ETN_Gitte_Decat

2024-04-23

...

Background information on the traineeship topic

Biocypher is created to retrieve information about certain interactions and create a knowledge graph. You can retrieve information about gene-gene interactions, protein-protein interactions, gene-disease interactions, metabolic interactions and biological processes. With this information biocypher will create a visual representation of the complexes.

To be able to create a knowledge graph you need results from a database. Biocypher uses a collection of reusable "adapters" for different sources of biomedical knowledge. Due to the use of adapters, a reproducible knowledge graph can be created and shared for every specific task.
VIB works with the database IRefIndex and would like to be able to use biocypher to create a knowledge graph. The problem is that there is not a adapter available for the IRefIndex database. In this adapter has to be defined where you can find specific information and what has to happen with that information. In particular they want to get information about the protein-protein interactions when creating the knowledge graph.

Examples that will be used to create the adapter file are from CROssBAR. CROssBAR is a comprehensive system that integrates large-scale biological/biomedical data from various resources and stores them in a NoSQL database. CROssBAR is enriched with the deep-learning-based prediction of relationships between numerous data entries, which is followed by the rigorous analysis of the enriched data to obtain biologically meaningful modules. These complex sets of entities and relationships are displayed to users via easy-to-interpret, interactive knowledge graphs within an open-access service. CROssBAR knowledge graphs incorporate relevant genes-proteins, molecular interactions, pathways, phenotypes, diseases, as well as known/predicted drugs and bioactive compounds, and they are constructed on-the-fly based on simple non-programmatic user queries. These intensely processed heterogeneous networks are expected to aid systems-level research, especially to infer biological mechanisms in relation to genes, proteins, their ligands, and diseases.

https://biocypher.org/ https://academic.oup.com/nar/article/49/16/e96/6310792?login=true

TDP

Traineeship documentation plan (TDP)

Student: Gitte Decat Traineeship supervisor: Alexander Botzki

---

Tentative planning:

During this traineeship I will work with Biocypher. Biocypher is created to retrieve information about certain interactions and create a knowledge graph. You can retrieve information about gene-gene interactions, protein-protein interactions, gene-disease interactions, metabolic interactions and biological processes. With this information biocypher will create a visual representation of the complexes.

To be able to create a knowledge graph you need results from a database. Biocypher uses a collection of reusable "adapters" for different sources of biomedical knowledge. Due to the use of adapters a reproducible knowledge graph can be created and shared for every specific task.

VIB works with the database IRefIndex and would like to be able to use biocypher to create a knowledge graph. The problem is that there is not an adapter available for the IRefIndex database. In this adapter file, has to be defined where you can find specific information and what has to happen with that information. In particular they want to get information about the protein-protein interactions when creating the knowledge graph.

These tasks will be performed during the traineeship from 22/04/2024 till 14/06/2024.

Week 1:

Exploring project

Installation of biocypher and poetry

Understanding the scripts that are provided by biocypher (tutorial and CROssBAR)

Creating a workflow of all the steps that happen when running biocypher.

Week 2:

Creating an input file for the IRefIndex database

Creating the adapter file for IRefIndex based on the biogrid/string adapter file from CROssBAR

Week 3:

Creating the adapter file for IRefIndex based on the tutorial (basic version)

Understanding every line of code in the tutotial (add logger functions for information)

Week 4:

Creating the adapter file for IRefIndex based on the tutorial + integrate parts of the biogrid/ intact adapter file from CROssBAR

Week 5:

Creating the adapter file for IRefIndex based on the tutorial + integrate parts of the biogrid/ intact adapter file from CROssBAR

Week 6:

Creating an tutorial so the adapter file that has been created by me, can also be used together with other adapter files (with the same running script)

Week 7:

Creating an tutorial so the adapter file that has been created by me, can also be used together with other adapter files (with the same running script)

Installing/ using neo4j to create a graph from the data

Week 8:

Using neo4j to create a graph from the data

Adjusted the adapter file based on the feedback I got to make it more sufficient and take less memory spece when running it

Data management

All the information that is collected during this traineeship can be found in this github page: https://github.com/GitteDecatBIT/Internship_VIB_2024_GitteDecat.git

In this github repository you can find everything from tutorials, to a workflow and code that is written during the traineeship. The code will be created based on the tutorials that are given by biocypher itself and are found on the website. The github repository that includes the tutorial can be find here: https://github.com/biocypher/project-template.git

