

PUBLIC

SIXTH FRAMEWORK PROGRAMME
PRIORITY 1
LIFE SCIENCES, GENOMICS AND BIOTECHNOLOGY FOR HEALTH



DIAMONDS PROJECT

Contract no: LSHG-CT-2004-512143

EU Deliverable
D5.4
Bioinformatics platform - Final Product

Date: 11th June 2008
Partner responsible: Noray Bioinformatics S.L.



TABLE OF CONTENTS

1	INTRODUCTION.....	4
2	WP5 PROGRESS.....	5
2.1	PLATFORM PROGRAMMING	5
3	FINAL PRODUCT DESCRIPTION.....	7
3.1	GENERAL DESCRIPTION	7
3.2	WORKSPACES.....	8
3.3	EXPRESSION PROFILER TAB	8
3.4	KNOWLEDGE BASE NAVIGATION	10
3.5	VISUALIZATION TOOL TAB	10
3.5.1	LEFT PANEL - TABS	11
3.5.2	RIGHT PANEL – NETWORK VISUALIZATION AREA.....	12
4	TECHNICAL DESCRIPTION	14
4.1	LANGUAGE AND ENVIRONMENT	14
4.2	ARCHITECTURE	15
5	INSTALLATION & MAINTENANCE MANUAL.....	17
5.1	INSTALLATION.....	17
5.2	SET UP & MAINTENANCE.....	19
6	USER MANUAL.....	24
6.1	PORTAL HOME PAGE - LOGIN	24
6.2	PRINCIPAL DISTRIBUTION	25
6.3	WORK SPACES	26
6.4	EXPRESSION PROFILER TAB.....	27
6.4.1	DATA UPLOADING.....	27
6.4.2	TRANSFORMATION – MISSING VALUE IMPUTATION	29
6.4.3	TRANSFORMATION – DATA TRANSFORMATION	29
6.4.4	TRANSFORMATION – DATA SELECTION	30
6.4.5	TRANSFORMATION – DATA NORMALIZATION.....	31
6.4.6	EP - RESULTS.....	31
6.5	KNOWLEDGE BASE NAVIGATION TAB	31
6.6	VISUALIZATION APPLET TAB.....	33
6.6.1	TABS	33
6.6.2	DRAWING AREA.....	37
6.7	LOG OUT	38

TABLE OF FIGURES

Figure 1. Platform Redefinition.....	5
Figure 2. Programming Roadmap.....	6
Figure 3. Final Product – General View	8
Figure 4. EP Tab overview	9
Figure 5. OpenLink Virtuoso SPARQL Query page.....	10
Figure 6. Visualization Tool Tab overview	11
Figure 7. DIAMONDS architecture	16
Figure 8. Tomcat manager screenshot	18
Figure 9. Platform Home Page.....	24
Figure 10. New user form.....	25
Figure 11. Principal distribution overview.....	25
Figure 12. File navigation	27
Figure 13. Uploading – Tabular Data	28
Figure 14. Uploading – Affymetrix files	28
Figure 15. EP – Missing Value Imputation	29
Figure 16. EP – Data Transformation.....	29
Figure 17. EP – Data Selection	30
Figure 18. EP – Data Normalization.....	31
Figure 19. Knowledge Base Navigation	32
Figure 20. Knowledge Base Navigation - Results.....	32
Figure 21. Visualization Applet – General View	33
Figure 22. Visualization Applet – Properties detail.....	33
Figure 23. Visualization Applet – Network Tab	34
Figure 24. Visualization Applet – Tree Tab	35
Figure 25. Visualization Applet – Add Tab	35
Figure 26. Visualization Applet – Query Tab.....	36
Figure 27. Visualization Applet – Palette Tab	36
Figure 28. Visualization Applet – Drawing Area.....	37

1 INTRODUCTION

According to project description, last deliverable of the Work Package 5 is the Final Product of the Bioinformatics Platform.

The platform has been developed and installed correctly in its final location. In order to complete this deliverable, attached can be found this document with a double objective: first, to explain the evolution of the project in WP5 and second, to describe the functionalities of the developed Final Product.

The Platform itself has been created as a .WAR Java file and it has been deployed in a VIB server. Currently it is accessible following next URL:

<http://crunch.fvms.ugent.be:8080/diamonds/faces/welcome/index.jsp>

First project planning described that the main objective of WP5 was to design and develop a platform focussed on cell cycle analysis data. Two major aims were defined for the Platform:

- Design and Development of the GUI (Graphical User Interface), where Cell Cycle tools will be embedded.
- Optimize the GUI capabilities in close collaboration with software engineers and end-users.

Due to methodological and technical requirement changes during the project, important re-steering of the Platform took place (described in Deliverable 5.3). After this decision, the previous version was discarded and a new Platform was developed based on new requirements. These changes implied to refine the list of tools for data analysis, to focus the effort on basic necessity tools, but kept intact the original objectives. The evolution of the new platform is described point 2.- *WP5 Progress*.

This GUI has been integrated with a workspaces part and some functionalities, grouped by tabs. One of these tabs is formed by web-services for the EBI's Expression Profiler (EP) to enable data analysis, storing, sharing and management. Several functionalities of this analysis tool have been integrated to allow users to treat their expression data. A second tab can be used to navigate in the Knowledge Base using SPARQL and in the third, a Visualization Applet is specifically created to navigate graphically in the database generated and fed by the VIB group. This issue is described in more depth in the third point of this document (3.- *Final Product Description*).

The new Platform has been developed using two main technologies: *RichFaces* over JSF (for the GUI) and *Java* (for the visualization applet). The use of these two technologies separates the two main parts of the new Platform. 4th and 5th points of the document have been written to describe other technical issues related to the Platform. They have been separated in two parts: a technical description and a manual to install and maintain the implementation.

As a final thing, a user manual has been included in the document, focused on web-lab users of the Platform (6.- *User Manual*).

2 WP5 PROGRESS

As a result of to the re-steering done after the validation of the pre-Beta version of the Platform, a series of new requirements were described (see Deliverable 5.3).

Work-package internal meetings were made to describe more in depth and validate the new requirements and technologies that should be used to adapt to Expression Profiler (see Figure 1). The minutes that were made after Hinxton and Bilbao meetings (July and August 2007) were validated with the project coordinator.

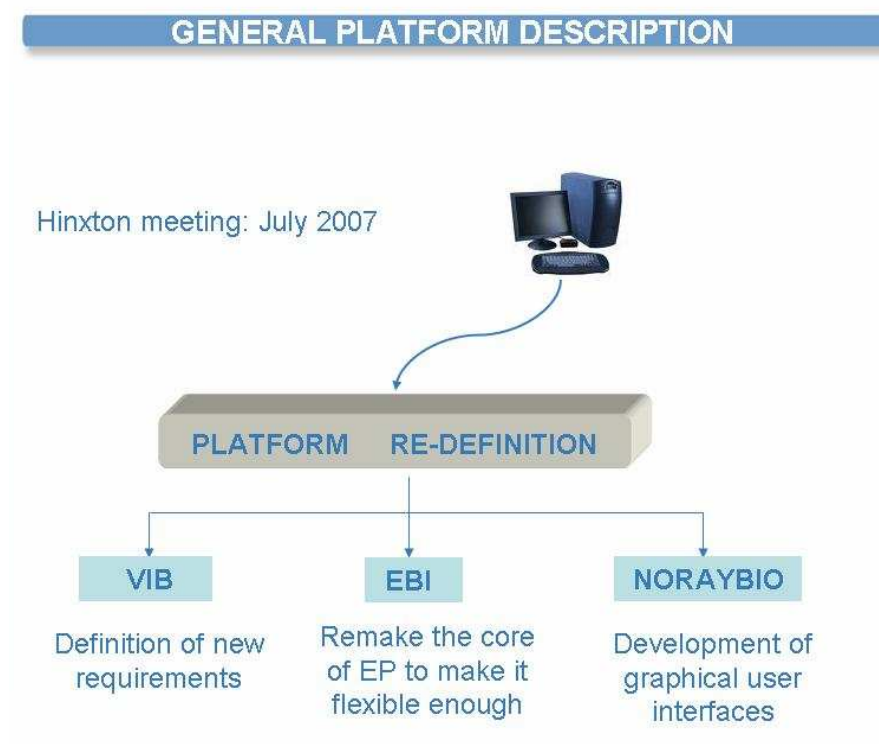


Figure 1. Platform Redefinition

These decisions were presented in the Oslo meeting in November 2007 to the other partners and they were accepted.

2.1 PLATFORM PROGRAMMING

During the year 2007 a hard programming task has been carried out, after the decision of re-steering the Platform. New functionalities and the inflexibility of Expression Profiler have forced to use new programming technologies that were not supposed to be used. Also, another change happened when it was necessary to change from Backbase to Richfaces because the first one was not an open source environment.

The new programming environment (Richfaces) has shown very little documentation and limited capability to adapt to the new requirements.

