POLITÉCNICA

TECHNICAL UNIVERSITY OF MADRID

HIGHER TECHNICAL SCHOOL OF AGRONOMIC, FOOD AND BIOSYSTEMS ENGINEERING

MASTER'S DEGREE IN COMPUTATIONAL BIOLOGY

DEPARTMENT OF BIOTECHNOLOGY - PLANT BIOLOGY

CanGraph: a python utility to study and analyse cancer-associated metabolites using knowledge graphs

FINAL MASTER THESIS

Author: **Pablo Ignacio Marcos López**

Tutors : **Adam Amara**
María Poveda Villalón

Carried out at the **International Agency for Research on Cancer**

July 2022

TECHNICAL UNIVERSITY OF MADRID

HIGHER TECHNICAL SCHOOL OF AGRONOMIC, FOOD AND BIOSYSTEMS ENGINEERING

MASTER'S DEGREE IN COMPUTATIONAL BIOLOGY

CANGRAPH: A PYTHON UTILITY TO STUDY AND ANALYSE CANCER-ASSOCIATED METABOLITES USING KNOWLEDGE GRAPHS

FINAL MASTER THESIS

Pablo Ignacio Marcos López

MADRID, 2022

Cotutors: Adam Amara, International Agency for Research on Cancer
María Poveda Villalón, Department of Artificial Intelligence, School of Computer Engineering, Technical University of Madrid

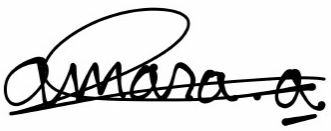
TÍTULO - CANGRAPH: UN PROGRAMA ESCRITO EN PYTHON PARA ESTUDIAR Y ANALIZAR METABOLITOS ASOCIADOS AL CÁNCER MEDIANTE GRAFOS DE CONOCIMIENTO

Memoria presentada por Pablo Ignacio Marcos López para la obtención del título de Máster en Biología Computacional por la Universidad Politécnica de Madrid

Fdo:
Pablo Ignacio Marcos López
Alumno de Máster
ETSIAAB-UPM & IARC

VºBº Cotutor

M. Adam Amara,
Researcher,
International Agency for Research on Cancer



VºBº Tutor UPM

Dña. María Poveda Villalón,
Department of Artificial Intelligence,
School of Computer Engineering,
Technical University of Madrid

Madrid, a 24 de junio de 2022

Agradecimientos

A los que estais ahora y espero que esteis siempre:
A Isa, a Marc, a Carlo, a Gabi, a Sergio, a Carlos, y a mi mismo

A los que sólo estuvisteis un rato y no volvereis a estar

A mi familia y a mis amigos, por estar siempre ahí y apoyarme cuando más los he necesitado. Sin ellos esto no habría sido posible

Y a la Caisse d'Allocations Familiales du Rhône por sus facilidades y rapidez a la hora de financiarme.

CanGraph: a python utility to study and analyse cancer-associated metabolites using knowledge graphs

CanGraph: un programa escrito en python para estudiar y analizar metabolitos asociados al cáncer mediante Grafos de Conocimiento

Pablo Ignacio Marcos López

June 24, 2022

Contents

.

Abstract

Abstract

Research on cancer, one of the most lethal diseases in the world today, is an expensive, complex process, usually carried out manually in laboratories. In this publication, we present CanGraph, a software solution that allows its users to annotate and interpret unknown metabolites by making use of five pre-existing databases (HMDB, SMPDB, DrugBank, ExposomeExplorer and Wikidata) and five search criteria (InChI, InChIKey, Structural Similarity, HMDB_ID, Name and ChEBI ID), resulting in an output database in GraphML format containing the associations to the different metabolic pathways, tissues and organisms to which these molecules may belong. Although it still presents problems, such as the long processing time, we hope that this program will be useful in automating the search for potential relationships between compounds and various diseases (specially cancer, as is the mission of International Agency for Research on Cancer (IARC), the Institution where this project has been carried out), with a view towards generating a web service that will make this program, and all its knowledge, available to the scientific community at large.

Keywords: metabolomics, cancer, python, bioinformatics

Resumen

La investigación sobre el cáncer, una de las enfermedades más letales del mundo en la actualidad, es un proceso caro y complejo, que suele llevarse a cabo manualmente en laboratorios. En este trabajo, presentamos CanGraph, una utilidad de software que permite a sus usuarios anotar e interpretar metabolitos desconocidos haciendo uso de cinco bases de datos preexistentes (HMDB, SMPDB, DrugBank, ExposomeExplorer y Wikidata) y cinco criterios de búsqueda (InChI, InChIKey, Similaridad Estructural, HMDB_ID, Nombre y ChEBI ID), dando como resultado una base de datos en formato GraphML que contiene las asociaciones a las diferentes rutas metabólicas, tejidos y organismos a los que estas moléculas puedan pertenecer. Aunque todavía presenta problemas, como el largo tiempo de procesamiento, esperamos que este programa sea útil para automatizar la búsqueda de posibles relaciones entre compuestos y varias enfermedades (especialmente el cáncer, como figura en la misión de la Agencia Internacional de Investigación sobre el Cáncer (IARC), institución donde se ha llevado a cabo este proyecto), con vistas a generar un servicio web que ponga este programa, y todo su conocimiento, a disposición de la comunidad científica en su conjunto.

Palabras clave: metabolómica, cancer, python, bioinformática

Introduction

Currently, cancer is one of the most devastating diseases in existence. Worldwide, more than 19 million diagnoses and almost 10 million deaths occur each year from this family of diseases,[1] accounting for 20% of deaths (and therefore being the most prevalent cause of it) in developed countries, and 13% of deaths in the developing world.[2] Because of this, billions of euros are spent annually on cancer research,[3] trying to find potential associations between cancer-causing compounds and drugs that may be able to treat said diseases.

A modest part of this work is carried out at the International Agency for Research on Cancer, the World Health Organisation's cancer research agency, which regularly publishes Monographs on some substances and their whether they might be classified or not as carcinogenic. As part of this work, and as in the rest of the industry, one of the activities that consumes most resources and effort is precisely the identification of new metabolites as potentially carcinogenic and their annotation and interpretation through the different pathways, tissues and organisms to which these substances may belong. This work is mainly done by hand, employing enormous amounts of human and personal resources which, if automated, could be freed up, allowing us to further expand the Agency's research efforts.

To solve this problem, metabolomics is being increasingly employed in cancer research.[4] This discipline, consisting on the global analysis of small molecule metabolites, can provide critical information about the cancer state that are otherwise not apparent, and makes automatizing the discovery of this new information easier and simpler.

