Course project in Algorithms in Bioinformatics

Lovisa Franzén (lovisa.franzen@scilifelab.se)
October 2019

Project outline

A normal workflow in bulk RNA-seq analysis is to first identify differentially expressed genes (DEGs) between two (or more) groups, and thereafter study the most highly and significant DEGs, for instance using pathway enrichment methods. Pathways can in turn be analysed and visualised as networks, since many cellular pathways may overlap or belong to the same group of pathways – pathways such as "macrophage polarisation" and "t cell receptor signalling" could for instance belong to the same broader "immune system" category and could therefore form its own subnetwork.

My idea for this course project was to perform DE and pathway analysis, and thereafter do pathway visualisation as a graph, using *networkX*. The data used is a publicly available dataset which can be found and downloaded at Gene Expression Omnibus, accession number GSE83888 [1]. The data consists of three groups, with three samples in each group, which corresponds to 1) pre-cultured ("PC") human foetal lungs, 2) human foetal lung explants after four days culture with budesonide ("Bud") treatment, or 3) without ("Way") budesonide treatment. For the purpose of this course project work, I have chosen to focus on the comparison between the "Bud" and the "Way" groups.

Environment, code, and data

For this project, I set up a conda virtual environment in which I run Python 3.7.4. All necessary python modules are installed within this environment, and the environment was exported as a .yml file which can be used to install a new instance of this environment.

All the code was written in a Jupyter Notebook (`results/main_analysis.ipynb`), which mixes code, plots, and markdown text. Everything is shared and available at GitHub (https://github.com/lfranzen/algbio-course-project), and the Jupyter Notebook can be viewed in nbviewer here.

The RNA-seq data files are located in the `data/` folder, or they can be downloaded directly from GEO.

Brief outline of the method

This project turned out to include a lot more obstacles than first imagined. First of all, I thought there would already exist a python method for performing differential expression analysis (DEA), however after searching around a lot I didn't manage to find any method that compares to the ones available in R. Since I was determined to do this project exclusively in Python, I ended up having to write my own function for running the DEA. Luckily, we already worked a bit with that (q-value calculations) during the course, which made it a bit easier and quicker. The next major obstacle was to find a good way to convert the pathway information for each gene into an adjacency matrix. Since the pathway data obtained for each gene from MyGene was formatted in a quite unique manner, I could not find any other pre-made, quick and easy, function to create the adjacency matrix but instead I wrote my own algorithm for it. It turned out to be for the best anyway, since that way I had full control over how I wanted my data to represented.

The performed workflow for the analysis can be summarized in the following steps:

1. Merge data to form one dataset which will form the basis for the analysis
    - Only data from the "Bud" and "Way" groups are kept
2. Formulate a simple differential gene expression analysis (DEA) function
    - Based on student's t-test and q-value computations. Gene-wise log2 fold change is computed between the group averages.
3. Perform DE analysis between the treated, "Bud", and untreated, "Way", groups
    - Filtering of genes was performed before running the DEA to remove too lowly expressed genes
4. Identify differentially expressed genes based on cut-offs:
    - P-value < 0.01
    - Absolute FC ≥ 1.5
5. Identify pathways in which each gene is a part of
    - Using Reactome database, and fetching the information using MyGene
6. Create adjacency matrix from gene-pathway information which describes the connectivity between all identified pathways based on number of shared genes among the DEGs.
7. Create a graph of the pathway adjacency matrix using networkX, and visualize it in different ways

Discussion of the outcomes

Differential gene expression analysis

In the R programming language, a wide variety of well established DEA methods, such as DESeq2 and edgeR, exists. However, Python is still lagging behind within this field and lacks user-friendly and robust DEA implementations for RNA-seq analysis. The implementation that I wrote for the DEA is a quite simple and straightforward approach. For statistical analysis between the two groups, I used the Student's t-test, which provided a t-statistic and a p-value for each gene. Other DEA methods normally uses a lot more refined statistical testing for their comparison, for instance does DESeq2 first apply an internal normalization of the counts and thereafter it fits a negative binomial generalized linear model for each gene and performs a Wald test for significance testing between groups.