Next to the tutorials I will created the adapter file based on the adapter files that can be found in the CROssBARv2 github repository. In this repository, you can find multiple adapter files for various types of information. The adapter files that are created for Biogrid and stringdb fwill be used as an example to create the adapter file for IRefIdex, because there is some similarity between them. This will make it easier to understand the code/logic behind the adapter file. The github repository that includes the CROssBARv2 files can be find here: https://github.com/HUBioDataLab/CROssBARv2.git

FAIR principles:

Findable: The data that is used comes from the IRefIndex database. This can be found on https://irefindex.vib.be/wiki/index.php/iRefIndex . In this database has each interactor a unique identifier and has metadata available. Further a key is generated for each protein interaction record and for each participant protein.
At the end of the traineeship, a zip file will be created of the Internship_VIB_2024_GitteDecat github repository and will be given to VIB.

Accessible:
The iRefIndex database can be accessed via the following link: https://irefindex.vib.be/wiki/index.php/iRefIndex . This is open and free and a PSI-MITAB tab-delimited format is used. My github repository is made public so everyone has access to it. This means that both my internal and external supervisor have access to it

Interoperable:
The iRefIndex data is available in PSI-MITAB 2.6 tab-delimited format. This is the default format or the golden standard in this field. This format is used in most cases.

Reusable:
By providing well-structured, well-documented, and openly accessible protein-protein interaction data is IRefIndex reusable and can be integrated with other databases and tools. For the traineeship, an ETN will be maintained where every step will be written down. Furthermore, the readme file on the github page will be completed at the end of the traineeship so that VIB knows what the github repository contains.

Traceability of steps and methods The ETN with the project steps can be found on the same github page as mentioned before. Everyday the different steps that were performed are documented in the markdown file called ETN_Gitte_Decat.Rmd.

Version control of code The status of my github is set on public, so that the github page is available for everyone. At the end of the traineeship, a zip file will be created from the github repository and this will uploaded/send to the traineeship supervisor from VIB and shared with Howest.

https://github.com/GitteDecatBIT/Internship_VIB_2024_GitteDecat.git

Learning outcome

Learning outcome (LO)

Student: Gitte Decat Traineeship supervisor: Alexander Botzki

---

Development goal:

Improve my Python coding skills and understanding the logic behind the codes. Along with understanding how all the scripts used in Biocypher are connected to each other. This also means understanding what every package/library does and where the information comes from.

Development activity

I will be working with biocypher and the IRefIndex database during this traineeship. The goal is to provide VIB with an adapter file for IRefIndex that can be used with Biocypher. I have to do a lot of research on how these adapters are formulated. They use the packages pypath and this is new to me, so I have to analyse all the components that can be found in this packages and where to use it/ adapt it for mine use.

At last I should be in contact with a person from Biocypher to let them know that I might have an extra adapter that can be used with Biocypher. These people should than look at my code and say if it is correct and if they even want to include to Biocypher and make it available for other users.

Desired results

At the end of the traineeship I would like to be able to write a Python script (adapter file) that makes it able to use IRefIndex in Biocypher.

Schedule

These activities will be performed during the traineeship from 22/04/2024 till 14/06/2024.

Necessary support and facilities

To obtain my goal I will need my traineeship working hours and learning materials such as the IREfIndex database and the Biocypher website that includes: information about the concept, tutorials and the CROssBAR project.

SMART principles

Specific: Creating a python script to be able to use IRefIndex together with Biocypher.

Measurable: At the end of the traineeship I should be able to use the adapter file for IRefindex together with Biocypher and create a knowledge graph.

Achievable: With the time and help I can get at VIB I should be able to create the adapter file.

Relevant: In the context of the traineeship, this learning outcome can be considered relevant. iRefIndex not only brings together data from these different sources but also standardizes it, making it consistent and easier to query. This is crucial because it allows us to efficiently map out protein interactions and understand their roles in biological processes and disease mechanisms

Time-bound: This learning outcome should be completed by the end of the traineeship.

Result

After 8 weeks of this traineeship, I was able to create an adapter file that makes it possible to use the IRefIndex database with Biocypher. I now understand so much more about the scripts that had to be created and where everything comes from.

