BIANA Tutorial

Contents

1 Getting Started **6**

1.1 Starting BIANA 6

1.2 Creating a new *BIANA Database* 8

1.3 Parsing *External Databases* 9

1.4 Creating my first working session (without data unification) . 12

1.5 Creating a *Unification Protocol* 18

1.6 Creating a working session using a *Unification Protocol* . . . 21

1.7 Creating unification protocols with rules having multiple attributes 23

1.8 Creating my first network 24

1.9 Creating different kind of networks 28

1.9.1 Different ways of network representation 28

1.9.2 Creating a network of attributes 33

1.9.3 Making predictions: Expanding relationships between nodes sharing one or more attributes 35

1.10 Removing *Unification Protocol*s 36

1.11 Removing *BIANA Database*s 37

2 Working with Multiple Data Sets **39**

2.1 Generate different datasets with initial sets of biomolecules of interest 40

2.2 Union and intersection of different data sets 41

2.3 Selecting a subset of entities by having some attributes or tags 41

2.4 Selecting a subset of entities from an existing set 43

2.5 Tagging biomolecules using different criteria 43

3 Handling Indirect Biological Relations **44**

4 Introducing User Defined Data 48
4.1 Introducing a user specific database 49
4.2 Create a network using your own data 52

5 Generating and Using Sequence Similarity Information 53
5.1 Calculating similarity of non redundant sequences in the *BIANA Database* 54
5.1.1 Removing sequence redundancy in the *BIANA Database* 54
5.1.2 Calculating sequence similarities in *BIANA Database* 55
5.2 Making predictions using sequence similarity: Sequence interologs 55

6 Putting It All Together 59
6.1 Creating a comprehensive BIANA database and an extensive unification protocol 59
6.2 Analysis on the mouse interactome 60
6.3 Analysis on the specie-wide interactome 62
6.4 Analysis on the homology extended interactome 66

Preface

Prerequisites

This tutorial assumes that you have properly installed BIANA Cytoscape Plugin. For BIANA download and installation please refer to *http://sbi.imim.es* or to the *BIANA* reference manual.

If you are using BIANA only for testing purposes, you can skip the MySQL server installation and the database creation and unification, because we provide a database in our server to be used for those tests. Check in the section .

Data files needed for following this tutorial can be downloaded from *http://sbi.imim.es/web/BIANA.php?page=biana.documentation*. It is also found in the directory `{biana/doc/tutorial}`.

Intended Audience

Users without previous BIANA experience. Users are expected to have basic knowledge on biological databases, protein protein interactions.

Aims

The tutorial aims to introduce basic acquaintance on BIANA usage.

Organization

This document is divided into 5 chapters, each of which demonstrates distinct capabilities of BIANA through simple examples. In the first chapter

(**Chapter 1**), we introduce how to use BIANA Cytoscape Plugin for creating sets of proteins if interest and their networks. Then, in **Chapter 2**, we will go over some common operations provided by BIANA to manage multiple data sets. Next, **Chapter 3**, shows how BIANA could benefit in the biological analysis of certain biological relations for a particular case, PTGS2. Following that chapter, **Chapter 4** explains basics of inserting user provided data. Finally, homology based interology network creation will be elaborated in **Chapter 5**.

Introductory Concepts

External Database Data source that contains biologic or chemical data parsed by BIANA. Examples include UniProt, IntAct, Reactome, Kegg, Pfam...

External Entity Entries given in *External Database*s.

BIANA Database Main data repository of BIANA storing biological/chemical information parsed from various *External Database*s.

Unification Protocol Set of rules that determine equivalence of data in various data sources (to cross data spread over distinct external databases).

User Entity Set of *External Entity*s that are decided to be "equivalent" with respect to a given *Unification Protocol*.

User Entity Set A group of *User Entities*.

Throughout text these special concepts are *highlighted with italic text.*

Testing BIANA Tutorial Database

You can access the online BIANA database for testing purposes. It allows skipping the installation step of installing a MySQL server and the initial steps of database creation and population. However, the online BIANA database has the following limitations:

1. It has only READ permissions.

2. New databases, parse new data or create unification protocols are not allowed. If user tries to execute some of these operations, an exception will occur.

3. It is a reduced database, which only contains necessary data to execute the tutorial exercises.

The online BIANA database has the following connection parameters (it is case-sensitive!):

- **User:** biana_user
- **Password:** biana_password
- **Database server:** sbi.imim.es
- **Database Name:** online_bianaDB

The online BIANA database contains the following *External Database*:

- UniProt-SwissProt database
- Gene Ontology (GO)
- NCBI Taxonomy database
- PSI-MI ontology
- IntAct database
- MINT database
- Reactome
- Tutorial databases (theoretical example)
- SBI datasets:
  - Inferred Metabolic Network
  - Transcription Factor cooperation and Regulated genes
  - Protein-protein interaction network predicted from sequences/structure distant patterns

Chapter 1

Getting Started: Preparing and Executing BIANA for a Typical Analysis Session

Under the context of this chapter, you will familiarize you with the basic usage of BIANA, building a BIANA data repository, defining equivalence of data in the database (*Unification Protocol*), creating sets of proteins of interest and constructing their networks.

1.1 Starting BIANA

From the Cytoscape menus at the top, go to **Plugins** and then click on **BIANA** as demonstrated in Fig 1.1.

Once BIANA is started, you should view the following section in the Cytoscape window (Fig 1.2).

1. Inside left Cytoscape panel:

   BIANA Menu: Main BIANA menu buttons.

   BIANA Session Panel: When a working session is started, all working elements can be accessed in this panel.

2. Inside bottom Cytoscape panel:

   BIANA Interpreter: Python interpreter where you can execute manually all BIANA commands and see how the graphical interface generates them.

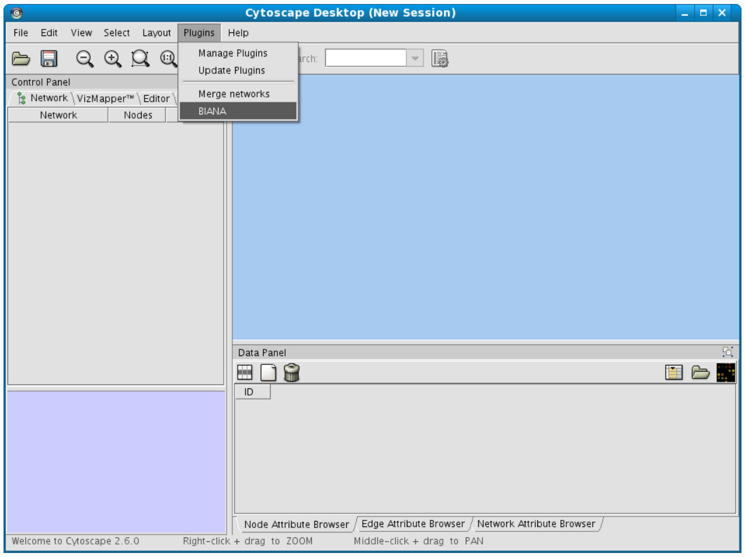


Figure 1.1: Starting BIANA from Cytoscape

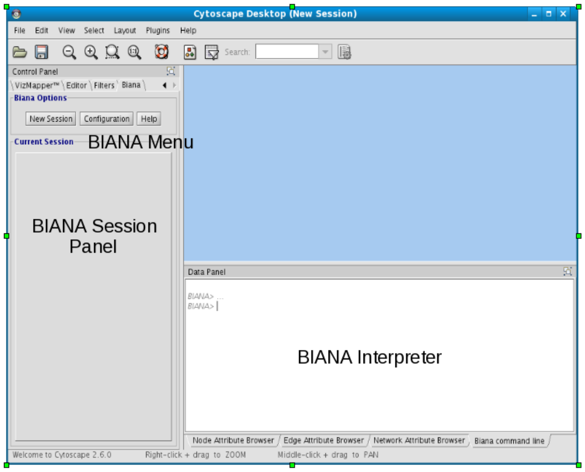


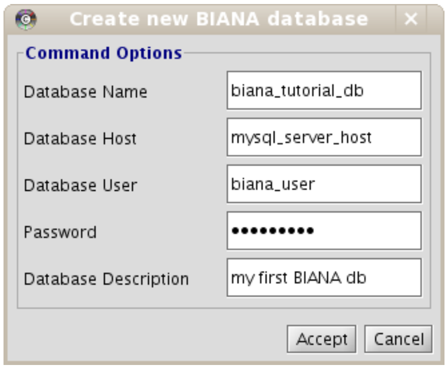
Figure 1.2: BIANA inside cytoscape

1.2 Creating a new *BIANA Database*

If you are running BIANA for the first time, the first thing to do is to create an empty *BIANA Database*. This MySQL database is going to be used to store all the data that is going to be parsed from *External Databases* later on.
If you are using our online *BIANA Database* for testing purposes, you must skip this section, as you don't have database creation permissions. You must execute the option *Add existing BIANA Database* with the parameters described at . Then, skip this section.

To create a new *BIANA Database*

1. Click on **Configuration** menu at the top of the BIANA panel on the left side of the *Cytoscape*.
2. Click on **Create new BIANA Database**.
3. In the pop-up window (Fig 1.3), enter database details.
4. Click on **Accept**.



Database Name: biana_tutorial_db
Database Host: mysql_server_host
Database User: biana_user
Password: •••••••••
Database Description: my first BIANA db

Figure 1.3: Creating a new BIANA Database.

> Remember, you have to use your MySQL server host and your MySQL configuration parameters: user and password. If you don't know how to manage it, take a look at www.mysql.com or ask your system administrator. Also, you can have your MySQL Server in your own machine or in a remote machine. If it is in your local machine, you can put as host "localhost" or "127.0.0.1" (not in all machines work both).

1.3 Parsing *External Databases*

Once a new *BIANA Database* is created, it only contains the structure of the database, but it is completely empty. So, you need to insert data available in various data sources (*External Databases*) into this database. See the BIANA manual to check which external databases and/or generic file formats are supported by BIANA. If you are using our test *BIANA Database* for testing purposes, you must skip this section (online database BIANA user account does not have permissions to insert data in our database server).

Usually, external databases contain biologic entities (as proteins, genes, etc.) and relations (as interactions, pathways, etc.) between them. Our examples contain following types of information:

- *input_database1.txt*: Contains a list of proteins and genes, with different types of identifiers for them (UniprotEntry, TaxonomyID, GeneID, AccessionNumber and GeneSymbol).
- *input_database2.txt*: Contains a list of proteins identified by their UniprotEntry code specifying which PFAM domains they contain.
- *input_database3.txt*: Contains a list of proteins identified by their NCBI AccessionNumber specifying their TaxonomyID and sequence.
- *input_database4.txt*: Contains a list of genes identified by their GeneID codes, and a list of interactions between them.

To be able to parse an *External Database*:

1. Click on **Configuration** menu.
2. Click on **Parse External Database**.
3. In the pop-up window (Fig 1.4), enter the details of the data source going to be parsed, parser type and folder or files the data is provided (Under the context of this tutorial, you will parse the 4 hypothetical databases containing various kind of information):
   (a) Select the *BIANA Database* you have just created to insert the *External Database*.
   (b) Select the type of the *External Database* format as "Generic Parser".

   (c) Choose the path of the *External Database* file as "tutorial/data/theoretical_example/input_database1.txt".

   (d) Put a name and version for the *External Database* that is going to be used to identify this *External Database* inside *BIANA* such as "database 1" and "version 1".

   (e) Choose default attribute type of the current *External Database*. Select "*uniprotentry*" for databases 1 and 2, "*accessionnumber*" for database 3 and "*geneid*" for database 4.

   (f) Click **Execute** button.

4. Parser will start automatically, and it will appear in the table at the bottom of the window. You can follow the status of the current parsing process in the same table. You can check parser outputs by clicking twice on that table row. Repeat the same process for each *External Database*. It is not necessary to wait previous parsing processes to finish. They are added into a queue and executed consecutively.

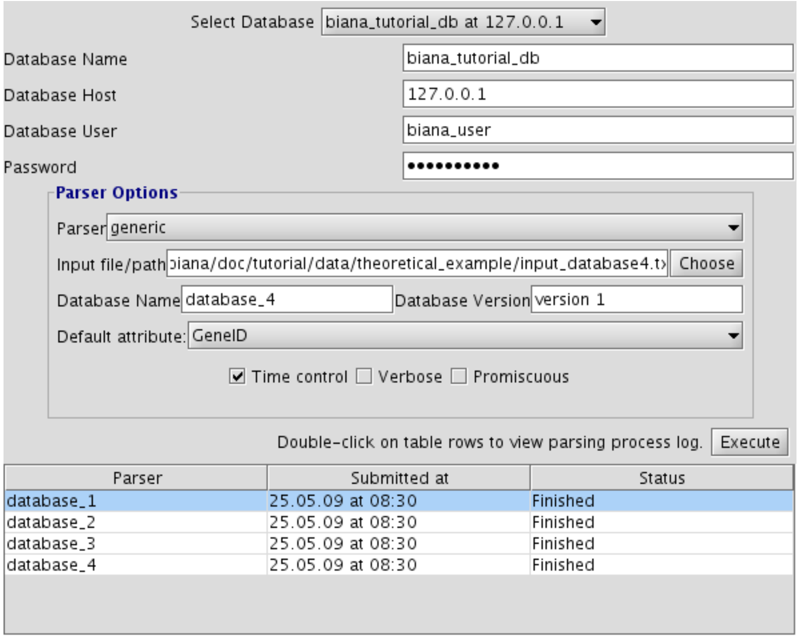
5. Once all 4 *External Databases* have been inserted, close the window.

6. Click on **Configuration** menu.

7. Click on **Update available BIANA databases** in order to update the information recently added.

Note: You can also create a new *BIANA Database* and insert *External Databases* from a command line script. Check *BIANA* reference manual to know how to use this option if you are interested.

BIANA offers by default several parsers to be able to insert most common databases. Most parsers do not have a default attribute selection option, as they know for each database which is the primary identifier. However, in data standard based parsers such as "generic", "biopax_level_2" or "psi_mi_2.5", as the default identifiers differ between different databases, you have to specify them manually.



Select Database: biana_tutorial_db at 127.0.0.1

Database Name: biana_tutorial_db

Database Host: 127.0.0.1

Database User: biana_user

Password: ••••••••••

Parser Options

Parser: generic

Input file/path: biana/doc/tutorial/data/theoretical_example/input_database4.tx

Database Name: database_4 Database Version: version 1

Default attribute: GeneID

☑ Time control □ Verbose □ Promiscuous

Double-click on table rows to view parsing process log.

Parser	Submitted at	Status
database_1	25.05.09 at 08:30	Finished
database_2	25.05.09 at 08:30	Finished
database_3	25.05.09 at 08:30	Finished
database_4	25.05.09 at 08:30	Finished

Figure 1.4: Parsing External Databases.

BIANA offers insertion of an *External Database* as **promiscuous**. If a database is specified as promiscuous using the checkbox in parser option window, the entries coming from this database will be treated differently during data unification. Entries coming from promiscuous databases, when unified, can belong to multiple *User Entities* if they satisfy the equivalence conditions (imposed by the unification protocol) with any non-promiscuous entry belonging to the same set. A useful example of a promiscuous database is SCOP, database of protein structural domains, where a domain can be assigned to more than one protein (*User Entity* in our analogy). See *BIANA* Manual for more information about a schematic explanation of promiscuity.

If you close *Cytoscape* before all parsers have finished, the process inserting the current *External Database* will continue until it finishes, but not the rest! So, it is not recommended to close *Cytoscape* before all *External Database*s have been inserted!

1.4 Creating my first working session (without data unification)

Once you have populated your *BIANA Database*, to start working with *BIANA*, you must create a *BIANA Session* explained below. A BIANA session is a container for all user generated sets of biomolecules and their networks. All query/data retrieval operations are going to be handled over this session using a *BIANA Database* created previously.

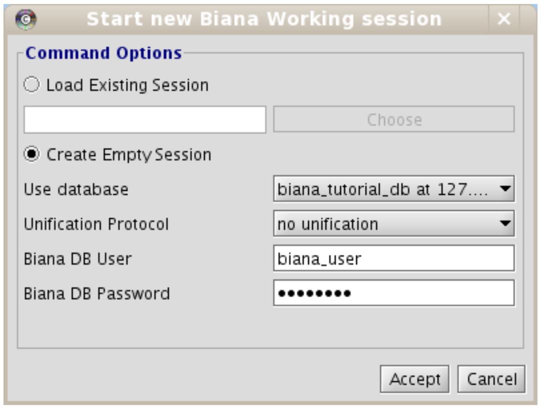
1. Click on **New Session** menu button at the top of the *BIANA* panel on the left side of the Cytoscape.

2. In the pop-up window, Select **Create empty session** option (Fig 1.5).

3. Select your *BIANA Database*(If you are using our online test *BIANA Database*, you must select it).

4. Select your *Unification protocol*. As you have not created any unification protocol, you will only be able to select **No unification**. If you are using our online test *BIANA Database*, select the *Unification Protocoltutorial_exercise1_no_unification*)

5. Add your MySQL Server access permissions for selected *BIANA Database* if necessary.



● Create Empty Session

Use database: biana_tutorial_db at 127....