For the fold change calculation, I followed the standard way of computing it by dividing the gene average expression of the test group (group A) with the gene average expression of the reference group (group B), and thereafter base 2 logarithmize the fold change; $log2\left(\frac{\bar{x}_A}{\bar{x}_B}\right)$. DESeq2 also applies something they term "shrinkage" to their fold change calculations, which in short applies a prior distribution to "shrink" fold changes where the statistical information is more uncertain (i.e. for very low count values) while fold changes with more statistical information remains not shrunk. For this project, however, I did not have time to investigate any modifications to the fold change calculations to mitigate the effects of low counts on the fold change. The DEA function I wrote is therefore relatively crude approach which may increase the risk for false positives, but I believe it works fine for the purpose of this analysis.

For computation of q-values, I reused the algorithm written by Lukas Käll (source) and which was discussed during the course. After having filtered the lowly expressed genes which I used as input for the DEA, I had a total of 12918 genes for which p-values were computed. This is a very large number of statistical tests, and the q-values I obtained from these p-values were all relatively high. The lowest q-value obtained was ~0.067. To select a list of differentially expressed genes (DEGs) I therefore used a p-value cut off of <0.01 instead, which corresponded to a q-value of ~0.108 for the gene with the highest q-value. This can be interpreted as having approximately 10% of false positive genes among 634 genes which had q-values equal or lower to ~0.108 (634 ◻ 0.108 = 68.5), which is quite high. After applying a p-value cutoff, I also filtered genes based on their log2(FC), where I only kept genes that had a |FC| ≥ 1.5, resulting in a final list of 406 DEGs (*figure* 1).

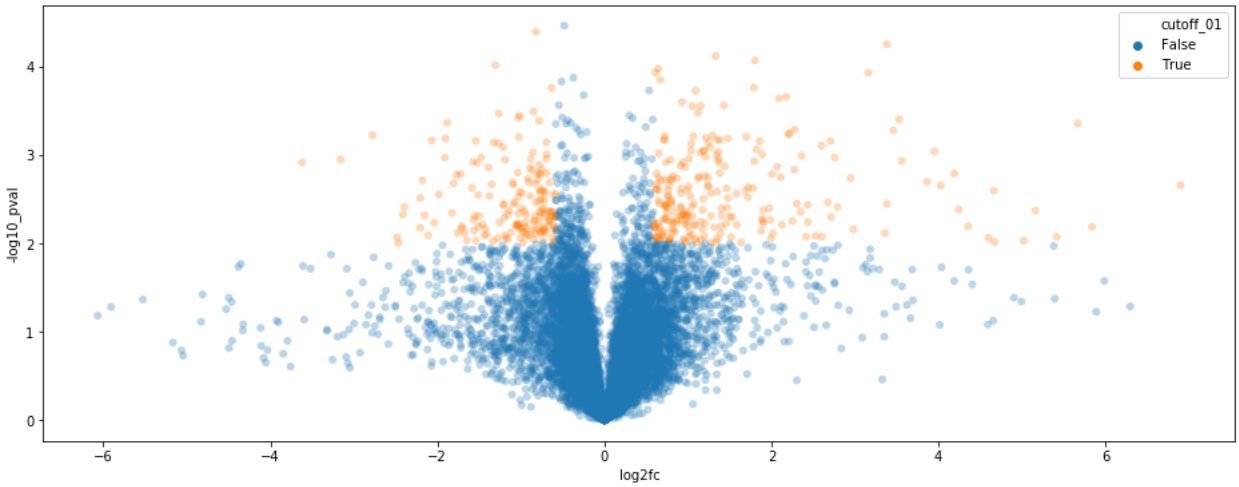

Figure 1. Volcano plot of the DEA results. Log2 fold change (FC) for every gene plotted against the -log10(p-value). In orange are genes with a p-value < 0.01 and |FC| ≥ 1.5 (approximately |log2 FC| ≥ 0.58).

Pathway annotations

There are many databases available that provides pathway annotations of genes. The most widely and commonly known may be the Gene Ontology (GO) annotations which provide a wide range of general pathway annotations for a number of model organism genes, or the KEGG database which is a well established option when studying metabolic pathways. Another such resource is the Reactome pathway database, which is neatly assembled and provides a wide range of pathway annotations – from cell signaling to disease processes.