ETN

Github

Github page where you can find all the data/info that is discussed in this ETN : https://github.com/GitteDecatBIT/Internship_VIB_2024_GitteDecat.git

Week 1

Monday 22/04/2024

Exploring code from Biocypher (adapter)
First, I looked up some information to get more insight of what Biocypher exactly does. After this I looked for code that could help me understand what happens when an adapter is created. This code I found on the github page linked on the biocypher website. https://github.com/HUBioDataLab/CROssBARv2/tree/main/bccb –> string_adapter.py

Creating and updating ETN:
Start working ETN in Markdown

Tuesday 23/04/2024 Exploring code from Biocypher (adapter)

I tried to understand this string_adapter.py code and tried to run it in my virtual studio code. i came accros a few problems that i had to figure out. First of all i had to install Biocypher and create a poetry environment. In this environment i could run the script after i downloaded all the tools/packages that were needed and did not have installed on my system yet. github: CROssBARv2/bccb/string_adapter.py

Installation:

- Install pipx –> sudo dnf install pipx, pipx ensurepath
- Install poetry –> pipx install poetry
- Install biocypher via poetry –> create a environment and add biocyher there:poetry new Biocypher_adapter, cd Biocypher_adapter, poetry add biocypher

Install packages:

- Install bioregestry –> pip install bioregistry
- Install pypath–> pip install pypath-omnipath

Problem: the script did not run completely, when running it it stops after a few step and give the prompt back

Wednesday 24/04/2024 Exploring code from Biocypher (tutorial_adapter)

On the biocypher website you can find a tutorial on how the scripts should work. i performed this tutorial on my system and looked at different scripts that are provided there. i tried to figure out how all the scripts are connected to each other. tutorial on github: Internship_VIB_2024_GitteDecat/Biocypher_tutorial/project-template

Creating a workflow (creating a adapter)

i started creating a workflow based on the tutorial.

Creating github page

Started to create my github page to collect all my data/information

####Thursday 25/04/2024 ##### Updated ETN
i updated my ETN for the past days (22-23 april)

Creating a workflow (creating a adpater)
i finished the first version of the workflow based on the tutorial workflow on github: Internship_VIB_2024_GitteDecat/Workflow/

Added everything to my github Added all the files i changed to my github Added new files to my github

####Friday 25/04/2024 ##### Updated ETN
iupdated my ETN for the past days

IRefIndex code for biocypher i tried figuring out the code on how biocypher is performed and how the adapter is created, with this information i tried to create the code for the IRefIndex database

Run biocypher on Crossbar data
i tried running biocypher with the crossbar data. i wanted to make sure that the code worked on my virtual studio code but i came across a few problems.

Problem: try running crossbar –> could not import validation_class –> pypandtic and curies where not working together –> updated curies did not work

Week 2

Monday 29/04/2024

Updated ETN
i updated my ETN for the week before friday

Exploration of the Biogrid code from Crossbar
I tried figuring out which columns i need to use from the IRefIndex database. I did this based on the code from the Biogrid database. github:Internship_VIB_2024_GitteDecat/Biocypher_IRefIndex

Tuesday 30/04/2024

Updated ETN
i updated my ETN for the day before

Write TDP
i started writing my documentation plan for the 8 weeks of the traineeship github: Internship_VIB_2024_GitteDecat/ETN/

Creating an adapter file for IRefIndex
I created a script that takes the columns you need as input for the adapter file. i tried adapting the biogrid_adapter file to make it work for irefindex github:Internship_VIB_2024_GitteDecat/Biocypher_IRefIndex

Wednesday 01/05/2024

Labour Day Day off

Thursday 02/05/2024

Submitted TDP (external supervisor)
i send my first version of the TDP to my external supervisor of VIB and he gave me some feedback github: Internship_VIB_2024_GitteDecat/ETN/

Creating adapter file for IRefIndex
I tried figuring out how to map the refseq accession number to the uniprot id (using biogrid as a example) github:Internship_VIB_2024_GitteDecat/Biocypher_IRefIndex

Firday 03/05/2024

Rewrite TDP
i took the feedback from my external supervisor into account and rewrited my TDP to make it more clear and uploaded it to the one drive

Write LO i wrote my learning outcome and uploaded it to the one drive

Updated ETN Updated ETN for the last few days

Upload everything to github
I uploaded everything to github so that all the information is there for the first time we had to submit the TDP/ETN

Creating adapter file for IRefIndex
I tried figuring out how to map the refseq accession number to the uniprot id (using biogrid as a example) github:Internship_VIB_2024_GitteDecat/Biocypher_IRefIndex

