

User Manual

Simon Hackl

February 14, 2019

Preface

BioNGraph is a database application to store property graphs and merge several networks based on a specified node property. It was implemented as part of the bachelor thesis *Graph Databases for Biomolecular Networks* at the Eberhard Karls-Universität Tübingen.

Quickstart

BioNGraph currently only works on Redis servers with a loaded RedisGraph module. Either a established server instance or a local server can be used to connect to.

Installing RedisServer and RedisGraph

The Redis server and RedisGraph are both freely available at <https://redis.io/> and <https://oss.redislabs.com/redisgraph/>. To establish a Redis server with the RedisGraph module the instructions on the respective web pages should be followed. To load the RedisGraph module the path to the `.so` file - created by building RedisGraph - has to be specified in the Redis server configuration file.

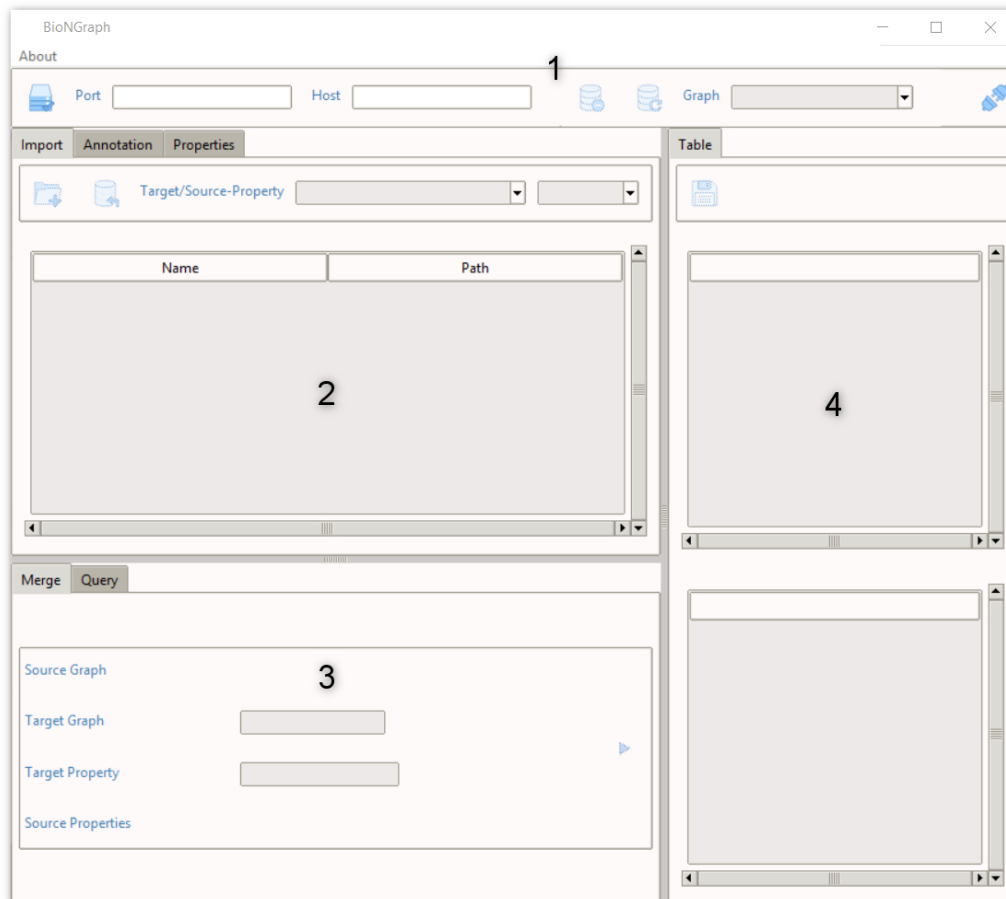
The server can be started by running the downloaded `redis-server` from console. The respective configuration file where the RedisGraph module is specified can be passed as a argument. More information on how to use Redis with Windows can be found here <https://redislabs.com/blog/redis-on-windows-10/>. However, one easy way is to use the Linux subsystem available for Windows 10.

Building BioNGraph

BioNGraph is freely available at <https://github.com/s-t-h/BioNGraph>. PyBuilder is used as building tool - after the repository was cloned or downloaded one can navigate to the corresponding path and simply run `pyb`. For Windows systems the command `pyb_` has to be used. After the build process has finished, a new file path `target/dist/biongraph` is created in the main directory of BioNGraph. To run BioNGraph the python script `application.py` has to be started at the file path mentioned above.

PyBuilder can be installed via `pip install pybuilder`. More information about PyBuilder can be found on the projects repository.