Nevertheless, NorayBio proposed and followed a roadmap for the programming tasks that started after the Bilbao and Hinxton meetings and has finished now (see Figure 2.).

- July-December 2007 – Development (in EBI) and implementation (in NorayBio) of web services related to selected functionalities of Expression Profiler.
- February 2008 – Installation in Ghent facilities of a first Beta version with the general structure and the Expression Profiler tab.
- March 2008– Installation in Ghent facilities of a second Beta version with the general structure, the Expression Profiler tab, the workspace running and the Knowledge Base Navigation Tab.
- April 2008– Installation in Ghent facilities of the last Beta version with the general structure, Expression Profiler tab, workspace running, Knowledge Base Navigation Tab and the Visualization Applet.
- May 2008: Trial period of the complete platform in Gent by potential end-users.
- End of May 2008 – Implementation of latest modifications and implementation of Final Product.

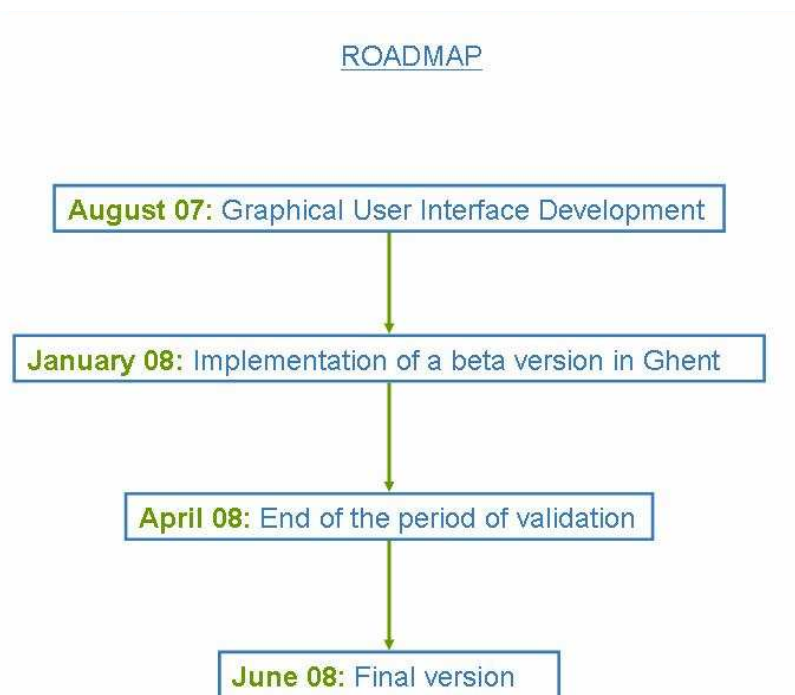


Figure 2. Programming Roadmap

NorayBio has recovered the feedback from the end-users (EBI and VIB) and has implemented the recommendations in the Final Product that is installed in its final location and presented in this document.

3 FINAL PRODUCT DESCRIPTION

3.1 GENERAL DESCRIPTION

DIAMONDS Platform is a web based tool that brings together a number of existing technologies to build a knowledge warehouse in a relational database designed to contain cell cycle regulatory network information, accessible through an intuitive user platform (GUI) with embedded data treatment tools.

It is focused to:

- Partners of the consortium who are generating gene expression data in their experiments and want to treat and store them.
- Users who want to extract information stored in the CCO (Cell Cycle Ontology) Knowledgebase about genes or processes related to their experimental aims.
- Users who want to generate and visualize new interaction networks based in their own knowledge.

Any user who has a username and a password is able to connect to the server where the Platform has been installed in:

<http://crunch.fvms.ugent.be:8080/diamonds/faces/welcome/index.jsp>

Platform has two main parts divided by their technology and the data that they can use: a part based in Expression Profiler to treat expression data, and a Java based applet to manage and visualize networks.

Users can upload expression files and treat them, using tools that have been developed by the EBI and implemented in the Platform. They can start from expression raw data, normalize and transform it directly in the EP part. Any action is stored and historical data can be consulted anytime.

The web applet has a list of functionalities that are focused to manage the information stored in the CCO Knowledgebase and to manage networks, based on a user-adapted visualization tool.

According to the design and spatial distribution, the platform is composed by:

- a) A top-left panel with the consortium logo
- b) A left panel, containing the workspaces and the "Log out" option.
- c) A work area divided in 5 tabs:
 - Expression Profiler tab
 - Knowledge Base Navigation tab
 - Applet Visualization tab
 - User Management tab
 - Credits tab

In this point, a Platform's introductory description is provided to have a general overview of each part's functions. More details are described in the 6.- *User Manual* point.

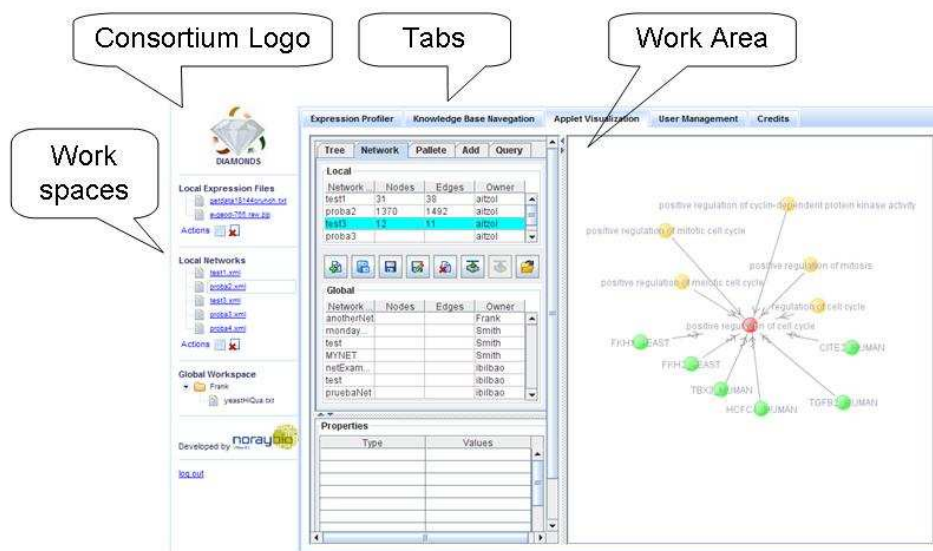


Figure 3. Final Product – General View

3.2 WORKSPACES

A workspace is accessible from any tab of the Platform and its objective is to allow the management of the information saved previously. It has two parts: a local workspace (a personal space for each user) and a global workspace (with special interest files).

Same structure of workspaces is replicated in the visualization applet, but only interaction networks and gene lists are stored in this space.

The local workspaces store and manage the expression files for the EP part and the networks created in the Visualization Applet. Global Workspace allows sharing networks that are uploaded to it.

3.3 EXPRESSION PROFILER TAB

As mentioned before, in this tab the user finds a list of tools are implemented via web services by EBI to treat expression data files. Those functionalities are:

1. Upload Expression Data
 - Tabular
 - Affymetrix

2. Transformations – Data Selection
 - Value ranges
 - Missing values
3. Transformations – Missing Value Imputation
 - Replace with zeroes
 - Replace with row averages
 - KNN imputation
4. Transformations – Data Normalization
 - RMA
 - GCRMA
 - Li & Wong
 - VSN
5. Transformations – Data Transformations
 - Intensity -> LogN Ratio
 - Ratio -> LogN Ratio
 - Average row identifiers
 - KNN imputation
 - Transpose
 - Abs -> Rel
 - Mean Center

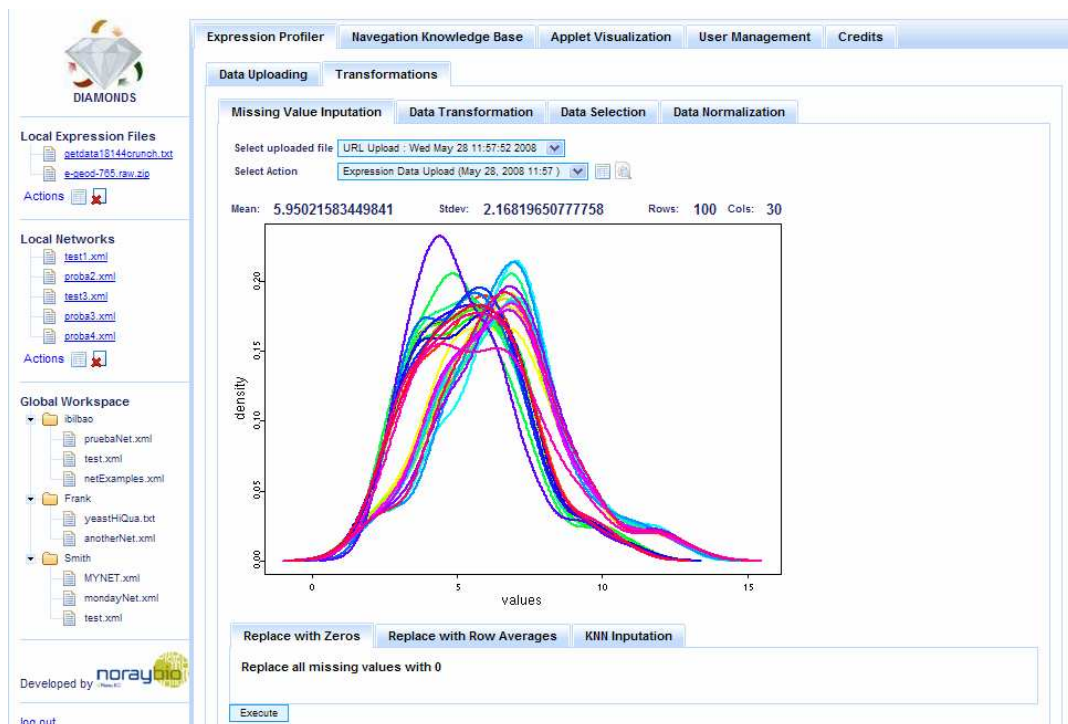


Figure 4. EP Tab overview

Uploaded data and subsequent files, generated after executing these functionalities with the data, are stored in the local workspace and in the action lists of the EP Tab.