Objectives

Given the need to automate, as far as possible, the search for new, potentially cancer-causing substances, and suitable drugs to treat cancer, and given the ease with which machines can find associations based on complex patterns (such as structural similarity) that can be difficult for humans to discover, the IARC Metabolism and Nutrition team has decided to create the CanGraph project. This consists of a series of software utilities that use metabolomics data to automatically annotate and interpret metabolites found in cancer research, discovering potential associations with cancer and associating them with potential membership in known metabolomic pathways in humans. The objectives of this project are several:

- To begin with, we want to create a Python program for internal use in the Agency, which allows the scientists who collaborate with it to obtain a series of Knowledge Graphs (that is, some graph-structured data models) that present a clear visualization of the function of the subject metabolite inside the human metabolome as a whole.
- Then, we would like to automate the analysis of these resulting networks. For instance, we would like to develop a way to find associations with cancer or other diseases, potential membership to known regulatory mechanisms and pathways on the metabolome, or interactions with other metabolites.
- Finally, we would like to, eventually, offer the program as Software as a Service (SaaS), inside a web utility that allows interested researchers to make use of our work.

Matherials and Methods

As explained, the intention of the solution herein described is to annotate and interpret a series of metabolites recently discovered inside a laboratory, trying to associate them to those tissues, metabolic pathways and/or organisms to which these metabolites might belong. For this, we have used a series of pre-existing resources, which we have fined-tuned to better suit our needs:

The Databases

In order to automate the annotation and interpretation of metabolites, the first thing we will need is a list of pre-existing, high-quality databases in which to search for compounds similar to those that may be identified in a laboratory. Thus, we have chosen the following five databases, which we hope will provide a comprehensive overview of the human metabolome and the interactions that the molecules in it have with various types of cancer:

- The **Human Metabolome Database (HMDB)** is an open-access database containing detailed information on small molecule metabolites found in the human body, intended for applications in metabolomics, clinical chemistry, biomarker discovery and general education.[5] The database includes chemical as well as molecular and biochemical data, including over 41,000 metabolite entries and approximately 7200 protein and DNA sequences, and is provided by the Wishart Research Group, a laboratory led by Dr. David Wishart in the Departments of Biological Sciences and Computer Science at the University of Alberta, in Edmonton, Canada.
- **DrugBank** is a bioinformatics and chemoinformatics resource that combines detailed drug data with comprehensive drug target information, containing over 7,800 drug entries and nearly 2,200 FDA-approved small molecule drugs, 340 FDA-approved biotech drugs, 93 nutraceuticals and >5,000 experimental drugs, which can be linked to protein sequences (drug targets) and product data.[6] The database is provided by OMx Personal Health Analytics Inc, a spin-off of the Wishart Research Group founded at the University of Alberta.
- The **Small Molecule Pathway Database (SMPDB)** is an interactive database containing over 618 small molecule pathways found in humans, over 70% of which are not found in any other pathway database.[7] It is designed to support pathway elucidation and discovery in metabolomics, transcriptomics, proteomics and systems biology by providing detailed representations of the pathways, metabolites and proteins it contains. It has also been developed by the Wishart Research Group.
- **Exposome Explorer** is the first database dedicated to biomarkers of exposure to environmental risk factors for disease, with a particular focus on cancers. It aims to provide comprehensive data on all known biomarkers of exposure to dietary factors, pollutants and pollution measured in population studies by collecting information on more than 800 peer-reviewed publications containing more than 10,000 measurements of different metabolites to test whether they can be used as biomarkers for a given disease.[8] This database is an internal development of IARC (where this work is carried out), but is publicly accessible on the Internet.
- **WikiData** is a free and open source knowledge base that can be read and edited by both humans and machines.[9] As a central repository for the structured data of the Wikimedia projects (including Wikipedia, Wikivoyage, Wiktionary and Wikisource) it is one of the world's largest collaboratively generated Open Data collections, which, although probably of lower quality due to being freely editable and not produced solely by experts, hopefully will include a large amount of generalist data, such as relationships between metabolites and diseases or a basic ontology of the different types of cancer.

Each of these databases has its pros and cons, which, as may be visualised in **Table 1**, and complement each other: for example, the SMPDB contains high quality data for pathways, but less information for metabolites than, for example, the Human Metabolome Database, which in turn is complemented by DrugBank, which has more information on drugs than on metabolites. On the other hand, Exposome Explorer allows us to find associations between all these metabolites, dietary intakes (the *exposome*, a novel concept defined by the CDC as "the measure of

all exposures of an individual over a lifetime and their relationship to health"[10]) and various diseases. Finally, WikiData allows us to add both general information (relationship of certain drugs and/or metabolites with diseases) and detailed information (for example, identifiers in external databases such as UniProt or Pubmed IDs for references, which are usually better detailed in this database than in, for instance, SMPDB, which however provides lots of information that is only present there) in a massive way, although with the drawback of being probably less accurate than other databases.

	HMDB	SMPDB	DrugBank	Exposome Explorer	WikiData
Data Quality	High	Huge	High	High	Low
Data Quantity	High	Medium	High	Medium	Huge
External IDs	Medium	Low	Medium	High	High
Info on Metabolites	Huge	Low	Medium	High	Medium
Info on Associated Nodes	Medium	Low	Medium	High	High

Table 1: Pros and Cons of each of the five databases being used.

The Database Management System (DBMS)

In order to define, create, maintain and control access to these databases, a Database Management System that can work with various types of input is needed. For this, we have chosen Neo4J, one of the most widely used DBMS in the world of computing, and which, unlike other more common ones such as MySQL or MariaDB, works in a "non-relational" way, i.e. the data is not structured as a series of interconnected tables with primary and secondary keys and a list of related values, but is presented, in this case, in a graph format.

In mathematics and computer science, a graph is a structure consisting of a series of objects (called "vertices" or "nodes") that may or may not be related by a series of "edges" or "arcs", which allow binary relationships between the elements of the set to be represented.[11] The advantage of this type of databases, which have re-emerged since the 1960s, is that they allow for more efficient processing on those areas of knowledge that can be represented as "networks" (e.g. a person's list of friends on Facebook, or, in the case in point, the human metabolism, which is made up of a series of metabolites (e.g. a the *nodes*), that are interrelated by enzymes and other non-enzymatic reactions (the *edges*)), as well as its greater efficiency in finding elements (because, instead of needing to find the index of each element in each table, the element itself directly points to related nodes in the database).[11] Because of these advantages, the Neo4J data model is simpler and more expressive than other RDMS such as MySQL, allowing us to query the graph almost in natural language.

The Common Schema

In order to obtain a coherent, useful and reliable result, we have designed the software in question so that it is capable of presenting its results according to a schema common to the 5 databases, which has been designed to simplify the result as much as possible while minimising the loss of information. This schema, which can be consulted at large in **Annex I**, has been designed taking into account the particularities of each of the 5 databases, their strengths and weaknesses, and unifying the types of nodes created and their properties in order to maximise the value of each data field. In **Figure 1** you can see a preview of it, as well as its differences and similarities with the original schema.

As can be seen, this scheme has been acquired by progressively merging different types of nodes; for example, all nodes dealing with diseases, whether cancers or not, have been renamed *Disease*.