Overview



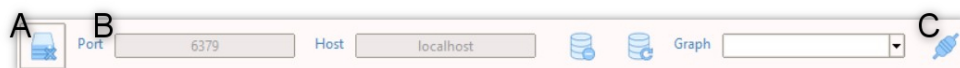
- 1 **Status bar:** Connect to server and manage graph keys.
- 2 **Import, Annotation, Properties:** Import graph files, annotate queries, display stored properties.
- 3 **Merge, Query:** Merge stored graphs, query stored graphs.
- 4 **Edge and Vertex view:** Display and export queries.

Exemplary Workflow

The main purpose of this application is the integration of several networks - represented as property graphs - into one coherent. In the following, an exemplary workflow demonstrates how to use this application.

Connect to a Redis Server

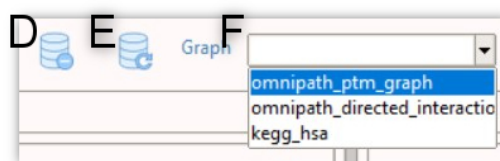
First and foremost, the application has to be connected to an established Redis server instance. To do this, navigate to **1**. The corresponding connection data can be specified in the port and host entry (**B**). In this case, a local server is connected to. The default connection data are port **6379** and host **localhost**. By pressing the adjacent connect button (**A**), BioNGraph tries to connect to the server.



If the connection is successful, the two entries mentioned are greyed out and the status indicator on the right (**C**) shows a new icon. In addition the other widgets become active.

Access Graph Keys

The entry field on the right (**F**) can be used to access existing graph keys. In this case, some graphs are already stored.

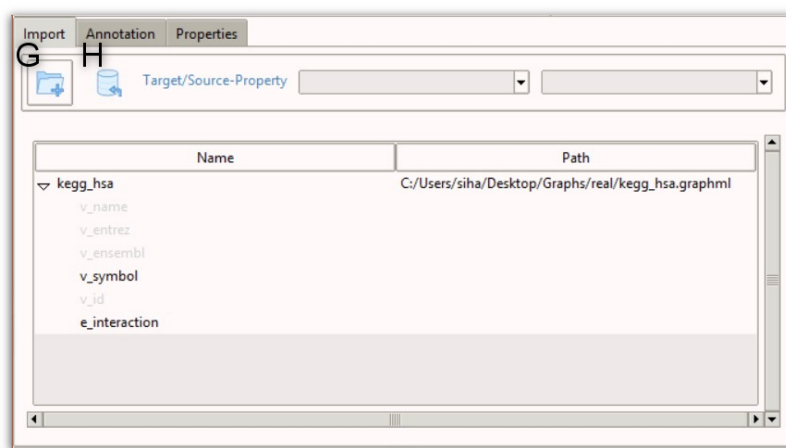


With the right button (**E**), the entire existing database can be saved on the hard drive. The left button (**D**) deletes the graph stored at the selected key.

💡 The option to save the database and the path where to save it is specified in the Redis server configuration file. This option may be disabled.

Upload Graphs

To upload new graphs, navigate to **2** and select the Import tab. The left button (**G**) opens a file manager. For the file to be added, a list of all possible properties is shown in the table below. If **Ctrl** is kept, properties can be deselected, these are greyed out and not imported. Pressing the right button (**H**) will upload the graph into the database. The file name is used as the key for the uploaded graph.



Currently BioNGraph supports the import of graphs in the form of **graphml** and **csv** files.

-
- 💡 If the source item of the table view is deselected the whole file will be ignored for the upload process - independent of the selected child items.
 - 💡 Currently there is no option to clear the view.
 - 💡 For the import of **csv** files the standard excel dialect should be used. In addition, the symbols () [] : - > < are reserved by the OpenCypher language and could lead to complications. In addition, a certain structure must currently be adhered to: the names in the header are interpreted as source node properties until the first reading of the keyword ID, as target node properties until the second read in of ID and all remaining names as edge properties. The names can be prefixed with _ as a separator symbol. This prefix will not be imported.

Merge Graphs

To merge stored graphs select the Properties tab of **2** and the Merge tab of **3**. At the Properties tab a list of all properties that are stored in all graphs appears. By holding **Ctrl** the properties can be selected, these are highlighted in light blue. The Merge tab now displays a list of the selected properties and the graphs they originate from. In the Target Graph field (**I**) the key for the merge graph must be specified. In the Target Property field (**J**) the name for the merge property must be specified. By pressing the Run button (**K**) on the right, the merge graph is created and stored in the database under the entered key.

The screenshot shows the BioNGraph software interface with two tabs: 'Properties' and 'Merge'.