Week 3

Monday 06/05/2024

Create adapter file based on biocypher tutorial
I tried creating the adapter file based on the tutorial from biocypher, to have a script that works. I try to run the adapter file based on the Crossbar files –> it does not work. Therefore i went back to an more easier example to take as a base. i added some parts from the crossbar based file to make it more specific for IRefIndex ##### Updated ETN Updated ETN for the this day

Tuesday 07/05/2024

Create adapter file based on biocypher tutorial
I tried including the input from the IRefIndex database. With the structure of the tutorial it is possible to run the scripts and create output. but the output does not overlap with the input from irefindex. In the tutorial they just take random ids, labels and so on... this needs to be adjusted for the irefindex input. i also added some logger functions to make it clear for myself what happens as what time.

Wednesday 08/05/2024

Create adapter file based on biocypher tutorial
I was able to include the input from the IRefIndex database. With the structure of the tutorial it is possible to run the scripts and create output. this is the output: 2024-05-08 16:59:55,013 INFO module:_logger This is BioCypher v0.5.37. 2024-05-08 16:59:55,013 INFO module:_logger Logging into `biocypher-log/biocypher-20240508-165955.log`. 2024-05-08 16:59:56,445 INFO module:CROssBAR_IRefIndex_input Running input script for IRefIndex 2024-05-08 16:59:56,446 INFO module:CROssBAR_IRefIndex_input Downloading interations from IRefIndex 2024-05-08 17:00:02,727 INFO module:CROssBAR_IRefIndex_input Getting information for the IRefIndex database: partner_a, partner_b, pumed_ids, method and organism 2024-05-08 17:00:02,837 INFO module:_get Loading cache file .cache/cache.json. 2024-05-08 17:00:02,838 INFO module:_get Use cached version from .cache/IRefIndex. 2024-05-08 17:00:02,839 INFO module:create_knowledge_graph_IrefIndex path: ['.cache/IRefIndex/7227.mitab.08-28-2023.txt.zip', '.cache/IRefIndex/7227.mitab.08-28-2023.txt.zip.unzip'] 2024-05-08 17:00:02,842 INFO module:_ontology Loading ontologies... 2024-05-08 17:00:02,843 INFO module:_ontology Instantiating OntologyAdapter class for https://github.com/biolink/biolink-model/raw/v3.2.1/biolink-model.owl.ttl. 2024-05-08 17:00:07,541 INFO module:_write Creating output directory

/home/guest/Github/Internship_VIB_2024_GitteDecat/Create_IRefIndex_adapter_based_on_tutorial/biocypher-
2024-05-08 17:00:07,541 INFO module:IRefIndex_adapter_03 Generating nodes. 2024-05-08 17:00:08,124 WARNING module:_deduplicate Duplicate node type protein found. 2024-05-08 17:00:08,126 INFO module:_write Writing 1 entries to Protein-part000.csv 2024-05-08 17:00:08,126 INFO module:IRefIndex_adapter_03 Generating edges. 2024-05-08 17:00:08,127 INFO module:IRefIndex_adapter_03 Generating nodes. 2024-05-08 17:00:08,868 INFO module:_write Writing 49 entries to ProteinInteraction-part000.csv 2024-05-08 17:00:08,869 INFO module:_write Writing neo4j import call to /home/guest/Github/Internship_VIB_ 2024-05-08 17:00:08,869 INFO module:_ontology Showing ontology structure based on https://github.com/biolink/biolink-model/raw/v3.2.1/biolink-model.owl.ttl 2024-05-08 17:00:08,870 WARNING module:_misc The ontology contains multiple inheritance (one child node has multiple parent nodes). This is not visualized in the following hierarchy tree (the child node is only added once). If you want to browse all relationships of the parsed ontology write a graphml file to disk and view this file. 2024-05-08 17:00:08,870 INFO module:_ontology

2024-05-08 17:00:08,870 INFO module:_core Duplicate node types encountered (IDs in log): protein

2024-05-08 17:00:08,870 INFO module:_core No duplicate edges in input. 2024-05-08 17:00:08,870 INFO module:_core No missing labels in input.

problems: - only takes 1 id (first one) - duplicates ar found - still takes random relationship id –> generating edges (id column in protein-interaction file) - second node id is not the right one yet - label column in protein -part000.csv -> dont know where this comes from - example property column in protein interaction file –> dont know what this needs to be / or that it needs to be included -

Updated ETN Updated ETN for the tuesday and wednesday

Thursday 09/05/2024

Ascension
Free day

Friday 10/05/2024

Create adapter file based on biocypher tutorial
Changed the way the data is downloaded in the adapter file. Use the url in the resource for downloading the data and parse the data in the adapter file other de def get_nodes.