Unification Protocol: no unification

Biana DB User: biana_user

Biana DB Password: ••••••••

Figure 1.5: Creating a new Working Session

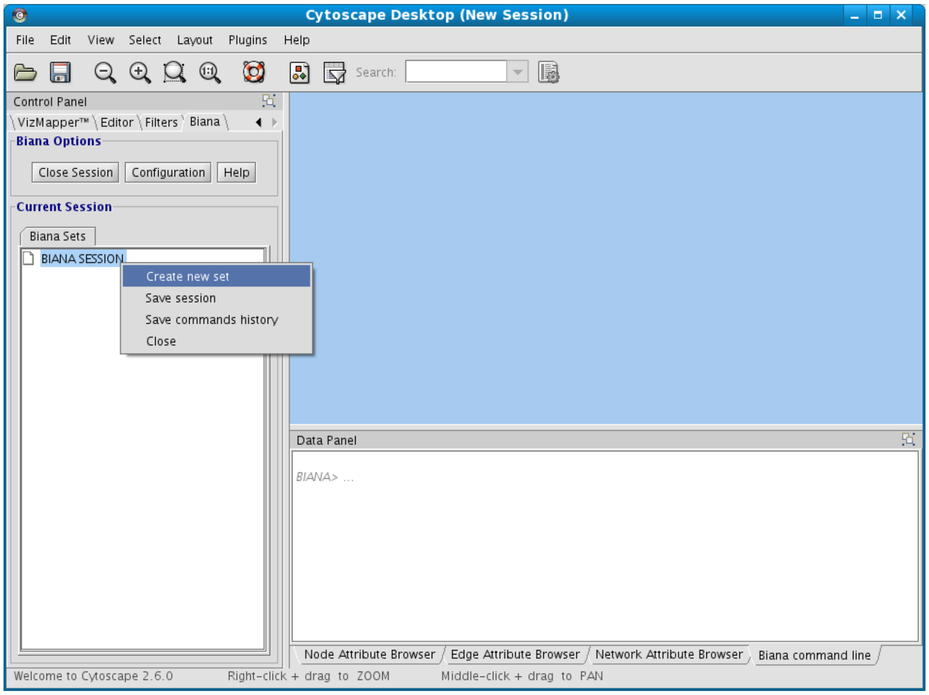
Once you start a new *BIANA Working Session*, the Session Panel becomes active and you can start creating your data sets (Fig 1.6). The basic working object in BIANA are data sets, called *User Entity Set*. In this example you are interested in set of proteins that contains the elements identified by *“uniprotentry” PROT1_HUMAN, PROT2_HUMAN and PROT1_MOUSE* in the databases you have parsed into your database. To create a *User Entity Set*:

1. Right-click on *BIANA Session* node
2. Select the option **Create New Set** (Fig 1.6).
3. In the appearing *“Create New User Entity Set”* window (Fig 1.7):
   (a) Put the name for the new data set. Name it *“my_first_set”*.
   (b) Select the *“uniprotentry”* attribute, and add the following identifiers to the right textarea: *PROT1_HUMAN, PROT2_HUMAN and PROT1_MOUSE.*
   (c) Click the **Add** button. A new row will appear in the **Added Attributes** table.
   (d) Repeat the steps (b) and (c) for all the elements you want to be in our initial data set. In this example, it is not necessary.

> When you execute the command to create a new data set, if there are no available attributes, probably you have forgotten to update databases information. Click Configuration:Update Available BIANA databases option.

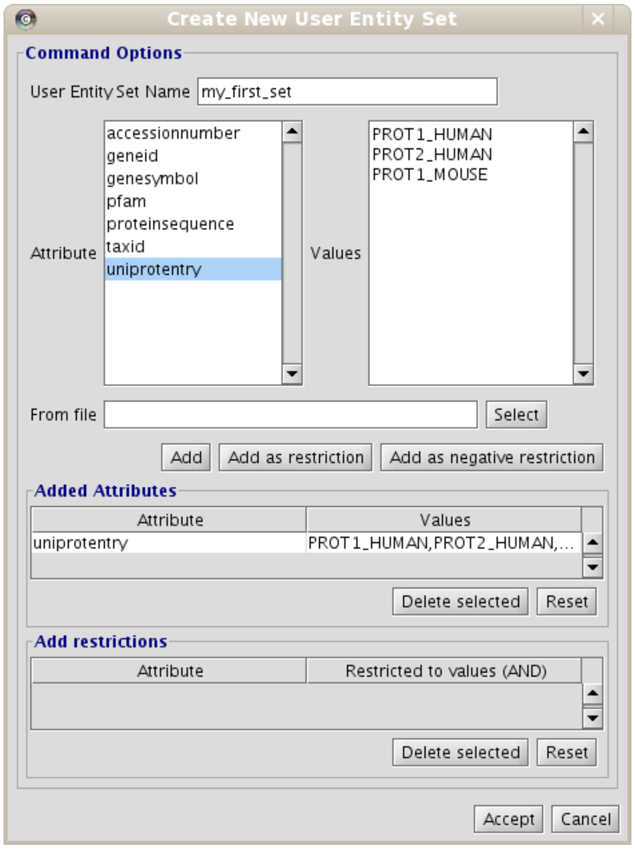
After that, you should observe the following changes (Fig 1.8):

- In the lateral panel, a new node has appeared in the *BIANA Session* tree, with the name of the new data set.
- In the BIANA terminal (*Cytoscape bottom panel*), the command of creating a new set has been executed.
- In the *Cytoscape Desktop*, a new network has been displayed (If you only observe a single node, you can change de layout by using *Cytoscape Layout algorithms* (Cytoscape → Layout → Cytoscape Layout. Check Cytoscape documentation to learn more about it).


Cytoscape Desktop (New Session)
File Edit View Select Layout Plugins Help
Search:
Control Panel
VizMapper™ Editor Filters Biana
Biana Options
Close Session Configuration Help
Current Session
Biana Sets
BIANA SESSION
Create new set
Save session
Save commands history
Close
Data Panel
Node Attribute Browser
Edge Attribute Browser
Network Attribute Browser
Biana command line
Welcome to Cytoscape 2.6.0
Right-click + drag to ZOOM
Middle-click + drag to PAN

BIANA> ...

Figure 1.6: Creating a new Set



User Entity Set Name: my_first_set

Attribute:
- accessionnumber
- geneid
- genesymbol
- pfam
- proteinsequence
- taxid
- uniprotentry

Values:
- PROT1_HUMAN
- PROT2_HUMAN
- PROT1_MOUSE

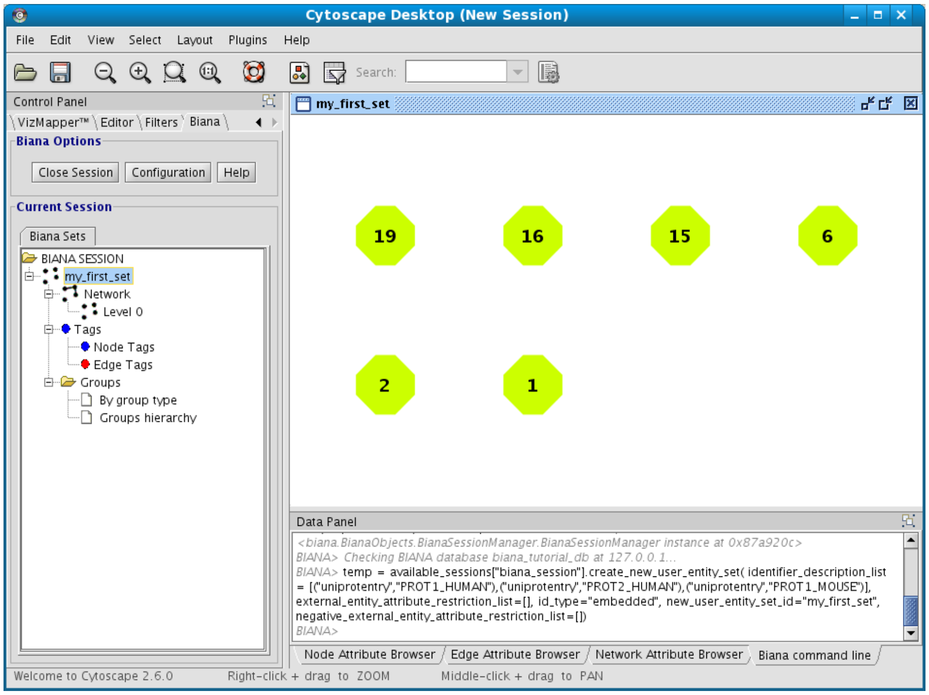
Added Attributes

Attribute	Values
uniprotentry	PROT1_HUMAN,PROT2_HUMAN,...

Add restrictions

Attribute	Restricted to values (AND)

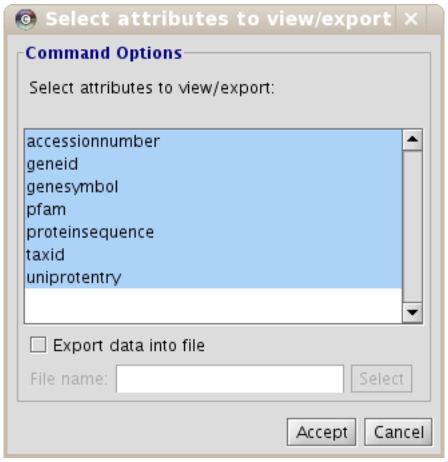
Figure 1.7: Details on creating a new set



```
<biana.BianaObjects.BianaSessionManager.BianaSessionManager instance at 0x87a920c>
BIANA> Checking BIANA database biana_tutorial_db at 127.0.0.1...
BIANA> temp = available_sessions["biana_session"].create_new_user_entity_set( identifier_description_list
= [("uniprotentry","PROT1_HUMAN"),("uniprotentry","PROT2_HUMAN"),("uniprotentry","PROT1_MOUSE")],
external_entity_attribute_restriction_list=[], id_type="embedded", new_user_entity_set_id="my_first_set",
negative_external_entity_attribute_restriction_list=[])
BIANA>
```

Figure 1.8: My first data set

What you should observe is a set of 6 nodes, identified by some numbers (*User entity IDs*, an internal *BIANA* identifier). Each node represents a *User Entity*. To look into detail what those nodes represent, right-click on *"my_first_set"* on the *BIANA Session* tree, and select the option **View/Export Set details**. A list of all available attributes will appear (Fig 1.9). Select all of them, and then click the button **Accept**.



Select attributes to view/export:

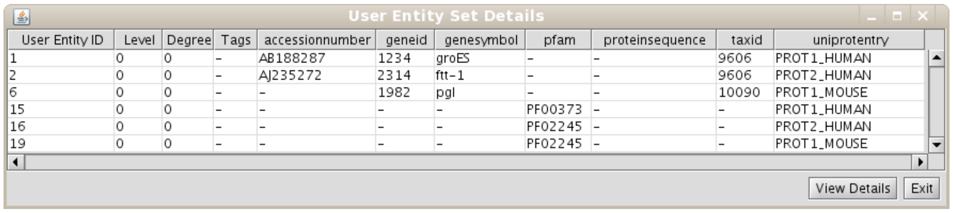
- accessionnumber
- geneid
- genesymbol
- pfam
- proteinsequence
- taxid
- uniprotentry

☐ Export data into file

File name:

Figure 1.9: View user entity set details

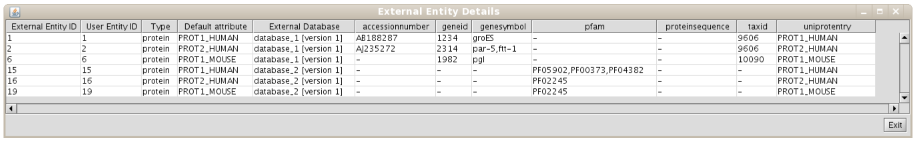
A new pop-up window will appear, showing the selected attributes for the all the nodes in the set (Fig 1.10). In this table, each row represents a node in our set (a ***User Entity***), and each column a property or attribute of it. You can observe all nodes have at least one of our selected "*uniprotentry*" attributes (*PROT1_HUMAN, PROT2_HUMAN and PROT1_MOUSE*). As you have not executed any *unification protocol*, each row consists in an entry in any of the *External Databases*.



User Entity ID	Level	Degree	Tags	accessionnumber	geneid	genesymbol	pfam	proteinsequence	taxid	uniprotentry
1	0	0	-	AB188287	1234	groES	-	-	9606	PROT1_HUMAN
2	0	0	-	AJ235272	2314	ftt-1	-	-	9606	PROT2_HUMAN
6	0	0	-	-	1982	pgl	-	-	10090	PROT1_MOUSE
15	0	0	-	-	-	-	PF00373	-	-	PROT1_HUMAN
16	0	0	-	-	-	-	PF02245	-	-	PROT2_HUMAN
19	0	0	-	-	-	-	PF02245	-	-	PROT1_MOUSE

Figure 1.10: View user entity set details

If you want to know more about these *User Entities*, you can look more detailed information if you select some rows and click the **View details** button. In figure Fig 1.11 you can observe the details of the resulting table.



External Entity ID	User Entity ID	Type	Default attribute	External Database	accessionnumber	geneid	genesymbol	pfam	proteinsequence	taxid	uniprotentry
1	1	protein	PROT1_HUMAN	database_1 [version 1]	AB188287	1234	groES	-	-	9606	PROT1_HUMAN
2	2	protein	PROT2_HUMAN	database_1 [version 1]	AJ235272	2314	par-5,ftt-1	-	-	9606	PROT2_HUMAN
6	6	protein	PROT1_MOUSE	database_1 [version 1]	-	1982	pgl	-	-	10090	PROT1_MOUSE
15	15	protein	PROT1_HUMAN	database_2 [version 1]	-	-	-	PF05902,PF00373,PF04382	-	-	PROT1_HUMAN
16	16	protein	PROT2_HUMAN	database_2 [version 1]	-	-	-	PF02245	-	-	PROT2_HUMAN
19	19	protein	PROT1_MOUSE	database_2 [version 1]	-	-	-	PF02245	-	-	PROT1_MOUSE

Figure 1.11: View external entity details table

You will observe again six rows detailing the information about these entries. In contrast with the last table, each row in this table consists of an ***External Entity***, an entry of an *External Database* (observe that a new column appeared, *External Database*, that indicates which is the source it). It is important to observe that the *External Entity ID*, which is an internal identifier for this entry, is equal to the *User Entity ID*. This is because you have not applied any *Unification Protocol*. Now, you can close the windows with tables.

As the *User Entity ID* is not useful for you, once you have executed the *View/Export set details*, you can change the *Vizmap* of Cytoscape display to label the nodes in the graphical representation with the attribute you want. To do this, select the **VizMapper** tab in the control panel of *Cytoscape* (left panel), and change the **Node label** property by selecting the desired attribute (Fig 1.12). (see *Cytoscape* documentation for more details on how modifying graphical display). In this example, you can select the attribute *uniprotentry* to label the nodes.

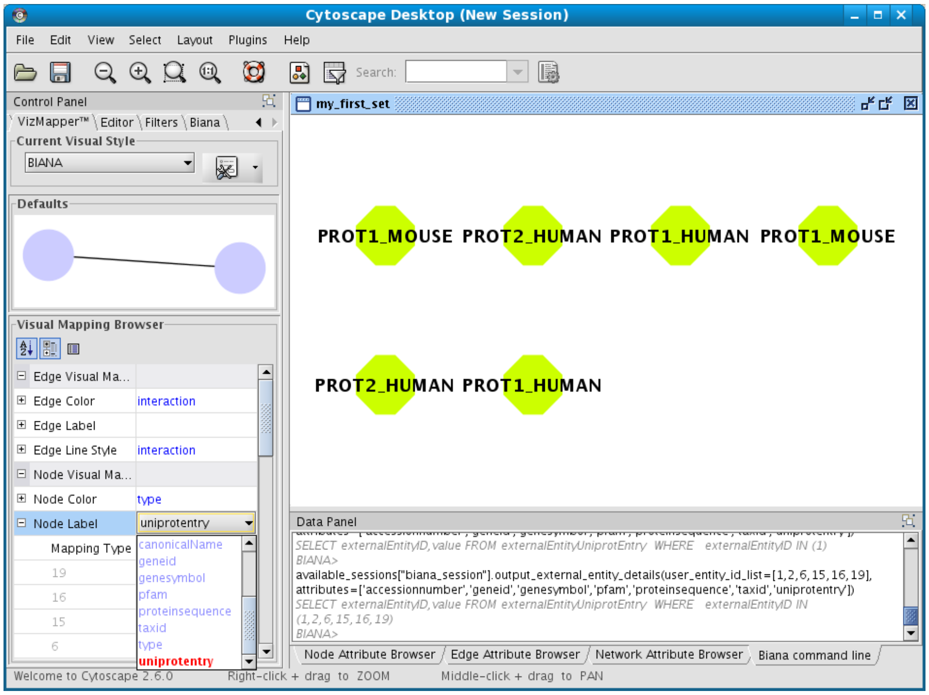
Finally, to close the current working session, click the **Close Session** button in the BIANA menu.