3.4 KNOWLEDGE BASE NAVIGATION

The Knowledge Base emerges from the CCO (Cell Cycle Ontology), developed in VIB. It is be stored in a server located at Ghent (<http://crunch.fvms.ugent.be>) as a RDF format. This format allows launching queries against the CCO using SPARQL included in an URL.

The CCO design, development and implementation have been VIB-PSB responsibility, as well as the contents management and updating.

Based on the OpenLink Virtuoso SPARQL protocol, the Platform provides in this tab the possibility to write any query and send it against the CCO. Advanced users have unlimited possibilities to extract information from the CCO.



OpenLink Virtuoso SPARQL Query

This query page is designed to help you test [OpenLink Virtuoso SPARQL protocol](#) endpoint. Consult the [Virtuoso Wiki page](#) describing the service or the [Online Virtuoso Documentation](#) section [RDF Database and SPARQL](#).

There is also a rich Web based user interface with sample queries. In order to use it you must install the ISPARQL package (fsparql_dav.vad).

Query

Default Graph URI

Security restrictions of this server does not allow you to retrieve remote RDF data. DBA may wish to grant "SPARQL_UPDATE" privilege to "SPARQL" account to remove the restriction.

Query text

```
select * where { ?s ?p ?o . }
```

Display Results As: Rigorous check of the query

Figure 5. OpenLink Virtuoso SPARQL Query page

3.5 VISUALIZATION TOOL TAB

The visualization module has been implemented through a Java Applet, where results obtained in previous tabs and information from the Knowledge Base is shown.

This web-applet is a multi-platform tool that has been designed to be used with any operating system. It only needs an updated web-browser with the Java plug-in. Compared with other visualization tools, it has a remarkable improvement due to its capability to be used online. Its design also allows to work independently from the rest of the Platform.

Main objective of the applet is to offer a graphical view of the CCO, but also to help users to create and modify personalized networks, based or not in the CCO database.

The network visualization has been reused and extended from a stand-alone application developed in VIB-PSB by Steven Vercruyse.

The following lines enumerate the functionalities that the Visualization Applet has:

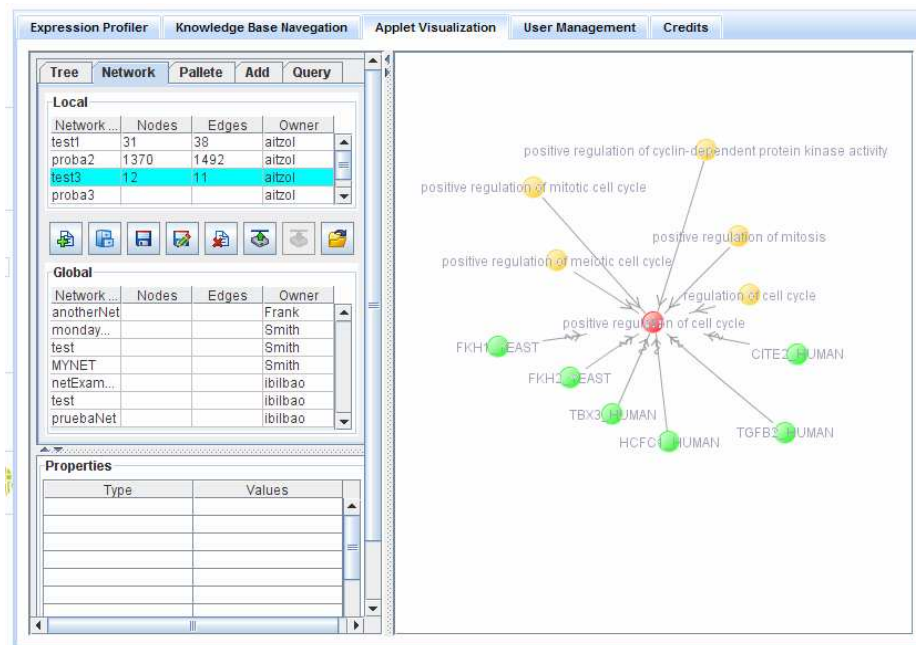


Figure 6. Visualization Tool Tab overview

3.5.1 LEFT PANEL - TABS

The user may see at the same time the graphical visualization and complementary options to modify the networks which are grouped in several tabs. These tabs are located in this part:

- Properties
- Workspace
- Palette
- Tree
- Queries

Description of the Properties

This tab will show the properties of a node selecting it and with a right-button option. The properties are always in the bottom part for any tab.

Description of the Workspace Tab

This tab is separated in two parts: an upper one with the Public Workspace, and the bottom one with the local Workspace. The system to represent the workspaces is the same as the described in the point 3.2. User will be able to add a pre-configured network from the global workspace to the Network Visualization Area.

It will be also possible to add a personal network to the visualization area. User will be able to modify the networks, save or overwrite existing files and start new networks. Representative buttons for these actions are included.

Description of the Palette Tab

It includes the options to select the colours of the nodes from the Network Visualization Area. The resulting new graph won't be automatically stored in the KB. Instead, the resulting graph could be stored as a file (xml file) in the user local workspace.

Description of the Tree Tab

The platform provides a hierarchy tree to navigate over the database contents, based on a SPARQL query provided by the PSB.

Description of the Queries Tab

User can search for new bioentities from the CCO. In this tab two searching fields have been included: the ID and the name. The result of the query is shown in a list and selected bioentities can be added to the Network Visualization Area.

Other queries are shown as mouse right-button options directly in the visualization area.

3.5.2 RIGHT PANEL – NETWORK VISUALIZATION AREA

This is the main work area to show bioentity interaction networks in a graphical way. Moreover, selecting one of the bioentities, the system makes a query to the knowledge base to characterize the bioentity getting its attributes (and show them in the "Properties" Tab). The user can save his/her own interaction networks as hypothesis in the personal workspace, but the user is not able to modify the KB contents.

The graphs show at the same time and with the same category: genes, proteins, other bioentities and ontology terms. All of them are represented as nodes and interactions between them are represented as arrows. Both have associated labels that are shown when the mouse is over them and their colour and shape depends on their type: node types differentiate between them by colour (modifiable in the Palette Tab) and interactions types differentiate by different arrow types.

There are some options to interact with the network using the mouse.

Left-button

Bioentities (nodes) shown in this area can be moved only selecting and moving them. Also, selecting the background allows to move the complete network to centre the interest area where user selects.

Centre Wheel

A zooming is included in the applet. Network labels are re-scaled to adapt to new size of the visualization area.

Right-button

Right button opens a list of options and functionalities related to selected item. In the case of the nodes, a user can expand it (draw nodes that interact with it), collapse and expanded node, delete the node, or draw its path to root (based on *is_a* relationships).

In the case of selecting the background using the right-button, the applet allows to execute network related functions as saving the network, sharing it in the Global Workspace or exporting an image with the network.

For a more detailed description go to 6.- *User Manual* point.

4 TECHNICAL DESCRIPTION

In this third point the technical specifications of the developed software are described. The description consists in three parts: Language and Environment, Architecture and Installation & Maintenance.

The Platform has been designed considering its maintenance and portability. With this document, the system administrator in charge of the Platform will be able to maintain and (if it is necessary) re-install the platform in the same or other location.

4.1 LANGUAGE AND ENVIRONMENT

Languages and environments used by NorayBio in the new GUI development have been chosen to:

- Allow the co-operation with EBI
- Facilitate the future support of the platform in VIB facilities.

In this sense, the new platform implementation will be based on Java language, due to its independence of the platform in which it is developed. This fact allows that the software developed can be run in any existing operating system.

The design of the GUI will be based on JAVA SERVER FACES. It is a framework for web based java applications. In collaboration with EBI we decided to use RichFaces because this jsf implementation has very useful components as tabs and filesystem access tree. That is needed in Diamonds application web.

