Tutorial : Querying the FHF-GKBox

The FHF-GKBox ('Fight Heart Failure' Graph Knowledge Box) is the knowledge graph developed by the EdgeLeap start-up under the supervision of MD Devignes, Malika Smaïl-Tabbone and Emmanuel Bresso during the FIGHT-HF project (RHU 2018-2022). It is available on-line for queryin as a Neo4J graph. For more information on Neo4J, visit the Neo4 web site or the Neo4J overview at tutorialspoint.com. FHF-GKBox will be named simply GKBox throughout this tutorial.

I. Introduction to the neo4J environment

1. Connection to the GKBox

https://fighthf-graphkb.loria.fr/browser/

Enter userId : BrazilStudent1, BrazilStudent2, BrazilStudent3, … BrazilStudent10 -> choose one per machine.

Password: d0n1zett1 (the same for everybody)

IMPORTANT: Ignore the error messages "Database access not available…." And "Cannot connect to Neo4J. Rather, immediately modify the Neo4J parameters : click on the "Parameters" icon (left panel, close to bottom).

- Activate "Do not use Bolt"
- Inactivate "Auto Complete" (last item)
- Click again on the parameters icon to close the panel
- Then refresh your browser

Click on the "Database" icon top of left panel. The content of the GKBox appears

2. Exploration of the GKBox content through the Neo4J interface

The graph database is composed of nodes and relations between the nodes. Each node is of a certain type or label: *protein*, *pathway*, *disease*, … and each relation has also a certain type: *interacts_with*, *consolidated*, …

The metagraph of the GKbox describes all the possible types of nodes contained in the GKBox and the existence or not of a relation between two types of nodes. For example, the protein node is linked to all other types of nodes including itself, but there is no direct relation between a disease node and a pathway node.

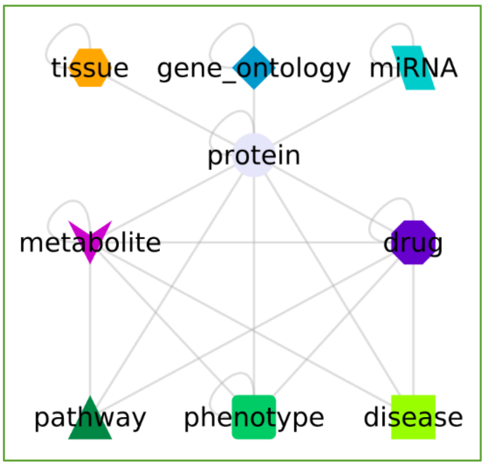
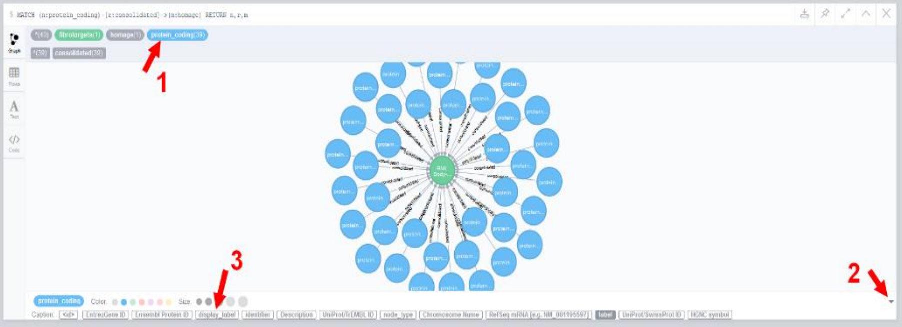


Figure 1 : Metagraph of the GKBox

The 2019 version of the GKBox contains 246,672 nodes and 24,601,110 relations, pertaining from the integration of 18 public data resources. It contains 20,431 protein nodes restricted to human species.