1.5 Creating a *Unification Protocol*

We know that different *External Databases* reference equivalent entities, so it would be very useful to integrate them. In this example, the four *External Databases* have different identifiers, but they share some of them. So, you can use them to integrate the data. First, we remember which attribute or identifiers are used in each of our parsed sample databases:

input_database1.txt : uniprotentry, taxonomy ID, geneid, NCBI Accession Number and GeneSymbol

input_database2.txt : uniprotentry and PFAM domains



```
attributes=['accessionnumber','geneid','genesymbol','pfam','proteinsequence','taxid','uniprotentry'])
SELECT externalEntityID,value FROM externalEntityUniprotEntry  WHERE   externalEntityID IN (1)
BIANA>
available_sessions["biana_session"].output_external_entity_details(user_entity_id_list=[1,2,6,15,16,19],
attributes=['accessionnumber','geneid','genesymbol','pfam','proteinsequence','taxid','uniprotentry'])
SELECT externalEntityID,value FROM externalEntityUniprotEntry  WHERE   externalEntityID IN
(1,2,6,15,16,19)
BIANA>
```

Figure 1.12: Change display properties of the set with Cytoscape

input_database3.txt : NCBI Accession Number, taxonomy ID and sequence.

input_database4.txt : geneid

If you want to integrate these databases, you cannot use a single identifier, as different *External Database*s are giving different identifiers. For example, if you only use *uniprotentry* code, only databases 1 and 2 will be unified, while *external entities* in databases 3 and 4 won't be unified with any other. If you decide to use *geneid* code to unify, then only databases 1 and 4 will be unified, etc. To avoid this problem, *BIANA* offers the possibility to create a *Unification Protocol* composed by several unification rules (called *Unification Protocol Atoms*). In this example, it would be interesting create the following unification:

- Unify entries of databases 1 and 2 by using *uniprotentry*
- Unify entries of databases 1 and 4 by using *geneid*
- Unify entries of databases 1 and 3 by using *accessionnumber*

Here, we show how to integrate different databases by using Unification Protocols. Creating a *Unification Protocol* means creating a set of rules to be used to integrate several *External Database*s. (If you are using our online test *BIANA Database*, skip this section, as you don't have permissions for executing it).

To do this, execute the following steps (Fig 1.13):

1. Click on **Configuration** menu and then click on **Create new unification Protocol**.
2. Enter the details of the unification protocol:
   (a) Select the *BIANA Database* in which you want to create a *Unification Protocol*
   (b) Put a name to the protocol (as you will be able to generate multiple *unification protocols* in the same *BIANA Database*)
   (c) Select the attribute (or combination of attributes) and the databases you want to use to integrate data.
      i. Select *uniprotentry* attribute and databases 1 and 2
      ii. Click the **Add** button.

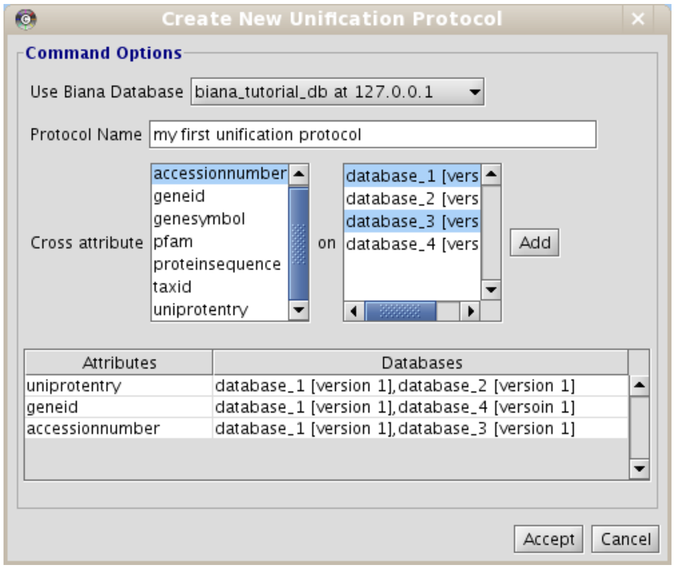
iii. Select *geneid* and databases 1 and 4

iv. Click the **Add** button.

v. Select *accessionnumber* and databases 1 and 3

vi. Click the **Add** button.

3. Click the **Accept** button.



Use Biana Database: biana_tutorial_db at 127.0.0.1

Protocol Name: my first unification protocol

Attributes	Databases
uniprotentry	database_1 [version 1],database_2 [version 1]
geneid	database_1 [version 1],database_4 [versoin 1]
accessionnumber	database_1 [version 1],database_3 [version 1]

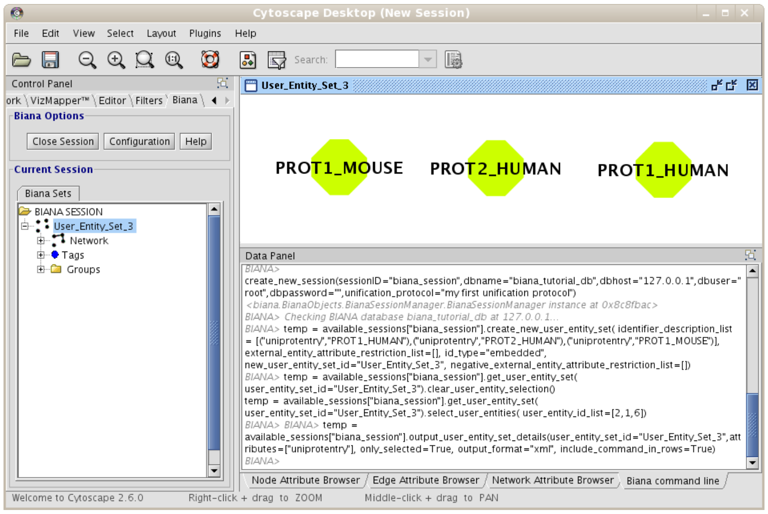
Figure 1.13: Create a unification protocol

> It is not mandatory to use all *External Databases* in the *BIANA Database* to create a *Unification Protocol*. Databases not used won't be used for the integration, and their entries won't be accessible for the sessions using this *unification protocol*. If you want to create a *unification protocol* and add an *External Database* without being integrated with the rest, simply add the database without specifying any attribute.

1.6 Creating a working session using a *Unification Protocol*

Repeat the process of starting a working session (see 1.4), but select the recently created *Unification Protocol*(if you are using our test *BIANA Database*, select the *Unification Protocoltutorial_exercise1_unification1*). Then, create a new *User Entities*, with the same uniprotEntry identifiers as used in the last example. Now, you should obtain a set with only 3 *User Entities* (Fig 1.14).

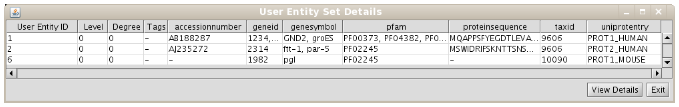
Note: If you do not see any labels for nodes it is because you have changed VizMapper to display uniprotentry identifiers. BIANA will load information associated with entities whenever you request them explicitly from the database using **View/Export set details**. You will see the uniprotentry ids will appear after asking for set details.


Cytoscape Desktop (New Session)
File Edit View Select Layout Plugins Help
Search:
Control Panel
ork VizMapper™ Editor Filters Biana
Biana Options
Close Session Configuration Help
Current Session
Biana Sets
BIANA SESSION
User_Entity_Set_3
Network
Tags
Groups
User_Entity_Set_3
PROT1_MOUSE
PROT2_HUMAN
PROT1_HUMAN
Data Panel
Node Attribute Browser
Edge Attribute Browser
Network Attribute Browser
Biana command line
Welcome to Cytoscape 2.6.0
Right-click + drag to ZOOM
Middle-click + drag to PAN

```
BIANA>
create_new_session(sessionID="biana_session",dbname="biana_tutorial_db",dbhost="127.0.0.1",dbuser="
root",dbpassword="",unification_protocol="my first unification protocol")
<biana.BianaObjects.BianaSessionManager.BianaSessionManager instance at 0x8c8fbac>
BIANA> Checking BIANA database biana_tutorial_db at 127.0.0.1...
BIANA> temp = available_sessions["biana_session"].create_new_user_entity_set( identifier_description_list
= [("uniprotentry","PROT1_HUMAN"),("uniprotentry","PROT2_HUMAN"),("uniprotentry","PROT1_MOUSE")],
external_entity_attribute_restriction_list=[], id_type="embedded",
new_user_entity_set_id="User_Entity_Set_3", negative_external_entity_attribute_restriction_list=[])
BIANA> temp = available_sessions["biana_session"].get_user_entity_set(
user_entity_set_id="User_Entity_Set_3").clear_user_entity_selection()
temp = available_sessions["biana_session"].get_user_entity_set(
user_entity_set_id="User_Entity_Set_3").select_user_entities( user_entity_id_list=[2,1,6])
BIANA> BIANA> temp =
available_sessions["biana_session"].output_user_entity_set_details(user_entity_set_id="User_Entity_Set_3",att
ributes=["uniprotentry"], only_selected=True, output_format="xml", include_command_in_rows=True)
BIANA>
```

Figure 1.14: Creating a *User Entity Set* after applying a *Unification Protocol*

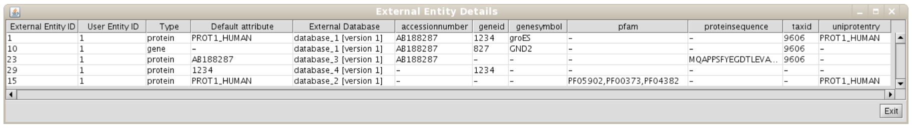
If you click on **View/Export set details**, and select all attributes, you will observe the 3 *User Entities* (Fig 1.15).



User Entity ID	Level	Degree	Tags	accessionnumber	geneid	genesymbol	pfam	proteinsequence	taxid	uniprotentry
1	0	0	-	AB188287	1234,...	GND2, groES	PF00373, PF04382, PF0...	MQAPPSFYEGDTLEVA...	9606	PROT1_HUMAN
2	0	0	-	AJ235272	2314	ftt-1, par-5	PF02245	MSWIDRIFSKNTTSNS...	9606	PROT2_HUMAN
6	0	0	-	-	1982	pgl	PF02245	-	10090	PROT1_MOUSE

Figure 1.15: Set details after applying a *Unification Protocol*

Finally, if you want to observe the *External Entity* details of one or multiple *User Entities*, select the rows on the table and click on the **View Details** button. For example, if you look at the details for the *User Entity* with ID 1, you will observe this *User Entity* is composed by *External Entity*s coming from all parsed *External Databases* (Fig 1.16). You can check that all unification rules you have applied have been used.



External Entity Details

External Entity ID	User Entity ID	Type	Default attribute	External Database	accessionnumber	geneid	genesymbol	pfam	proteinsequence	taxid	uniprotentry
1	1	protein	PROT1_HUMAN	database_1 [version 1]	AB188287	1234	groES	–	–	9606	PROT1_HUMAN
10	1	gene	–	database_1 [version 1]	AB188287	827	GND2	–	–	9606	–
23	1	protein	AB188287	database_3 [version 1]	AB188287	–	–	–	MQAPPSFYEGDTLEVA...	9606	–
29	1	protein	1234	database_4 [version 1]	–	1234	–	–	–	–	–
15	1	protein	PROT1_HUMAN	database_2 [version 1]	–	–	–	PF05902,PF00373,PF04382	–	–	PROT1_HUMAN

Figure 1.16: Viewing external entity details after the unification protocol. Note that all *External Entitys* belong to the same *User Entity*, and that *External Entitys* 1 and 15 share *uniprotentry* code, *External Entitys* 1, 10 and 23 share *NCBI Accession Number code*, and 1 and 23 share *geneid code*.

1.7 Creating unification protocols with rules having multiple attributes

In a single *BIANA Database*, you can create multiple *Unification Protocols*. Let's try other examples of *Unification Protocols*.

In this example, you only have an *External Database* that tells you information about protein sequence. Let's try to create a *Unification Protocol* that only uses an *External Database*, and unifies by the attribute *proteinsequence*. To do it, repeat the process of creating a unification protocol adding the following parameters:

***Unification Protocol* name** : protein sequence unification

Crossed Attribute : proteinsequence

External Databases : database_3

Then, follow the following steps:

1. Start a new *BIANA Session* with the tutorial database and the *Unification Protocol protein sequence unification* (if you are using our test *BIANA Database*, select the *Unification Protocoltutorial_exercise1_protseq*).
2. Create a new *User Entity Set*, with the nodes with *Accession Number* AB188287 (you should obtain a single *User Entity* with id 1).
3. View details of the set
4. View the details of the node

You will notice *User Entity* with ID 1 is composed by 3 entries coming from the same *External Database*. All of them have been unified because they share exactly the same sequence. However, two of the external entities are from human (taxonomy ID 9606) and the other from mouse (taxonomy ID 10090). In some cases it can be interesting to do this unification, but imagine you don't want to merge different *External Entity*s with the same sequence but different specie. In this case, you can create a *Unification Protocol* rule that incorporates more than a single attribute. To do this, follow the following steps:

1. Create a new unification protocol as before, but select two *Crossed Attributes* at the same time: proteinsequence and taxid. Name it *sequence and taxid unification*.

2. Start a new *BIANA Session* with the tutorial database and the *sequence and taxid unification Unification Protocol*(if you are using our test *BIANA Database*, select the *Unification Protocoltutorial_exercise1_protseq_taxid*).

3. Create a new *User Entity Set*, with the nodes with *Accession Number* AB188287 (you should obtain a single *User Entity* with id 1).

4. View details of the set

5. View the details of the node

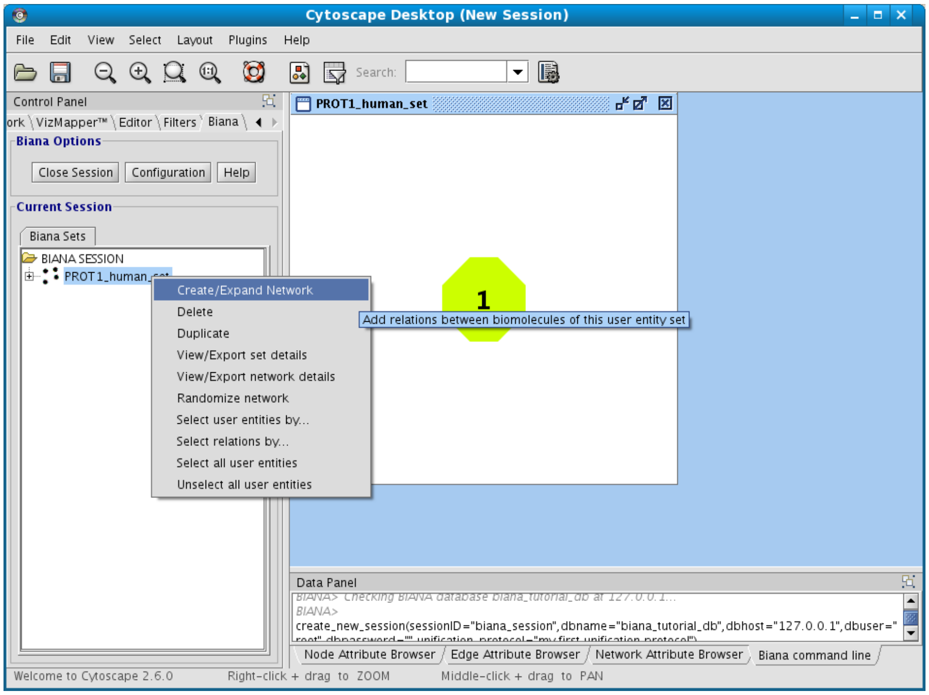
Now, you will notice *User Entity* with ID 1 is composed by only the 2 *External Entity*s that share at the same time the protein sequence and the taxonomy ID.

1.8 Creating my first network

If you have an opened *BIANA Session*, close it and start a new *BIANA Session* using the *biana_tutorial_db* and the first *Unification Protocol* you have created (*my first unification protocol*) (if you are using our online test *BIANA Database*, select the *Unification Protocoltutorial_exercise1_protseq_taxid*).

First of all, you will create a network of interactions. Follow next steps:

1. Create a new *User Entity Set*, using as initial attributes the *uniprotentry* identifiers *PROT1_HUMAN* and *PROT2_HUMAN*. Name it *my first network*.



Add relations between biomolecules of this user entity set

```
BIANA> Checking BIANA database biana_tutorial_db at 127.0.0.1...
BIANA>
create_new_session(sessionID="biana_session",dbname="biana_tutorial_db",dbhost="127.0.0.1",dbuser="
root",dbpassword="",unification_protocol="my first unification protocol")
```

Figure 1.17: Creating a new BIANA Session.

2. Right-click on the set (in the *BIANA Tree*, in the Cytoscape left panel), and select the option **Create/Expand Network** (Fig 1.17).

3. Click on **Add relations** checkbox. A new pop-up window for details will appear. If it does not appear, click on **Details** button.