The *MicrobialMetabolite* and *Component* nodes, which come from Exposome Explorer, have been merged with the rest of *Metabolite*s, eliminating a number of unhelpful properties present in the original database (those ending in * _count *). For the drugs present in DrugBank, the patent holders have not been added, but only the *Company* that is responsible for the manufacturing and packaging of the drugs, in order to filter them if necessary; prices have also been excluded, since we did not consider them relevant. There are also nodes that have been designed from the beginning to have common keys: for example, the *ExternalEquivalent*s, which appeared in WikiData, have been uniformly designed to represent **exact** equivalents of a given node.

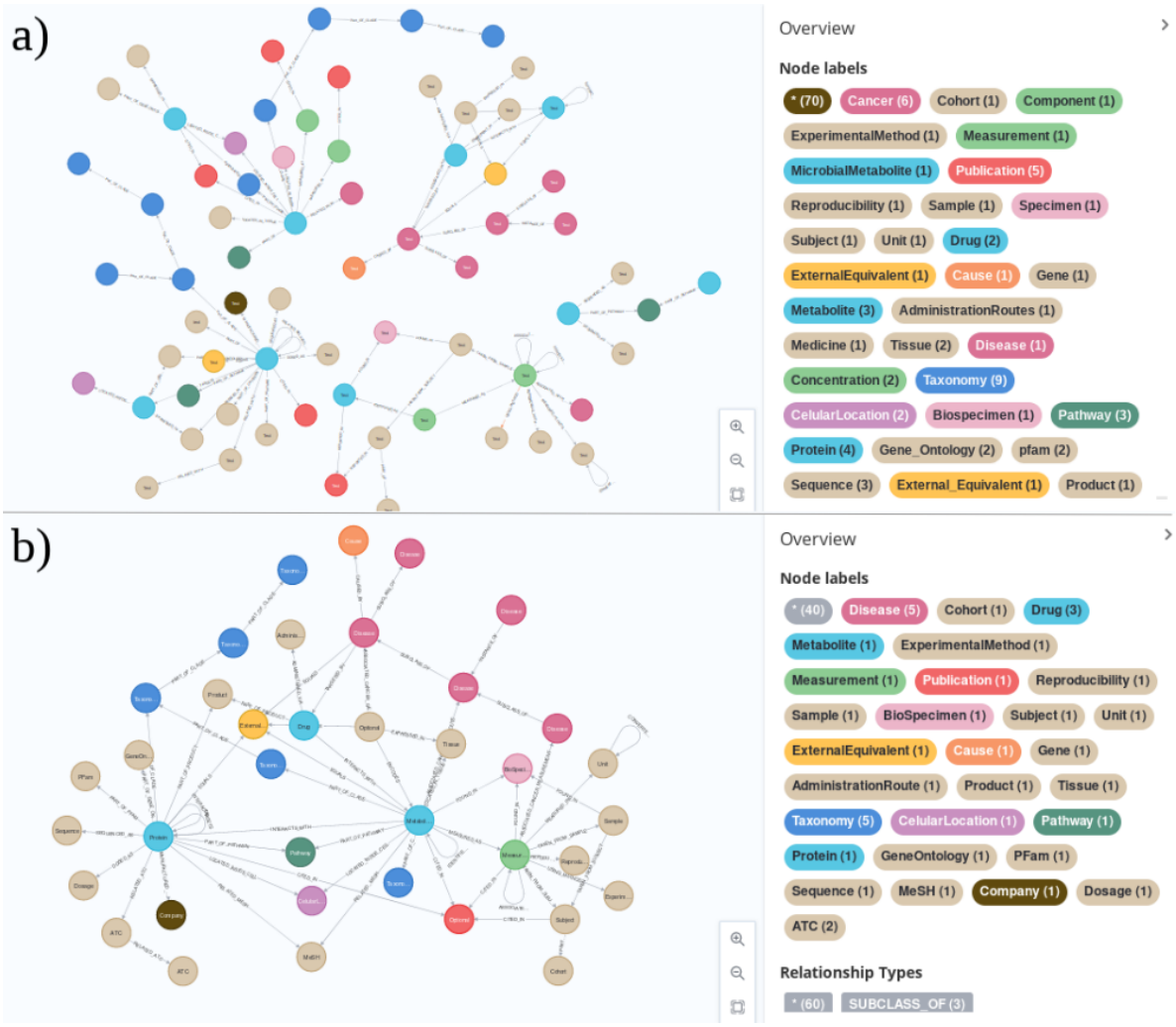


Figure 1: Schemas for our databases **a)** Before and **b)** After unifying them into a common schema. Before, we had five separated patterns; whereas, now, we have a unified one, in which information is condensed, keys are unified and predictability, expected. The number of data fields, nodes (from 70 to 29 (may be repeated in the image for better representation)) and relations (from 98 to 35) has also been reduced.

In general, a thorough analysis of the different fields present in the five databases has been carried out, considering those equivalent to merge them and those different to keep them separate. For example, while the *Kegg_Pathway_ID* and *Kegg_Component_ID* can be merged into a single *KEGG_ID* (as they are prefixed and values cannot collide), the *FooDB_IDs* necessarily need a separate field for the *FooDB_Compound_ID* and the *FooDB_Food_ID*.

The Software

This software has been designed as a Python[12] script originally tested on a computer running Python 3.8.10 inside KDE Neon 5.25, a GNU/Linux distribution based on Ubuntu 20.04 LTS and using Linux Kernel version 5.13.0-51; although, in general it should be compatible with all systems capable of running Python. It works by processing the five databases mentioned above to produce, for each metabolite for which information is requested, a file in GraphML format (an XML-based file format for graphs)[13] containing all the nodes associated with it. This requires the user to provide a CSV with a number of optional metabolite identifiers. Although more detailed instructions can be found in the README of the project itself on Git,[1] the program supports the following data as input identifiers:

- **InChI**: The International Chemical Identifier, an identifier designed by IUPAC and NIST to provide a standard, readable way of encoding molecular information and to facilitate searching for information in databases and on the web.[14] This identifier can be generated autonomously for any small metabolite using openly licensed software, and its known structure and formula. It is highly recommended that this is calculated and provided to the program (in case of hitherto unknown metabolites, this might be the only identifier availaible on the list)
- **InChIKey**: This is a hashed version of the InChI Identifier, i.e., a mathematically condensed version of it shaped as a 27-character string that is simpler, and thus easier to use, than the original.[15] Providing it is opcional.
- **Identifier**: The identifier in the Human Metabolome Database, if available.
- **Name**: A commonly accepted name for the metabolite. It may be more imprecise that the other identifiers and even lead to false positives, so it is recommended that you enter it only if it is standardised according to IUPAC nomenclature.[2]
- **ChEBI**: The ChEBI database identifier, a resource for small chemical entities of biological interest.[16] It is also optional.

Other identifiers, such as SMILES or MonoisotopicMass, were considered as potential for inclusion, but were discarded, the former because it provides less information than InChI, which can unerringly (99.5% accuracy)[14] identify a metabolite, and the latter because it could lead to many false positives (as we theorise that there will be many metabolites with a similar mass).