- Click on the database icon
    - The various types/labels of nodes, types of relations and their properties are presented. Identify the 9 nodes labels from the metagraph.
    - Ignore the *metadata* labels. These are used to generate the documentation of the GKBox. In particular it is possible to know the origin of each node and each relation in the database.
- Click on the ***protein*** node label
    - A set of 25 nodes is displayed in the main window.
    - When passing the pointer on a node, information is displayed at the bottom of the window. Click on the node once to stabilize the display and click on the small arrow on the right to display more information (by default you see only one line).
    - You can change the identifier which is displayed on a node. Click on the type of node on top left of the display (**1** in Figure 2) and go to the bottom, expand information by clicking on the small arrow on the right (**2** in Figure 2) and select Uniprot_name or something else in the bottom information (**3** in Figure 2). See what is changed depending on your selection. Keep what you prefer (***Uniprot_name*** is nice for proteins, ***Display_label*** is convenient for any type of nodes)



```
$ MATCH (n:protein_coding)-[r:consolidated]->(m:homage) RETURN n,r,m
```

*(40)
fibrotargets(1)
homage(1)
protein_coding(39)
*(39)
consolidated(39)
Graph
Table
Text
Code
1
3
2
protein_coding
Color:
Size:
Caption: <id> EntrezGene ID Ensembl Protein ID display_label identifier Description UniProt/TrEMBL ID node_type Chromosome Name RefSeq mRNA [e.g. NM_001195597] label UniProt/SwissProt ID HGNC symbol

Figure 2 : Screenshot of the GKBox browser showing how to select what identifier should be displayed on each node.

-
    - When clicking once on the node, you see three functionalities around the node.
        - On the right an X meaning “Remove node from visualization”,
        - On the left, a padlock, meaning “Unlock the node to re-layout the graph”
        - Below, a large cross meaning “Expand child relationships”.
        - Click on this last functionality “Expand child relationships“. What do you see ?
    - **Avoid clicking twice** on a node as this has the same effect as expand child relationship and it is not reversible (to my knowledge), so it can ruin your efforts to show a well organized graph !)
- Do the same exploration with a type of relation. for example click on ***drug_targets*** relationship type in the panel showing the database content.
    - A set of 25 drug targets relations is displayed. At the top left you can see how many distinct nodes, drugs, metabolites and proteins are involved. Drugs and metabolites are two overlapping labels for the same type of node. Are there more metabolites than drugs in your display ? Can you identify the nodes which are only metabolites and not drugs (just pass over the nodes with the pointer and look at the information displayed at the bottom).
    - Try to change the identifiers displayed on drugs or metabolites.

Note on “consolidated” relations

There are as many relations between two nodes as different data sources integrated in the GKBox. Many of them are redundant. To simplify the graph display, there exists in most cases a unique

« consolidated » relation between two nodes. These consolidated relations have been constructed by the GKBox designers (EdgeLeap start-up and the Loria) according to various rationales depending on the data sources. This is documented elsewhere. One just needs to keep in mind that these are the most reliable relations of the GKBox.

II. Construction of cypher queries

Cypher Query Language (CQL) is the query language adapted to query the Neo4J knowledge graphs. A **cypher query** is generally composed of three parts :

- **`MATCH`** : description of the sub-graph that will be queried
  - Example :
    - `MATCH (n:protein)-[r:interacts_with]-(d:drug)`
  - `(n:proteins), (d:drug)` : node variables, with their type
  - `[r:consolidated]` : relation variable, with its type
  - Nodes and relations are linked with a dash “ – “ when the relation is not oriented, or with an arrow “->” when oriented.
- **`WHERE`** : conditions on the requested properties of nodes and relations
  - Example
    - `WHERE n.identifier in [`*`list of identifiers`*`]`
- **`RETURN`** : describe precisely what should be returned and displayed
  - Example
    - `RETURN n, r, d` : returns the nodes corresponding to n and d in the query and the relations r which connect the nodes.

IMPORTANT : for the exercises below you should create two types of files for your records

- a simple text editor to prepare and save your queries. You can number then according to the tutorial. When the query is ready, copy-paste it in query field of the GKBox browser.
- a Draw / Powerpoint file to copy-paste and save screenshots of the visualization and keep track of the results.