4. In the relation types list, select *interaction* type, and then click the **Accept** button (Fig 1.18).

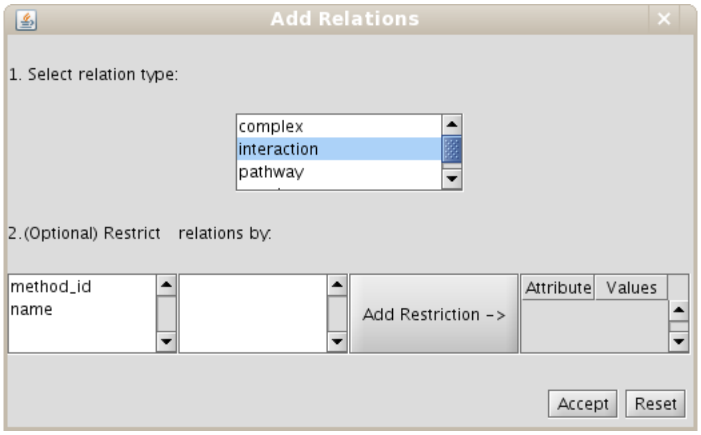


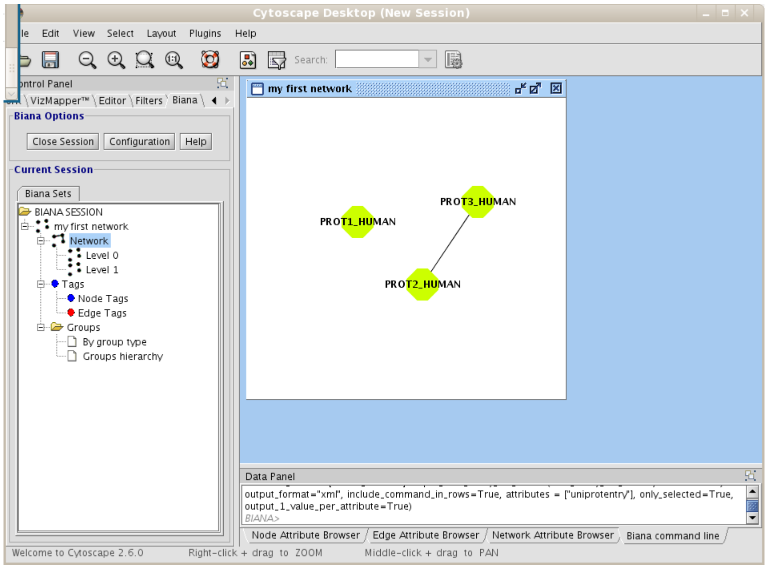
Figure 1.18: Creating a new network details.

5. Don't change any other parameter. Notice that the default network depth is 1, and the option of adding the relations between nodes at the last level(depth) is selected.

6. Click on the **Accept** button.

As a result, the original data set, which only had 2 nodes, will have a new node and an edge joining two of the nodes. You can use the option **View/Export set details** and change the *Cytoscape* VIZMAP to label the nodes according to *uniprotentry* codes. You should observe the same results as in Fig 1.19).

You should notice that the two initial nodes have been used as **seed**s to generate the network (also called **root** nodes). They belong to **Level 0**. If you expand the set node in the *BIANA Tree* (see Fig 1.19), and expand the **Network** tree node, you will observe that the network contains two levels. If you focus any of these nodes (*Level 0* or *Level 1*), the nodes belonging to the selected level will be highlighted and selected.



```
output_format="xml", include_command_in_rows=True, attributes = ["uniprotentry"], only_selected=True,
output_1_value_per_attribute=True)
BIANA>
```

Figure 1.19: My first network using *BIANA*.

You can notice also that there is no interaction for protein *PROT1_HUMAN*, and an interaction for protein *PROT2_HUMAN*, that interacts with protein *PROT3_HUMAN*. The number of edges of each node is called the **degree** of the node in the network: *PROT1_HUMAN* has degree 0 and *PROT2_HUMAN* and *PROT3_HUMAN* have degree 1. These node descriptors are shown when you execute the option **View/export set details**.

If you are interested in adding further levels of interactions to the network, you can click again on **Create/Expand Network** on the set options. If you click it again, you will notice *BIANA* adds a new protein (*PROT4_HUMAN*) that is interacting with *PROT3_HUMAN*) without asking for network creation criteria again (thus using the same criteria).

You can repeat the command **Create/Expand Network** command iteratively. In this example, you will be able to create a network with 6 levels (from 0 to 5) (Fig 1.20). You can also view specific network node details by selecting nodes with the mouse (see *Cytoscape* documentation on selecting network nodes), and then right-clicking on the selected nodes. In the pop-up menu that appears, select the option *BIANA* → **View entity details** (Fig 1.21).

Just as an introduction (you will see more in details in other examples), you can select nodes by different criteria, and tag them. Try to add and see if you can do it... If not, don't worry, you will return tagging later.

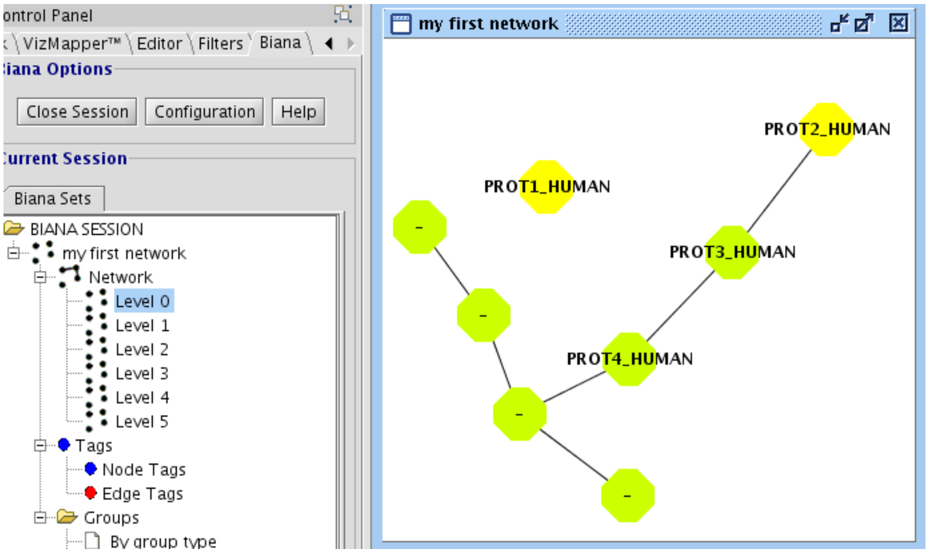


Figure 1.20: My first network to level 5. Note that nodes have been labeled using the code *uniprotentry*. Some of the nodes have not any value for this attribute, so, it is not showed.

> Once you create a network, you cannot modify its parameters. If you want to create a network with different criteria, you must create a new set and then create the network or combine different sets with created networks.

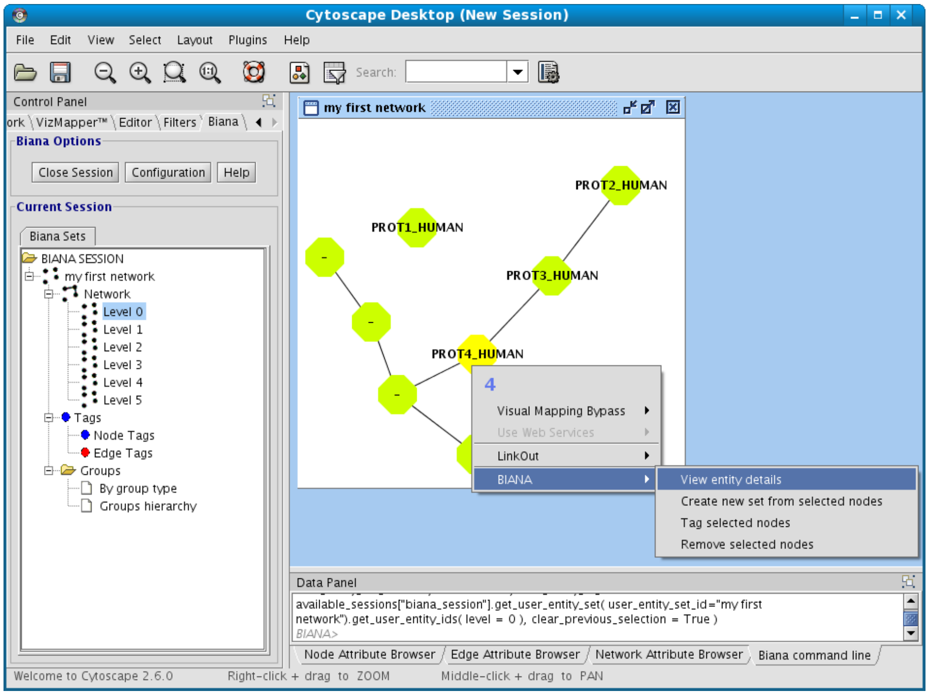
1.9 Creating different kind of networks

1.9.1 Different ways of network representation

In the last example, you have seen how to create a network of relationships of the type *"interaction"* between *User Entities*. However, it would be interesting to create networks that contain different kind of relations. In our example database (see details on file input_database4.txt), you have inserted the following relation types: *"interaction","complex","reaction", and pathways"*.

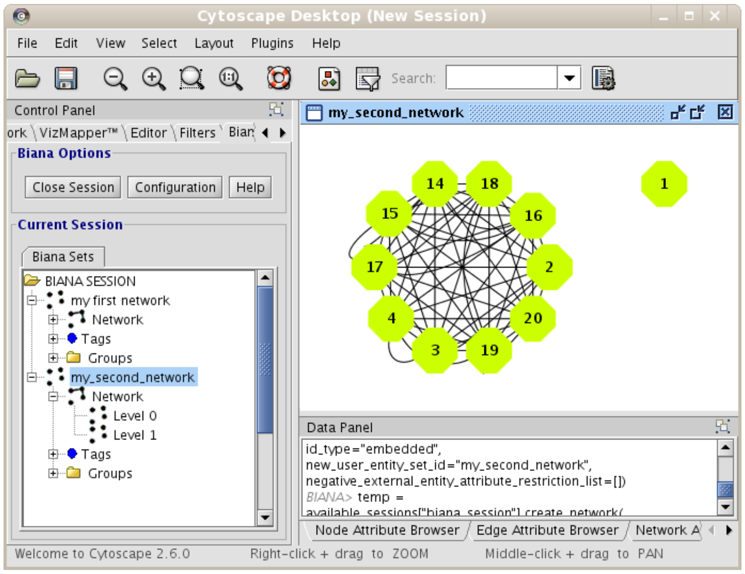
If you repeat all the last steps of the previous example, and select all this type of relations, you should observe a network with 11 nodes an 57 edges (see Fig 1.22).

Why is the network represented like this? Because in *BIANA*, all re-



```
available_sessions["biana_session"].get_user_entity_set( user_entity_set_id="my first
network").get_user_entity_ids( level = 0 ), clear_previous_selection = True )
BIANA>
```

Figure 1.21: Node pop-up menu.



```
id_type="embedded",
new_user_entity_set_id="my_second_network",
negative_external_entity_attribute_restriction_list=[])
BIANA> temp =
available_sessions["biana_session"].create_network(
```

Figure 1.22: Network at level 1 using all types of interactions. A Circular Layout has been applied for the representation.

lations between *User Entities* are stored as edges. In our example, there is a complex (named *ABC Complex*) composed by 4 proteins. In *BIANA*, these four nodes are all linked between them because the are part of the same complex. In the graphical representation, it means having 6 edges for this relation. Furthermore, you have specified to create networks also using *pathway* relations. In our example, you have inserted three pathways that contain most of the entities. All entities belonging to the same pathways will be linked between them because they are involved in the same *pathway*.

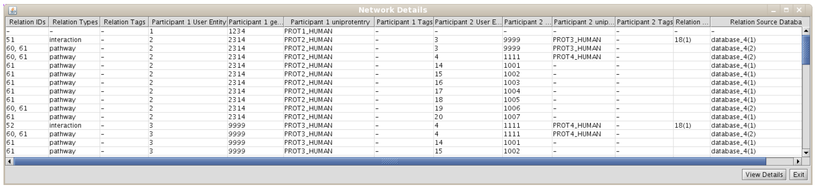
To look into the details of the interactions:

1. Right-click on your set you want to see network details in the *BIANA Tree*.

2. Select the **View/export Network Details** in the pop-up menu.

3. Select the participant and relation attributes to be printed and click **Accept**.

4. A new window will appear displaying the detailed info about the relations (Fig 1.23)



Relation IDs	Relation Types	Relation Tags	Participant 1 User Entity	Participant 1 ge...	Participant 1 uniprotentry	Participant 1 Tags	Participant 2 User E...	Participant 2 ...	Participant 2 unip...	Participant 2 Tags	Relation ...	Relation Source Databa
-	-	-	1	1234	PROT1_HUMAN	-	-	-	-	-	-	-
51	interaction	-	2	2314	PROT2_HUMAN	-	3	9999	PROT3_HUMAN	-	18(1)	database_4(1)
60, 61	pathway	-	2	2314	PROT2_HUMAN	-	3	9999	PROT3_HUMAN	-		database_4(2)
60, 61	pathway	-	2	2314	PROT2_HUMAN	-	4	1111	PROT4_HUMAN	-		database_4(2)
61	pathway	-	2	2314	PROT2_HUMAN	-	14	1001	-	-		database_4(1)
61	pathway	-	2	2314	PROT2_HUMAN	-	15	1002	-	-		database_4(1)
61	pathway	-	2	2314	PROT2_HUMAN	-	16	1003	-	-		database_4(1)
61	pathway	-	2	2314	PROT2_HUMAN	-	17	1004	-	-		database_4(1)
61	pathway	-	2	2314	PROT2_HUMAN	-	18	1005	-	-		database_4(1)
60, 61	pathway	-	2	2314	PROT2_HUMAN	-	19	1006	-	-		database_4(2)
61	pathway	-	2	2314	PROT2_HUMAN	-	20	1007	-	-		database_4(1)
52	interaction	-	3	9999	PROT3_HUMAN	-	4	1111	PROT4_HUMAN	-	18(1)	database_4(1)
60, 61	pathway	-	3	9999	PROT3_HUMAN	-	4	1111	PROT4_HUMAN	-		database_4(2)
61	pathway	-	3	9999	PROT3_HUMAN	-	14	1001	-	-		database_4(1)
61	pathway	-	3	9999	PROT3_HUMAN	-	15	1002	-	-		database_4(1)

Figure 1.23: Network details table.

It is important to notice that, in difference with the last network, all nodes (except roots) have appeared at **level 1**. This is because all nodes are found to possess a relation with a root protein at the first level using the relation types you have specified.

This information can be useful, but of course not represented in this way. There are two ways of changing it:

1. Change the way *Cytoscape* shows a relation. For example, a relation of the type *complex* can be interesting to be represented as a node in the graph, and all participants linked to this node. To change the way a type of relationship is represented:

   (a) Click on **Configuration** button in *BIANA* menu, and select **Preferences**.

   (b) Select the relation types you want to represent as nodes.

   This option only changes the way *Cytoscape* represents the relations. It does not affect anything on how *BIANA* works.

2. Add relationships as **groups** instead of as edges. In this case, new nodes are added to the network, but no new edges are added. A new *group* is created for each relationship belonging to these relationship types, and all *User Entities* participating in these relationships will be added to the groups respectively. To do this, add the relation types in the category *Add relations as GROUPS* instead of *Add Relations*, in the **Create/Expand Network** window.

