FATHI Victoria

Heterogeneous data Integration

Unstructured data management - Post-genomics Applications

Introduction

Post-Genomics analysis goal is to extract information from heterogeneous data. Such analyses are challenging because they request integration of different types of data (description, experimental measure, litterature, ...) from a plurality of sources.

Here, we focused on the integration of E. coli K-12 MG1665 genome and associated data of Uniprot, Ecocyc and String databases. Then, we examined four different metrics of enrichment on a benchmark dataset to better understand how and when to use them. Eventually, we analyzed the unknown data set '20' as part of an example of post-Genomics analysis.

1 - Database Creation

a) Accessibility

This project is a collaboration with Lou Duron.

Data, scripts and documentation are accessible in gitlab here

This repository is divided into three branches. The main branch contains:

- necessary files and scripts for database creation
- general documentation
- query sets and benchmark sets
- original script from our supervisor Roland Barriot

The 'vic' and 'lou' branches correspond respectively to Victoria FATHI and Lou DURON analysis.

b) Environment

Integration of diverse multi omics data is currently a big challenge for bioinformatics analysis. Graph Oriented Databases have proven to be an effective approach to represent complex biological relationships (Thapa & Ali, 2021). Here, we used Neo4J, a graph oriented database to integrate E. coli K-12 MG1665 genome annotation.

We worked with **Neo4j Desktop 1.4.8** and the database management system **Graph DBMS 4.3.1**. We navigated through the database with **Neo4j Browser**.

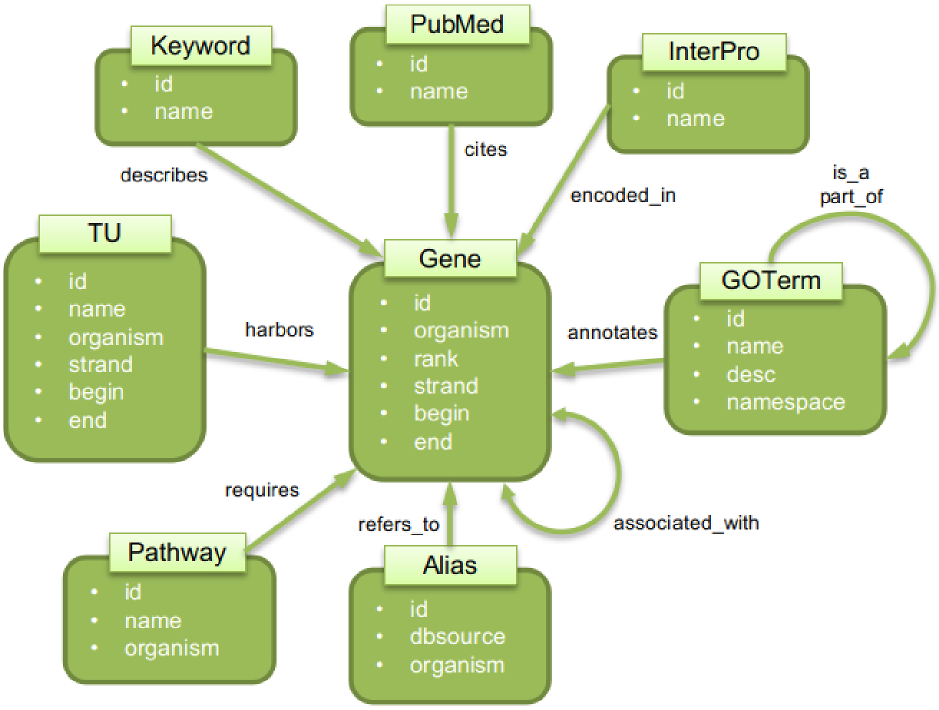
Annotation files were parsed and analyzed in **R 4.0.5 (and Rstudio 1.4.171)** using the **Tidyverse 1.3.1** package. Enrichment analysis algorithm requires **python 3.X** and the following libraries.

- **numpy 1.19.5**
- **scipy 1.5.3**
- **py2neo 2021.2.3**
- **pandas 1.1.5**

Note that we used conda 4.10.1 during this project. An .yml file is available on the gitlab project to recreate the environment.

c) Database structure:

The following scheme presents the structure of **E. coli K-12 MG1665** genome annotation:



d) raw data

Uniprot:

E.coli K-12 MG1665 proteome was retrieved from Uniprot database by selecting the following columns.