III. Exercises with a set of biomarkers extracted from Fibrosis Predictive CMFIs[1]

Corresponds to following article : Bresso E, Ferreira JP, Girerd N, Kobayashi M, Preud'homme G, Rossignol P, Zannad F, Devignes MD, Smaïl-Tabbone M. (2022) Inductive database to support iterative data mining: Application to biomarker analysis on patient data in the Fight-HF project. J Biomed Inform. 135:104212.

Table 1: List of 9 relevant biomarkers

Acronyme	FullName	UniProt_ID	GKBox UniProt name
CD4	CD4 molecule	P01730	CD4_HUMAN
FGF23	fibroblast growth factor 23	Q9GZV9	FGF23_HUMAN
PTX3	pentraxin 3	P26022	PTX3_HUMAN
REN	renin	P00797	RENI_HUMAN
Sdf1A	Fas ligand	P48061	SDF1_HUMAN

[1] CMFIs : Contrasted Maximal Frequent Itemsets

TF	coagulation factor III, tissue factor	P13726	TF_HUMAN
TNFRSF11A	TNF receptor superfamily member 11a	Q9Y6Q6	TNR11_HUMAN
TRAILR2	TNF receptor superfamily member 10b	O14763	TR10B_HUMAN
XCL1	X-C motif chemokine ligand 1	P47992	XCL1_HUMAN

1. Extract and display the 9 proteins from the GKBox

Query 1: isolated nodes

```
MATCH (n:protein)  WHERE n.identifier in
["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
RETURN n
```

Copy and paste after the $ sign in the query field of the GKBox browser.

⤢ The display window can be enlarged fullscreen by clicking on the enlargement icon at the top right of the window

Check the rich information available for each biomarker in the properties field at the bottom of the display when passing on each node.

2. Look for direct relationships between the 9 proteins

Query 2: Consolidated relationships

```
MATCH (n:protein)-[r:consolidated]->(m:protein)  WHERE n.identifier in
["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
AND m.identifier in ["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
RETURN n, r, m
```

How many nodes are displayed ? Which ones ? Do you find consolidated relationships between all pairs of biomarkers ? List the pairs of biomarkers that are connected.

Query 3: Any type of relationship

```
MATCH (n:protein)-[r]->(m:protein)  WHERE n.identifier in
["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
AND m.identifier in ["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
RETURN n, r, m
```

How different is this result from the preceding one ? List the additional proteins that are now connected. Do you find some relationships between all pairs of biomarkers ? List the pairs of biomarkers that are connected.

3. Look for indirect relationships between the 9 proteins

Query 4: Protein-Pathway-Protein

```
MATCH (n:protein)-[r:consolidated]-(p:pathway)-[s:consolidated]-(m:protein)
WHERE
n.identifier in
["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
AND m.identifier in ["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
RETURN n, r, p, s, m
```

Arrange the visualization to compare with Figure 5 from the publication (reproduced at the end of this tutorial (Note that this figure was arranged for publication with the Cytoscape software).

Query 4 : Protein-GO_term-Protein

```
MATCH (n:protein)-[r:consolidated]-(g:gene_ontology)-[s:consolidated]-
(m:protein)
WHERE
n.identifier in
```

```
["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
AND m.identifier in ["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
AND g.namespace = "biological_process"
RETURN n, r, g, s, m
```

Arrange the position of the 9 BMs as before to compare with the preceding graph. What supplementary information do you get with Gene Ontology Annotation ?

4. Optional : import network into Cytoscape

Export should be a csv format with following fields

```
Node 1 (Source), Node 2 (Target), Relation type, Property of Node 1
(optional), Property of Node 2 (optional).
```

In practice this means that you can only export one relation at a time.