Week 4

Monday 13/05/2024

Create adapter file based on biocypher tutorial
i changed the way that the url is specified. The url is now specified in the biocypher config file. the donwload of the data happens via the resource and the parsing of the file happens in the adapter file.

integrating the process step of biogrid in the adapter file: errors: 1) File "/home/guest/.cache/pypoetry/virtualenvs/biocypher-project-template-fuLsphMc-py3.11/lib64/python3.11/site-packages/pandas/core/indexes/base.py", line 6251, in _raise_if_missing raise KeyError(f"{not_found} not in index") KeyError: "['uniprot_a', 'uniprot_b'] not in index"

–> uniport_a and uniprot_b are not specified in the input, changed it to partner_a and partner_b

2) File "/home/guest/Github/Internship_VIB_2024_GitteDecat/Create_IRefIndex_adapter_based_on_tutorial/templat line 252, in aggregate_fields element = "|".join([str(e) for e in set(element.dropna())]) ^^^^^^^^^^^^^^^^^^^^^^^^ TypeError: unhashable type: 'list'

–> the uniprot column was not correctly parsed, there were some special cases (eg: pubmed:10.1038/nsmb.2049|pubmed:21552260 and these were recognised as a list. –> Adapted the code for parsing that column

Tuesday 14/05/2024

Update ETN

Updated ETN for the day before ##### Create an overview of the steps that need to happen in the adapter file
Created an overview for myself of which steps happen in the adapter file while getting the nodes and edges ##### Create adapter file based on biocypher tutorial
Created an adapter file that runs fully. the nodes and edges are not correctly specifief yet, because their is no output created.

1) Traceback (most recent call last): File "/home/guest/Github/Internship_VIB_2024_GitteDecat/Create_IRefIndex_ada line 73, in bc.write_nodes(adapter.get_nodes()) File "/home/guest/.cache/pypoetry/virtualenvs/biocypher-project-template-fuLsphMc-py3.11/lib/python3.11/site-packages/biocypher/_core.py", line 279, in write_nodes nodes = peekable(nodes) ^^^^^^^^^^^^^^^^ File "/home/guest/.cache/pypoetry/virtualenvs/biocypher-project-template-fuLsphMc-py3.11/lib/python3.11/site-packages/more_itertools/more.py", line 320, in **init** self._it = iter(iterable) ^^^^^^^^^^^^^^^ TypeError: 'NoneType' object is not iterable
2) ERROR wile writing edge data

Wednesday 15/05/2024

Create adapter file based on biocypher tutorial

Get nodes from the adapter file works + output gets created. The information in the output file is still random. Trying to figure out how to specify the correct information per node

Thusday 16/05/2024

Create adapter file based on biocypher tutorial

Trying to figure out how to specify the correct information per node. Came accros a problem that the dataframe is too small. Way to many rows are getting removed –> filters to strict based on the swissprot ids Removed the part from the code where they remove rows from the dataframe if uniprot_a or uniprot_b are not in swissprot ##### Update ETN
Updated ETN for the day before

Friday 17/05/2024

Create adapter file based on biocypher tutorial

Trying to figure out how to specify the correct information per node (pmid and taxon) ##### Intermediate evaluation Got feedback from my internal supervisor on my skills and got feedback on my ETN from my external supervisor

Intermediate evaluation traineeship

The feedback i got from my supervisor was expected by me. I can still grow a lot in my coding skills and i am verry happy to hear that my supervisor thinks that this growth will be there at the end. I still have a lot to work on, but i know/feel that I am going to get there with little steps at a time.

Week 5

Monday 20/05/2024 #####PentecostMonday
Free day

Tuesday 21/05/2024

Create adapter file based on biocypher tutorial
Trying to figure out how to specify the correct information per node (method)

problems: 1) ERROR – Error while writing edge data.