Once these identifiers are received in the appropriate format, the program tries to find exact matches for Name, Identifier, InChI, InChIKey or ChEBI; and, if it does not find them, it searches for metabolites with a structural similarity of at least 95% with the original metabolite about which information is sought. To calculate this structural similarity, we have used **rdkit**, a Python module focused on chemoinformatics and open-source machine learning that allows us to calculate the structure of a molecule from its InChI (if it is present among the parameters presented to the program).

Once the relevant structure has been calculated, we obtain the MACCS fingerprint of both the "Query" molecule and all the "Subject" molecules present in our five databases using rdkit. These fingerprints, whose acronym stands for "Molecular ACCess System", are 166-bit 2D structure fingerprints that are commonly used to measure molecular similarity. Since each bit is either on or off, MACCS keys can represent more than 9.3×10^{49} different molecules,[17] which should give us enough confidence that they correctly represent the molecules we are working with. Next, for each pair of query and subject molecules, we calculate their Sørensen-Dice similarity index; if the match is greater than 95%, we import both the subject and the query into our knowledge graph.

[1] Github Repository: https://github.com/OMB-IARC/CanGraph
[2] IUPAC Nomenclature Guide: https://iupac.org/what-we-do/nomenclature/

This Sørensen-Dice similarity index is a measure of similarity between two vectors widely used in computer science and machine learning, and takes values from 0 to 1 where 1 is most similar and 0 is least similar.[18] It was developed and published independently by Thorvald Sørensen and Lee Raymond Dice, who published it in 1948 and 1945, respectively.

The original metabolite will be imported and marked as `OriginalMetabolite`, and all those related to it will be marked by the relationship `-[r:ORIGINALLY_IDENTIFIED_AS]->`, with the properties of the relationship explaining the basis for this identification.

Sample Workflow

A potential workflow, as depicted in **Figure 2**, would be as follows: a researcher, in their laboratory, finds a new metabolite in the course of their research. Intrigued about it, they calculate its InChI, and provide it to our program, along with any other identifiers they may have collected from it. After letting the program run, the researcher will obtain a GraphML file, which they can open in Neo4J or any other data management program (e.g. CytoScape) allowing them to investigate the associations between the provided metabolites and those that have been identified as the same or with a high degree of similarity. This exploration is currently left to the researcher, and can be done manually; however, due to the large number of nodes present in the exports, we are considering developing a complementary program for this.

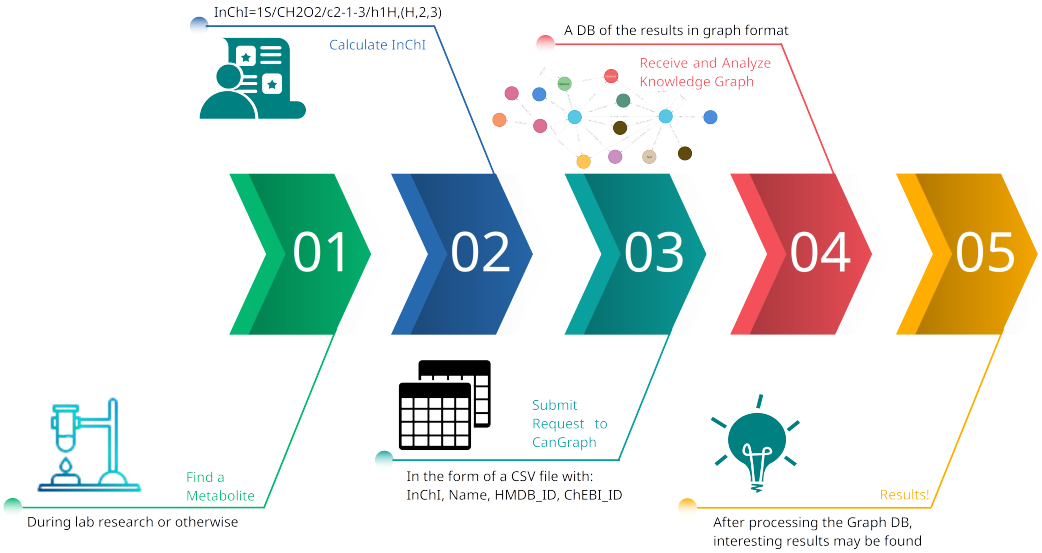


Figure 2: A sample workflow. After finding a metabolite, its InChI is calculated, and it, together with more info, is presented to the program in CSV format. The resulting KG can be analyzed for new insights.

Understanding the Output

To understand and illustrate the kind of results that can be obtained with this program, as well as to demonstrate its correct functioning, we have performed a test run with a series of metabolites discovered by the IARC Nutrition and Metabolism team, to which the author of this paper belongs. Also, to better understand the output of the program, we have created a series of graphs that count the number of times data is retrieved from each individual database, as well as the reasons for choosing to import a particular field (as defined in this same section).

Results

Sample Outputs

For the time being, the solution we have given to the problem explained above (the need to annotate and interpret newly discovered metabolites) takes the form of a Minimal Viable Product Software that does not yet enjoy all the functionalities we would like for it to have. However, from a list of metabolites provided by the IARC Nutrition and Metabolism Team, we have managed to generate a series of Knowledge Graphs, demonstrating the viability of the project and its usefulness. As can be seen in **Figure 3**, these graphs maintain the scheme designed in **Figure 1** and explained in more detail in **Annex I**; we can see that, for a given *OriginalMetabolite*, almost 200,000 related nodes and 250,000 relationships appear, allowing us to navigate the graph and understand the different information sources.

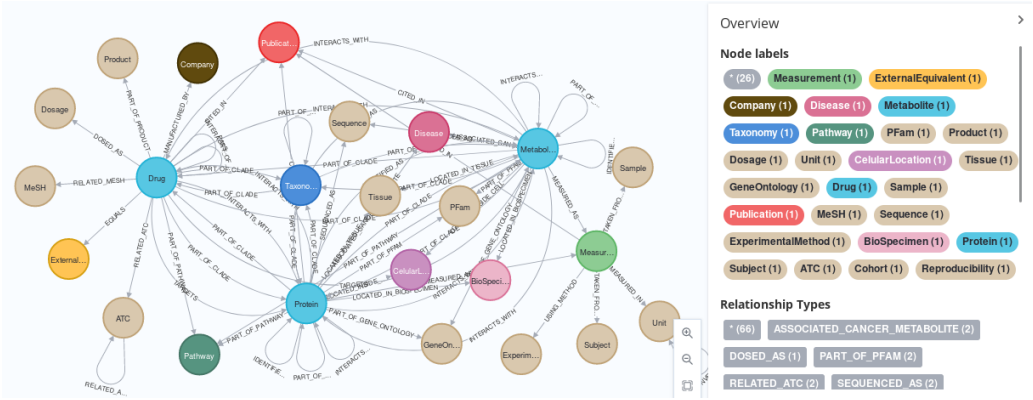


Figure 3: Schema of CanGraph's output. This is coherent with the schema presented in **Figure 1**, showing 26 different types of nodes and 66 types of relations. The numbers may not add up since some node and relation types might be missing or be duplicated.

This provides a great variety of information, such as the publications in which it has been found (**Figure 4a**), the pathways it can take part in, (**Figure 4b**), the diseases it is related with (**Figure 4c**), and, in the case of proteins, their Genomic and Proteic sequences (**Figure 4d**).