Properties Tab:

Name	Source	Type
v_hgnc	omnipath_directed_interactions	vertex
v_hgnc	omnipath_ptm_graph	vertex
v_symbol	kegg_hsa	vertex
e_residue_offset	omnipath_ptm_graph	edge
e_residue_type	omnipath_ptm_graph	edge
e_modification	omnipath_ptm_graph	edge
e_is_inhibition	omnipath_directed_interactions	edge
e_is_stimulation	omnipath_directed_interactions	edge
e_interaction	kegg_hsa	edge

Merge Tab:

Source Graph: omnipath_directed_interactions, omnipath_ptm_graph, kegg_hsa

Target Graph (**I**): OmnipathKegg

Target Property (**J**): hgnc

Source Properties: omnipath_directed_interactions_v_hgnc, omnipath_ptm_graph_v_hgnc, kegg_hsa_v_symbol

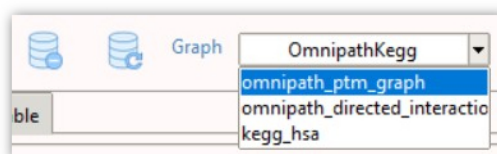
Run button (**K**)

For the implemented merge algorithm, only one property can be selected per graph. For this, the first selected property per graph is used. Selected edge properties are ignored.

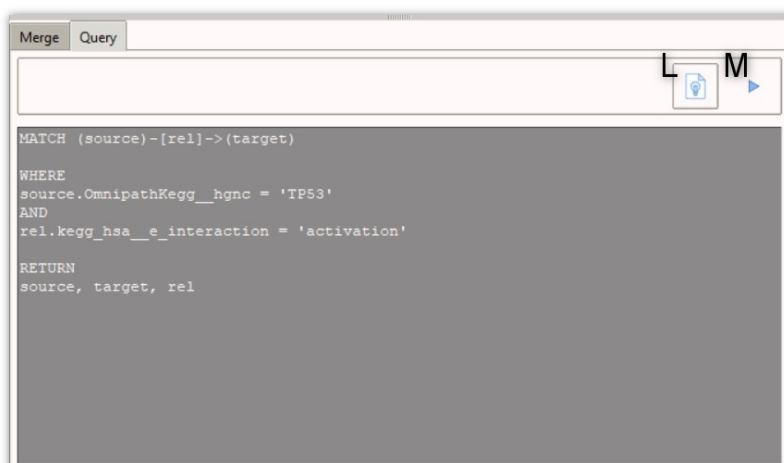
💡 BioNGraph uses `_` and `__` as internal separation symbols. These characters will be removed from the text entered in **I** and **J**.

Query Graphs

To query stored graphs, an existing graph key must first be selected. This is done in the corresponding entry (F) in 1. In this case, the merge graph key created in the last step was chosen.



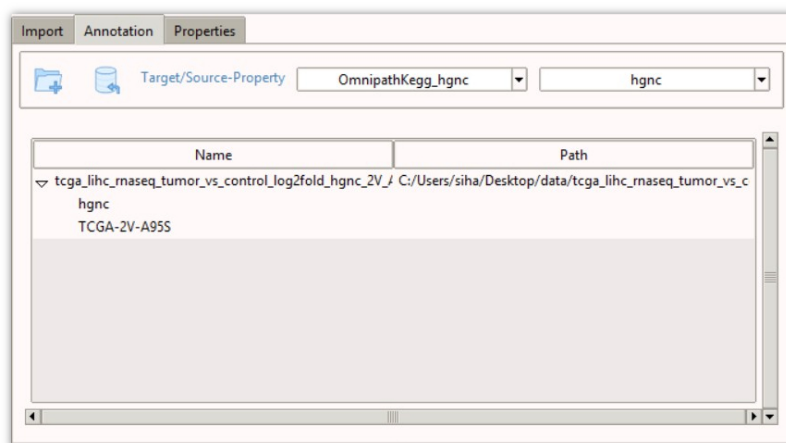
To start a query, navigate to 3 and select the Query tab. The gray field is used as input space for OpenCypher syntax queries. The explain button (L) in the image below generates information about the execution plan of the query. The query can be started by the right button (M). The result of a query is displayed in 4. There the upper table shows queried edges and the lower table shows queried nodes.



-
- 💡 BioNGraph uses `__` as internal separation symbols for graph and property names. For example if a property is listed in 1 as *Name: Attribute* and *Source: Graph* the full name of the property is `Graph__Attribute` and this name should be used for queries.

Annotate Query Results

A query result can be annotated with additional data. For this navigate to **2** and select the Annotation tab. In addition to the buttons **G** and **H** the two entry fields on the right become available. In the left entry the property to use as identifier property in the current selected graph has to be specified. In the right entry the property to match from has to be specified. In the process of annotation all nodes with a specific value of the identifier property will inherit all properties that are stored with the respective match property value. The match property value will not be uploaded. Properties can be deselected and the annotation can be uploaded as it is done for importing new graphs. BioNGraph currently supports the annotation with `json` and `csv` files.