- Entry name
- Gene names (ordered locus)
- Gene Ontology IDs
- Interpro
- Keywords

This file is available in: raw_data/uniprot-proteome_UP000000625.tab.gz

EcoCyc:

Metabolic pathways and Transcription Units (TU) annotations are from Ecocyc Database and were obtained from this link: https://www.ecocyc.org/group?id=biocyc17-55140-3842483872.

Associated files are in *raw_data/*

- All_instances_of_Genes_in_Escherichia_coli_K-12_substr._MG1655.txt : Genes (for id mapping purposes)
- All_instances_of_Pathways_in_Escherichia_coli_K-12_substr._MG1655.txt: Pathway
- all_transcription-units.txt: Transcription Units

PubMed:

PubMed references to each gene were provided directly from Roland Barriot (supervisor).

e) Parsing raw data files

From raw data, we generated two types of files:

nodes files: describes nodes id and properties

relationships files: describes nodes relationships and properties

The R script is available on gitlab: *get_data.rmd.* **To reproduce data parsing you will need to change the working directory: setwd('/<path to Part2>/Part_2/')**

Note that output files are stored in the *import* directory. This directory can be directly copied in the *import* directory of Neo4j for database creation.

The necessary queries for the database creation are stored in *data_integration_procedure.html.* You can also create the database from the BDD.dump file.

f) Descriptives Statistics

Annotation quality and completeness are essential for enrichment analysis but these conditions are rarely met. Annotations Database can be imprecised even incorrect and biased towards more studied gene sets. They also suffer from terminology problems which are challenging for data integration. That is why the choice of target sets is important for enrichment analysis.

In the database, GOTerms nodes (fig.1) and Is_a relationship (fig.1) are overrepresented. This could be explained by the annotation of upstream GOTerms in the hierarchy (less specific terms). However, it is more likely that certain sets of genes are more described than others since on average a gene is associated with 5.44 GOTerms.