This can be obtained by structuring the RETURN part of the query

For example Query 4 should be modified first for the first relation between protein and pathway

Query 5

```
MATCH (n:protein)-[r:consolidated]-(p:pathway)-[s:consolidated]-(m:protein)
WHERE
n.identifier in
["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
AND m.identifier in ["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
RETURN n.display_label, p.display_label, type(r), labels(n), labels(p)
```

Save the result as a csv file named Query4_part1forCytoscape.csv

Then adapt the query to export the relation between the pathways and the other proteins

Query 6

```
MATCH (n:protein)-[r:consolidated]-(p:pathway)-[s:consolidated]-(m:protein)
WHERE
n.identifier in
["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
AND m.identifier in ["http://identifiers.org/uniprot/P01730",
"http://identifiers.org/uniprot/Q9Y6Q6",
"http://identifiers.org/uniprot/O14763",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P26022",
"http://identifiers.org/uniprot/P48061",
"http://identifiers.org/uniprot/P47992",
"http://identifiers.org/uniprot/P13726",
"http://identifiers.org/uniprot/Q9GZV9"]
RETURN p.display_label, m.display_label, type(s), labels(p), labels(m)
```

Save the result as a csv file named Query4_part2forCytoscape.csv

Open Cytoscape

Import file -> first import Query4_part1forCytoscape.csv.

Assign n (node 1 column) as Source, p (node 2 column) as Target, type(r) (column 3) as relation type, labels(n) (column 4) as Source property, labels(p) (column 5) as Target property, and import.

Then import Query4_part2forCytoscape.csv

Assign p (node 1 column) as Source, m (node 2 column) as Target, type(s) (column 3) as relation type, labels(p) (column 4) as Source property, labels(m) (column 5) as Target property, and import.

Then use the merge tool to merge the two networks.

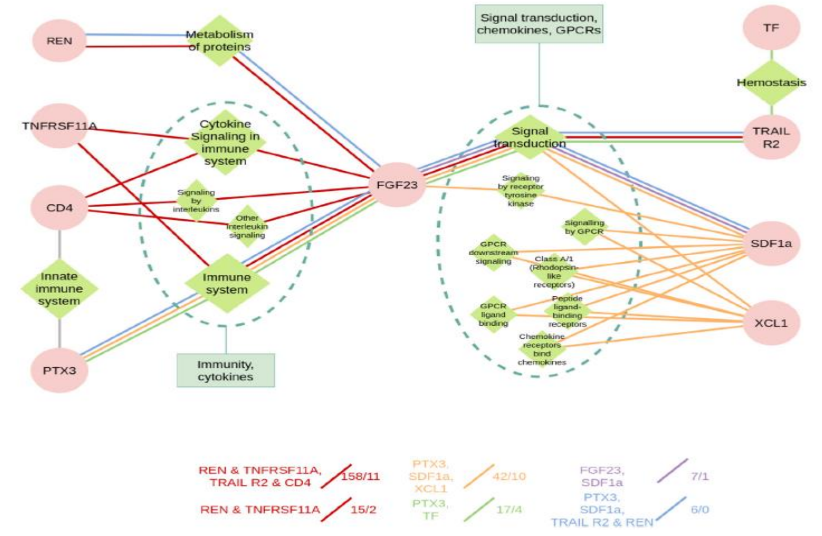

REN
Metabolism of proteins
TNFRSF11A
Cytokine Signaling in immune system
CD4
Signaling by interleukins
Other interleukin signaling
Innate immune system
Immune system
PTX3
Immunity, cytokines
FGF23
Signal transduction, chemokines, GPCRs
Signal transduction
Signaling by receptor tyrosine kinase
Signalling by GPCR
GPCR downstream signaling
Class A/1 (Rhodopsin-like receptors)
GPCR ligand binding
Peptide ligand-binding receptors
Chemokine receptors bind chemokines
TF
Hemostasis
TRAIL R2
SDF1a
XCL1
REN & TNFRSF11A, TRAIL R2 & CD4
158/11
REN & TNFRSF11A
15/2
PTX3, SDF1a, XCL1
42/10
PTX3, TF
17/4
FGF23, SDF1a
7/1
PTX3, SDF1a, TRAIL R2 & REN
6/0

Figure 3: Protein-Pathway-Protein network from article (Bresso et al. 2022). The network was extracted from the GKBox through cypher queries, imported into Cytoscape to modify the figure, which was later annotated under PowerPoint to create the final Figure. Green diamonds are Pathways and pink circles are Proteins. Colored edges refer to rules leading to fibrosis as displayed in the decision tree.