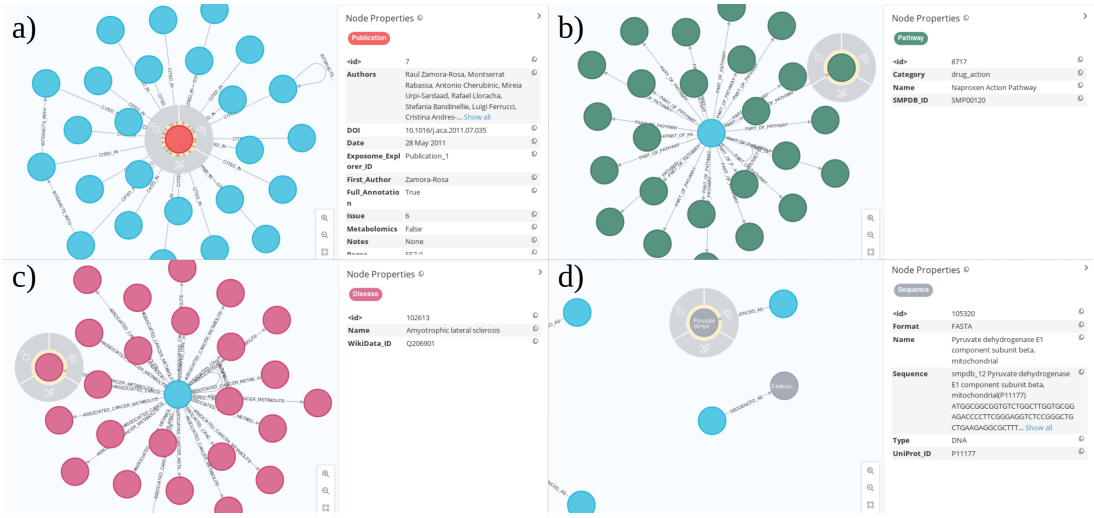


Figure 4: Example of some result nodes. **a)**: Some *Drug*s and *Protein*s, showing the *Publication*s they might be CITED_IN (and their keys) **b)** A *Metabolite* and all of the *Patway*s it takes part in **c)** A *Disease*, its name and a WikiData_ID **d)** Some pairs of *Protein*s and their *Sequence*s, which can either be proteic ("PROT") or genomic ("DNA")

Provenance of the Data

In **Figure 5a**, we can see the provenance of the data present in CanGraph's outputs, categorised according to the method that has been used to consider these data as "valid for import" from one of our five databases. This has been calculated by noting each time the program considered a record present in a database as a suitable match for import into the result, writing down both the reason why it was considered valid and the database from which it was taken.

All five input identifiers have been invoked a similar amount of times, which indicates that, most probably, they complement each other, so that, when one of them is missing, it is still possible to obtain information, as the others could make up for its absence. In this way, a researcher using our program would not have to indicate all the 5 proposed identifiers, but, by indicating just one of them (*InChI*) or even more if they are availaible, it is likely that she would find relevant matches in our 5 databases.

On the other hand, with respect to the distribution of the data, the "structural similarity" criterion clearly stands out for its breadth. This makes sense, as "structural similarity" is, next to "name", the least strict field we have defined; that is, for a given metabolite, there is only one InChI, one ChEBI, one HMDB_ID, and, if well defined, one name; however, a single metabolite can be similar to many others, which explains the disproportionate impact of this feature on the selection of fields for import. Moreover, this is quite positive, as it implies that, for unknown metabolites (for which all other data sources are less likely to be present) there is a higher probability that we will find one with structural similarity in our database. Also worth mentioning is the ChEBI field, which has a much smaller mean and width than the other fields. This is because, in the file provided by the IARC Nutrition and Metabolism Team that was used to test the software, the ChEBI IDs were annotated in a non-standard way (prefixed), and so they could not find matches in all databases. This is a problem that should be tackled in new versions of the software, but it also demonstrates its versatility: despite input problems, the program managed to generate Knowledge Graphs successfully.

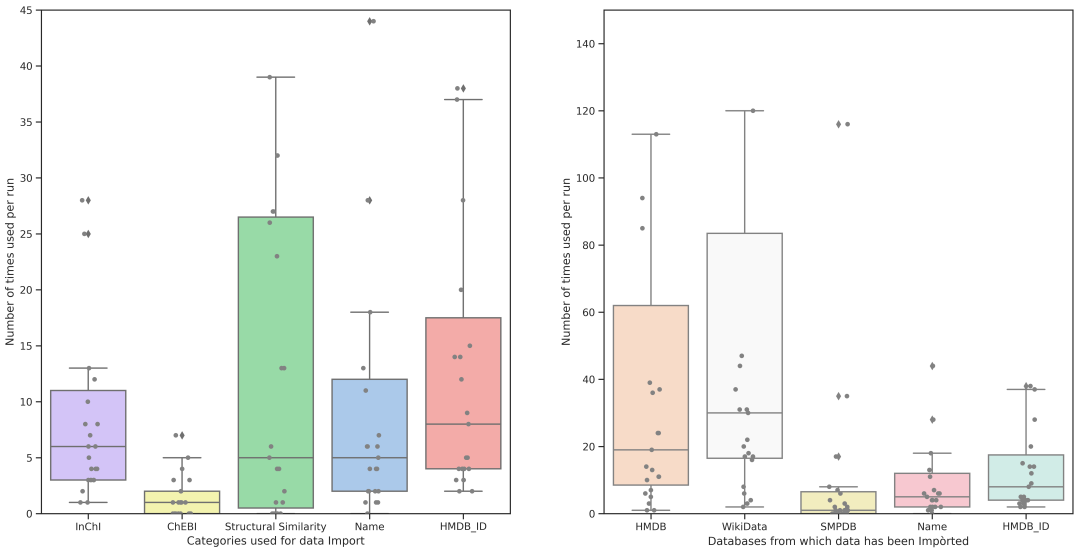


Figure 5: **a):** Provenance of the data in CanGraph's outputs by data source. Since some outliers are present, the y axis has been cut to ommit them if they are over 3 standard deviations. **b)** Procedence by import method. The y axis has also been cut to facilitate data visualization, removing outliers over 3 standard deviations within each column.

Finally, **Figure 5b** shows the origin of the data present in the outputs we have generated, this time categorised according to the database from which they have been extracted. Unlike in the previous figure, here there are two databases that stand out when it comes to providing information: WikiData and the HMDB. This makes sense: on the one hand, as we have explained, WikiData is the more general database, and the scripts we have designed to extract information from it are triggered whenever there are matches by name or other common identifiers such as UniProt ID, CAS Number and others, which are likely to be present. Regarding the HMDB, its disproportionate impact on providing information is also logical, since, in the input provided, almost all metabolites came with a non-null "Identifier" value (which, as explained, is the ID in the HMDB). This can also be seen in **Figure 4a**, where the average usage for the "HMDB_ID" criterion is also the highest (although just slightly) when importing data.