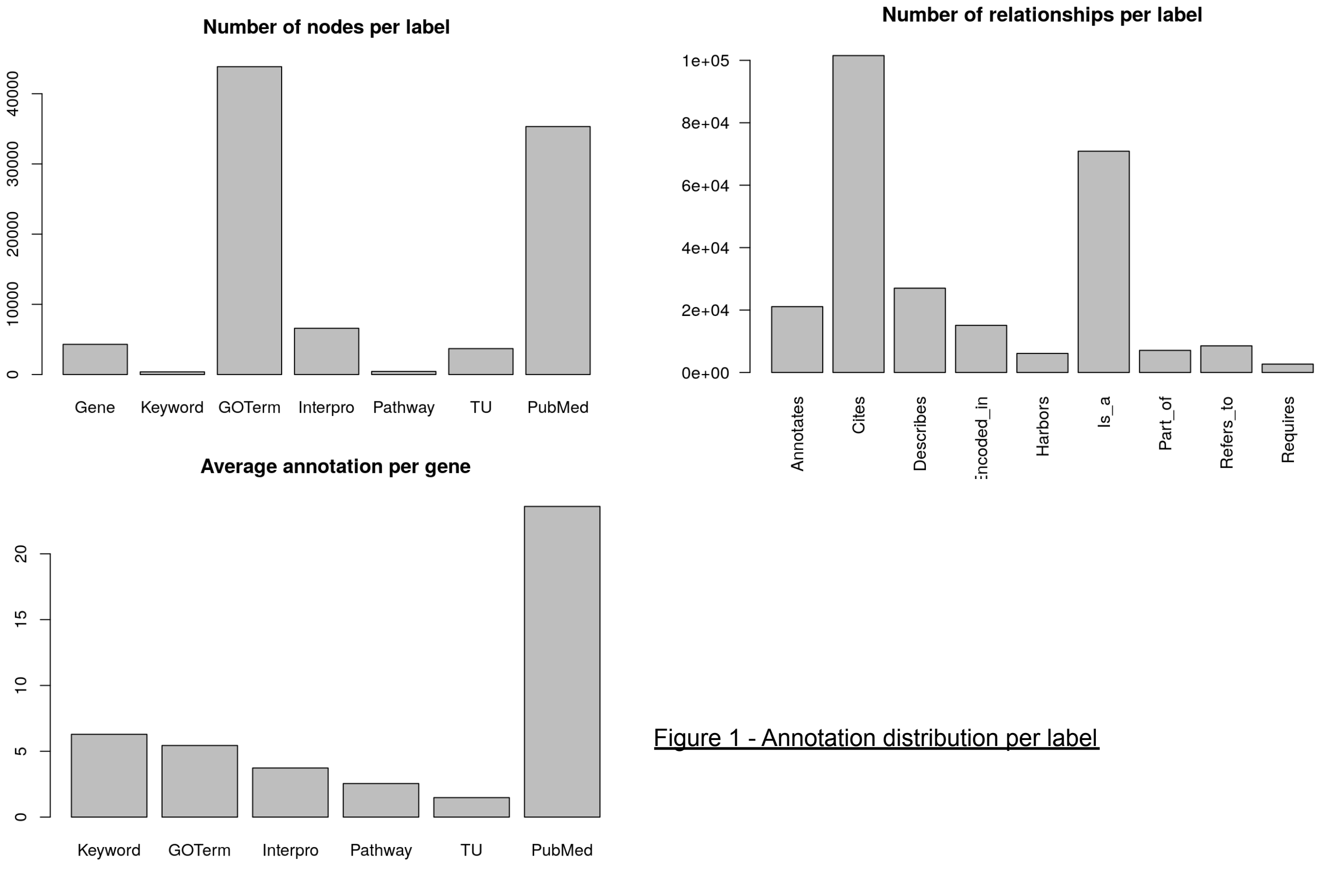


Figure 1 - Annotation distribution per label

PubMed also shows a high level of annotation with more than 35000 total references and 23 on average per gene (fig.1). Such level of annotations is already exploited in enrichment analysis with text mining approaches (Frijters et al., 2008). However, without tools, PubMed enrichment analysis is time consuming and therefore can not be the primary approach.

Interestingly, even though Uniprot Keywords are less diverse than GOTerms, they are on average more uniprot keywords than GOTerms per gene (fig. 1). In addition, Uniprot Keywords are not structured as an ontology so this annotation does not suffer from general annotation like GOTerms.

Furthermore, Pathway, TU and Interpro Domains present fewer nodes and relationships in comparison with the precedent target sets. In consequence, these annotations can not be used alone for enrichment analysis.

2 - Enrichment analysis

a) Implemented functionalities

From the original script, new measures of enrichment were implemented:

- Coverage : based on the number of shared elements
- Pearson's chi-squared test
- Hypergeometric
- Binomial

Two arguments and allowable values checks for the arguments were added:

- -p, --password: password of neo4j database
- -w --write path and names for results (tsv file)
- Target set (-t) and metrics type (-m) are checked to be an allowable value.

To use enrichment.py:

```
enrichment.py -p [PASSWORD] -q [QUERY] -t [NODE_TYPE] -m [METRICS] [OPTION]
-p, --password [required] password neo4j database
-q, --query [required] path to query set
-t --target_type [required] Target sets node type [GOTerm, Interpro, Keyword,Pathway,
PubMed, TU"]
-m --measure [required] Dissimilarity index: binomial (default), hypergeometric, chi2 or
coverage
-s --species taxon id (default=511145)
-a --alpha [optional] Significance threshold (default = 0.05)
-c, --adjust [optional] Adjust for multiple testing (FDR)
-l --limit [optional] Maximum number of results to report.
-w, --write [optional] path and name for results (tsv file)
-r --revigo [optional] only if --write, write id and p value for revigo
-v, --verbose [optional] print intermediary results/queries
```

Example:

```
./enrichment -p pwd -q sets/set01.txt -t Keyword -m chi2
```

b) Comparative analysis of metrics and target sets

In order to compare the four metrics of enrichment (Pearson's chi-squared test, Hypergeometric, Binomial and coverage) we created benchmark datasets. The aim is to create dataset with a known enrichment and to compare enrichment results from the four metrics to the ground truth.

