of the metabolites in the model is connected to. Subsequently a plot is generated showing the chance of having a number of connections as a function of the number of connections is generated. These analysis are standard in network analysis to check what type of network structure the system has; is there a random distribution of the number of connections or does the network show a scale free structure (see e.g. [15]).

Installs the web services and queries the wsdl file:

```
InstallService['http://jjj.biochem.sun.ac.za/axis/services/QueryJWS?wsdl']
```

---

Returns the web services that are currently available for JWS Online:

```
{getRates, getAllModels, getAllBiomodels, getAllBiomodels-
sIds, getModelsByOrganism, getModelsByCategory, getModelIn-
fo, getNmat, getKmat, getLmat, getSteadyStateTable, getTime-
course, getJacob, getEigenv, getCmat, getEmat, getRateEqua-
tions, getRateEquationFormulae, getExtVar, hasFunction}
```

Retrieve the current model names from JWS Online and assign the names to the variable JWSmodels:

```
JWSmodels=getAllModels[]
```

The output of the query (a list with all model names) is not shown.

Define a function “queryFunction” that checks whether a given function is present in a model:

```
queryFunction[modelname_, function_] := (response = Invoke-
ServiceOperation[hasFunction, ToString[modelname], ToS-
tring[function]]; response[[2, 3, 1, 3, 1, 3, 1, 3, 1]])
```

Checks which of the JWS Online models have the stoichiometric analysis function:

```
queryJWS=queryFunction[#, Nmat] &/@ JWSmodels
```

Define a function to retrieves the stoichiometric matrix and returns a list of the variables, stoichiometry matrix and the rate names:

```
Nmat[modelname_] := (response = InvokeServiceOperation[getN-
mat, ToString[modelname]]; rates = Table[response[[2, 3, 1, 3,
2, 3, 4, 3, i, 3, 1]], {i, Length[response[[2, 3, 1, 3, 2, 3, 4,
3]]]}]; stochmatraw = response[[2, 3, 1, 3, 2, 3, 5, 3]]; smrows =
Length[stochmatraw]; smcols = Length[stochmatraw[[1,
3]]]; stochmat = Table[stochmatraw[[i, 3, j, 3, 1]], {i,
smrows}, {j, smcols}]; varsraw = response[[2, 3, 1, 3, 2, 3, 6,
3]]; numvars = Length[varsraw]; vars = Table[varsraw[[i, 3,
1]], {i, numvars}]; {vars, stochmat, rates})
```

An example of the use of the N matrix function for the Teusink model:

```
nmat = Nmat[teusink]
```

And its output:

```
{{(ACE d)/dt, (BPG d)/dt, (d F16P)/dt, (d F6P)/dt, (d G6P)/dt,
(d GLCi)/dt, (d NAD)/dt, (d NADH)/dt, (d P2G)/dt, (d P3G)/dt, (d
PEP)/dt, (d Prb)/dt, (d PYR)/dt, (d TRIO)/dt},
{{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, -2, 0, -1, 0, 0}, {0, 0, 0, 0, 0,
0, 1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 1, -1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0}, {0, 1, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{1, -1, -1, -2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {-1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0}, {0, 0, 0, 0, 0, 0, -1, 0, 0, 0,
```

---

```

0, 0, -3, 0, 1, 1, 0}, {0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 3, 0, -1, -
1, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 1, -1, 0, 0, 0, 0, 0, 0}, {0, 0, 0,
0, 0, 0, 0, 1, -1, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0,
1, -1, 0, 0, 0, 0, 0, 0}, {-1, 0, -1, -1, -1, 0, 0, 1, 0, 0, 1, 0, 0,
0, 0, 0, -1}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, -1, 0, 0, 0, 0, 0}, {0,
0, 0, 0, 0, 2, -1, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0}},

{vGLK, vPGI, vGLYCO, vTrehalose, vPFK, vALD, vGAPDH, vPGK, vPGM,
vENO, vPYK, vPDC,
vSUC, vGLT, vADH, vG3PDH, vATP}}

```

Defining a function to determine the number of reactions that each of the metabolites in a given model is connected to, note that this function calls the `Nmat[]` function defined above:

```

degreedistribution[modelname_] := (nmat = ToExpression[N-
mat[modelname]]; Table[Length[Cases[nmat[[2, i]], Ex-
cept[0]], {i, Length[nmat[[2]]}])

```

And the application of the function to all the models:

```

degree = (degreedistribution[#] &)/@ querymodels

```

Now we make a bin count for each of the number of connections in all the models:

```

dataDegreeJWS = N[BinCounts[Flatten[Join[degree]], {1, 40,
1}]]

```

And we determine the total number of connections in all the models:

```

totalNodes = Total[dataDegreeJWS]

```

Finally we can plot the chance of having a number of connections  $P(k)$ , against the number of connections:

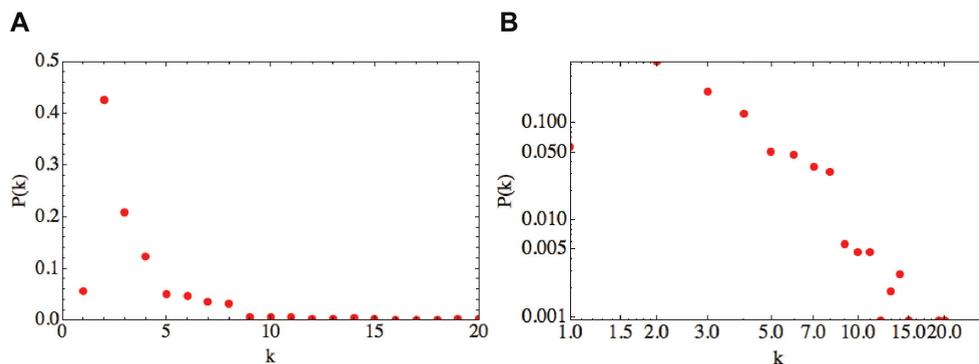
```

ListPlot[dataDegreeJWS/totalNodes, Frame -> True, FrameLabel
-> {'k', 'P(k)'}, PlotStyle -> {Red, PointSize[0.02]},
BaseStyle -> {FontSize -> 20}, PlotRange -> {{0, 20}, {0,
0.5}}, ImageSize -> 500]

```

The resulting plot is shown in Fig. 4A, together with a plot in double logarithmic space in Fig. 4B. From the plot in Fig. 4A it can be depicted that the connections do not show a normal distribution which would be indicative for a random connected network, rather the connections appear to have more of a scale free type of structure (Fig. 4B), which would result in a linear relation in double log space [15].

---



**Figure 4.** The results of a Mathematica<sup>®</sup> workflow example.

(A) the distribution of the chance for a number of connections for the metabolites in the JWS Online models is indicated. (B) the same distribution but now plotted in double logarithmic space. See the text for details on the workflow; 82 models were analysed with a total number of connections of 1081.

Clearly, this is a very specific question that is addressed, but the example serves to show that it is very simple to integrate workflows in an environment that is web service enabled. The standardisation of the queries and responses make it possible to automate requests, even to multiple databases and to make updates, for instance when new models are added to the database, or when a new DNA sequence needs to be analyzed via a certain workflow, trivial.

## CONCLUSIONS

In this contribution we have highlighted some new developments in the JWS Online project. In addition to further extensions of JWS Online simulations functionality, such as the implementation of a scanning option and the saving options for simulation results, we have focused on web services. Web services form a very useful tool, by which one can process standardized requests, which can easily be automated and extended to large numbers of models, as is illustrated in two simple workflow examples in this chapter.

**REFERENCES**

- [1] Olivier, B.G. and Snoep, J.L. (2004) Web-based kinetic modelling using JWS Online. *Bioinformatics* **20**:2143 – 2144.
  - [2] Silicon Cell Project, SiC, <http://www.siliconcell.net>
  - [3] Yeast Systems Biology Network, YSBN, <http://www.ysbn.org>
  - [4] Systems Biology for Micro Organisms, SysMO, <http://www.sysmo.net>
  - [5] 7<sup>th</sup> Framework EU program on Systems Biology of eukaryotic unicellular organisms, UniCellSys, <http://www.unicellsys.eu>
  - [6] JWS Online, <http://jjj.biochem.sun.ac.za>
  - [7] Wolfram Research, 100 World Trade Center Drive, Champaign, IL. <http://www.wolfram.com>
  - [8] Hofmeyr, J-H. S. (2001) Metabolic control analysis in a nutshell. In: *2<sup>nd</sup> International Conference on Systems Biology*, (eds. T.-M. Yi, M. Hucka, M. Morohashi and Kitano, H.) Omnipress, Madison, pp. 291 – 300.
  - [9] Kyoto Encyclopedia of Genes and Genomes, KEGG, <http://www.genome.jp/kegg>
  - [10] Braunschweig Enzyme Database, BRENDA, <http://www.brenda-enzymes.info/>
  - [11] Biomodels, <http://www.ebi.ac.uk/biomodels/>
  - [12] SABIO-RK, <http://sabio.villa-bosch.de/SABIORK/>
  - [13] Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A. and Li, P. (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* **20**:3045 – 3054.
  - [14] Li, P., Oinn, T., Soiland, S. and Kell, D.B. (2008) Automated manipulation of systems biology models using libSBML within Taverna workflows. *Bioinformatics* **24**:287 – 289
  - [15] Barabasi, A.-L. and Oltvai, Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* **5**:101 – 113.
-