Potential Improvements

One of the main problems with CanGraph is its long running time: for the list of 25 metabolites on which the program was tested, the estimated running time is about 160 hours. While some of this slowness may be due to rdkit, which takes a long time to generate the MACCS fingerprints and the Sørensen-Dice index on which some of the database search criteria are based, most of it is due to the poor transaction times offered by the Neo4J Python driver, which is generally inefficient, and the Neo4J database itself, which is rather unstable, at least on GNU/Linux. Perhaps these problems can be solved by using the IARC High Performance Computing system, which has supercomputers for which available resources or driver instability may be less of a problem. Another problem that has required addressing is the instability of WikiData's SPARQL point (from which we get the information for this database): as this cannot be available 24 hours a day, we have had to design the code to be able to retry those requests that have not been able to pass through on the fist attempt.

Another major detail that remains to be worked out is the post-processing of the generated networks. At present, this work is left to the researchers using the software, although there is the possibility at IARC of generating a program that allows this to be done in a comprehensive and automated way, simplifying the process even further.

Conclusions

As set out on the beggining, and as part of the CanGraph project, we have been able to design a software capable of acting as a "search engine" in existing databases, achieving our goal of annotating and interpreting hitherto unknown metabolites, as well as their potential relationships with certain diseases, in a simple, fast and convenient way. Although there is still work to be done, such as generating a script to automate the post-processing of the generated networks, or fixing minor details in the integration of certain search criteria, this paper has definitively proven the usefulness and success of our project, which we hope will help IARC to continue its indispensable work against cancer and for the benefit of humanity.

In addition, the complex and time-consuming process of manually optimising the five schemas of five very different databases has allowed us to greatly improve our knowledge of the different identifiers in existing databases on the internet, enabling us to generate a schema that is both reliable for the future and adaptable to new databases that we could add to this "search engine" that we are creating.

Looking into the future, we would also like to provide this program as Software as a Service (SaaS) in the cloud, available at least to all IARC scientists and potentially to external researchers who find the service useful. Using pre-existing software such as cytoscape.js, it might be possible

to present the resulting Knowledge Graphs in an attractive, simple and informative way to potential users. We would also like to try running the service on the IARC High Performance Computing system, which could potentially help reduce the two major roadblocks encountered to date: the instability of the DataBase Management System and the slow processing times.

References

1. Sung H, Ferlay J, Siegel RL, Laversanne M, Soerjomataram I, Jemal A, et al. Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: A Cancer Journal for Clinicians [Internet] 2021 [cited 2022 Jun 21];71(3):209–49. Available from: https://onlinelibrary.wiley.com/doi/abs/10.3322/caac.21660

2. Boffetta P, Parkin DM. Cancer in developing countries. CA: A Cancer Journal for Clinicians [Internet] 1994 [cited 2022 Jun 24];44(2):81–90. Available from: https://onlinelibrary.wiley.com/doi/abs/10.3322/canjclin.44.2.81

3. Eckhouse S, Lewison G, Sullivan R. Trends in the global funding and activity of cancer research. Mol Oncol [Internet] 2008 [cited 2022 Jun 21];2(1):20–32. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5527789/

4. Schmidt DR, Patel R, Kirsch DG, Lewis CA, Vander Heiden MG, Locasale JW. Metabolomics in cancer research and emerging applications in clinical oncology. CA: A Cancer Journal for Clinicians [Internet] 2021 [cited 2022 Jun 24];71(4):333–58. Available from: https://onlinelibrary.wiley.com/doi/abs/10.3322/caac.21670

5. Wishart DS, Guo A, Oler E, Wang F, Anjum A, Peters H, et al. HMDB 5.0: The human metabolome database for 2022. Nucleic Acids Research [Internet] 2022 [cited 2022 Jun 23];50(D1):D622–31. Available from: https://doi.org/10.1093/nar/gkab1062

6. Wishart DS, Feunang YD, Guo AC, Lo EJ, Marcu A, Grant JR, et al. DrugBank 5.0: A major update to the DrugBank database for 2018. Nucleic Acids Research [Internet] 2018 [cited 2022 Jun 23];46(D1):D1074–82. Available from: https://doi.org/10.1093/nar/gkx1037

7. Jewison T, Su Y, Disfany FM, Liang Y, Knox C, Maciejewski A, et al. SMPDB 2.0: Big improvements to the small molecule pathway database. Nucleic Acids Res 2014;42(Database issue):D478–484.

8. Neveu V, Nicolas G, Salek RM, Wishart DS, Scalbert A. Exposome-explorer 2.0: An update incorporating candidate dietary biomarkers and dietary associations with cancer risk. Nucleic Acids Research [Internet] 2020 [cited 2022 Jun 23];48(D1):D908–12. Available from: https://doi.org/10.1093/nar/gkz1009

9. Wikidata: A free and open knowledge base [Internet]. [cited 2022 Jun 23];Available from: https://www.wikidata.org/wiki/Wikidata:Main_Page

10. Exposome and exposomics NIOSH CDC [Internet]. 2022 [cited 2022 Jun 21];Available from: https://www.cdc.gov/niosh/topics/exposome/default.html

11. Rodriguez MA, Neubauer P. The graph traversal pattern [Internet]. 2010 [cited 2022 Jun 21];Available from: http://arxiv.org/abs/1004.1001

12. Python reference manual. Python.org [Internet]. [cited 2022 Jun 23];Available from: https://www.python.org/doc/

13. GraphML specification [Internet]. [cited 2022 Jun 23];Available from: http://graphml.graphdrawing.org/specification.html

14. Heller SR, McNaught A, Pletnev I, Stein S, Tchekhovskoi D. InChI, the IUPAC international chemical identifier. Journal of Cheminformatics [Internet] 2015 [cited 2022 Jun 24];7(1):23. Available from: https://doi.org/10.1186/s13321-015-0068-4

15. Karol PJ. The InChI code. J Chem Educ [Internet] 2018 [cited 2022 Jun 24];95(6):911–2. Available from: https://doi.org/10.1021/acs.jchemed.8b00090

16. Hastings J, Owen G, Dekker A, Ennis M, Kale N, Muthukrishnan V, et al. ChEBI in 2016: Improved services and an expanding collection of metabolites. Nucleic Acids Res [Internet] 2016 [cited 2022 Jun 24];44(D1):D1214–9. Available from: https://europepmc.org/articles/PMC4702775

17. Kuwahara H, Gao X. Analysis of the effects of related fingerprints on molecular similarity using an eigenvalue entropy approach. Journal of Cheminformatics [Internet] 2021 [cited 2022 Jun 21];13(1):27. Available from: https://doi.org/10.1186/s13321-021-00506-2

18. Dice LR. Measures of the amount of ecologic association between species. Ecology [Internet] 1945 [cited 2022 Jun 24];26(3):297–302. Available from: https://onlinelibrary.wiley.com/doi/abs/10.2307/1932409

Annex I - Schema Definition

From any software, a standardised, unified and reliable output is expected, so that it can be worked with in a simple and comfortable way. For this reason, and as explained in the rest of the paper, we have considered it necessary to define a common schema for the search results offered by this program, so that, when we extract information from each of the five different databases, we obtain a result with the minimum number of duplications and complexities possible, but which maintains as much of the information and usefulness present in each data field as possible.