I found a module in python named "mygene", which fetches gene annotation information from the MyGene.info database. The gene identifier used in the current RNA-seq dataset was ENSEMBL IDs, and by querying the MyGene database I was able to fetch full gene names and gene symbols for most of the genes. It was also possible to fetch information regarding pathways in which the queried gene is annotated to be part of (*Table 1*). For the sake of simplicity in this project, I decided to exclusively obtain pathway annotation data from the Reactome database. Some of the genes lacked gene and/or pathway annotations in the MyGene database, so for the downstream analysis I decided to remove these genes since I am focusing on the pathway annotations. After filtering, I had a list of 232 DEGs which all were fully annotated with gene names and pathways. After extracting all the available pathways among the DEGs, I obtained a list of 846 unique pathways which will form the nodes of the network.

query	_id	_score	name	pathway	symbol	notfound
ENSG00000130707	445	19.113523	argininosuccinate synthase 1	{'reactome': [{'id': 'R-HSA-1430728', 'name': ...	ASS1	NaN
ENSG00000124440	64344	19.658031	hypoxia inducible factor 3 subunit alpha	{'reactome': [{'id': 'R-HSA-1234158', 'name': ...	HIF3A	NaN
ENSG00000152642	23171	20.469358	glycerol-3-phosphate dehydrogenase 1 like	{'reactome': [{'id': 'R-HSA-1430728', 'name': ...	GPD1L	NaN
ENSG00000185813	5833	20.074652	phosphate cytidylyltransferase 2, ethanolamine	{'reactome': [{'id': 'R-HSA-1430728', 'name': ...	PCYT2	NaN
ENSG00000144810	1295	21.300198	collagen type VIII alpha 1 chain	{'reactome': [{'id': 'R-HSA-1442490', 'name': ...	COL8A1	NaN
...	...	...	...	...	...	...

Table 1. The first five out of 406 entries of the MyGene query results. For each gene (provided as an ENSEMBL ID), the gene name and gene symbol are fetched along with all Reactome pathways for which the gene is annotated to. The "_id" field is the NCBI gene ID (Entrez ID) for each gene. The "_score" column corresponds to MyGene's internal score which represents how well the query matches the returned gene object.

Adjacency matrix

When creating a graph, one can do so using an adjacency list or adjacency matrix. In this case, I created a weighted adjacency matrix with a dimension of 846□846. The code I wrote for creating the adjacency matrix loops over each unique pathway and searches for the occurrence of that pathway within a nested list of pathways with the length 232, where each item corresponds to a gene, and where each sublist is a list of pathways annotated to each gene. For every unique pathway, I extract the pathway sublists for all the genes where the queried pathway is found. Scores of 1 will then be added in the matrix corresponding to the position of the queried pathway and the positions all other pathways which are found within the same gene's pathway list. The number of genes identified to contain the specified pathway is also stored in a separate dictionary, and will serve as node weights or sizes for the network. In the end, the adjacency matrix will contain high scores at coordinates between pathways that are more often recurring together, while pathways which do not share any genes will have a score of 0.

Network generation

From the adjacency matrix created, it was possible to initialize a graph using *networkX*. By looking at various network metrics such as node degree, it was possible to make the

conclusion that there is a high connectivity between the nodes and that one can cross the whole network without taking a lot of steps. On average, each node is connected to 41.6 other nodes, although the distribution of the node degree is positively skewed. Also, the node size distribution is positively skewed (*figure* 2), where the average pathway contains approximately 2.68 DEGs.