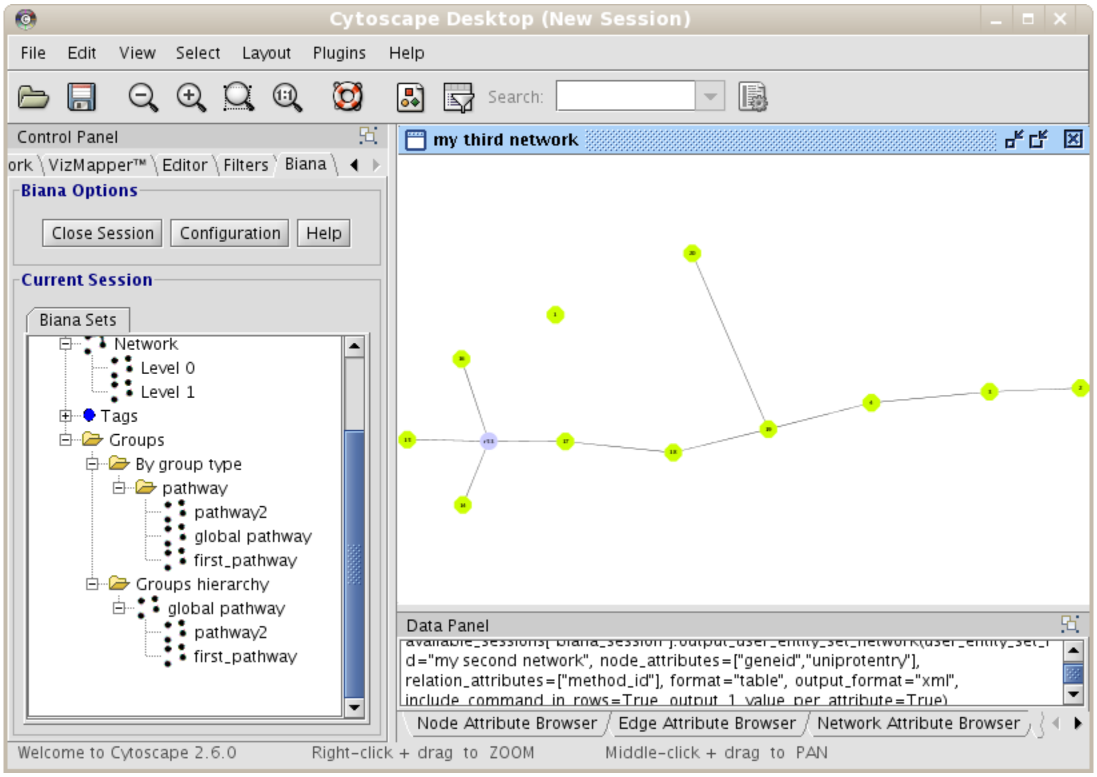
In our example, follow the following steps:

1. Create a new data set with the same root proteins (*PROT1_HUMAN and PROT2_HUMAN*).

2. Change the *Complex* relation type to be represented as a node.

3. Create the network. In the network dialog window, select *interaction, complex and reaction* in relation types in the **Add Relations** category, and *pathway* in the **Add relations as GROUPS** category.

You should obtain a network as in Fig 1.24. You must notice that a network of **level 1** has been created, and that groups have appeared in the *BIANA Tree* (left *Cytoscape* panel). If you click on the groups of nodes, nodes belonging to these groups are selected. In this example, you have introduced two pathways (*first_pathway* and *pathway2*), which belong to another pathway (*global pathway*). This hierarchy is represented in the *BIANA Tree*.


Cytoscape Desktop (New Session)
File Edit View Select Layout Plugins Help
Search:
Control Panel
ork VizMapper™ Editor Filters Biana
Biana Options
Close Session Configuration Help
Current Session
Biana Sets
Network
Level 0
Level 1
Tags
Groups
By group type
pathway
pathway2
global pathway
first_pathway
Groups hierarchy
global pathway
pathway2
first_pathway
my third network
Data Panel
Node Attribute Browser Edge Attribute Browser Network Attribute Browser
Welcome to Cytoscape 2.6.0 Right-click + drag to ZOOM Middle-click + drag to PAN

```
available_sessions["biana_session"].output_user_entity_set_network(user_entity_set_i
d="my second network", node_attributes=["geneid","uniprotentry"],
relation_attributes=["method_id"], format="table", output_format="xml",
include_command_in_rows=True, output_1_value_per_attribute=True)
```

Figure 1.24: Network at level 1 for *PROT1_HUMAN and PROT2_HUMAN*. *Complex* relations have been represented graphically as nodes (purple node), and *pathway* relations have been inserted as groups instead of edges of the network.

Relationships added as **groups** are not added as edges in the network. Consequently, they won't affect node degree, network topology, etc.

1.9.2 Creating a network of attributes

BIANA offers the possibility to create a network by defining an edge between two entities that share one or more attributes. To illustrate this, you will create a set with all entities that have a *uniprotentry* identifier, and a network of relationships by sharing the same specie (Fig 1.25).

1. Create a new set of data with all entities that have a *uniprotentry* identifier. To do this, select the *uniprotentry* attribute and add the character ***** as value.
2. Go to the **Create/Expand Network** menu, select the option **Add attribute relations** and add the attribute *taxid*.
3. Select the option to create a network at **level 0**. This will create the network of relationships using only the nodes from the initial set.
4. Click the **Accept** button.

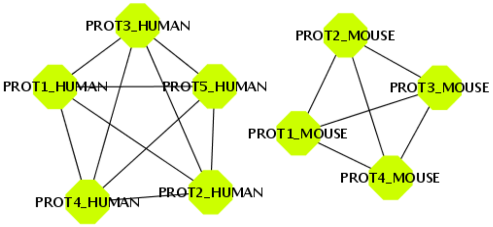
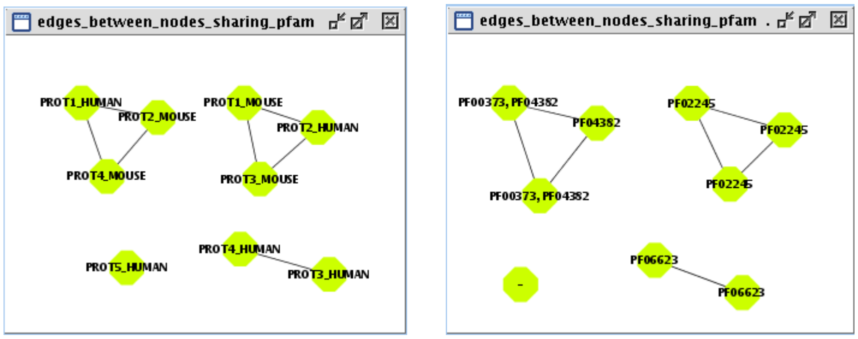


Figure 1.25: Taxonomy ID network at level 0 for all entities which have at least one *uniprotentry* identifier.

Let's try a more complicated example. Create the same data set as before (all entities that have a *uniprotentry* identifier), and a network at level 0 of attribute relations by selecting the attribute *pfam*. An edge will appear for those entities that share at least a *PFAM* domain (Fig 1.26). You can see that *PROT1_HUMAN* shares a *pfam* domain with *PROT4_MOUSE*.

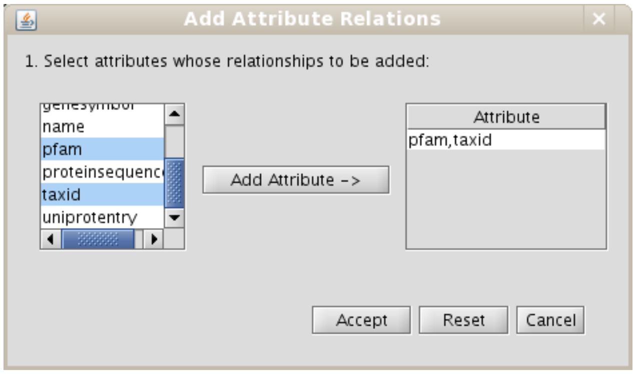


(a)

(b)

Figure 1.26: *PFAM* network at level 0 for all entities which have at least one *uniprotentry* identifier. **(a)** Network nodes labeled with *uniprotentry* codes. **(b)** The same network with nodes labeled with *pfam*.

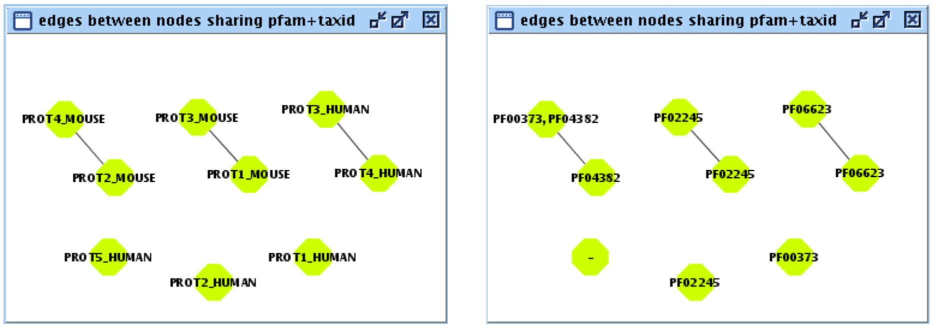
Let's imagine you want to see these edges because they share more than a single attribute at the same time. To illustrate this, execute exactly the same example, but selecting attributes *pfam* and *taxid* at the same time (Fig 1.27). It is not the same adding one attribute and consecutively the other, than selecting them at the same time: In the first option (add consecutively different attributes), an edge will be created if they share ANY of the selected attributes. In contrast, in the second option, an edge will be created if they share ALL selected attributes at the same time. In the new network (Fig 1.28), you should notice that some edges have disappeared compared with the last network (Fig 1.26).


Add Attribute Relations

1. Select attributes whose relationships to be added:

Attribute
pfam,taxid

Figure 1.27: Details of selecting attributes for creating an attribute network.



(a) (b)

Figure 1.28: Network at level 0 for all entities which have at least one *uniprotentry* identifier. Edges are created for entities sharing *PFAM* and *taxid* at the same time.**(a)** Network nodes labeled with *uniprotentry* codes. **(b)** The same network with nodes labeled with *pfam* domains.

1.9.3 Making predictions: Expanding relationships between nodes sharing one or more attributes

In some cases, it may be interesting to extrapolate edges between two entities to other two entities because they share some attributes (for example, an interaction has been reported between two proteins in a specie, and you want to extrapolate this interaction to their homologous proteins in other specie (*interologs*). *BIANA* allows to extrapolate any type of relationships between nodes if they share one or more attributes at the same time (attributes can be: *sequence homology, PFAM domains, structural domains*, etc... In **Chapter 5**, we show a practical example on how expand by sequence homology. In the current example, you will expand edges between nodes sharing *PFAM* domains.

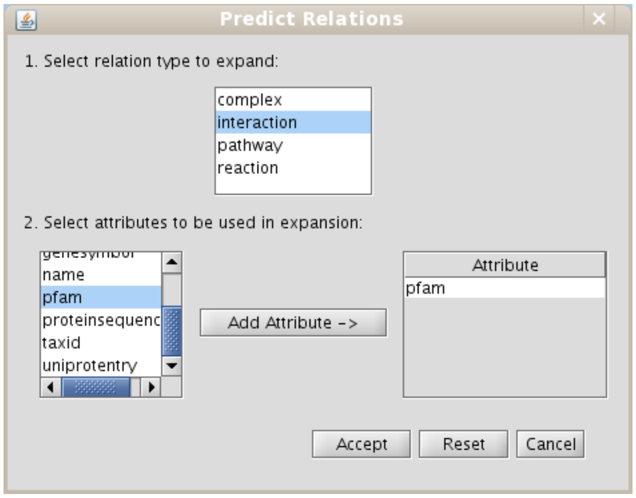
Follow next steps:

1. Create a new set for all human proteins (adding the *taxid* 9606). Add also *taxid* 9606 as a restriction (this restriction means all nodes in this set must have this attribute. Initial restrictions are applied in all operations in the data set).

2. Select the command **Create/Expand network**. In the network creation dialog:

   (a) Select the *interaction* relation type in the section **Add relations**

(b) Select the *interaction* relation type in the section **Make predictions by sharing attributes** and add the *PFAM* attribute to do expansions (Fig 1.29)



Predict Relations

1. Select relation type to expand:

complex
interaction
pathway
reaction

2. Select attributes to be used in expansion:

genesymbol
name
pfam
proteinsequenc
taxid
uniprotentry

Attribute
pfam

Figure 1.29: Predict relations by expansions dialog

(c) Select **Level 0**

3. Click **Accept**

After that, you should obtain a network similar to Fig 1.30. If you compare with the previous network, three new interactions are predicted (between *PROT1_HUMAN* and *PROT2_HUMAN*; *PROT2_HUMAN* and *PROT4_HUMAN*; *PROT2_HUMAN* and *PROT3_HUMAN*). For instance, a new edge corresponding to a prediction has appeared between *PROT1_HUMAN* and *PROT2_HUMAN*, because they share *PFAM domains* with two mouse interacting proteins: *PROT2_MOUSE* and *PROT1_MOUSE* respectively (you can check they interact by creating a new network of interactions for these proteins or for all mouse proteins).

1.10 Removing *Unification Protocols*

Before continuing, you are going to delete last *Unification Protocols* you have created. If you are using our test *BIANA Database*, skip this section, as you don't have permissions to delete a *Unification Protocol*.

Follow the following steps to delete one or more unification protocols:

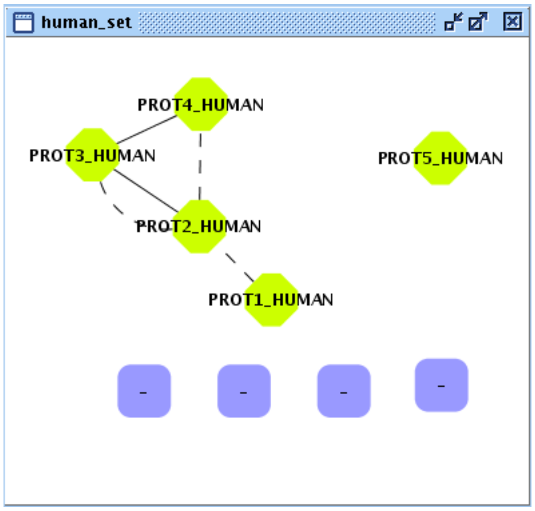


Figure 1.30: Network for human proteins. *Interactions* are showed as solid black lines and predictions by sharing *PFAM* are showed as dashed black lines.

- Click on **Configuration** menu and then click on **Delete *Unification Protocol***.
- Select the *BIANA Database* in which you want to delete one or more *Unification Protocol*.
- Click the **Accept** button.

> This action is not reversible. Once you delete a *Unification Protocol*, you can not revert the changes. If you want a backup, see the example on the tutorial.

1.11 Removing *BIANA Databases*

Finally, to finish this exercise, you are going to delete *BIANA Database* you have created for this exercise. If you are using our test *BIANA Database*, skip this section, as you don't have permissions to delete a *BIANA Database*.

Follow the following steps:

- Click on **Configuration** menu and then click on **Delete *BIANA Database***.

- Select one or multiple *BIANA Database*s you want to delete.
- Click the **Accept** button.

This action is not reversible! Once you delete a *Unification Protocol* or *BIANA Database*, you can not revert the changes! If you want a backup, see the example on the tutorial.

Chapter 2

Working with Multiple Data Sets

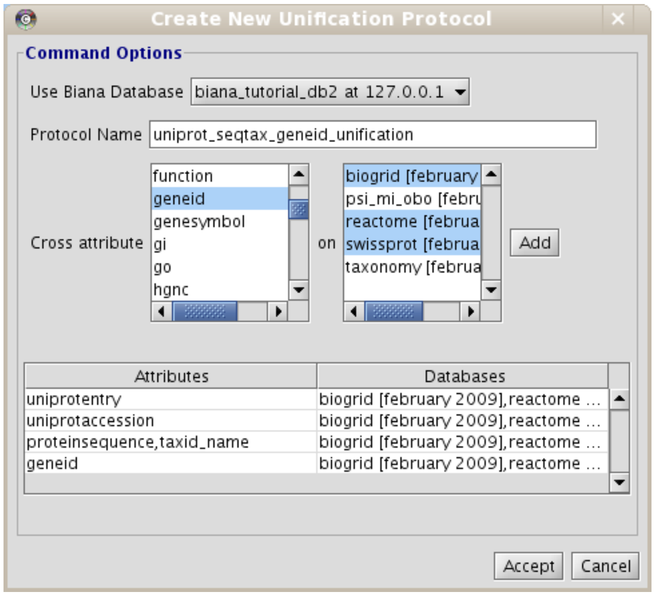
In this chapter, you will go over most of the operations that can be done with one or multiple data sets, such as union, intersection, selection, tagging, etc.

Before you start, you will create a new *BIANA* database with real biological data (skip this section if you are using the online test *BIANA Database*):

1. **Create a new BIANA database**: Name it *biana_tutorial2*.
2. **Parse external databases**: Parse the following databases.
   (a) Uniprot SwissProt
   (b) Reactome
   (c) Taxonomy
   (d) PSI-MI OBO
   (e) IntAct

   To know which parser you have to select for each database and which files are needed, check the *BIANA* reference manual *Approximate parsing time is less than 1h on a computer having 4 64bit 2.33GHz speed dual core processors with 2GB RAM.*
3. **Create the Unification Protocol**: Use the following unification protocol and name it *uniprot_seqtax_geneid_unification* (Fig 2.1):
   - Attribute *uniprotentry* between *SwissProt,Reactome and IntAct.*

- Attribute *uniprotaccession* between *SwissProt,Reactome and IntAct*.
- Attributes *proteinsequence* and *taxid* between *SwissProt,Reactome and IntAct*.
- Attribute *geneid* between all databases.



Use Biana Database: biana_tutorial_db2 at 127.0.0.1

Protocol Name: uniprot_seqtax_geneid_unification

Attributes	Databases
uniprotentry	biogrid [february 2009],reactome ...
uniprotaccession	biogrid [february 2009],reactome ...
proteinsequence,taxid_name	biogrid [february 2009],reactome ...
geneid	biogrid [february 2009],reactome ...

Figure 2.1: Unification Dialog screenshot

2.1 Generate different datasets with initial sets of biomolecules of interest

Once you have unified, start a *BIANA* working session with the *Unification Protocol* you have created (if you are using our online test *BIANA Database*, use the *Unification Protocoltutorial_exercise2_uniprot_seqtax_geneid_unification*). Then, create the following *User Entity* sets using the following data:

1. Alzheimer set: keyword *Alzheimer* in their *Description, Name and Disease* attributes.
2. Diabetes set: keyword *Diabetes* in their *Description, Name, Function and Disease* attributes.