The Relationships

There are 35 different kind of nodes, which can be seen with their associated nodes here:

Origin Node	Relation type	Destination Node
Drug	ADMINISTERED_VIA	AdministrationRoute
Disease	ASSOCIATED_WITH_CANCER_GENE	Gene
Disease	ASSOCIATED_CANCER_MEASUREMENT	Measurement
Disease	ASSOCIATED_CANCER_METABOLITE	Metabolite
Measurement	ASSOCIATED_WITH_MEASUREMENT	Measurement
Disease	CAUSED_BY	Cause
Drug, Protein, Metabolite, Subject or Measurement	CITED_IN	Publication
Unit	CONVERTED_INTO	Unit
Drug	DOSED_AS	Dosage
Gene	ENCODES	Drug or Metabolite
Drug, Protein, Metabolite, Disease or Gene	EQUALS	ExternalEquivalent
Gene	EXPRESSED_IN	Tissue
Drug, Metabolite, Measurement, Sample	FOUND_IN	BioSpecimen
Metabolite	IDENTIFIED_AS	Metabolite
Disease	INSTANCE_OF	Disease
Drug	INTERACTS_WITH	Metabolite or Protein
Drug, Protein, Metabolite	LOCATED_INSIDE_CELL	CelularLocation
Drug, Metabolite	LOCATED_IN_TISSUE	Tissue
Drug	MANUFACTURED_BY	Company
Metabolite	MEASURED_AS	Measurement
Measurement	MEASURED_IN	Unit
Protein, Metabolite or Taxonomy	PART_OF_CLADE	Taxonomy
Subject	PART_OF_COHORT	Cohort
Protein	PART_OF_GENE_ONTOLOGY	GeneOntology
Drug, Protein or Metabolite	PART_OF_PATHWAY	Pathway
Protein	PART_OF_PFAM	Pfam
Drug or Protein	PART_OF_PRODUCT	Product
Protein, ATC	RELATED_ATC	ATC
Drug, Protein or Metabolite	RELATED_MESH	MeSH
Measurement	REPODUCIBILE_WITH_CONDITIONS	Reproducibility
Protein	SEQUENCED_AS	Sequence
Disease	SUBCLASS_OF	Disease
Measurement	TAKEN_FROM_SAMPLE	Sample
Measurement or Sample	TAKEN_FROM_SUBJECT	Subject
Disease	TARGETED_BY	Drug
Drug or Protein	TARGETS	Drug or Protein
Measurement	USING_METHOD	ExperimentalMethod

Figure 6:

The Nodes

There are 29 different kind of nodes, each with their own properties. These are:

Node Type	Description	Primary Key	PK Description	Keys
ATC	The *Anatomical Therapeutic Chemical Classification System Code* for a given **Protein** node	Code	A given ATC Code	Name
AdministrationRoute	The different routes (oral, topical...) a given **Drug** may be administered via	WikiData_ID	The node's identifier in the WikiData Database	Name, URL
BioSpecimen	A labpratory specimen (blood, urine...) a **Sample** and/or **Measurement** can be taken from	Name	A text chain that shortly defines the node	Specimen_Type, URL
Cause	The cause of a given **Disease**	WikiData_ID	The node's identifier in the WikiData Database	Name, URL
CelularLocation	The cellular compartment where a given **Drug**, **Metabolite** or **Protein** may be found	Name	A text chain that shortly defines the node.	-
Cohort	A group of **Subject** nodes that share some common characteristics	Name	A text chain that shortly defines the node.	Abbreviation, Citation, Country, Description, Exposome_Explorer_ID, PubMed_ID, Study_Design_Type, URL
Company	The company that manufactures, produces or packages some **Drug**	Name, URL	The company's name and URL of its website	Generic, Manufacturer, Packager
Disease	An illness that can afflict humans	Name	A text chain that shortly defines the node.	Alternative_Names, Anatomical_Location, Disease_Ontology_ID, Exposome_Explorer_ID, HPO_ID, ICD_11, ICD_10, KEGG_ID, OMIM_ID, URL, WikiData_ID
Dosage	A dosage a given **Drug** is approved to be administered as. **NOTE**: Due to problems with the DBs this nodes come from, the PK has been stablished as "Route", which is not the most coherent, but was the only key present in all nodes.	Route	The route a given dose is administered as	Strength, Form
Drug	A medicament that might be used to treat a given **Disease**, due to it targeting some **Protein** or **Metabolite** **NOTE**: Since all nodes in this category come originally from DrugBank, their DrugBank_Ids can be used as PK	DrugBank_ID	The node's identifier in the DrugBank Database.	Average_Mass, CAS_Number, ChEBI_ID, CHEMBL_ID, ChemSpider_ID, , Formula, InChI, InChIKey, KEGG_ID, LiverTox, Name, Pregnancy_Category, PubChem_ID, SMILES, URL, WikiData_ID
ExperimentalMethod	The method used during a given study. **NOTE**: Since all nodes in this category come originally from Exposome Explorer, their Exposome_Explorer_Ids can be used as PK	Exposome_Explorer_ID	The node's identifier in the Exposome Explorer DB	Alternative_Names, Method_Type, Name
ExternalEquivalent	An exact external equivalent for an item in our database	URL	The URL of the resource	Resource_Name, WikiData_ID
Gene	A gene that may codify a given **Protein** or **Metabolite**	WikiData_ID	The node's identifier in the WikiData Database	Chromosome_Location, Cytogenetic_Location, Ensembl_Gene_ID, Entrez_ID, HGNC_ID, HomoloGene_ID, Name, OMIM_ID, URL, Strand_Orientation, Genomic_End, Genomic_Start
GeneOntology	A controlled vocabulary describing a **Gene** and the attributes of its gene products **NOTE**: Since the only category present in all nodes is "**Description**", this was set as PK	Description	A text chain that shortly defines the node.	Category, GO_ID
Measurement	A measurement taken of a **Metabolite** from a given **Sample**	Value	The value of a given measurement	Adjusted_On, Adjustment_Type, Ancestry, Comments, Concentration_Detected_Min, Concentration_GeoMean, Concentration_GeoSD, Concentration_InterQuartile_Range, Concentration_Max, Concentration_Mean, Concentration_Median, Concentration_Min, Concentration_Percentile_05, Concentration_Percentile_10, Concentration_Percentile_25, Concentration_Percentile_75, Concentration_Percentile_90, Concentration_Percentile_95, Concentration_SD, Confidence_Interval_95_Geo_Lower, Confidence_Interval_95_Geo_Upper, Confidence_Interval_95_Lower, Confidence_Interval_95_Upper, Detected_Only, Detected_Proportion, Detected_Size, Exposome_Explorer_ID, Food_Items, Normal, Regressed_On, Size, Supplement_Inclusion
MeSH	The *Medical Subject Heading* of a given item	Category	A text description of the MeSH category	MeSH_ID
Pathway	A linked series of chemical reactions occurring within a cell	SMPDB_ID	The node's ID in the Small Molecule Pathway DB	Category, PW_ID, Description, Kegg_ID, Name
PFam	A database of **Protein Fam**ilies that includes their annotations and multiple sequence alignments generated using hidden Markov models	PFAM_ID	The node's ID in the PFAM database	Name
Node Type	**Description**	**Primary Key**	**PK Description**	**Keys**