To obtain such datasets we retrieved helicase genes of E.Coli K-12 from Uniprot database. In addition, only genes manually reviewed were conserved. Here is the associated query:

helicase AND reviewed:yes AND organism:"Escherichia coli (strain K12) [83333]

The associated file is in benchmark_data/uniprot-helicaseorganism__EscherichiacolistrainK12ECOLI--.tab

Dataset were parsed with the same method used for data integration (see get_benchmark_data.r). An additional analysis (not presented here) of the benchmark data is also available in gitlab *benchmark_analysis.Rmd.*

From the original helicases gene set we created two sub datasets to assess the impact of set size. The first subset is a small set of 20 genes (*benchmark_data/small_set.txt)* and the other has 350 genes *(benchmark_data/big_set.txt)* both composed of a fraction of helicase genes and randomly picked genes (without redundancy) in the database.

Then, enrichment analysis was performed using the python script *enrichment.py.* We restrict the analysis to GOTerms and Keywords target sets to not induce a bias towards incomplete target sets (TU, Pathway, etc.). The results are available in *bencharmark_results/ and were obtained with the following command:*

```
./enrichment.py

             -p pwd

             -q benchmark_data/<big or small>_set.txt

             -t GOTerm or Keyword

             -m metric

             -c

             -w benchmark_results/<big or small>_<GOTerm or Keyword>_<metric>.tsv
```

First, the two sub-datasets presented the same top results for hypergeometric and binomial. This is because hypergeometric approximates binomial when the population size is significatively larger than the query size.

In addition, coverage metric applied in the small dataset didn't point out GOTerms and Keywords associated with helicases. Only nonspecific GOTerms (biological regulation for example) were returned because a small dataset will more likely share a lot of common elements with the target set. Moreover, interestingly χ^2 returned the same top results as binomial and hypergeometric distribution (not in the same order though). Another consideration is the threshold choice to select interesting enrichment, χ^2 p values are always lower than binomial or hypergeometric values.

Regarding the big dataset, all the metrics failed to return GOTerms and Keywords associated with helicases. An explanation could be that selecting too many genes from an experience results in (differential expression experiment for example) hidden subgroups.

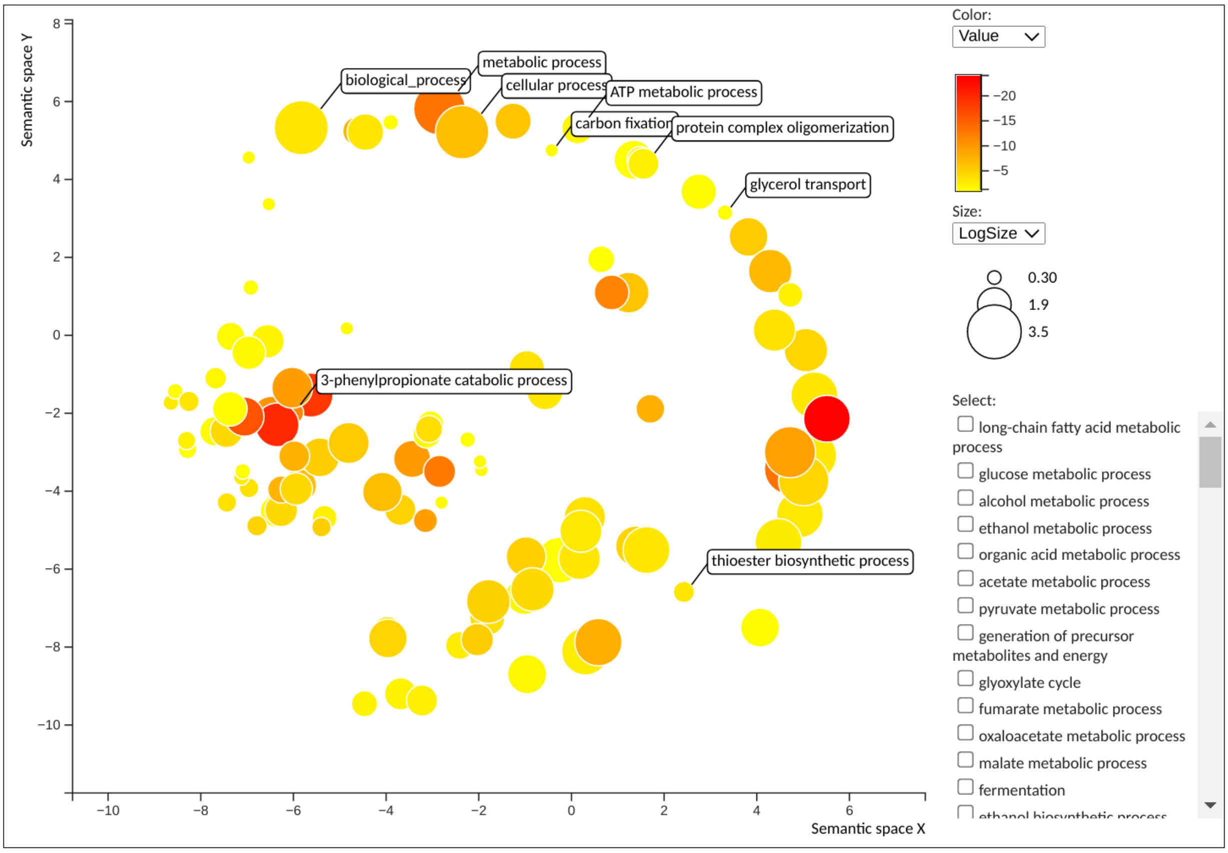
To go further, it would be necessary to use standardized benchmarking methods as suggested in Geistlinger et al. (2020).