–> biocypher/biocypher/write/_batch_writer.py line 380

edge_list: in code: edge_list.append((None, _source, _target, "Interacts_With", _props)) in terminal: (None, 'uniprot:X2JFU8', 'uniprot:P40793', 'Interacts_With', {'pubmed_ids': '36436559', 'taxon': '7227', 'method': 'pull down'})]

2) WARNING – The ontology contains multiple inheritance (one child node has multiple parent nodes). This is not visualized in the following hierarchy tree (the child node is only added once). If you want to browse all relationships of the parsed ontology write a graphml file to disk and view this file.

–> biocypher/biocypher/_misc.py line 117

Wednesday 22/05/2024

Create adapter file based on biocypher tutorial
Fixed the problem when writing edge data –> specified the right label in the config file (these 2 were not the same in the separate scripts)

Thusday 23/05/2024

Create adapter file based on biocypher tutorial
Integrated refseq and genbank in the adapter file so that these ids were not skipped in the process step. tried mapping the refseq and genbank ids to uniprot ids, but this was to difficult because not all refseq ids had a uniprot id that they could be mapped to.

Friday 24/05/2024

Create adapter file based on biocypher tutorial
Integrated the refseq and genbank ids in the config file so that the scripts creates separate output files for the different types of id. adapted the get_nodes function to make it possible to create those separate files per id type

solution: - Function to determine if an ID is a UniProt ID def is_uniprot_id(node_id): return bool(re.match(r'^([A-N,R-Z][0-9]([A-Z][A-Z, 0-9][A-Z, 0-9][0-9]){1,2})|([O,P,Q][0-9][A-Z, 0-9][A-Z, 0-9][A-Z, 0-9][0-9])(.+)?$', node_id))

- Function to determine if an ID is a refseq ID def is_refseq_id(node_id): return bool(re.match(r'^(((AC|AP|NC|NG|NM|NP|NR|NT|NW|NZ)_[0-9]+)|((XM|XP|XR)_[0-9]{6,9}))(.+)?', node_id))
- Function to determine if an ID is an entrez ID
  def is_entrez_id(node_id): return bool(re.match(r'[1]+[0-9]+(.+)?$', node_id))

Tried specifying the right prefix to the id when getting the edges, but this part does not work yet. the right prefix (type of id) is not integrated in the output file yet. ##### ETN updated
updated my ETN for the past few days

Friday 24/05/2024

[1] A-Z

Create adapter file based on biocypher tutorial

Fixed the problem when specifying the protien type in the protein interactions output file. add.prefix def –> if self.add_prefix and identifier: if is_uniprot_id(identifier): prefix = “uniprot” elif is_refseq_id(identifier): prefix = “refseq” elif is_entrez_id(identifier): prefix = “entrez” else: prefix= “ “) ##### ETN updated updated my ETN for the past few days

Week 6

Monday 27/05/2024

Created a tutorial

Checked if my running script and the irefindex adapter file could go together with the uniprot adapter file from crossbar. Tried downloading the uniprot data using the same logic as the irefindex adapter file –< donwloading via the resource from biocypher –> too difficult

Tuesday 28/05/2024

Created a tutorial

Used the uniprot adapter file from crossbar to download and proces the data –> had some problems when trying to execute the uniprot adapter file. Had to change some versions of packages that are specified in the toml file: downgraded pypath-omnipath and upgraded pydantic? ##### Feedback on adapter file
Got some feedback on my adapter file from James, he told me some things i needed to change to make the code better/more logic

feedback: - create the url based on taxon_id and release version –> changed this in the create_knowledge_graph - change absolute path to the config file in the create_knowledge_graph and the absolute path to the inpit file in the adapter file from irefindex - specify the organism if you selected to donwload the file that contains all the organisms - relationship_id is still in the node output csv, this does not need to be there - error while writing edge data when downloading the file that contains all the organisms - defs to get the protein types are specified twice, maybe change this so you dont repeat things restrict to 1 organism –> sometimes there are more taxon_id in the output files - QC!!!!

Wednesday 29/05/2024

Create adapter file

fixed somethings from the feedback! i changed the absolute paths using os. i created the url where you can specify the taxon_id and the relase version. this works for the files that contains 1 organisms, but when i want to run the all organisms file i get an error while writing the edge data.

performing the all organism file–> error while writing the, it only gives the source or the target id not both

Thusday 30/05/2024

Create adapter file

Tried fixing this error from wednesday, but with no success

Friday 31/05/2024

Create adapter file

Fixed the error by filtering on organism in the data file that contains all organisms. This was to big the process as a whole. Added some error messages if the taxon_id is not in the file. Fixed the problem that the relationship id is not in the output file from the nodes, deleted it from the properties in the config file

Week 7

Monday 03/06/2024

Create tutorial with irefindex and uniprot

Fixed the errors when executing the uniprot adapter file, got some errors with the esm2_embedding part. Ignored this for now (delted ot from the code). now the adapter file runs, but still gives some errors when writing the edge data (headers) Still a problem that i always downkoads it again.