RichFaces and JSF are composed by the following set of features:

- A set of APIs to represent components of a user interface and to administrate its states, handle events, validate entrance, define a navigation scheme of the pages and to give support for internationalization and accessibility.
- A set of components for the user interface.
- Multitabs flow application.
- A model of events in the server side.
- States Administration.
- Managed beans.

In order to improve the use of JSF, it was decided to use RichFaces AJAX JSF libraries. RichFaces is an implementation of JSF with a set of advanced objects for its use in web applications that allow users' approach to an environment similar to the one of the desktop applications.

The java development is separated into two different parts:

First, the web development made with NetBeans.

- Product Version: NetBeans IDE 6.0 M10 (build 200706281431)
- Java: 1.5.0_11; Java HotSpot(TM) Client VM 1.5.0_11-b03
- System: Windows XP version 5.1 running on x86; Cp1252; es_ES (nb)

The second one is the visualization java applet with MyEclipse.

Eclipse is the IDE (integrated development environment) chosen for the implementation of DIAMONDS visualization, because it is an open source system used for the creation of Web projects in Java Server Faces (among other options) which allows a perfect integration of BACKBASE in its environment.

In order to improve the IDE functionalities we have used MyEclipse 6.0 license product to develop the visualization java applet and the Java Sun version that is going to be used is *Java 1.5 update 11*.

This IDE is a very recognized developer tool in java world. In MyEclipse 6.0 version, code can be compiled while being written, providing instant feedback on syntax errors. Like most modern IDEs, this one has a graphical implementation. This aspect reduces the time dedicated to learn the language, and increases developer productivity. MyEclipse, an example of a multiple-language IDE uses Java as a base installed language. MyEclipse incorporates today's most innovative open-standard technologies to provide a development environment for J2EE WEB, XML, UML and databases and a wide array of application server connectors to streamline development, deployment, testing and portability.

The Java applet is a Swing application and has the most intuitive and advanced layout and sizing controls of any Swing visual designer today.

To make easier the collaboration in development processes between EBI and NorayBio and share information and code, a version control system has been implemented, specifically the version control system SVN (Subversion):

<https://diamonds.googlecode.com/svn/trunk>

4.2 ARCHITECTURE

The Platform structure is based on three blocks. For the correct development of the different parts of the platform the tasks assignment is described below:

- EBI:
 - Web services development.
- VIB:
 - Requirements supervision
 - Knowledge base definition
 - Database implementation
- NORAYBIO:
 - Graphical integration
 - Visualization tool
 - Internal validation

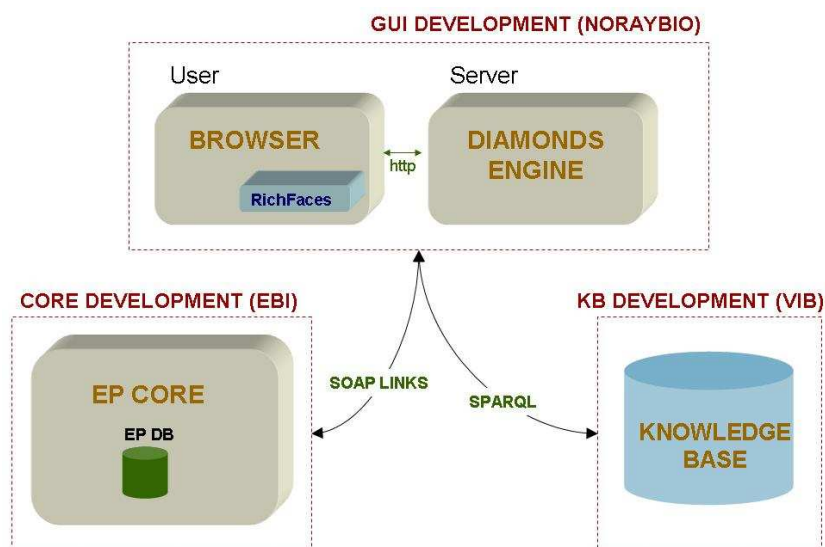


Figure 7. DIAMONDS architecture

EP integration in the platform is made using web services (SOAP links). EBI has developed and adapted some of the functionalities of EP to be used and implemented by NorayBio.

The CCO (Cell Cycle Ontology) Knowledge Base is stored in RDF format (<http://www.w3.org/RDF/>) and will be exploited using SQL based SPARQL query sentences (<http://www.w3.org/TR/rdf-sparql-query/>). VIB has developed and fed this Knowledge Base and they have also written the queries that have been implemented in the server engine by NorayBio.

The Platform comprises the tools to interact with EP web services, the tools to query the CCO Knowledgebase, a graphic user interface to manage these tools and a Java based web applet to visualize networks.

5 INSTALLATION & MAINTENANCE MANUAL

The Platform development has been designed to be installed in an easy way in the final server. Nowadays, the software is installed but in this document installation process is explained to allow the portability of it. Also, issues concerning to Platform maintenance are included in the document.

5.1 INSTALLATION

The complete GUI Platform and the visualization tools have been compressed in a unique web archive distributions file (WAR) that accomplishes the web application rules to be deployed in an application server. This type of file allows creating applications that are deployed in a server using Tomcat Manager. The file is included in the CD annexed to this document and is part of Deliverable 5.4.

To deploy and install de WAR file, Tomcat 6 distribution (or higher) is needed in the application server.

Tomcat Installation

The following paragraphs describe how to install a Tomcat server in your local machine in a few easy steps. In order to build a binary distribution version of the container from a source distribution, do the following:

1. In case you don't have a Java environment installed in your machine, download and install a Java Development Kit (JDK) release (version 1.5.11 or later) from <http://java.sun.com/j2se/>.

Set an environment variable called JAVA_HOME to the pathname of the directory where you have installed the JDK release.

2. Download and Install the Tomcat Binary Distribution (version 6 or higher) from: <http://tomcat.apache.org>

Unpack the binary distribution into a convenient location so distribution resides in its own directory (conventionally named "apache-tomcat-[version]"). In this document, the symbolic name "\$CATALINA_HOME" is used to refer to the full pathname of the release directory.

3. Start Up Tomcat. Tomcat can be started by executing the following commands:

```
$CATALINA_HOME\bin\startup.bat      (For Windows servers)
$CATALINA_HOME/bin/startup.sh      (For Unix servers)
```

After start-up, the default web applications included with Tomcat will be available by visiting: <http://localhost:8080/>

Further information about configuring and running Tomcat can be found in the documentation included here, as well as on the Tomcat web site: <http://tomcat.apache.org>

- To Shut Down Tomcat: Tomcat can be shut down by executing the following command:

\$CATALINA_HOME\bin\shutdown (For Windows servers)

\$CATALINA_HOME/bin/shutdown.sh (For Unix servers)

WAR file Deployment

This administration web console makes easier the application deployment from a Java WAR file. It can be done in two steps: select the file and click on "Install" button. Tomcat uploads and installs the software automatically.

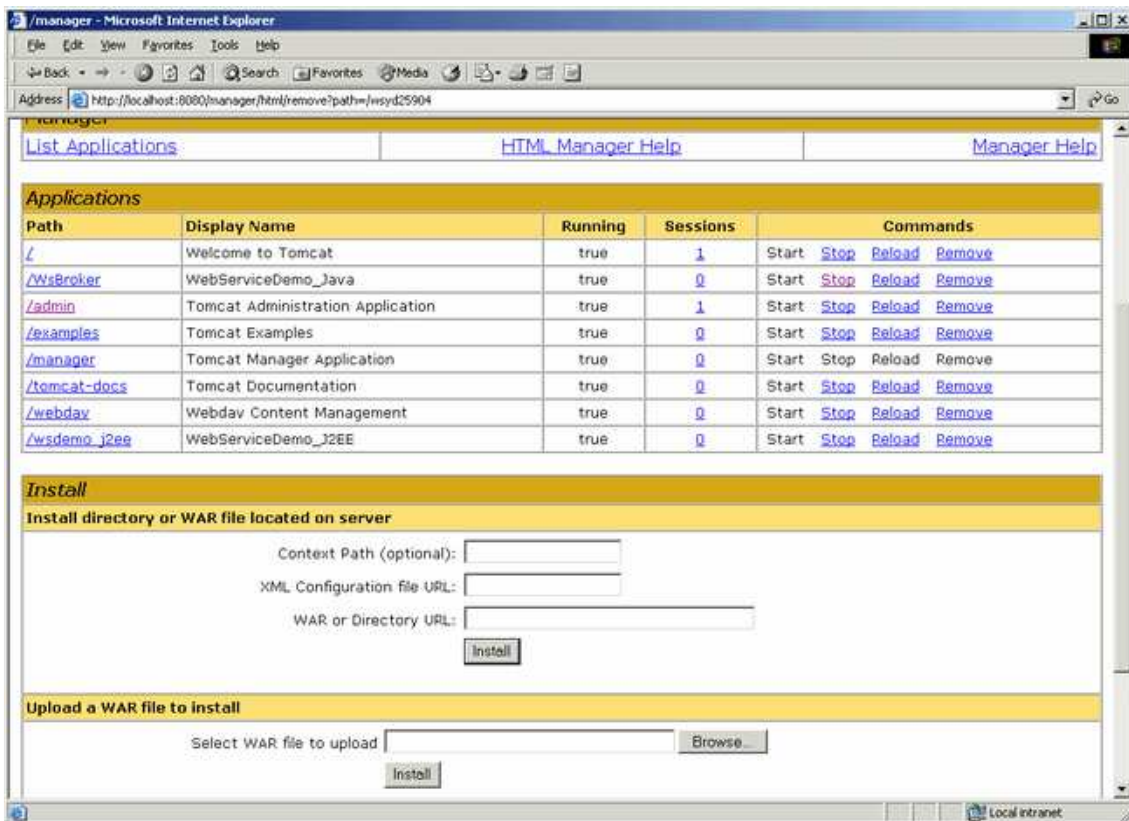


Figure 8. Tomcat manager screenshot

The tomcat manager runs in 8080 port under the web context named manager. For example: <http://127.0.0.1:8080/manager/html>