IV. Exercise to study gene-disease relationships

Three groups of patients with different phenotypes (Hypertension, Obesity and Diabetes) have been studied thanks to proteomics data. All these phenotypes lead to heart failure which also means Fibrosis. Important biomarkers have been identified in each group, with increased or decreased expression in the patients compared to healthy people. We wanted to study the relationships between these biomarkers and Fibrosis in the different groups of patients. The whole study is described in Ferreira et al., 2019[2].

The final Protein-Pathway-Disease graph is represented in Figure 4 of the original article and here below in Figure 4.

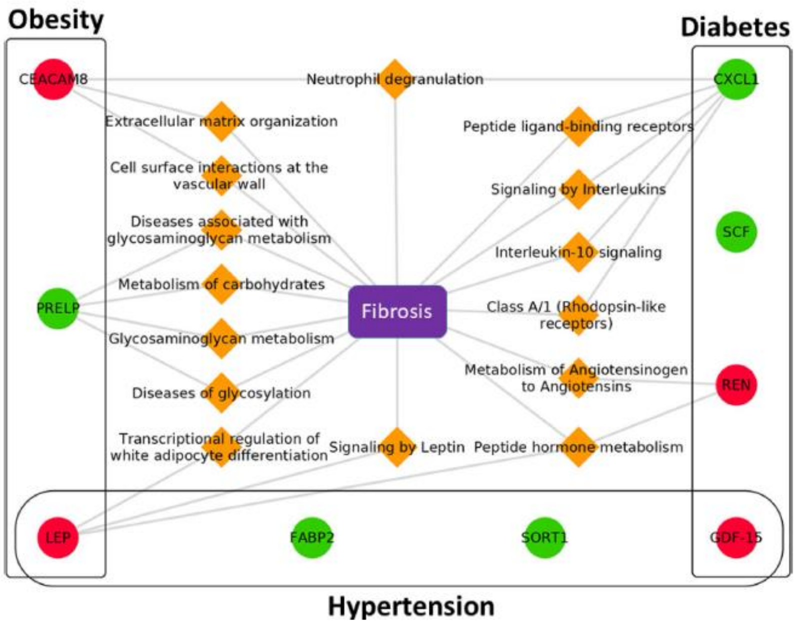

Obesity
Diabetes
CEACAM8
Neutrophil degranulation
CXCL1
Extracellular matrix organization
Peptide ligand-binding receptors
Cell surface interactions at the vascular wall
Signaling by Interleukins
Diseases associated with glycosaminoglycan metabolism
SCF
Interleukin-10 signaling
Metabolism of carbohydrates
PRELP
Fibrosis
Class A/1 (Rhodopsin-like receptors)
Glycosaminoglycan metabolism
Metabolism of Angiotensinogen to Angiotensins
REN
Diseases of glycosylation
Transcriptional regulation of white adipocyte differentiation
Signaling by Leptin
Peptide hormone metabolism
LEP
FABP2
SORT1
GDF-15
Hypertension

Figure 4 : Biomarkers and their associated pathways leading to fibrosis. Legend: “circles” are proteins biomarkers (red, increased expression ; green, decreased expression in the various phenotypic groups), orange diamonds are pathways, and violet square is disease.