Tuesday 04/06/2024

Installation of neo4j

Did not work –> w not working with the right lines of code for a red hat distribution

Wednesday 05/06/2024

Installation + use of neo4j

1) rpm –import https://debian.neo4j.com/neotechnology.gpg.key cat « EOF > /etc/yum.repos.d/neo4j.repo [neo4j] name=Neo4j RPM Repository baseurl=https://yum.neo4j.com/stable/5 enabled=1 gpgcheck=1 EOF
2) yum install neo4j-5.20.0
3) sudo /usr/bin/neo4j-admin database import full –nodes="/home/guest/Github/Internship_VIB_2024_GitteDecat/Crea out/20240605113010/Uniprot.Protein-header.csv,/home/guest/Github/Internship_VIB_2024_GitteDecat/Create_IRefI out/20240605113010/Uniprot.Protein-part." *–nodes="/home/guest/Github/Internship_VIB_2024_GitteDecat/Create_ out/20240605113010/Refseq.Protein-header.csv,/home/guest/Github/Internship_VIB_2024_GitteDecat/Create_IRefInd out/20240605113010/Refseq.Protein-part.*" –nodes="/home/guest/Github/Internship_VIB_2024_GitteDecat/Create_I out/20240605113010/Entrez.Protein-header.csv,/home/guest/Github/Internship_VIB_2024_GitteDecat/Create_IRefIn out/20240605113010/Entrez.Protein-part." *–relationships="/home/guest/Github/Internship_VIB_2024_GitteDecat/Cr out/20240605113010/Protein_protein_interaction-header.csv,/home/guest/Github/Internship_VIB_2024_GitteDecat/C out/20240605113010/Protein_protein_interaction-part.*"

Thusday 06/06/2024

Installation + use of neo4j

Dont install neo4j in terminal, do it with desktop. Tried importing data into neo4j, but got an error that some nodes are missing. looked back at the code and output from the adapter file to make sure that all the nodes are in there. Found some mistakes in the code when checking if a certain node is a uniprot/refseq/entrez protein.

Friday 07/06/2024

Use of neo4j

Tried importing the data into neo4j! nodes were imported, but there were 0 relationships. Tried adjusting code in adapter file to make the format like it should be, but no success.

WEEK 8

Monday 10/06/2024

Use of neo4j
Figured out the problem why the importing would not work. The ids were not the same in the nodes file and edge files. Got the query to create a graph in neo4j

Got feedback from James to make my code more efficient and take less memory when executing it. –> Issues in github page: - Tighten up memory usage - Add a way to not filter on taxonID (or filter on *all* taxon ids) - Uncontrolled file directory walking and file input - Import strangeness - Use of no-op functions - Hard coded taxon IDs

Tuesday 11/06/2024

Integrated feedback in adapter file
Tried fixing some issues from github:

- Changed code so the taxon_id is asked in the terminal: def get_taxon_id_from_arg(): “ ““Get taxon_id from command line arguments.”“ ” if len(sys.argv) < 2: raise ValueError(“Please provide a taxon_id as a command line argument.”) return sys.argv[1]
- Change the interaction collections to a tuple: @dataclass class Interaction: partner_a: str partner_b: str pmid: set[str] method: set[str] taxon: set[str] relationship_id: set[str]

  interactions: dict[tuple[str, str], Interaction] = {}

Doesnt get further then the process step–> interactions is empty

Wednesday 12/06/2024

trial presentation
Finished ppt ad created a text to present. Presented for my mentor and got some feedback and answered some questions

Thusday 13/06/2024

Integrated feedback in adapter file
Fixed issues from github:

removed no-op functions in create_knowledge_graph remove import from create_knowledge_graph to get ride of cycle import –> imported as parameters when calling the function set an if statement to if the option to not filter on taxon_id hen entering “*”

Self-assesment

Learning outcome 1

The student writes, adjusts or uses code on the command line to process (biological) data, to create or manipulate a database structure, to create a (web) interface, ... independent of the programming language –> Good: After copying example code from the biocypher tutorial and CROssBAR, i was able to created an adapter file for IRefIndex by adapting parts of the code.