5.2 SET UP & MAINTENANCE

Diamonds java web application architecture is fully designed to install both in Windows and Unix server systems. This feature is set up into a web configuration file called *application.properties* located into *WEBINF* folder. This is the file content and the configurable parameters:

If the deployment is carried out in a Unix server, third line must be quoted and second line must be unquoted. If the deployment is in a Windows server, *vice versa*.

```
#required to make the webservice call
webfile.os=unix
#webfile.os=win
```

Web folderpaths are configured in this part.

```
webfile.unixfolderpath=/usr/local/tomcat/6/webapps/diamonds/resources/experiment
webfile.winseparator = \\
webfile.unixseparator = /
webfile.winfolderpath=C:\\NetBeansProjects\\diamonds\\build\\web\\resources\\experiment
```

These parameters can be quoted/unquoted to activate production-mode, if the Platform must be reprogrammed.

```
#ftp.do=no in production mode.
#ftp.do=yes in local mode into no web open internet
ftp.do=no
#ftp.do=yes
```

These parameters can be quoted/unquoted and dominion URL's redefined.

```
#data to make the ftp
ftp.machine =produccion.noraybio.com
ftp.webfolder =diam
ftp.user =xxx
ftp.password =xxx

# THIS IS THE PRODUCTION MODE.
# No ftp in production mode.
url.dominion=crunch.fvms.ugent.be:8080/diamonds/faces
#url.dominion=enoray34:8084/diamonds/faces/
url.experimentfolder=resources/experiment
```

Variables

There are some variables that can be changed by the system administrator. To do this, Java programming software is needed. FrontEnd desktop program can be used to open the file *constants.java*, in the *com.diam.util* folder, change the variables and

compile it. Obtained *constants.class* must be located in the same place overwriting previous file.

One of these variables in the applet class *constants.java*, is the password required to share networks, uploading them to the Global Workspace. A default password has been assigned: "diamonds2008".

```
public static final Object_PWD = "diamonds2008";
```

The other group of static variables is related to Expression Profiler instance. If it changes the location of the instance, system administrator can change these variables and the Platform will continue running.

```
static      public      String      END_POINT      =
"http://www.ebi.ac.uk/microarray-srv/EPdev/cgi-
bin/epws_soap.pl";

static      final      public      String      QNAME      =
"http://www.ebi.ac.uk/EP/WS/SOAP";
```

END_POINT is used to associate the URL where the webservices are located. QNAME is used to define the method that will be executed.

Example:

```
Service service = new Service();
Call call = (Call) service.createCall();

call.setTargetEndpointAddress(      new      java.net.URL(
Constantes.END_POINT      ));
call.setOperationName(      new      QName(      Constantes.QNAME      ,
functionName      ));
```

Next two variables are used to get access to the action related resources (Histogram, legend, heatmap, text data, etc.) and, in case of EP instance de-location, must also be modified.

```
static      final      public      String      EBI_userDataURL      =
"http://www.ebi.ac.uk/microarray-srv/EP/data/USER_DATA/";

static final public String EBI_SERVER = "http://www.ebi.ac.uk";
```

Java Applet Signature

Visualization applet needs to connect between servers to share information and read local file system to copy images. These actions throw a `SocketException` (which can be seen in the Java Console window). `SocketException` is a subclass of `IOException` that stops the applet connections.

The solution to this problem is to obtain a digital certificate and use it to sign the applet. When a browser downloads a page that contains a signed applet, before running the applet it displays a certificate in a message box. If users trust on the signed certificate, the applet will run without problems.

This certificate claims that the applet comes from the party named within and contains the digital signature of a certificate authority. The certificate contains identifying information about the certificate holder and the certificate issuer, or trusted authority. A certificate authority is a third party that is trusted to verify a certificate applicant's credentials. When the authority is satisfied with its applicant's credentials, it issues a digitally signed certificate.

The certificate authority may also have its own certificate that was generated by an even more trusted authority that verified its credentials. Each certificate will refer to the certificate of its issuing authority. This may continue for several levels and is called a chain of trust. The chain of trust ends with a top-level authority that issues its own certificate based on its own reputation.

For the Diamonds project, we have created a self-signed certificate for the applet. However, this signature expires in 6 months and must be signed again to certify its trustworthiness.

The following section will cover the basic steps to creating a signed applet.

1. Package the applet into a JAR file. The applet must be in a JAR file before a certificate can be attached to it. Use the `jar` JDK utility. If the applet was previously referenced with the help of a `codebase` attribute in `<applet>` tag, replace the `codebase` attribute with the `archive` attribute. The value of the `archive` attribute is a URL of a JAR file.
2. Create a public/private key pair. The command for this is:

```
keytool -genkey.
```

Keytool is another SDK utility. It will prompt you for a password to your keystore and for the remaining parameters, one of which is `alias`, whose value is the name of the key. The keystore is a file that contains your public/private key-pairs, and the public-keys of others with whom you exchange information. See the documentation in the above link.

3. Create a certificate for the key you created in the previous step.

```
keytool -selfcert
```

Again, keytool will prompt you for a keystore password and remaining parameters. This certificate is now self-signed by you, meaning that it has not been validated by any third party. This is suitable for demo purposes, and may be acceptable to yourself and those who know you because if there is any doubt that the certificate is really yours they can always call you up and ask you for the digest to verify that it is really you and not some impostor that created the certificate. However, if this applet is to be widely distributed, and you want it to be accepted by those who do not know you personally, you would certainly want to pay a modest fee to obtain a certificate that is validated by a trusted

certificate authority. The procedure for this is straightforward, but beyond the scope of this simple tutorial.

4. Run `jarsigner` associate this certificate with the JAR file that contains your applet. You will need to give the name of the public key of the certificate you just created. This creates a digest for each file in your JAR and signs them with your private key. These digests or hashes, the public key, and the certificate will all be included in the "WEB-INF" directory of the JAR.

Your applet is now signed. The next time you or someone else downloads it in it's page the browser will present a dialog box displaying the credentials you just created for it and asking the user permission to run it. If he/she chooses not to, the applet will throw the same `AccessControlException` that we saw in the Java Console window the first time we tried to run it in our browser. The difference is that now the user gets to make an informed decision as to whether or not they trust your applet to not harm his/her system.

Example:

```
C:\Java\jdk1.5.0_11\bin>keytool -genkey -alias tomcatserver -
keystore c:\java\k
```

```
keystore
```

```
Write the password: diamonds
```

```
Name and surname:
```

```
[Unknown]: Noray Bioinformatics
```

```
Department:
```

```
[Unknown]: R+D
```

```
Organization:
```

```
[Unknown]: Noray BG
```

```
City:
```

```
[Unknown]: Bilbao
```

```
State:
```

```
[Unknown]: Bizkaia
```

```
Country:
```

```
[Unknown]: es
```

```
Is correct?: CN= "Noray Bioinformatics", OU= "r+d", O= "Noray BG", L=Bilbao,
ST=Bizkaia, C=es?
```

```
[no]: yes
```

```
Write <tomcatserver> password:
```

```
(INTRO if it is the same as keystore's): diamonds
```

```
C:\Java\jdk1.5.0_11\bin>keytool -selfcert -alias tomcatserver -
keystore c:\java\
```

```
Kkeystore
```

```
Write keystore password: diamonds
```

```
C:\Java\jdk1.5.0_11\bin>jarsigner -keystore C:\Java\kkeystore  
diamonds20080429
```

```
jar tomcatserver
```

```
Enter Passphrase for keystore: diamonds
```

Following these steps, platform can be installed and configured for its use in any server.

6 USER MANUAL

The final product has been installed in VIB's server and it is fully functional. The Platform is formed by several parts that will be described in this point.