1. Selection of source and target nodes

LIST OF 6 RELEVANT BIOMARKERS (NOTE THAT LEP OCCURS IN TWO PHENOTYPIC GROUPS)

Phenotype	Acronyme	FullName	UniProt_ID	GKBox UniProt name
Hypertension	LEP	Leptin	P41159	LEP_HUMAN
Obesity	CEACAM-8	Carcinoembryonic antigen-related cell adhesion molecule 8	P31997	CEAM8_HUMAN
Obesity	PRELP	Prolargin	P51888	PRELP_HUMAN
Obesity	LEP	Leptin	P41159	LEP_HUMAN
Diabetis	REN	renin	P00797	RENI_HUMAN
Diabetis	CXCL1	Growth-regulated alpha protein	P09341	GROA_HUMAN
Diabetis	SCF	Kit ligand	P21583	SCF_HUMAN

[2] Study described in the FIBRO-Target paper ; Ferreira et al, 2019, Clinical Research in Cardiology, 109:22-33. https ://doi.org/10.1007/s0039 2-019-01480 -4

Build and save the cypher query that retrieves and displays all these biomarkers from the GKBox (analogous to Query 1 above, just change the UniProt Ids). You can number this query Query 1F.

You can test if there are direct protein-protein interactions (consolidated or not) among pairs of biomarkers in this set as before. Just adapt Queries 2 and 3 (name these queries Query 2F and Query 3F)

IDENTIFYING DISEASE OR PHENOTYPE NODES

The disease or phenotype to study is Fibrosis. The question is how Fibrosis is represented in the FHF-GKBox ?

a) As a disease ?

Query 7

```
MATCH (d:disease)  WHERE any (x in d.display_label where x contains
"fibrosis" or "Fibrosis") RETURN d.display_label, d.identifier
```

Save the Results Table as a csv file. Various anatomical locations of fibrosis are retrieved (58 answers in total). Select the 3 concerned by cardiac location and note their identifiers.

Select a set of identifiers to proceed further.

For example, one group of students could be focus on the 3 diseases nodes related to cardiac fibrosis

- Cardiac fibrosis http://identifiers.org/umls/C1397307
- Encomyocardial fibrosis http://identifiers.org/umls/C0553980 , http://identifiers.org/doid/DOID:12932
- Myocardial fibrosis http://identifiers.org/umls/C0151654

Another group can also use the unique node Fibrosis which is part of the answers (for comparison purpose)

- Fibrosis http://identifiers.org/umls/C0016059

b) As a phenotype ?

Query 8

```
MATCH (ph:phenotype)  WHERE any (x in ph.display_label where x contains
"fibrosis" or "Fibrosis") RETURN ph.display_label, ph.identifier
```

Save the Results Table as a csv file. Compare with preceding results (Total number of results ? How many concerned by cardiac location ? Note their identifiers). A third group can use these nodes as target nodes for comparison purpose.

2. Retrieve direct relationships between BMs and disease ?

Query 9 (example with the three nodes related to cardiac fibrosis as a disease)

```
MATCH (bm:protein)-[r:consolidated]-(d:disease) WHERE bm.identifier in
["http://identifiers.org/uniprot/P41159",
"http://identifiers.org/uniprot/P31997",
"http://identifiers.org/uniprot/P51888",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P09341",
"http://identifiers.org/uniprot/P21583"]
AND d.identifier in ["http://identifiers.org/umls/C1397307",
"http://identifiers.org/umls/C0553980",
"http://identifiers.org/umls/C0151654]
RETURN bm, r, d
```

Interpret and conclude

3. Retrieve Protein-Pathway-Disease paths (indirect relationships between BMs and disease)

Query 10 (example with Cardiac fibrosis as a disease)

```
MATCH (bm:protein)-[r:consolidated]-(p:pathway)-[s:consolidated]-
(d:disease) WHERE bm.identifier in
["http://identifiers.org/uniprot/P41159",
"http://identifiers.org/uniprot/P31997",
"http://identifiers.org/uniprot/P51888",
"http://identifiers.org/uniprot/P00797",
"http://identifiers.org/uniprot/P09341",
"http://identifiers.org/uniprot/P21583"]
AND d.identifier in ["http://identifiers.org/umls/C1397307",
"http://identifiers.org/umls/C0553980",
"http://identifiers.org/umls/C0151654]

RETURN bm, r, p, s, d
```

Interpret and conclude.
Compare your final graph with the one obtained in Figure 4.

Conclusion

Enjoy querying further the FHF-GKBox !