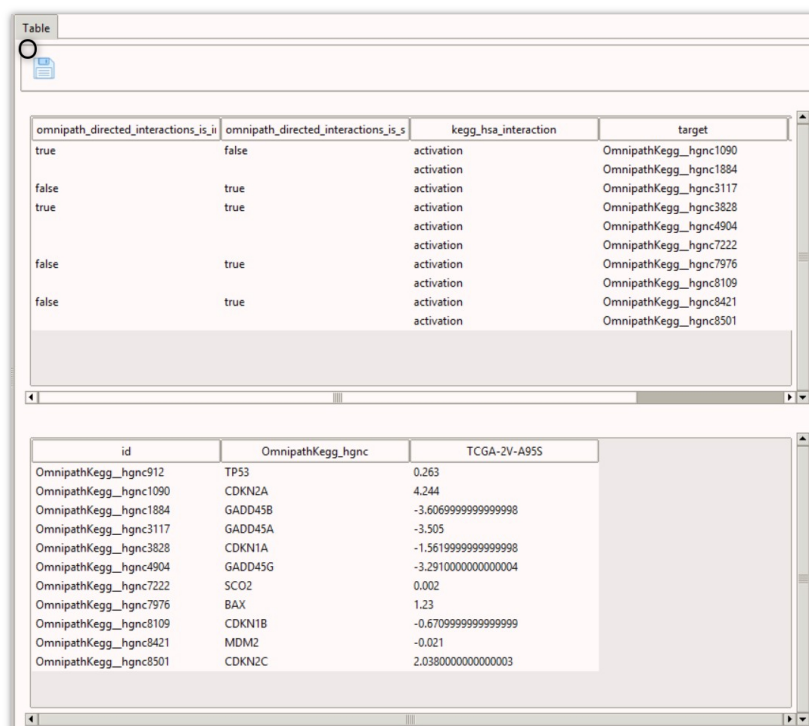


💡 For the annotation with `csv` files the same restrictions apply as for importing data. For `json` files the expected syntax is

```
[
  { "some property": "foo", "some other property": "bar", ... } ,
  ...
]
```


Export Query Results

Finally, the annotated data appears as a new column in the lower node table at 4. To export the data the button on top (O) can be used. This will open a file manager to specify the path where to save the file. Currently only the export of `graphml` files is supported.



omnipath_directed_interactions_is_i	omnipath_directed_interactions_is_s	kegg_hsa_interaction	target
true	false	activation	OmnipathKegg_hgnc1090
		activation	OmnipathKegg_hgnc1884
false	true	activation	OmnipathKegg_hgnc3117
true	true	activation	OmnipathKegg_hgnc3828
		activation	OmnipathKegg_hgnc4904
		activation	OmnipathKegg_hgnc7222
false	true	activation	OmnipathKegg_hgnc7976
		activation	OmnipathKegg_hgnc8109
false	true	activation	OmnipathKegg_hgnc8421
		activation	OmnipathKegg_hgnc8501

id	OmnipathKegg_hgnc	TCGA-2V-A955
OmnipathKegg_hgnc912	TP53	0.263
OmnipathKegg_hgnc1090	CDKN2A	4.244
OmnipathKegg_hgnc1884	GADD45B	-3.6069999999999998
OmnipathKegg_hgnc3117	GADD45A	-3.505
OmnipathKegg_hgnc3828	CDKN1A	-1.5619999999999998
OmnipathKegg_hgnc4904	GADD45G	-3.2910000000000004
OmnipathKegg_hgnc7222	SCO2	0.002
OmnipathKegg_hgnc7976	BAX	1.23
OmnipathKegg_hgnc8109	CDKN1B	-0.6709999999999999
OmnipathKegg_hgnc8421	MDM2	-0.021
OmnipathKegg_hgnc8501	CDKN2C	2.0380000000000003

- 💡 The property names will be revised automatically to remove the separator symbols. In addition, the columns can be renamed by right-clicking on the respective header.
- 💡 In some cases - when queries are limited to a specific amount of responses - the table headers do not fit with the displayed properties. However, the internal data structure remains consistent.

Roadmap

Suggestions for the further developement of BioNGraph and improvement of its usability are listed below.

- Change Import and Annotation tab to a top-level window for clarity.
- Change Properties tab to a hierarchical layout for clarity.
- Add support to auto-revise the internal naming of properties.
- Add support for more file types.
- Add blueprints for queries.
- Add support for Redis server security.
- Add support for user specified identifiers for importing csv files.
- Add support for graph types and a hierarchical graph key view.
- Add support for user specified node and edge labels.
- Add support for uploading annotated query results.
- Implement a script for multi-graph upload.
- Implement import automation e.g. with PyPath.