6.1 PORTAL HOME PAGE - LOGIN

The platform has a home page where users will be addressed to access to it (see Figure 5). This is the URL to access to the Platform:

<http://crunch.fvms.ugent.be:8080/diamonds/faces/welcome/index.jsp>

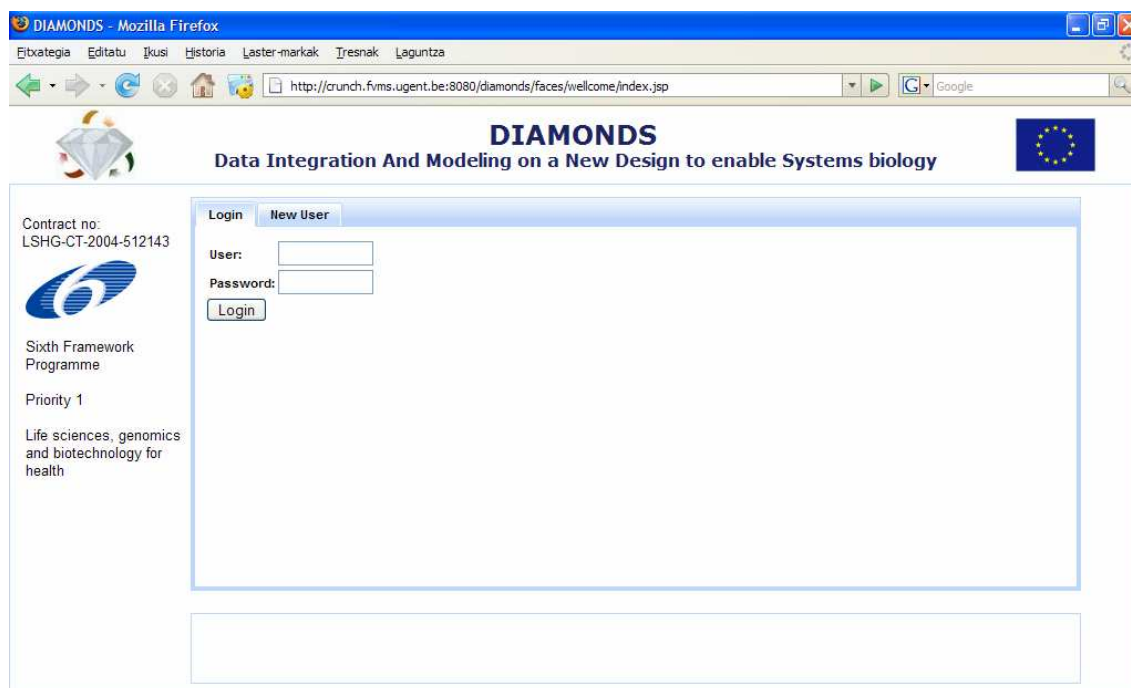


Figure 9. Platform Home Page

Once in the Home Page, there are two options for the users:

- Login (for previously registered users)
- Create a new user

Users that have been registered in the Platform or in EP before, can access directly entering the username and the correct password.

To create a new user, the Platform shows a list of fields that must be filled:

- First and second name
- Organisation and role
- E-mail
- Username and password

The form is titled 'New User' and contains the following fields:

- Forename (first name) *
- Surname (last name) *
- Organisation
- Role
- Email *
- Username *
- Password (must be at least 6 chars long) *
- Retype password *

Below the fields is an 'Execute' button and a note: '* fields are required'.

Figure 10. New user form

New users will be registered and stored in the database for future sessions. For each registered user, a personal Local Workspace will be created, both for the Expression Files and for the Network XMLs.

6.2 PRINCIPAL DISTRIBUTION

Logged users can access to the main page of the Platform, where it can be seen the distribution of the principal frames and functions.

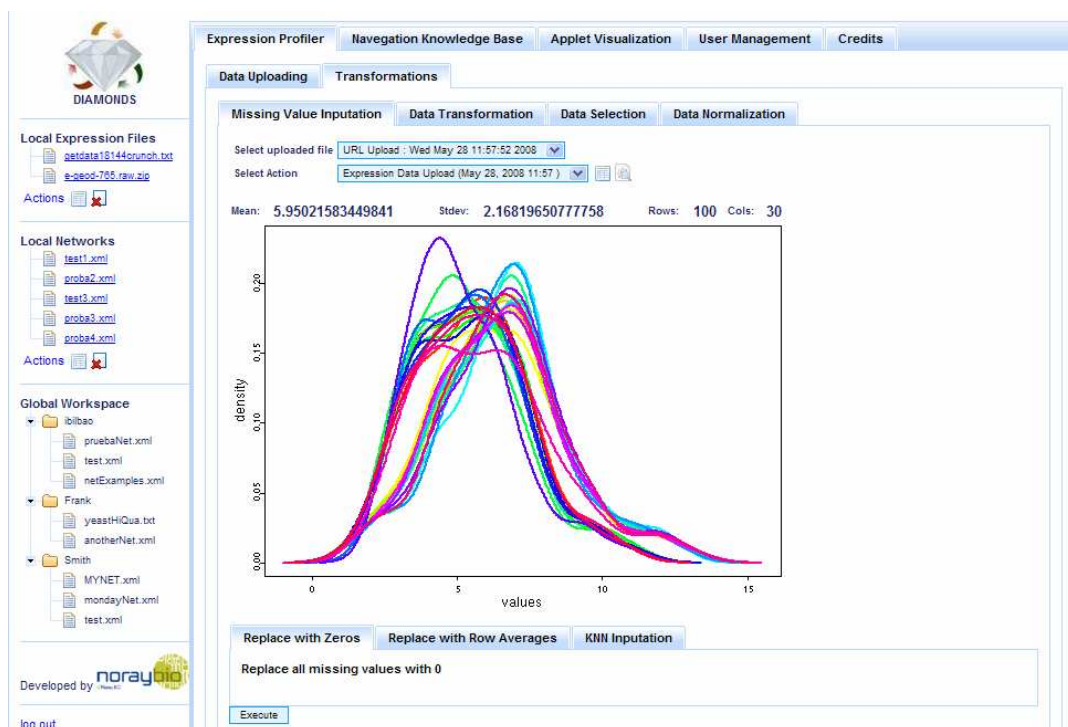


Figure 11. Principal distribution overview

These are the Platform main parts:

- Workspaces
- Expression Profiler Tabs
- Knowledge Base Navigation Tab
- Applet Visualization Tab

One by one, all these parts will be described in this document.

6.3 WORK SPACES

This part of the Platform is split in three spaces:

- Local Expression Files
- Local Networks
- Global Workspace

The local workspaces store and manage the expression files for the EP part and the networks created in the Visualization Applet. Global Workspace allows sharing networks that are uploaded to it.

Local Expression Files

For every user registered and logged in the Platform, a local workspace is generated in the VIB server. One part of this local space is reserved for the expression files uploaded to be used with the EP tools. These files are both listed in the EP File-List and in the local work-space.

Files stored in this list can be opened or downloaded using the “Open” button or deleted from the DIAMONDS server with “Remove” button.

Local Networks

Second part of the local workspace is dedicated to the networks created using the Visualization Applet. These networks can be interconnected nodes based on the Knowledge Base, new networks based on user’s knowledge or gene/protein lists obtained from a selection process.

As it happens with the expression files, networks can be opened as text files using the “Open” button or deleted from the server with “Remove” button. Clicking over the network file name directly opens the Visualization Applet representing graphically the selected network.

Global Workspace

Last part of the workspaces is dedicated to store and share networks that are accessible for any user logged to the Platform. Only advanced users will be able to add networks to this space. Clicking over a global workspace network executes the Visualization Applet and represents graphically the selected network.

6.4 EXPRESSION PROFILER TAB

First tab in the work area is the group of EP tools. These tools work calling to web services located in the EBI servers using SOAP links. The tools are divided in this tree structure:

- Data Uploading
- Data Transformation
 - Missing Value Imputation
 - Transformation
 - Selection
 - Normalization

In each case, a list of methodologies is included to work with the expression files. In this document, a brief description of the tools is provided. For more information direct to EBI's EP Home Page (<http://www.ebi.ac.uk/expressionprofiler/>).

All actions carried out in this part are stored in two fields (see Figure 8): "Select uploaded file" and "Select Action". In the first field, uploaded files are stored and they are used to classify the actions taken from that file. Any method applied to that file adds automatically a new record in the "Select Action" field allowing consulting this result anytime.

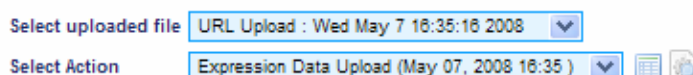


Figure 12. File navigation

6.4.1 DATA UPLOADING

Two types of expression data can be uploaded to the Platform using these options: tabular data and Affymetrix files. In each case, a similar but different method must be followed.

Uploaded files are validated to ensure their format. An error message is shown if it does not comply with predefined formats.

Tabular Data

Expression Data stored in a text file can be uploaded using two methods. First method is to select the file from the computer and second method is to enter a URL where the file is located. There are some fields to define file format (see Figure 9) and there is also the possibility to assign a name. The "Execute" button uploads the file to both servers.