3. Human Diabetes set.
   (a) keyword *Diabetes* in their *Description, Name, Function and Disease* attributes.
   (b) Only human (by adding attribute *taxid 9606* as restriction OR *taxid_name human*) Notice that here you can use the attribute *taxid_name* because you have inserted *Taxonomy* database into *BIANA*. Otherwise, you would not be able to use this attribute. (See "special" attributes section in the *BIANA* manual).

2.2 Union and intersection of different data sets

You can obtain a new *User Entities* by the combination of others. For this, you can select multiple *User Entity Set* in the *BIANA* tree. Then, right-click and select the operation you want to perform. For example, you can do:

- Intersection between *Alzheimer Set* and *Diabetes Set.*
- Create a network of interactions at level 1 for *Alzheimer Set* and *Diabetes Set* and the obtain again a new *User Entities* with the intersection between them.

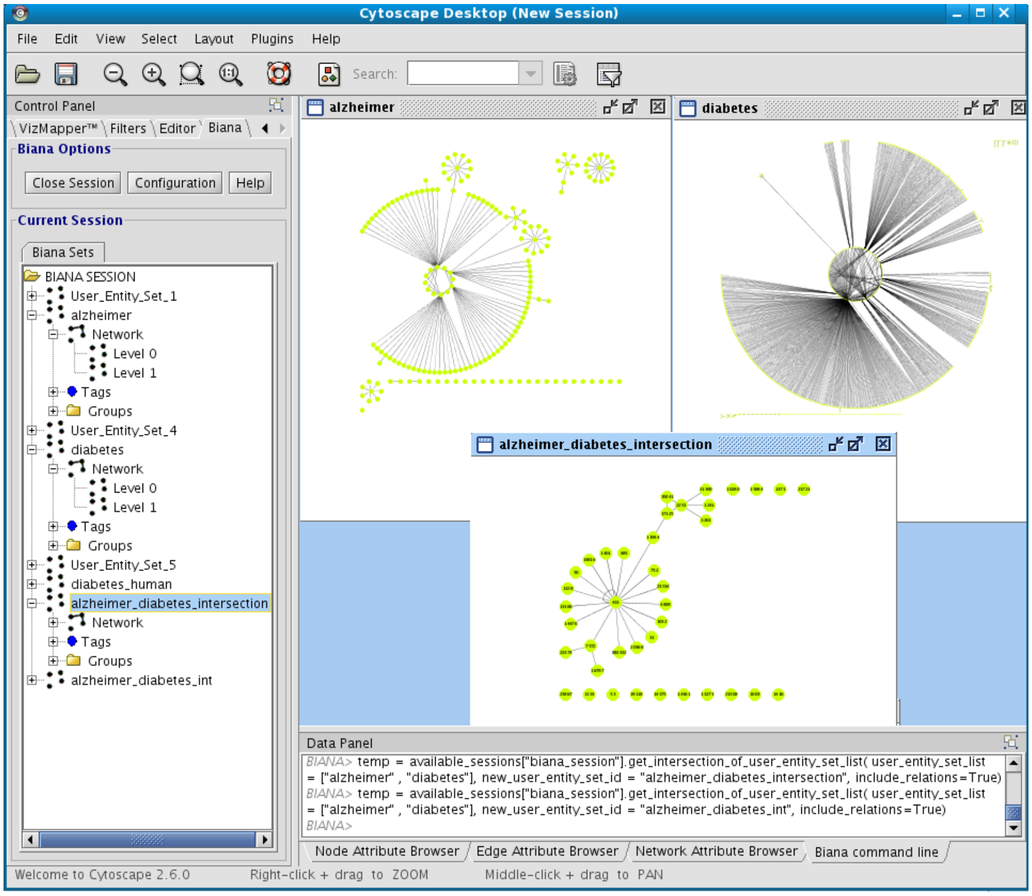
The resulting networks would be similar to the ones demonstrated in Fig 2.2.

2.3 Selecting a subset of entities by having some attributes or tags

You can select *User Entities* in a *User Entity Set* using different criteria. For this:

1. Right-click on the *User Entity Set* whose *User Entities* you want to select.
2. Select the option **Select user entities by attribute/tag**.
3. Specify the attributes

For example, select in the *Alzheimer Set* all entities that have the keyword *Diabetes* in their *Description, Name, Function and Disease* attributes. You must observe selected the same *User Entities* that have appeared in the intersection between *Alzheimer Set* and *Diabetes Set.*



```
BIANA> temp = available_sessions["biana_session"].get_intersection_of_user_entity_set_list( user_entity_set_list
= ["alzheimer" , "diabetes"], new_user_entity_set_id = "alzheimer_diabetes_intersection", include_relations=True)
BIANA> temp = available_sessions["biana_session"].get_intersection_of_user_entity_set_list( user_entity_set_list
= ["alzheimer" , "diabetes"], new_user_entity_set_id = "alzheimer_diabetes_int", include_relations=True)
BIANA>
```

Figure 2.2: Alzheimer and Diabetes networks and their intersection.

2.4 Selecting a subset of entities from an existing set

You can select *User Entities* in a *User Entity Set* that also belong to another *User Entity Set*. This can be thought as intersection of the *User Entities* (nodes) of two *User Entity Set* without the need of creating a new set (usually used in combination with tagging). For this:

1. Right-click on the *User Entity Set* whose *User Entities* you want to select.
2. Select the option **Select user entities from existing set**.
3. In the pop-up window, select *User Entity Set* whose *User Entities* will be checked for existence in the current *User Entity Set*.

For example, select in the *Alzheimer Set* that also belong to *Diabetes Set*. You must observe selected the same *User Entities* that have appeared in the intersection between *Alzheimer Set* and *Diabetes Set*.

2.5 Tagging biomolecules using different criteria

You can tag selected nodes and edges to get them again later without having to select them again.

To tag nodes or edges, you must Right-click on one selected element in the *Cytoscape* plugin, and select the option **BIANA** → **Tag selected nodes**.

For example, tag the last selection you have done with the tag *Intersection*.

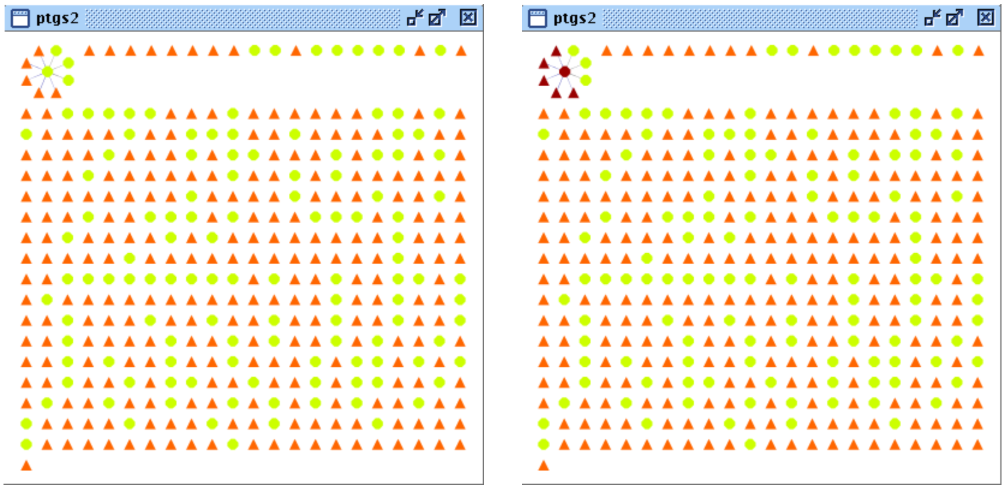
Chapter 3

Handling Indirect Biological Relations: Pathways as Groups of Biomolecules

In this chapter, you will exploit capabilities of *BIANA* to gather and analyze interaction information of a molecule of interest when the interaction information is in the form of indirect relation such as belonging to a particular pathway. In this example you will work with an inducible isozyme of Prostaglandin-endoperoxide synthase, PTGS2 that plays a key role in biosynthesis of the prostanoids. Use the *BIANA Database* you have created in **Chapter 2**.

You will first create the set of biomolecules associated with PTGS2 and then create its network.

1. **Create a *BIANA* working session** with the *Unification Protocol* that has been created in the second chapter.
2. **Create a new *User Entity Set*** with entries associated with "PTGS2" in their *name* attribute and call it as "ptgs2".
3. **Create/Expand Network** for the *User Entity Set*
   (a) Add all relation types but "pathway"s excluding last level relations in **Add relations**.
   (b) Add "pathway" relations as groups in **Add relations as groups**.
4. **View/Export set details** selecting *uniprotentry* and change the *Cytoscape* VIZMAP to label the nodes according to *uniprotentry* codes.



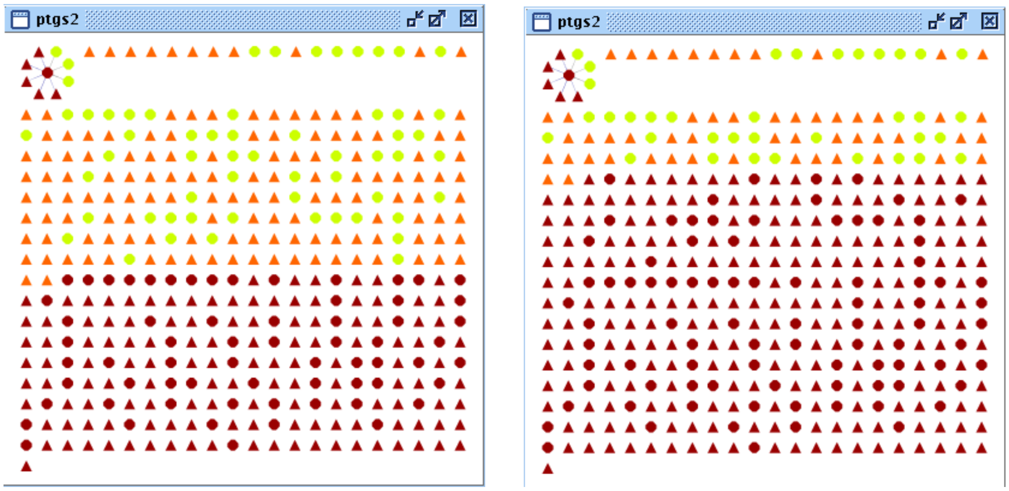
(a) Initial view.

(b) Snapshot of PTGS2 network where REACTOME pathway 13433 is selected.

Figure 3.1: PTGS2 network where relations other than pathway type are included and pathways are included as groups.

You should be observing a network similar to the one in Fig 3.1-a. Now go to *Groups* on the *BIANA Tree* and expand *Groups hierarchy*. Select the nodes with ids reactome:13433, reactome:13705 and reactome:1396 respectively. Note that different nodes corresponding to selected pathways are selected (Fig 3.1-b, 3.2-a, 3.2-b).

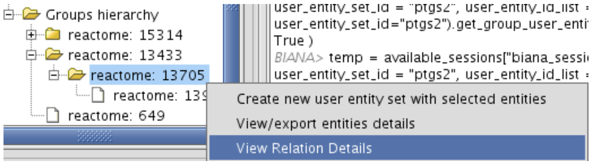
To get details of a relation (in our case a specific pathway), go to *Groups hierarchy* on the *BIANA Tree* and right-click on the tree node, e.g. reactome:13705, corresponding to the relation (Fig 3.3). Then, on the pop-up menu click on **View Relation Details**. Information associated with this relation will be displayed in a new window as demonstrated in Fig 3.4.



(a) Snapshot of PTGS2 network where REACTOME pathway 13705 is selected.

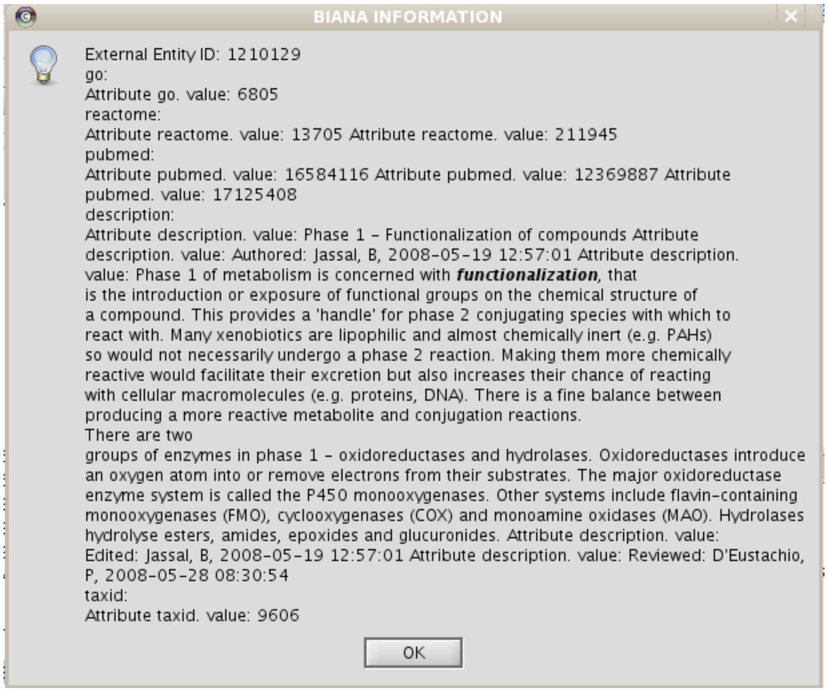
(b) Snapshot of PTGS2 network where REACTOME pathway 1396 is selected.

Figure 3.2: PTGS2 network where relations other than pathway type are included and pathways are included as groups.



```
user_entity_set_id = "ptgs2", user_entity_id_list =
user_entity_set_id="ptgs2").get_group_user_enti
True )
BIANA> temp = available_sessions["biana_sessi
user_entity_set_id = "ptgs2", user_entity_id_list =
```

Figure 3.3: Displaying relation details inserted as a group.



```
External Entity ID: 1210129
go:
Attribute go. value: 6805
reactome:
Attribute reactome. value: 13705 Attribute reactome. value: 211945
pubmed:
Attribute pubmed. value: 16584116 Attribute pubmed. value: 12369887 Attribute
pubmed. value: 17125408
description:
Attribute description. value: Phase 1 - Functionalization of compounds Attribute
description. value: Authored: Jassal, B, 2008-05-19 12:57:01 Attribute description.
value: Phase 1 of metabolism is concerned with functionalization, that
is the introduction or exposure of functional groups on the chemical structure of
a compound. This provides a 'handle' for phase 2 conjugating species with which to
react with. Many xenobiotics are lipophilic and almost chemically inert (e.g. PAHs)
so would not necessarily undergo a phase 2 reaction. Making them more chemically
reactive would facilitate their excretion but also increases their chance of reacting
with cellular macromolecules (e.g. proteins, DNA). There is a fine balance between
producing a more reactive metabolite and conjugation reactions.
There are two
groups of enzymes in phase 1 - oxidoreductases and hydrolases. Oxidoreductases introduce
an oxygen atom into or remove electrons from their substrates. The major oxidoreductase
enzyme system is called the P450 monooxygenases. Other systems include flavin-containing
monooxygenases (FMO), cyclooxygenases (COX) and monoamine oxidases (MAO). Hydrolases
hydrolyse esters, amides, epoxides and glucuronides. Attribute description. value:
Edited: Jassal, B, 2008-05-19 12:57:01 Attribute description. value: Reviewed: D'Eustachio,
P, 2008-05-28 08:30:54
taxid:
Attribute taxid. value: 9606
```

Figure 3.4: Details of reactome:13705 pathway.

Chapter 4

Introducing User Defined Data

BIANA provides some default database parsers for most common databases and formats. *BIANA* has been designed to be able to store any kind of biologic database, relying on the needs of the user to integrate data between databases by choosing which combinations of attributes must be shared. However, due to the large number of different databases, formats and versions, not all databases with biologic data have a current BIANA Parser. Despite existing interchange standard formats, databases often change their formats, so parsers are not guaranteed to work in all database versions. In order to solve this problem, we provide a set of default parsers, that will be updated in the *BIANA* web page (http://sbi.imim.es/web/BIANA.php).

However, probably you are interested in having a parser for another *External Database*, or inserting your own data into *BIANA* to integrate it with other *External Database*.

There exist two ways to insert your own data into *BIANA*:

1. Convert your data to a specific format. *BIANA* proposes a tabulated format (see example files of the *theoretical example* you have seen in the first tutorial exercise). For more details on how is it the format, check the *BIANA* reference manual.

2. Create your own parser and add it to *BIANA*.

4.1 Introducing a user specific database

If you are using our test *BIANA Database*, skip *BIANA Database*creation and unification steps. You must use the test *BIANA Database Unification Protocol inferred_metabolic_network*.