3 - Analysis of set 21

To do the enrichment analysis, we used binomial for each target sets available in the database

```
./enrichment.py -p a -q set.M2.20.txt -t GOTerm -m binomial -w
results_set20/<target set>_binomial.tsv -c
```

We then used Revigo to visualize GOTerm enrichment.



Color: Value

Size: LogSize

Select:

☐ long-chain fatty acid metabolic process
☐ glucose metabolic process
☐ alcohol metabolic process
☐ ethanol metabolic process
☐ organic acid metabolic process
☐ acetate metabolic process
☐ pyruvate metabolic process
☐ generation of precursor metabolites and energy
☐ glyoxylate cycle
☐ fumarate metabolic process
☐ oxaloacetate metabolic process
☐ malate metabolic process
☐ fermentation
☐ ethanol biosynthetic process

From the REVIGO visualization, it seems (from the more specific GOTerms) that the 20th set is enriched in genes involved in aromatic compound synthesis. GOTerms enrichment is supported by additional enrichment in ‘Aromatic hydrocarbons catabolism’ and ‘Pyrimidine biosynthesis’ in the Keywords annotation.

Feedback

Roland Barriot’s projects are always the most difficult ones among all the projects that we have. However, they are also the ones where we learn the most and this project is no exception. We worked with R and python, learning commonly used libraries (tidyverse, pandas, scipy). We also decided to improve our git knowledge by using git branches for the first time.

The two remarks I have in mind are: we could have used more time to go deeper into the topic and we learn better in groups.

References

Frijters, R., Heupers, B., van Beek, P., Bouwhuis, M., van Schaik, R., de Vlieg, J., Polman, J., & Alkema, W. (2008). CoPub : a literature-based keyword enrichment tool for microarray data analysis. Nucleic Acids Research, 36(Web Server), W406‑W410. https://doi.org/10.1093/nar/gkn215

Thapa, I., & Ali, H. (2021). A Multiomics Graph Database System for Biological Data Integration and Cancer Informatics. Journal of Computational Biology, 28(2), 209‑219. https://doi.org/10.1089/cmb.2020.0231

Geistlinger, L., Csaba, G., Santarelli, M., Ramos, M., Schiffer, L., Turaga, N., Law, C., Davis, S., Carey, V., Morgan, M., Zimmer, R., & Waldron, L. (2020). Toward a gold standard for benchmarking gene set enrichment analysis. Briefings in Bioinformatics, 22(1), 545‑556. https://doi.org/10.1093/bib/bbz158