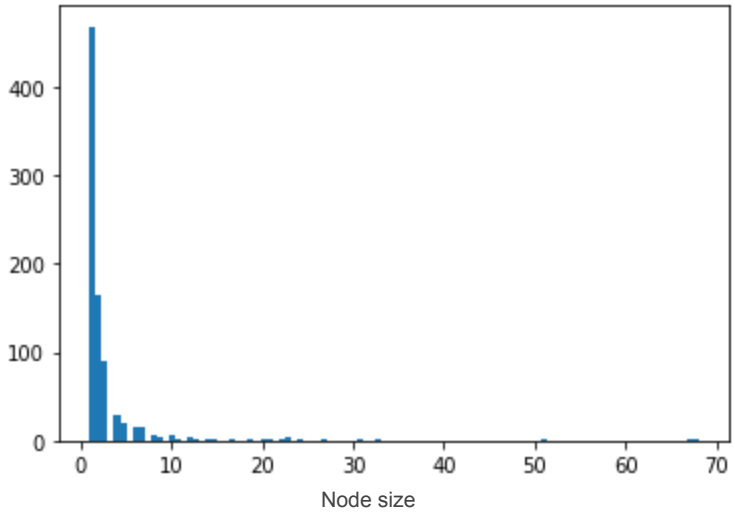

Figure 2. Node size distribution for the network. The node size corresponds to the number of genes among the selected DEGs which shares the same pathway. Most pathways have a size < 5 (average is 2.68 DEGs per pathway), while a few of them are very large.

When drawing the network, I set the node weight to control the node size and color, and the edge weight to determine the edge thickness, which gave the network a more distinct shape (*figure* 3). However, since there is still a very large number of pathways to keep track of, it is often useful to group the pathways into larger categories. Here I took a simple approach to identify a seven selected "categories" by looking for keywords in the pathway names and thereby classifying them as for instance "Metabolism" or "Signaling". I selected these categories based on which pathways were among the largest ones in the data, as well as what I thought would be of interest for this particular study. As we are dealing with lungs, I included "Hypoxia", and as the cells are from fetal tissues I also thought it interesting to look for pathways related to "Development", and so on. By making the graph interactive, using the python module *mpld3*, I enabled the possibility to see the full pathway name of a node by hovering above it, which makes it substantially easier to study the details of the network. I visualized both the full network, but also the minimum spanning tree of the network which makes the network less cluttered and it is easier to study the "core" of the network (*figure* 4A).

In the network, it was possible to see a distinct group of the hypoxia-related pathways (*figure* 4B), and there was also another branch in the network which had a group of pathways related to DNA-repair. In addition, the authors of the data discussed the

antiinflammatory effect of Budesonide treatment, and in my network there are a few big nodes corresponding to immune system-related pathways . However, the pathway of the largest size and with the highest degree of connectivity is the one corresponding to the annotated pathway “Signal Transduction”. This is a very vague pathway, which most likely have an incredibly high number of genes annotated to it, so it is not strange that we pick it up as the most prominent pathway in our data.