Expression Profiler | Navigation Knowledge Base | Applet Visualization | User Management | Credits

Data Uploading | Transformations

Tabular Data | Affymetrix

Enter the location of the data matrix on your computer

Or enter the data matrix URL

Select the type of your data file

You can enter a custom delimiter here

Enter the position of first data column in the table

Enter the position of first data row in the table

Select data species

More Info:
Enter a name for your experiment (optional, but advised)

Figure 13. Uploading – Tabular Data

Affymetrix

Affymetrix ZIP files that are created as a result of gene expression experiments can be loaded in the same way as tabular text files. There are also two ways: directly from the local units or from a URL. Definition of these files needs less description fields (see Figure 10). Depending on the type of Internet connections and the size of the ZIP file, this process can take several minutes.

Expression Profiler | Navigation Knowledge Base | Applet Visualization | User Management | Credits

Data Uploading | Transformations

Tabular Data | Affymetrix

You can put your CEL files into one TAR/GZ or ZIP file and upload it

Enter the URL of an archive (TAR/GZ or ZIP) containing the Affymetrix CEL files you want to analyze

Select data species

More Info:
Enter a name for your experiment (optional, but advised)

Figure 14. Uploading – Affymetrix files

Any file that has been uploaded correctly will be shown in two places. On the one hand, a new file is shown in the Local Expression Files space. On the other hand, a new record is added to “Select uploaded file” field, in the EP tab. For each new record in this field, a list of actions is loaded in the “Select Action” field.

6.4.2 TRANSFORMATION – MISSING VALUE IMPUTATION

The Missing Value Imputation tab is separated in two parts: a top part that is common in all EP tabs (except file uploading) and a bottom part where there is a list of methods to apply:

- Replace with zeros
- Replace with Row Averages
- KNN Imputation. This method requires defining the number of neighbours to use in imputation.



Figure 15. EP – Missing Value Imputation

The objective of these methods is to complete the data to prepare it for further transformations.

Each method has its own parameters to configure it. Click on “Execute” button to apply the method to selected file. Incidences are shown in the bottom Note-Box. If the method has been applied correctly, a new action is added to the “Select Action” field and automatically results are shown the graphic representation box.

6.4.3 TRANSFORMATION – DATA TRANSFORMATION

Data Transformation tools are located in the tab that has the same name as the principal. It also has two parts: the top common part and a bottom part with the methods:

- Intensity - (log N) Ratio
- Ratio - (log N) Ratio
- Average row identifiers
- KNN imputation
- Transpose
- Mean centre

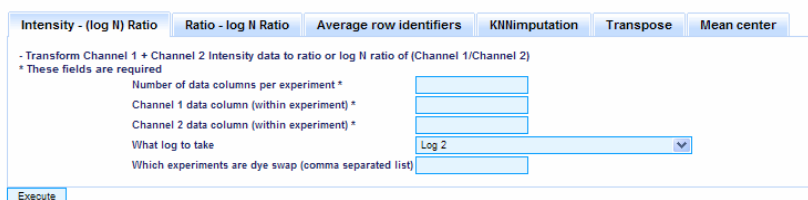


Figure 16. EP – Data Transformation

The objective of these methods is to treat the data to extract the maximum information from it. All of them work in the same way: selecting the file that should be treated,

selecting the method and clicking on the “Execute” button. Some of them have parameters that can be configured:

- Intensity - (log N) Ratio. Needs defining these parameters:
 - Number of data columns per experiment, Channel 1 data column, Channel 2 data column, which log to take, which experiments are dye swap.
- Ratio - (log N) Ratio. Needs defining these parameters:
 - What log to take.
- KNN imputation. Needs defining these parameters:
 - Estimate missing values.
 - Mean-center genes and/or arrays.
- Mean center. Needs defining these parameters:

If the method has been applied correctly, a new action is added to the “Select Action” field and automatically results are shown the graphic representation box.


6.4.4 TRANSFORMATION – DATA SELECTION

After or before applying treatment methods to the data a selection process can be done with the uploaded data. EP’s tools allow selecting the data using two methods and for each one, there is a list of parameters that can be configured:

- Value Ranges:
 - Above how many standard deviations should a gene's value fall for the gene to be selected
 - Minimum percentage of experiments for which the above STDEV criterion should hold
 - Alternatively, if not 0 - select rows with N greatest standard deviations
- Missing Values:
 - Genes that at most have a certain percentage of their values missing

Figure 17. EP – Data Selection

The system to apply the methods is the same as in previous tools: select the file, select the method, define the parameters and click on “Execute” button.

Results are shown as a new record in the “Select Action” field and a graphic representation. Text results can be analyzed opening them with the  button.

6.4.5 TRANSFORMATION – DATA NORMALIZATION

This tab comprises methods to normalize data that has been prepared in previous steps. There are 4 different normalization methods in the Platform and they are ordered by their complexity: from the simplest (and fastest) to the most complex (and slowest).

- RMA
- GCRMA
- Li & Wong
- VSN





Figure 18. EP – Data Normalization

As in other cases, bottom Note-Box notifies if the method has been executed properly and results are stored with a new record in the “Select Action” field.

6.4.6 EP - RESULTS

Results are divided and classified by the original-uploaded expression file. These files can be selected using the “Select uploaded file” field.

Any action done with the selected file is recorded as a new element in the “Select Action” field. The record consists of 4 elements:

- Name / date stamp: This information is shown for each element of the list.
- Results in text mode: on the right side of the field the button () allows opening a new window where results are shown as a text file.
- Dataset Heatmap / Dataset Lineplot: on the right side of the “Text-mode button”, there is another button () that allows opening a new window where a heatmap and a lineplot are represented. The format and the type of additional graphical representations depend on the type of data and the action.
- Histogram: appears in the centre of the screen. It represents the “density vs. values” histogram of the data after any action.

6.5 KNOWLEDGE BASE NAVIGATION TAB

This tab is focused to advanced users who want to extract information from the CCO knowledgebase. The CCO database has been generated and fed by the VIB and allows to be queried using SPARQL sentences. The process is based on the OpenLink Virtuoso SPARQL protocol.

The tab has a central main “Query text” box where any query can be written and launched. Except a limit of maximum 300 results, there is no restriction to launch any type of query.

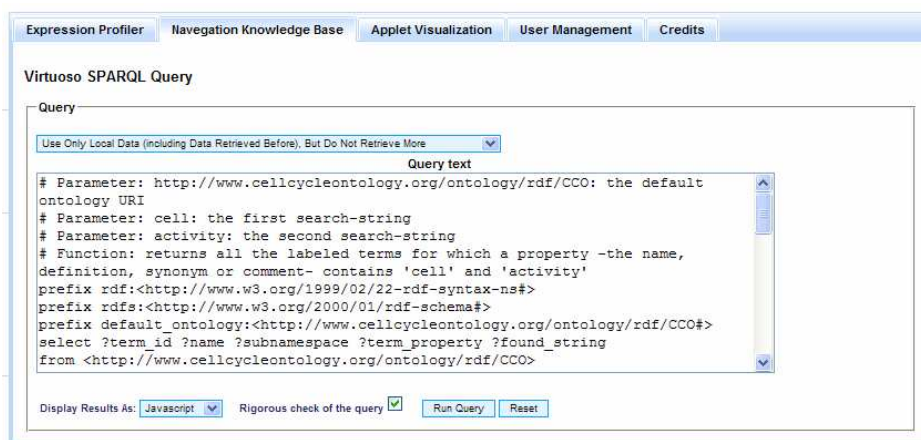


Figure 19. Knowledge Base Navigation

Results can be generated in several formats: XML, HTML, JSON and Javascript. The browser opens the results file automatically.

term_id	name
http://www.cellcycleontology.org/ontology/rdf/CCO#CCO_C0000043	pericentriolar material
http://www.cellcycleontology.org/ontology/rdf/CCO#CCO_C0000446	hydrogen%3Apotassium-exchanging ATPase complex
http://www.cellcycleontology.org/ontology/rdf/CCO#CCO_C0000505	protein kinase CK2 complex
http://www.cellcycleontology.org/ontology/rdf/CCO#CCO_C0000793	vacuolar proton-transporting V-type ATPase complex
http://www.cellcycleontology.org/ontology/rdf/CCO#CCO_C0001883	chitosome
http://www.cellcycleontology.org/ontology/rdf/CCO#CCO_C0001980	activin receptor complex

Figure 20. Knowledge Base Navigation - Results

6.6 VISUALIZATION APPLET TAB

The Visualization Web Applet has been developed from a stand-alone application created by the VIB group. The objective of the applet is to represent graphically the information stored in the CCO Knowledgebase. The applet has two parts: one for the tabs where applet functionalities are grouped, and another part for the graphical representation. There is a separation between the two parts that can be moved laterally and even one of the parts can be hidden completely.