In this example, insert a specific *External Database*by using a specific parser created specifically for it:

1. Download raw data from a database build in our lab (Dataset called *Metabolic Network* at *http://sbi.imim.es/web/BIANA.php?page=biana.datasets*.
2. Download specific parser for this dataset in the same webpage, and copy it under the biana/BianaParser directory (the location of this directory will depend on your installation procedure!).
3. Restart the Graphical Interface to make the new parser available.
4. Create a new Biana Database (see tutorial example 1) and populate it with:
   (a) Our specific *Metabolic Network* database using the downloaded parser (called *inferred_metabolic_network*).
   (b) Uniprot Swissprot
5. Unify using the following *Unification Protocol* atoms:
   - EC code

> If you are interested in learning how to create a parser, check at *http://sbi.imim.es/web/BIANA.php?page=biana.parsers*. You can use this parser as an example on how to create a *BIANA*Parser. It is advisable to check your new parsers in empty *BIANA Database*s to avoid inserting errors in existing *BIANA Database*s.

Introduction to our metabolic network database

This dataset corresponds to a metabolic network reconstruction that gives an score to each pair of possible chained enzymatic reactions. These chained reactions are possible when there is at least one chemical compound acting at the same time as product of enzyme A and substrate of enzyme B, a shared compound. The score is based in the plausibility of observing such chemical compounds, according to their own frequency and the frequency of other products of enzyme A and other substrates of enzyme B which do not take part in chaining reactions. Most of existing methods for reconstructing metabolic networks disregard relationships in which enzymes only share very frequent compounds. The method from which this dataset is derived tries to discern whether these relationships can be taken into consideration or not. Hence, cases in which a relationship between two enzymes can only be established due to the presence of shared common compounds are tagged as "special". Often, a given enzyme, described by an EC code, can perform a given reaction bidirectionally (from substrates to products and vice versa) and/or can perform more than a single reaction. The provided score is specific for the reaction each enzyme is performing in the relationship and their directions. It is recommended to use high reliability scores (pValue ¡= 0.05).

Database information description

In this *External Database*, we have the following information we want to insert into *BIANA*:

- ENZYME A: EC code of enzyme A
- ENZYME B: EC code of enzyme B
- Score: score for the relationship enzyme A enzyme B
- pValue: p-value associated to the score
- KEGG RA: KEGG reaction code for enzyme A
- KEGG RB: KEGG reaction code for enzyme B
- Dir A: direction of reaction of enzyme A (d: direct, from substrates to products; r: reverse, from products to substrates)
- Dir B: direction of reaction of enzyme B (d: direct, from substrates to products; r: reverse, from products to substrates)
- SPECIAL: True if the relationship corresponds to a special case; false if not.
- Common metabolites: number of metabolites allowing the relationship between the two reactions

With this information, we can determine:

- **External Entity Types**: *protein* for *External Entity* and *reaction* for *External Entity Relation*. They already exist in default *BIANA External Entity* types, it is not necessary to add them.
- **External Entity Attributes and External Entity Relation Attributes**:
  - *EC*. It already exists in default *BIANA External Entity* attributes, it is not necessary to add it.
- **External Entity Relation Attributes**:
  - *Score*. It already exists in default *BIANA External Entity* attributes, it is not necessary to add it.
  - *Pvalue*. It already exists in default *BIANA External Entity* attributes, it is not necessary to add it.
  - *Difficulty*. This is very specific for this database, and it does not exist in *BIANA* default *External Entity* attributes. To add it, add the value *(“Difficulty”,”enum(\”Special\”,\”Easy\”)”)* in the EXTERNAL_ENTITY_IDENTIFIER_ATTRIBUTES list in ***BIANA_GLOBALS*** file.
  - *CommonMetabolitesNumber*. This is very specific for this database, and it does not exist in *BIANA* default *External Entity* attributes. To add it, add the value *(“CommonMetabolitesNumber”,”integer(2)”)* in the EXTERNAL_ENTITY_IDENTIFIER_ATTRIBUTES list in ***BIANA_GLOBALS*** file.
- **External Entity Relation Participant Attributes**:
  - *Direction*. This is very specific for this database, and it does not exist in *BIANA* default *External Entity* relation participant attributes. To add it, add the value *(“Direction”,”enum(\”direct\”,\”reverse\””)* in the EXTERNAL_ENTITY_RELATION_PARTICIPANT_ATTRIBUTE_TYPES list in ***BIANA_GLOBALS*** file.
  - *KEGG*. It already exists in default *BIANA External Entity* attributes, it is not necessary to add it.

4.2 Create a network using your own data

Finally, you will create a network with your data.

Create an initial data set:

- All entities having an EC code
- All entities belonging to human (taxid 9606). Note that you cannot use the *taxid_name* as you have not inserted *Taxonomy External Database*.

Finally, create a network with the following criteria:

- Reactions having an score greater or equal to 1.5 (In **Add Relations** dialog add *score* as a relation restriction attribute with the value of “>= 1.5”)
- Reactions having an score greater to 1.5 and having the *difficulty* attribute ”special”

Chapter 5

Generating and Using Sequence Similarity Information for Predictions Based on Interologs

More often than not, we are interested in inferring relations through similarity between molecules. For this purpose it is necessary to know all vs all sequence similarity of biomolecules stored in BIANA. However, BIANA contains redundant sequence information coming from biological data sources in its data repository. To be able to make accurate predictions based on sequence similarity, first we need to eliminate duplicate sequences and then calculate sequence similarities. In this chapter, you will discover how to remove repeated sequences in a BIANA database and how to do sequence-based interaction predictions.

You will use the *BIANA Database* created in **Chapter 2**.

In order to execute this example, you must have installed in your computer the following software:

1. blastpgp and bl2seq (http://www.ncbi.nlm.nih.gov/BLAST/download.shtml)

2. CD-HIT (http://bioinformatics.ljcrf.edu/cd-hi/)

Then, it is necessary to modify the ***BIANA_GLOBALS*** file manually. It is located in our BIANA_INSTALLATION_PATH/biana_globals.py (if you installed *BIANA*as administrator, you will need to modify it as administrator!). At the end of this file, modify the following parameters:

1. FORMATDB_EXEC = 'YOUR_ABSOLUTE_BLAST_PATH/bin/formatdb'
2. BLASTALL_EXEC = 'YOUR_ABSOLUTE_BLAST_PATH/bin/blastall'
3. BL2SEQ_EXEC = 'YOUR_ABSOLUTE_BLAST_PATH/bin/bl2seq'
4. CD_HIT_PATH = 'YOUR_ABSOLUTE_CD_HIT_PATH'

5.1 Calculating similarity of non redundant sequences in the *BIANA Database*

As mentioned above, similarity of sequences in the database is of utmost interest while discovering inferred relationships between sequences. *BIANA* provides a shell script *(for POSIX based OSs)* to remove duplicated sequences and to calculate sequence similarity between all sequences in the database. This script, called *process_sequences.sh* can be found under `{scripts/administration/}` directory. The parameters of this script describing the database to use and the "identity-threshold" of CD-HIT clusters, are hard coded in the beginning of the file. Identity threshold of CD-HIT clusters (variable named "identity") can have values in-between 0 and 1 denoting the percentage of identity among members of the same cluster. All pairwise sequence alignment will be conducted between those elements in the same cluster afterwards. Once you set database-specific parameters and CD-HIT cluster identity parameter at the top of the file, make sure that program paths for BLAST and CD-HIT are properly set in ***BIANA_GLOBALS*** file. Then, execute the command line:

```
./process_sequences.sh
```

This script works in two steps as explained in the next sections (*Approximate running time for the example database with the specified configuration is less than one quarter of an hour*).

5.1.1 Removing sequence redundancy in the *BIANA Database*

First all the sequences in the *BIANA Database* are taken from database and duplicates are found using "shell" utilities like *sort* and *unique*. Then, non-redundant sequences are inserted back into the *BIANA Database*. At this step *eliminate_duplicate_sequences.py* python script is used to handle this task.

5.1.2 Calculating sequence similarities in *BIANA Database*

In the second step, calculate_sequence_similarities.py python script is used to get sequences from database, finding CD-HIT clusters, running BLAST and inserting the sequence similarity information to the database.

5.2 Making predictions using sequence similarity: Sequence interologs

In this example, you will exploit sequence interologs of a small signaling G protein (GTPase): RND3. Let's start with creating the set of user entities associated with this biomolecule.

1. **Create a *BIANA* working session** with the *Unification Protocol* that has been created in the second chapter.
2. **Create a new *User Entity Set*** with entries associated with "RND3" as geneSymbol and name it as "rnd3".
3. **Duplicate** created *User Entity Set* by right clicking on the newly created set on *BIANA Tree* and name it as "rnd3_copy".

Before predicting new interactions, first let's check which interactions are available for the molecules associated with RND3 gene. For this:

4. **Create/Expand Network** for the *User Entity Set* named "rnd3" adding relation type "interaction" excluding last level relations in **Add relations**.
5. **View/Export set details** selecting *uniprotentry* and change the *Cytoscape* VIZMAP to label the nodes according to *uniprotentry* codes.

You should be observing a network similar to the one in Fig 5.1. On the other hand if we create the network as follows:

6. **Create/Expand Network** for the *User Entity Set* named "rnd3_copy"
   (a) Add relation type "interaction" excluding last level relations in **Add relations**.

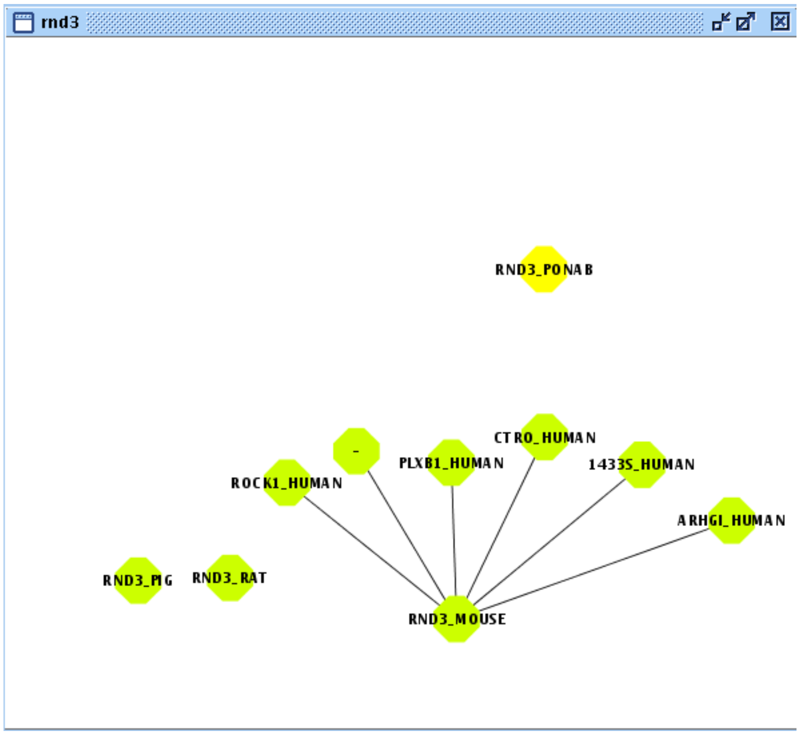
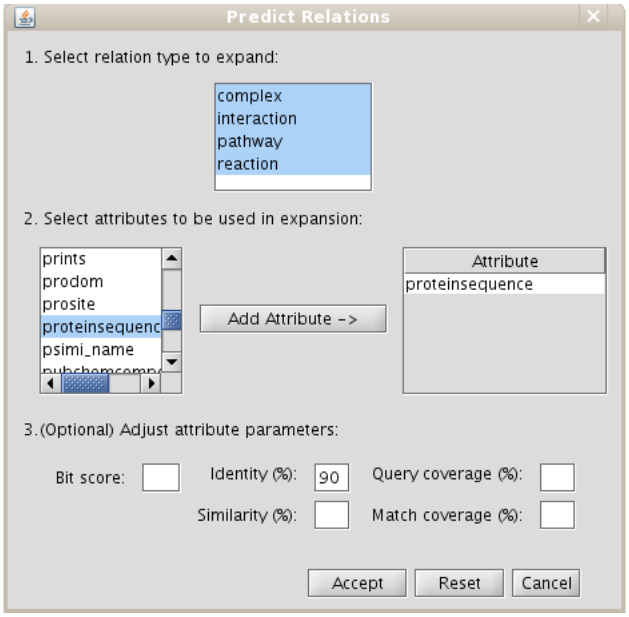


Figure 5.1: Interactions of RND3 products available in the *BIANA Database*.

(b) Add relation type "interaction" types excluding last level relations in **Make predictions by sharing attributes** also add *proteinsequence* attribute with ***identity*** parameter set at 90 as demonstrated in Fig 5.2.

7. **View/Export set details** selecting *uniprotentry.*



Predict Relations

1. Select relation type to expand:

2. Select attributes to be used in expansion:

proteinsequence

3.(Optional) Adjust attribute parameters:

Bit score:

Identity (%): 90

Query coverage (%):

Similarity (%):

Match coverage (%):

Figure 5.2: Adjusting sequence similarity parameters in network generation using **Make predictions by sharing attributes**.

We expect to see a network like the one shown in Fig 5.3.

Note that new interactions for RND3_PONAB were predicted between biomolecules because RND3_PONAB shares sequences at ≥ 90% identity level with some proteins that interacts with these biomolecules. If you add attribute "proteinsequence" without entering any parameters, exact sequence identity will be assumed. When you try this in this example you will see no new interactions are predicted because, similarity of sequences associated with biomolecules over which predictions are done have 99% identity (not 100%).

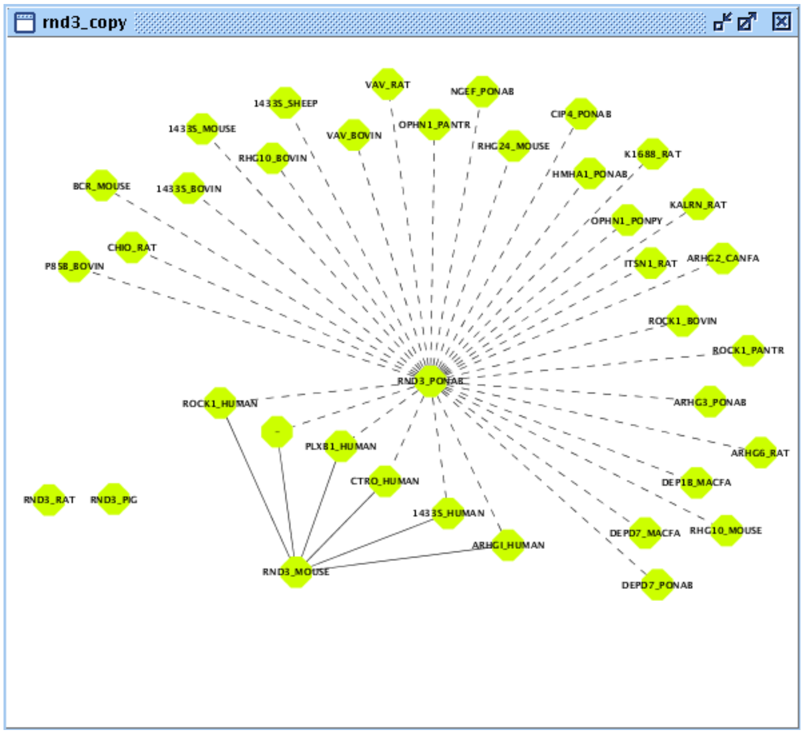


Figure 5.3: Predicted interactions of RND3 products based on sequence identity at 90% threshold.

Chapter 6

Putting It All Together: A case study of investigating common mediators in Alzheimer and Diabetes using network biology approach

This final chapter focuses on how BIANA can assist to investigate real life problems based on a case study. Please refer to the relevant section of the previous chapters if you can not figure out how to achieve explained behaviour.