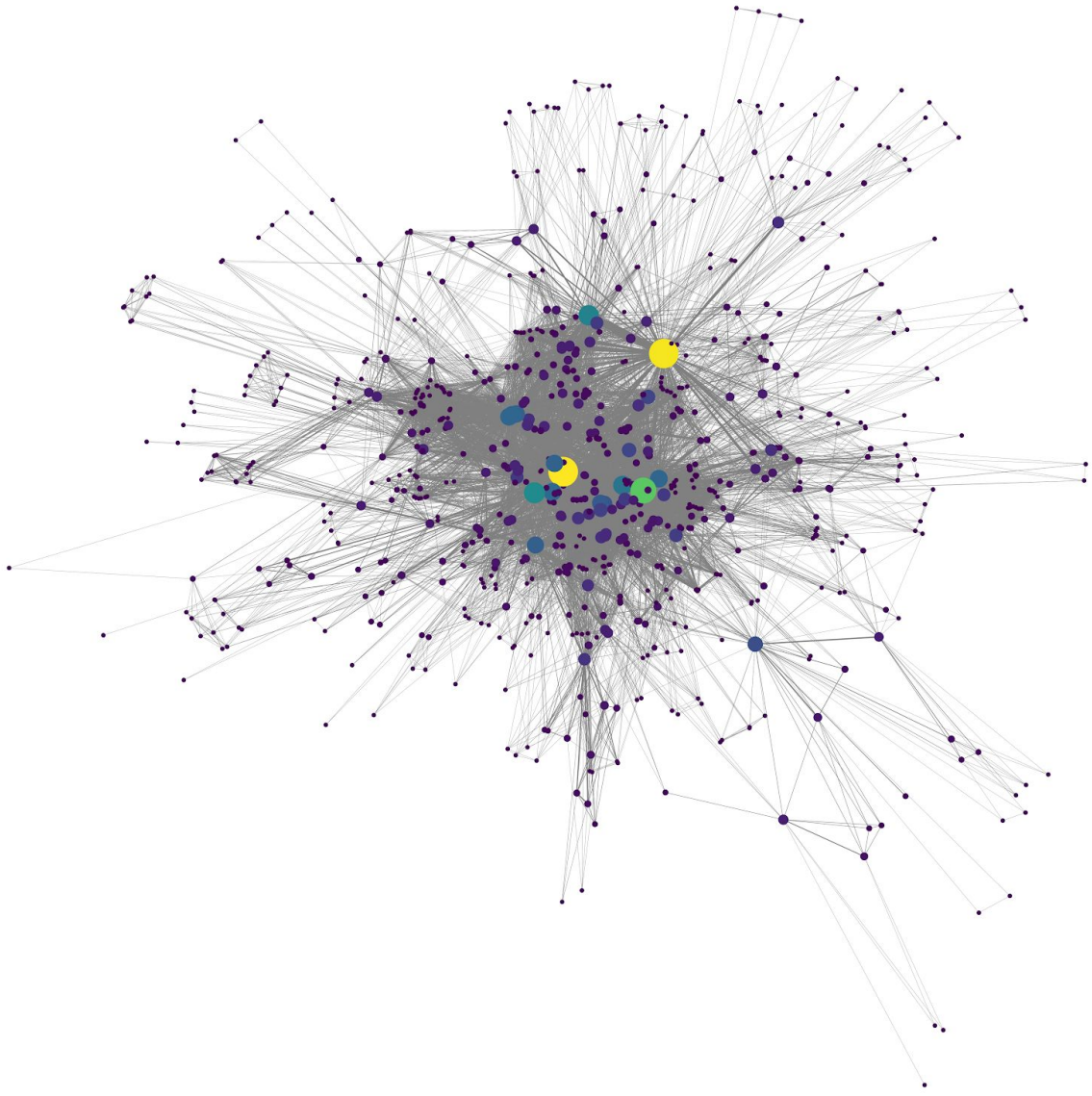

Figure 3. Network of pathways generated from the pathway annotations belonging to the selected DEGs. The distance between the nodes and the width of the edges are governed by the edge weights, i.e. the number of genes shared between the pathways. The size and color of the nodes are determined by the size, i.e. number of genes, of the pathway. The nodes in the center are highly interconnected and are also generally of larger size,

while nodes at around the network borders are generally small and have fewer connections. Smaller cliques of nodes can also be seen around the borders of the network.

Figure 4. Minimum spanning tree of the pathway network with nodes colored by pathway classification. B: At one corner of the network is a group of pathways where all are related to hypoxia. Similar groups of pathways which display high biological similarity can be found across the network. An interactive version of the network can be found in the *main_analysis.html* file.

Conclusions

In this course project, I was able to process a RNA-seq dataset from count matrices until the generation of a pathway network based on differentially expressed genes. The obtained network can be used to study what pathways are more highly "enriched" between the two sample conditions; Budesonide treated and untreated lung cultures. Genes belonging to metabolic pathways, signaling pathways, immune system related pathways, and hypoxia pathways are some examples of pathways that I was able to pick up and visualize in the network. Without much further details about whether the identified pathways are activated or deactivated in the treatment group, it is hard to draw any real conclusions from these results. However, with the time limitation of this course project the scope had to be limited, and I am still happy that I managed to get this far with this kind of analysis by exclusively working in Python and performing a lot of things from scratch, of which I had limited knowledge of beforehand.

References

[1] Anne Marie Barrette, Jessica K. Roberts, Cheryl Chapin, Edmund A. Egan, Mark R. Segal, Juan A. Oses-Prieto, Shreya Chand, Alma L. Burlingame, and Philip L. Ballard, "Antiinflammatory Effects of Budesonide in Human Fetal Lung", *American Journal of Respiratory Cell and Molecular Biology*, 2016