The code is written in an elegant way, is not redundant and/or makes use of loop structures and functions where possible
–> Good: The code that has been created uses loops if necessairy. The running script only contains lines of code that call defenitions from the adapter file. This makes it more clear to other users to follow the steps of the process

Learning outcome 2

The student adjusts an analysis pipeline or script in a correct way, with eye for detail, with the goal to apply the pipeline or script within a new research context or on new data
–>Excellent: By adding ways to download different files for different taxon_ids, I was able to handle various types of input files. IRefIndex has a list of files that can be downloaded for a few taxon_ids, if you want another taxon_id a file that contains all organsims will be downloaded

The student is able to work with files containing data, to read data and write data to new files, to filter and/or transform data, ... and to check the data for correctness
–> Good: IRefIndex data is donwloaded, processed and written to output files (nodes and edges, total of 9 output files)

Learning outcome 3

The student can deposit data in a (relational) database system and/or can query a database system, whether or not by using a programming language
–>Sufficient: Neo4j is used as a database to create a graph from the data. This query language was new to me

The student creates a (relational) database system and/or creates a directory structure to store and organize (biological) data to be able to efficiently select data from the database/directory structure
–>Good: Everything canb be found in my github repository, there is a link to the repositories a used as base to create the adapter file. Because of the readme file it is clear what to execute in which folder

The student makes use of (dynamic) programming to develop an interface to visualize, analyse or store (biological) data, independent of the programming language
–>Good: By creating nodes and edges with biocypher, i also get an output file that makes it possible to import data to neo4j. Neo4j is used to create a graph from that data

Learning outcome 4

The student looks independently for new possibilities to program more efficiently, to store, organize or process data, to analyze data, ... and is able to apply the new techniques, with or without results
–>Sufficient: This part has mostly been done by James, he took al look at my code and had some feedback to make the code more efficient

The student learns new programming and data processing techniques to store select, organize or analyze biological data in a more efficient way –> Excellent: I was able to use a new database, called neo4j. This is a graph database were i got import my data that was generated and i got use the cypher query language to create a graph from the proein interactions

The student knows and spontaneously applies the rules to protect non-public data
–> N/A /

Learning outcome 5

The student searches for and compares various programming and data processing techniques and has insight into the advantages and disadvantages of the different techniques
–> Sufficient: I did not really compare different techniques, i used the tutorial as a base an adjusted the code from there. Thelast week James took a look at my code and gave some feedback on how to make it more sufficient and he also helped with the small details to make that new version work.

The student himself can select the appropriate programming language or software to process data on the basis of advantages and disadvantages –> Good Biocypher has a tutorial and a general set-up on how to created the files. This files are .py, .yaml and .toml files . The main scripts that had to be created be me are written in python

Learning outcome 6

The student combines programming skills with computer science knowledge and bioinformatics techniques to solve a (biological) research question, but the focus of the project remains on the bioinformatics aspect
–>Good: At the and of my traineeship i was able to created an adapter file for iRefIndex (scripts are written in python and i used neo4j to create a graph)

The student can use his knowledge from multiple disciplines to summarize results in a scientifically correct way and to make a conclusion for the project
–> Good After did a trail presentatin I was able to answer the questions from Alexander and James, sometimes i needed some help to get the whole answer.

Learning outcome 7

The student organizes his code, information and documents in an orderly manner, uses comments to document code, keeps track of results in an (electronic) lab notebook and/or uses version control (e.g. GIT)
–> Excellent I wrote everything down in my ETN and i pushed almost daily new updates to my github page

The student works fluently with colleagues, helps or asks for help wherever possible or needed, does more than just the bare minimum, ... –> Excellent I ask for help when i need it. I wrote my questions down on paper for when my supervisor was here to answer my questions. I also had some meetings with James to check my code if i got stuck on the parts that he gave feedback on

International aspects

- Internal supervisor: Alexander Botzki, James collier
- James is an international college
- Alexander had some congresses he had to attend, which was in total for 10 days.
- We could rely on James Collier (our second supervisor) at all times,James works from home but we were able to communicate through chat or teams meetings very easily.
- We saw Alex on average once a day

Friday 14/06/2024

Integrated feedback in adapter file

When downloading taxon_id 9606 and 559292 it gives an error when writing edges–> was not able to fix this. Supervisor will take a further look to this.