In case you are interested in executing this example from command line or checking step-by-step all the executed BIANA commands, you can find the script at http://sbi.imim.es/data/biana/scripts/exercise6_script.py. You can also find a video of some parts of this example at http://www.youtube.com/user/bianatutorial

6.1 Creating a comprehensive BIANA database and an extensive unification protocol

Considering that Alzheimer's disease (AD) and diabetes are known to be coupled where having diabetes bears an increased risk for AD. Let's say, for instance, you are interested in proteins that may possibly play an important role both in Alzheimer's disease and diabetes. BIANA is perfectly suited for pursuing such kind of tasks where one needs to fetch specie-wide sequence annotation and interaction information spread across various data resources in a user-friendly and time inexpensive manner. First, you need to create a more comprehensive database for this task. The new *BIANA* database contains the following databases (see Sections 1.2 and 1.3, the databases inserted as promiscuous are marked with a star[*]):

1 SwissProt
2 Uniprot Trembl
3 NCBI Taxonomy
4 PSI-MI OBO
5 GO
6 SCOP*
7 HGNC
8 IPI
9 IntAct*
10 DIP*
11 MPACT*
12 HPRD*
13 MINT*
14 BioGrid*
15 Reactome*
16 KEGG Gene
17 KEGG KO
18 KEGG Ligand

Next, you have to create a unification protocol with the following unification protocol atoms (see Section 1.7):

- Attribute *uniprotaccession* between *all databases but Taxonomy, PSI-MI OBO and GO*
- Attributes *proteinsequence* and *taxid* between *all databases but Taxonomy, PSI-MI OBO and GO*
- Attribute *geneid* between *all databases but Taxonomy, PSI-MI OBO and GO*
- Attribute *scop* between *SwissProt, Trembl and SCOP*

6.2 Analysis on the mouse interactome

Once the database and the unification protocol is created, start a new BIANA working session (see Section 1.6) and create the following user entity sets (see Section 1.4):

- "alzheimer_mouse_0": set of proteins associated with Alzheimer's disease in mouse proteome (adding "alzheimer" as name, keyword, description, disease and function attributes and adding the restriction of "10090" as taxid attribute in **Create a new set** window)

- "diabetes_mouse_0": set of proteins associated with diabetes (adding "alzheimer" as name, keyword, description, disease and function attributes and adding the restriction of "10090" as taxid attribute in **Create a new set** window)

Next, try to duplicate (session tree right click → *Duplicate*) these sets changing their names to "alzheimer_mouse_1" and "diabetes_mouse_1" respectively, and create the relation networks as follows (see Section 1.8):

- "alzheimer_mouse_1": relations of proteins including "interaction", "complex" and (biochemical)"reactions" as relation types at level 1, without last level relations in **Create relation network** window

- "diabetes_mouse_1": relations of proteins including "interaction", "complex" and (biochemical)"reactions" as relation types at level 1, without last level relations in **Create relation network** window

Now, you have two networks of proteins, one containing proteins annotated as Alzheimer in parsed *External Database*s and proteins interacting with them and another containing proteins annotated as diabetes in parsed *External Database* and proteins interacting with them. Since you are interested in identifying common proteins which mediate AD and diabetes, try to get common proteins interacting with an Alzheimer or diabetes associated protein in the mouse proteome (considering that mouse is one of the most frequently used model organism in studies focusing on AD and diabetes) by intersecting these two sets "alzheimer_mouse_1" and "diabetes_mouse_1" (see Section 2.2) and naming it as "intersection_mouse_1". To see whether this interaction set contains proteins from the initial sets, try to select *User Entities* from existing *User Entity Set*s of "alzheimer_mouse_0" and "diabetes_mouse_0" respectively (see Section 2.4). Among the proteins contained in the intersection protein-protein interaction network, the product of Mitogen-activated protein kinase (MAPK8; aka JNK1) is distinguished for its highly relevance in both pathologies. Because, not only it interacts with an AD associated protein (product of APP) and a diabetes associated protein (product of MAPK8IP1) but also these two proteins interact with each other constituting a triangle of interactions. Strikingly, though included in neither of initial set of AD and diabetes proteins, JNK1 has been demonstrated to be involved in maintenance of neuronal mictrotubules [Karin, 2003, Dev Cell], beta-amyloid-mediated stabilization of p53 [Campbell, 2003, Biochem J], a protein attributed to cell death in the brain of patients with AD [Taniguchi, 1997, Biochem Biophys Res Com] as well as

Table 6.1: Proteins included in each *User Entity Set* in mouse.

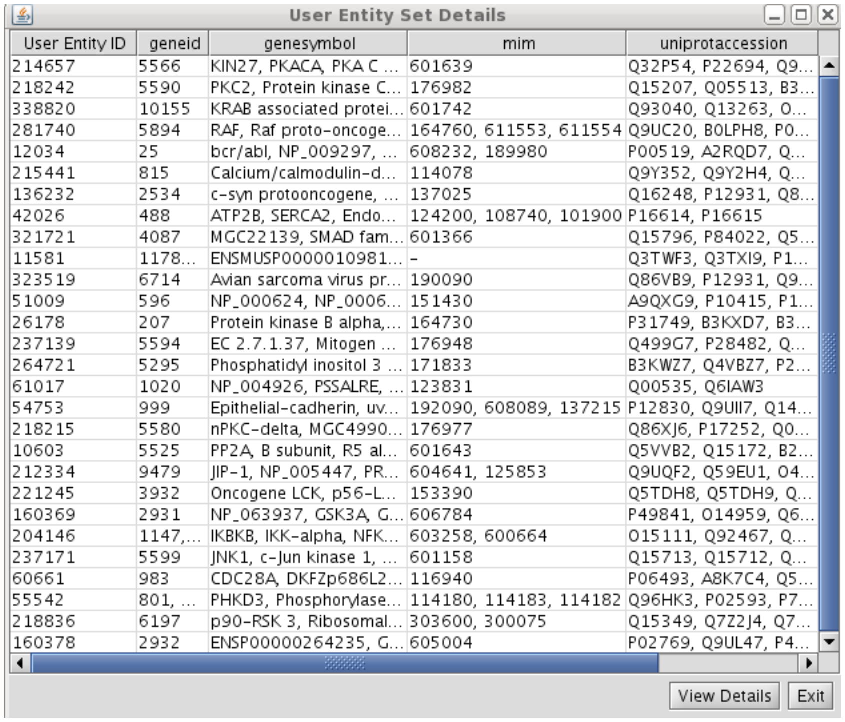
User Entity Set Name	Number of *User Entities*	Number of interactions
alzheimer_mouse_0	9	0
diabetes_mouse_0	23	0
alzheimer_mouse_1	760	784
diabetes_mouse_1	162	208
intersection_mouse_1	28	4

major risk factors of type 2 diabetes such as insulin resistance [Hotamisligil, 2002, Nature; Karin, 2007, Cell Metabolism], adiposity [Bhanot, 2008, Am J Physiol Endocrinol]. You can observe the number of *User Entities* and relationships included in each set by looking at Cytoscape Network Panel. We provide these numbers in **Table** 6.1 to give an overall picture. Next, you can look into detail which identifiers, description, mim codes, etc are associated to nodes in the intersection set (**Figure** 6.1). To do this, select the option **View/Export Set details** and then select the interested attributes you are interested in.

6.3 Analysis on the specie-wide interactome

Turning our attention to the interaction network of specie independent proteome (not restricting to mouse) to be able to increase number of possible candidates we create interaction networks for all proteins available in BIANA associated with AD and diabetes using the same keywords as above (except that we do not add restriction based on taxid attribute). Adopting the same naming convention as before, We name these *User Entity Sets* as "alzheimer_0" and "diabetes_0" to describe initial sets of AD and diabetes associated proteins and "alzheimer_1" and "diabetes_1" to describe their interaction networks. This time, intersection network itself does not give much information on the possible relevance of the pathologies of the included nodes with its considerable size (see **Table** 6.2) due to including specie-wide information. Next, you can filter the proteins contained in these two networks, so that only proteins linked at least to two Alzheimer-associated proteins or to two diabetes-associated proteins ar taken into accound. This is done by following these steps:

- Select *User Entities* initially associated with Alzheimer in "alzheimer_1"
  - Go to *Network*, click *Level 0* to select *User Entities* at level 0



User Entity ID	geneid	genesymbol	mim	uniprotaccession
214657	5566	KIN27, PKACA, PKA C ...	601639	Q32P54, P22694, Q9...
218242	5590	PKC2, Protein kinase C...	176982	Q15207, Q05513, B3...
338820	10155	KRAB associated protei...	601742	Q93040, Q13263, O...
281740	5894	RAF, Raf proto-oncoge...	164760, 611553, 611554	Q9UC20, B0LPH8, P0...
12034	25	bcr/abl, NP_009297, ...	608232, 189980	P00519, A2RQD7, Q...
215441	815	Calcium/calmodulin-d...	114078	Q9Y352, Q9Y2H4, Q...
136232	2534	c-syn protooncogene, ...	137025	Q16248, P12931, Q8...
42026	488	ATP2B, SERCA2, Endo...	124200, 108740, 101900	P16614, P16615
321721	4087	MGC22139, SMAD fam...	601366	Q15796, P84022, Q5...
11581	1178...	ENSMUSP0000010981...	-	Q3TWF3, Q3TXI9, P1...
323519	6714	Avian sarcoma virus pr...	190090	Q86VB9, P12931, Q9...
51009	596	NP_000624, NP_0006...	151430	A9QXG9, P10415, P1...
26178	207	Protein kinase B alpha,...	164730	P31749, B3KXD7, B3...
237139	5594	EC 2.7.1.37, Mitogen ...	176948	Q499G7, P28482, Q...
264721	5295	Phosphatidyl inositol 3 ...	171833	B3KWZ7, Q4VBZ7, P2...
61017	1020	NP_004926, PSSALRE, ...	123831	Q00535, Q6IAW3
54753	999	Epithelial-cadherin, uv...	192090, 608089, 137215	P12830, Q9UII7, Q14...
218215	5580	nPKC-delta, MGC4990...	176977	Q86XJ6, P17252, Q0...
10603	5525	PP2A, B subunit, R5 al...	601643	Q5VVB2, Q15172, B2...
212334	9479	JIP-1, NP_005447, PR...	604641, 125853	Q9UQF2, Q59EU1, O4...
221245	3932	Oncogene LCK, p56-L...	153390	Q5TDH8, Q5TDH9, Q...
160369	2931	NP_063937, GSK3A, G...	606784	P49841, O14959, Q6...
204146	1147,...	IKBKB, IKK-alpha, NFK...	603258, 600664	O15111, Q92467, Q...
237171	5599	JNK1, c-Jun kinase 1, ...	601158	Q15713, Q15712, Q...
60661	983	CDC28A, DKFZp686L2...	116940	P06493, A8K7C4, Q5...
55542	801, ...	PHKD3, Phosphorylase...	114180, 114183, 114182	Q96HK3, P02593, P7...
218836	6197	p90-RSK 3, Ribosomal...	303600, 300075	Q15349, Q7Z2J4, Q7...
160378	2932	ENSP00000264235, G...	605004	P02769, Q9UL47, P4...

Figure 6.1: Proteins included in intersection network of AD and diabetes associated proteins in mouse.

  - Alternatively you can use *User Entity Set* “alzheimer_0” and select *User Entities* from existing set (see Section 2.4

- Tag these selected *User Entities* with the keyword “alzheimer” (see Section 2.5)

- Select *User Entities* that are linking at least 2 of the tagged *User Entities*

  - Right-click on *User Entity Set* “alzheimer_1” and click *Select user entities by attribute/tag* (see Section 2.3)
  - In the pop-up window select *by tag linkage: “alzheimer”* (nodes connecting at least 2 tagged nodes)

- Create a new set from the selected *User Entities* with the name “alzheimer_1_linking”

  - Right-click a selected *User Entity* and follow BIANA → *Create new set from selected nodes*
  - Alternatively, select *User Entity Set* on the session tree and follow *Tags rightarrow Node Tags rightarrow* “alzheimer”. Then right-click and select *Create new set from selected nodes*

- Select *User Entities* initially associated with Diabetes in “diabetes_1”

  - Go to *Network*, click *Level 0* to select *User Entities* at level 0
  - Alternatively you can use *User Entity Set* “diabetes_0” and select *User Entities* from existing set (see Section 2.4

- Tag these selected *User Entities* with the keyword “diabetes” (see Section 2.5)

- Select *User Entities* that are linking at least 2 of the tagged *User Entities*

  - Right-click on *User Entity Set* “diabetes_1” and click *Select user entities by attribute/tag* (see Section 2.3)
  - In the pop-up window select *by tag linkage: “diabetes”* (nodes connecting at least 2 tagged nodes)

- Create a new set from the selected *User Entities* with the name “diabetes_1_linking”

Table 6.2: Proteins included in each *User Entity Set*.

User Entity Set Name	Number of *User Entities*	Number of interactions
alzheimer_0	74	0
diabetes_0	169	0
alzheimer_1	1616	1922
diabetes_1	2565	3667
intersection_1	457	805
intersection_1_linking	51	0

- Right-click a selected *User Entity* and follow BIANA → *Create new set from selected nodes*
- Alternatively, select *User Entity Set* on the session tree and follow *Tags rightarrow Node Tags rightarrow* "diabetes". Then right-click and select *Create new set from selected nodes*

Then, by intersecting these two subsets "alzheimer_1_linking" and "diabetes_1_linking", you will obtain proteins that have not been attributed to any of the pathologies in the initial sets. Next we conduct a GO functional enrichment analysis to describe common roles these proteins in the intersection set ("intersection_1_linking)"). We use GORILLA [Yakhini, 2009, BMC Bioinformatics] webserver and give the *User Entities* in *User Entity Set* "intersection_1_linking" as the set of gene list and all other *User Entities* included in *User Entity Set* "intersection_1" as the background gene list. Though we have a general idea, we are not able to deduce any significant clues on the roles of these *User Entities* (see **Table** 6.3. On the other hand, we found direct supporting evidence in the literature that some of the proteins in this set of potential mediators are products of genes that has been reported to play key roles in both AD (e.g. CamKII [Colbran, 1992, Neurochem Int; Yamauchi, 2003, Neuroscience Letters; Iqbal, 2007, Euro Jour of Neuroscience], GSK3b [Lovestone, 2004, neuroscience letters; O'Neil, 2005, Jour of Neurochem; Geschwind, 2008, Arch Neurol]) and diabetes (e.g. CaMKII [Akhtar, 2003, Autonomic & Autacoid Pharmacology; Vitale, 2009, Cellular Signalling], GSK3b [Bussiere, 2004, Current Pharmaceutical Design; Yamada, 2006, Nature Medicine])

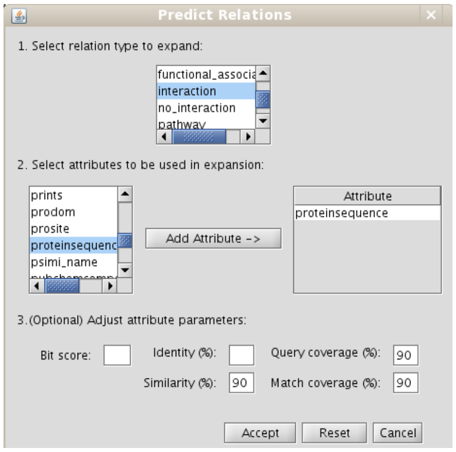
Table 6.3: GO enrichment analysis for proteins included in intersection network of AD and diabetes associated proteins using GORILLA[Yakhini, 2009, BMC Bioinformatics].

GO Term	Description	P-value	Enrichment (N, B, n, b)
GO:0019900	kinase binding	4.26E-5	3.83 (370,21,46,10)
GO:0019899	enzyme binding	2.68E-4	2.41 (370,50,46,15)
GO:0019901	protein kinase binding	3.24E-4	3.79 (370,17,46,8)
GO:0016773	phosphotransferase activity, alcohol group as acceptor	8.88E-4	2.19 (370,55,46,15)
GO:0045309	protein phosphorylated amino acid binding	9.64E-4	6.43(370,5,46,4)

6.4 Analysis on the homology extended interactome

In order to increase the list of potential candidates, you can repeat the analysis including predictions obtained using the interology concept (i.e. transferring interactions from proteins to their homologous pairs). Before starting this step, it is necessary to obtain sequence similarity data (see **Chapter 5**). Once homology data is calculated, we repeat the same steps as for specie-wide interactome, but adding transferred interactions by means of protein sequence similarity:

- Create initial “alzheimer” and “diabetes” as in the previous example (without restricting specie).
- Create network for both “alzheimer” and “diabetes” sets:
  - Add relations: Relations of proteins including “interaction”, “complex” and (biochemical)“reactions” as relation types at level 1, without last level relations in **Create relation network** window.
  - Make predictions by sharing attributes: Predictions of interactions (select “interaction” in the relation type to expand), and add the attribute “proteinSequence”. Set the percentage similarity value to 90, and query and match coverage to 90 (it means at least both similar proteins must match in its 90% sequence length) (Fig 6.2).



1. Select relation type to expand: interaction

2. Select attributes to be used in expansion:

Attribute
proteinsequence

3.(Optional) Adjust attribute parameters:

Bit score:
Identity (%):
Query coverage (%): 90
Similarity (%): 90
Match coverage (%): 90

Figure 6.2: Example of predictions by sharing attributes. In this particular case, interactions are transferred for proteins having a 90% of sequence in at least their 90% of sequence length.

Table 6.4: Proteins included in each *User Entity Set*.

User Entity Set Name	Number of *User Entities*	Number of interactions
alzheimer_0	74	0
diabetes_0	169	0
alzheimer_1	3466	3866
diabetes_1	6987	16447
intersection_1	2894	6846
intersection_1_linking	221	81

Now, if you compare previous networks with the new ones you can observe they contain more nodes and edges. Again, select only those proteins linking at least 2 Alzheimer associated proteins or 2 diabetes proteins. Now, you observe a larger list of potential candidates to study the relation between Alzheimer and Diabetes. The number of nodes and edges obtained are:

A visual inspection on the list (looking at function, description, disease, etc) yielded some new interesting candidates to be evaluated, such as calreticulin or drebrin. For example, chaperone calreticulin appeared to have a possible mediating relation with AD [Taguchi et al, Acta Neuropathol, 2000] and with insulin receptors [Bass et al., The Journal of Cell Biology, 1998]. Also drebrin has been related with both AD and diabetes [Burdo et al,. Neurobiology of aging, 2008].