Bioinformatics 2019-2020

Date: 15 January 2020

Project report

|  |
| --- |
| Bioinformatics@Data Science A.Y. 2019-2020  Malignant Mesothelioma: a study on the interactome  Giulia Cassarà1, Ivan Colantoni1  1Group no. 13 Abstract In this project we will analyze the interactome of the Malignant Mesothelioma disease in human organism. |

The disease

**Malignant mesothelioma** is a cancer of the thin tissue (mesothelium) that lines the lung, chest wall, and abdomen. The major risk factor for mesothelioma is asbestos exposure.

In the section below we will see the genes involved in the disease (the so called seed genes) and the interactions between seed genes and non seed genes in the human organism. We collected the interaction data from two different PPI (Protein-Protein Interaction) sources and integrated together to build the seed genes interactome, the union interactome and the intersection interactome. Furthemore, we analyzed the network graph obtained by different interactomes using a Python library, NetworkX. We applied clustering methods for disease modules discovery. Finally, we found putative disease genes using DIAMOnD tool.

Seed genes

To get the list of the seed genes involved in the disease, we explored the **DisGeNet** website which has a search engine that helps users to find gene-disease associations (GDAs). The gene-disease associations in **DisGeNET** is organized according to the types of source databases: for example, we get our GDAs from the CURATED dataset, which contains GDAs from UniProt, PsyGeNET, Orphanet, the CGI, CTD (human data), ClinGen, and the Genomics England PanelApp. From the browser we specified our disease of interest (Malignant Mesothelioma) and downloaded the dataset as tab separated text file.

Then, all of our data analysis is perfomed using a Python Library, Pandas, all configured under the Jupyter Lab framework. In Table 1 we show an example of the tab separated text file obtained from the procedure explained above. Of course, for the demonstration purpose we omitted some informations.

**Table 1. List of seed genes for the Malignant Mesothelioma disease obtained from the CURATED dataset of the DisGeNet database.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Gene | Gene\_id | UniProt | Gene\_Full\_Name | Protein\_Class |
| NAT2 | 10 | P11245 | N-acetyltransferase 2 | transferase |
| PARP1 | 142 | P09874 | poly(ADP-ribose) polymerase 1 | NaN |
| ANXA2 | 302 | P07355 | Annexin A2 | NaN |
| APOA1 | 335 | P02647 | Apolipoprotein A1 | NaN |

For all genes in the seed gene list, we have checked if the symbols were updated and approved on the HGNC website. All of the gene symbols were approved.

Then, we have collected the following information from Uniprot:

1. Official gene symbol.
2. Uniprot AC (a.k.a. ‘Uniprot entry’).
3. Protein name.
4. Entrez Gene ID (a.k.a ‘GeneID),
5. and a very brief description of its function

Every information was collected from Uniprot’s section Reviewed (Swiss-Prot). Anyway, for the last point, we had to clean the string of function’s description and truncate the string to keep it shorter and more readable.

In Table 2 we show an example of the tab separated csv file obtained from the procedure written above. Of course, for the demonstration purpose we omitted some informations.

We noticed that every gene has a function description in Uniprot Swiss Reviewed, except for the protein transducin beta like 1 X-linked receptor 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Symbol** | **Name** | **GeneID** | **UniprotAC** | **Function** |
| NAT2 | N-acetyltransferase 2 | 10 | P11245 | Participates in the detoxification of a plethora of hydrazine and arylamine drug |
| PARP1 | poly(ADP-ribose) polymerase 1 | 142 | Q6N069 | Auxillary subunit of the N-terminal acetyltransferase A (NatA) complex which displays alpha (N-terminal) acetyltransferase activit |
| ANXA2 | annexin A2 | 302 | Q9H2H9 | Functions as a sodium-dependent amino acid transporte |
| APOA1 | apolipoprotein A1 | 335 | P09874 | Poly-ADP-ribosyltransferase that mediates poly-ADP-ribosylation of proteins and plays a key role in DNA repair |

Summary on interaction data

For each seed gene, we collected all binary protein interactions from two different PPI sources:

* Biogrid Human, latest release available
* IID Integrated Interactions Database (experimental data only)

Table 3 Summarize the main results reporting:

no. of seed genes found in each different Dbs; total no. of interacting proteins, including seed genes, for each DB; total no. of interactions found in each DB.

|  |  |  |  |
| --- | --- | --- | --- |
| Source Database | No. of seed genes | No. of interacting proteins | No. of interactions |
| Biogrid | 104 of 109 | 17899 | 485380 |
| IID | 104 of 109 | 17278 | 270230 |

We have retrieved some informations from Uniprot about missing genes. These are the genes missing in Biogrid: **CCL27**, which has full name C-C motif chemokine 27 and is a Chemotactic factor that attracts skin-associated memory T-lymphocytes; **MIR125A, MIR126, MIR484, PWAR6** haven’t any Uniprot corrispondence.

These are the genes missing from IID: **GPR27** which is an Orphan receptor and possible candidate for amine-like G-protein coupled receptor. **MIR125A, MIR126, MIR484, PWAR6** haven’t any Uniprot corrispondence.

# Interactomes data

We have Build and stored three tables from all Dbs:

1. seed genes interactome: interactions that involve seed genes only
2. union interactome: all proteins interacting with at least one seed gene.
3. intersection interactome: all proteins interacting with at least one seed gene confirmed by both DBs

In the format:

*interactor A gene symbol, interactor B gene symbol, interactor A Uniprot AC, interactor B*

*Uniprot AC, database source*

**Seed genes interactome**

We iteratively build a Pandas dataframe that contains the seed genes that interact in both Biogrid and IID, indicating the source db from which the interaction originates. We then saved the interactome as a tab-separated csv file. After eliminating the redundancies, we obtained a total number of 480 seed genes that interact on both databases.

**Union Interactome**

The construction of the union interactome was more cumbersome. We had to consider all the interactions in which one of the interactor is a seed gene. A nonseed gene interacts at least once with a seed gene; then, in the union interactome we also considered interactions between nonseeds.

In the Python implementation, we first built the dataframe containing seed-nonseed interactions for both databases.

Then for the nonseed-nonseed interactions we first built a list containing the gene symbols of the nonseeds.

Iterating on both dataframes we memorized the positions where both interactors are nonseeds, and at the end of the cycle we took the rows positioned in those indexes and we built a new dataframe, one for each database.

Finally, we merged all the tables obtained in the previous steps (the table with the seed-nonseed interactions and the two tables with the nonseed-nonseed interactions) and saved in the usual tabular csv format.

**Intersection Interactome**

The construction of the intersection interactome was pretty straightforward.

We splitted the table with all proteins interacting with at least one seed gene confirmed

by both DBs by the 'source\_db' column. We dropped the 'source\_db' column and merged again the two tables. In pandas, the default type of merge will use all columns and is inner, so it returns a new dataframe with values present in both dataframes.

# Enrichment analysis

We Printed a list of genes -one for each row- of the intersection interactome for Enrichr. Than, we downloaded KEGG HUMAN 2019 and Ontologies tables of the overrepresented GO categories.

We did the same thing for the Union Interactome, but some symbols

# Notes and comments

References (if any, this is the format to be used)

Alexandrescu,A. (2001) Modern C++ Design: Generic Programming and Design Patterens Applied. Addision Wesley Professional, Boston.

Dormand,J.R. and Prince,P.J. (1980) A family of embedded Runge–Kutta formulae. *J. Comp. Appl. Math.*, **6**, 19–26.