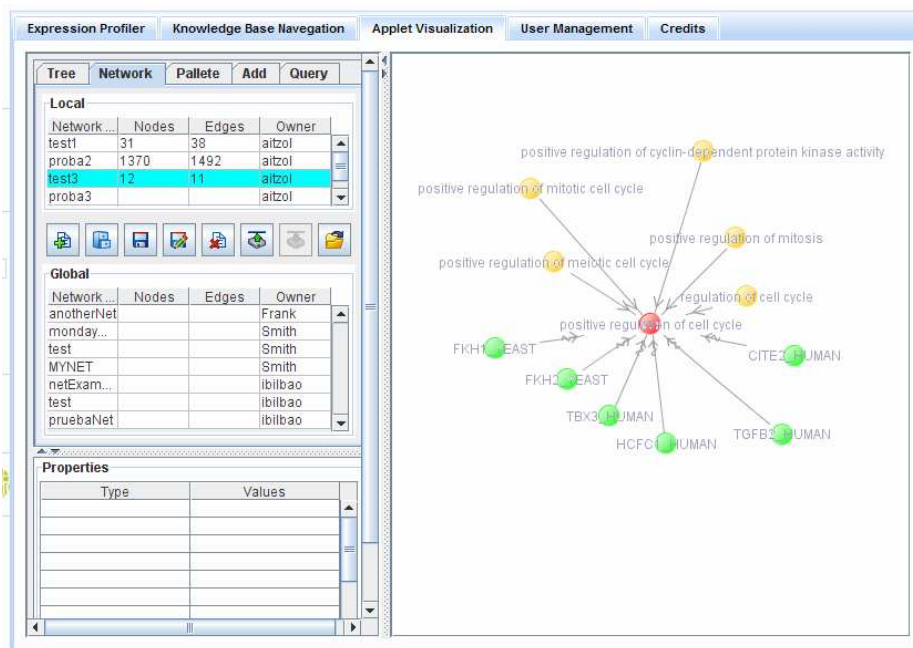


Figure 21. Visualization Applet – General View

6.6.1 TABS

There are five tabs located in the left side of the applet: Network, Tree, Add, Query, and Palette. All tabs have a common space in the bottom part: the properties list, which is formed by a list of characteristics of the selected element (in the query, in the tree or in the graph). This property list is extracted from the CCO.

Properties	
Type	Values
ID	CCO_B0000306
Name	FKH1_YEAST
Subnamespace	protein
Definition	Fork head protein homolog 1
references	P40466
references	AAA60938.1
references	AAZ22496.1
references	CAA86147.1
references	P40466
references	Q45U14

Figure 22. Visualization Applet – Properties detail

Network Tab

Local Networks and Global Workspace are represented here. It allows managing them directly, without having to close or to reload the applet.

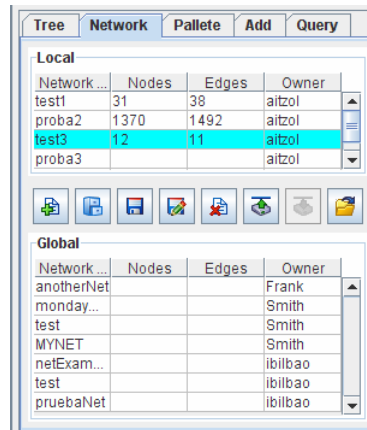










Figure 23. Visualization Applet – Network Tab

These are the actions that can be done with the networks:

-  **Create a new network.** Previously loaded/modified graph is deleted and a blank space is loaded to start with a new network.
-  **Open Selected Local Network.** It cleans the graph-space and loads the selected local network.
-  **Open Selected Global Network.** It cleans the graph-space and loads the selected global network.
-  **Save.** It saves changes done for the opened network. It only works for local networks.
-  **Save as.** It saves the modified network as a new network in the local workspace. It is very useful to personalize Global Networks.
-  **Delete Network.** It deletes selected network from the Local Workspace.
-  **Share Network.** Advanced users will be able to share self-constructed network with other users. This option uploads the networks to the Global Workspace.
-  **Open File.** Networks are stored in the server as XML files. Manually generated or modified files can be uploaded from the local units and saved in the server, as a local network.

Tree Tab

In this tab, the CCO Knowledgebase is represented in a tree-like list. Double clicking on any element expands its children and shows its properties. Opened elements can be closed again.

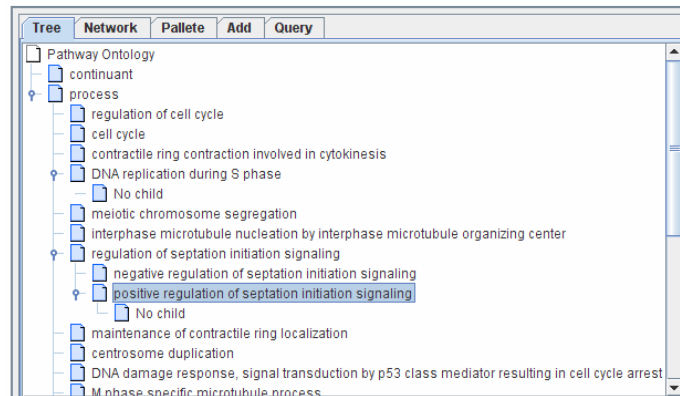


Figure 24. Visualization Applet – Tree Tab

Add Tab

Nodes and interactions can be added to the represented network. If these elements are not included in the CCO Knowledgebase, “Add Tab” allows adding it.

To add a new node, a node name must be entered and a node type selected. To add a new interaction, two nodes and an interaction type must be selected. In both cases, the node/interaction is added when user clicks on “Add element” button.

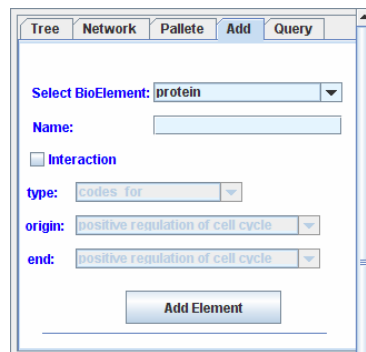


Figure 25. Visualization Applet – Add Tab

Query Tab

This tab can be used for KB querying and for add existing nodes to the network, Queries can be done based on the element's ID or by a keyword (or some keywords).

ID search method must be exact and it only recovers one element.

Keyword based search can use a word or a combination of words, and the result includes any element of the CCO that has this word/combination in the name, synonym or property fields.

Results are shown in a list located below. An element's properties can be consulted before adding it to the network.

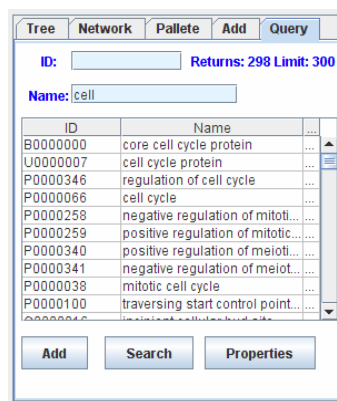


Figure 26. Visualization Applet – Query Tab

Palette Tab

Relationship between node types and colours is shown in this tab. The user can change and personalize the colour scheme of the networks. These changes are applied in the next network loading.

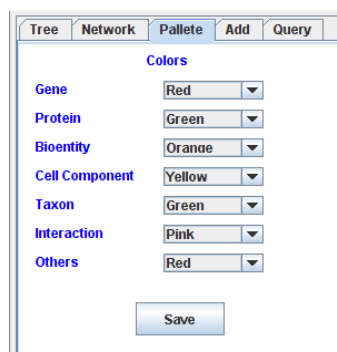


Figure 27. Visualization Applet – Palette Tab

6.6.2 DRAWING AREA

This area, located in the right side of the applet is designed for the graphical representation of networks. Previously interactions between tabs and this area have been described. Here it is a list of options that can be done in the Drawing Area using the mouse.



Figure 28. Visualization Applet – Drawing Area

Left button options:

- Select a node and move it.
- Select the background and drag the complete network.

Centre Wheel

- Zoom in and zoom out.

Right button options:

- Select a node and:
 - **Expand:** It looks for nodes related to this element and adds founded elements and its relationships.
 - **Collapse:** It does the inverse process. It deletes every node and interaction related with the selected node in an one-step distance. It does not delete items that have interactions with other elements.
 - **Delete:** It deletes the node and its interactions.
 - **Properties:** It loads the property list in the tabs bottom space.
 - **RootPath (is_a):** It looks for the shortest path to root based on “is_a” type interactions.

- Select the background and:
 - **Save Graph**: It saves the modifications done in the selected graph.
 - **Save Graph as**: It does the same, but creates a new network. User is asked for a new name for the network.
 - **Export Image**: It generates an image of the network in jpg, jpeg, png or bmp format. Image can be saved in the user's computer.
 - **Share Graph to Community**: Advanced users will be able to share self-constructed network with other users. This option uploads the networks to the Global Workspace.

6.7 LOG OUT

To log out from the Platform, there is a link in the bottom-left part of the screen. If the users spend 120 min. without using the Platform, session is finished automatically.