Node Type	Description	Primary Key	PK Description	Keys
Product	Something produced by a **Company**; usually, a drug or a medicine	Name	A text chain that shortly defines the node.	Approved, Country, DPD_ID, Dosage_Form, EMA_MA_Number, EMA_Product_Code, Ended_Marketing_On, FDA_Application_Number, Generic, Ingredient, Labeller, NDC_ID, NDC_Product_Code, Over_the_Counter, Route, Source, Started_Marketing_On, Strength, URL, WikiData_ID
Subject	A subject form which a **Measurement** has been taken	Exposome_Explorer_ID	The node's identifier in the Exposome Explorer DB	Age_Max, Age_Mean, Age_Median, Age_Min, Age_SD, Ancestry, BMI_Max, BMI_Mean, BMI_Median, BMI_Min, BMI_SD, Country, Description, Ethnicity, Female_Proportion, Weight_Min, Weight_SD Gender, Health_Condition, Height_Max, Height_Mean, Height_Median, Height_Min, Height_SD, Name, Nb_of_Cases, Nb_of_Controls, Size, Smoker_Proportion, Supplement_Exclusion, Weight_Max, Weight_Mean, Weight_Median,
Metabolite	An intermediate or end product of the human metabolism; usually, small molecules	ChEBI_ID, CAS_Number	The metabolite's identifier in the ChEBI and CAS databases	Acceptor_Count, Alternative_Names, Antibiotic, Average_Mass, Average_Molecular_Weight, Bacterial_Source, Bigg_ID, Bioavailability, Boiling_Point, CAS_Number, ChEBI_ID, ChEMBL_ID, ChemSpider_ID, Component_ID, Description, Donor_Count, DrugBank_ID, Exposome_Explorer_ID, FooDB_Compound_ID, FooDB_Food_ID, Formal_Charge, Formula, Function, Ghose_Filter, HMDB_ID, IUPAC, Identification_Method, InChI, InChIKey, KEGG_ID, KNApSAcK_ID, Level, MDDR_Like_Rule, METLIN_ID, Melting_Point, Metabolite_ID, Microbial_Metabolite, Monisotopic_Molecular_Weight, Name, Number_of_Rings, Organism, PDB_ID, Phenol_Explorer_Compound_ID, Physiological_Charge, Polar_Surface_Area, Polarizability, PubChem_ID, Publication_ID, Refractivity, Ro5, Rotatable_Bond_Count, SMILES, Secondary_HMDB_IDs, State, Status, Substrate, Synonyms, Transporter_DB_ID, URL, UniProt_ID, VMH_ID, Verber_Rule, Water_Solubility, WikiData_ID, WikiPedia_Article, logP, logS, Pka_Strongest_Acidic, pKa_Strongest_Basic
Sample	A sample from which a **Measurement** can be taken	Exposome_Explorer_ID	The node's identifier in the Exposome Explorer DB	Intake_Food_Coverage, Intake_Time_Coverage, Intervention_Dose, Repetitions, Ancestry, Time_Definition, Time, Intake_Tool
Publication	An academic reference for a given fact	Title	The title of the publication	Authors, DOI, Date, Exposome_Explorer_ID, First_Author, Full_Annotation, Intake_Count, Issue, Metabolomics, Notes, Pages, PubMed_ID, Public, Publication, Study_Design_Type, Title, Volume
Sequence	A sequence for a given **Protein**	Sequence	A genomic or proteic sequence	UniProt_ID, Type, Format, Locus, Chromosome_Location, Name
Taxonomy	A Taxonomic clade	Name	A text chain that shortly defines the node.	Type
Tissue	A diferenciated biologic material constitued by a group of cells	Name	A text chain that shortly defines the node.	URL, WikiData_ID
Protein	A macromolecule formed by a chain of aminoacids. It can also be labeled as **Metabolite**	UniProt_ID	The node's ID in the UniProt database	Absorption, Actions, Affected_Organisms, Alternative_DrugBank_Ids, Average_Mass, Average_Molecular_Weight, BindingDB, CAS_Number, ChEMBL_ID, ChemSpider_ID, Clearance, Description, DrugBank_ID, FDA_Label, Formula, Function, GenAtlas_ID, GenBank_Gene_ID, Genbank_Protein_ID, GeneCards_ID, Gene_Name, Groups, Guide_to_Pharmacology_ID, HGNC_ID, HMDB_ID, Half_Life, Hydrophobicity, IUPHAR_ID, Indication, Isoelectric_Point, KEGG_ID, Known_Action, LabelName, Locus, Mechanism_of_Action, Melting_Point, Metabolism, Name, Organism, PDB_ID, PharmGKB, Pharmacodynamics, Protein_Binding_Info, PubChem_ID, Residue_Number, Route_of_Elimination, RxCUI, SMILES, Safety_Data_Sheet, Secondary_HMDB_IDs, Source, Specific_Function, Stat, Synonyms, Taget_Name, Target_ID, Target_Organism, Target_Position, Theoretical_PI, Therapeutic_Targets_Database, Toxicity, Transmembrane_Regions, UNII, URL, Volume_of_Distribution, Water_Solubility, WikiPedia_Article, logP, pKa
Reproducibility	The conditions for reproducing a given measurement and some information to check for if the **Measurement** is, in fact, reproduced	Exposome_Explorer_ID	The node's identifier in the Exposome Explorer DB	Variance_Within, ICC_Confidence_Interval_95_Lower, Size, ICC_Confidence_Interval_95_Upper, CV_Within, ICC, CV_Between
Unit	A given unit of measurement	Name	A text chain that shortly defines the node.	Group, Converted_to_ID, Type, Name, Exposome_Explorer_ID
Node Type	**Description**	**Primary Key**	**PK Description**	**Keys**

For the sake of simplicity, it has been decided against presenting here a detailed description of each feature: for example, some features are Boolean, some are Integer, and some are text strings; however, such a detailed analysis seemed too much for this article.

Additional Comments

Please note that this schema can be interactively consulted in Neo4J by importing the corresponding GraphML file from the Git Repository. The command to use, after presenting `new-schema.graphml` in Neo4J's import path (availaible in your instance's config) is:

```
CALL apoc.import.graphml("new-schema.graphml", {useTypes:true, storeNodeIds:false,
readLabels:True})
```

This file can also be explored using other GraphML-compatible software, such as cytoscape.

Annex II - License Information

This PDF was generated using pandoc:

```
pandoc  --pdf-engine=xelatex --highlight-style tango --biblio Bibliography.bib
        --toc "TFM Body.md" -o "TFM Body.pdf"
```