

Softwarewerkzeuge der Bioinformatik

Prof. Dr. Volkhard Helms
PD Dr. Michael Hutter, Markus Hollander,
Andreas Denger, Marie Detzler, Velik Velikov

Saarland University
Department of Computational Biology

Winter semester 2021/2022

Tutorial 9 January 12, 2021

Gene annotation and Jupyter Notebooks

Exercise 9.1: Gene Set Enrichment Analysis

In this exercise, you are going to perform a pathway enrichment analysis on a dataset which contains microarray gene expression data from Adult T-cell leukemia/lymphoma (ATL) patients, as well as control samples from patients without ATL. Pathway terms are a form of gene annotation that describe the molecular interactions a gene is part of, in and around the cell. The goal of this analysis is to compare the cancer/non-cancer samples, and search for pathways that are significantly enriched or depleted in cancer patients when compared to the control samples. This can give us a deeper understanding of the cellular mechanisms of cancer cells, and how they differ from healthy cells.

(a) Preparation

- (1) Gene Expression Omnibus ([GEO](#)) is a database of gene expression datasets. The dataset of ATL patients and control samples we are going to use has the GEO identifier [GSE14317](#). Click on the identifier to see the entry in the GEO database. How many ATL- and control samples were analyzed as part of the study? What gene did they focus on in their analysis?
- (2) We going to use the webservice *Genetrail* for pathway analysis. Visit the website at <https://genetrail.bioinf.uni-sb.de/> and start a *Transcriptomics* analysis.
- (3) Next, you need to add a dataset. Genetrail allows us to provide a GEO identifier, and downloads the data automatically. Select *Enter a GSE file*, and write the identifier of the dataset into the field (GSE14317).
- (4) In the next window, the samples need to be assigned to groups. Assign the samples starting with *ATL* to the sample group, the samples starting with *CD* go to the reference group. *Hint: you can select multiple samples by holding shift on your keyboard*

(b) Setting up the tests

- (1) The *Identifier-level statistic* is used to find genes which show a significant differential expression between cancer and non-cancer patients. Select the *Independent Students t-Test*.
- (2) The *Set-level statistic* calculates whether the genes associated with a certain pathway are enriched in the data. Perform a gene set enrichment analysis (GSEA) with a Kolmogorov-Smirnov test. This test is non-parametric, which means that it makes fewer assumptions about the distribution of the data and is therefore more robust.
- (3) Select only *KEGG* and *Reactome* as your categories, scroll to the top of the page and press *Start analysis*.

(c) Analysis

- (1) Open up the results window by clicking on *View*.
- (2) According to the original paper, the genes **PCNA** and **BIRC5** were among the genes with the most significant differential expression between cancer and non-cancer samples. How many Reactome/KEGG pathways are each of them part of? *Hint: Type the gene symbols into the search bar that appears on the right after selecting **KEGG - Pathways** or **Reactome - Pathways***
- (3) Search for Reactome pathways that contain **BIRC5**. Select the *More...* button next to the *SUMOylation of DNA replication proteins* pathway, then search for **BIRC5** and click on it. This will take you to the *genecards.com* entry of that gene. Why could an up-regulation of this gene be related to cancer? What is its gene name?

Exercise 9.2: Pathway & 3D Structure analysis

In this second part of the exercise, we will look at one of the pathways in more detail, and examine the three-dimensional structure of a protein complex which participates in that pathway.

- (a) Visit reactome.org and search for *SUMOylation of DNA replication proteins*. Select the first result. Click on the name of the pathway in *Locations in the PathwayBrowser*.
- (b) As we found out in Exercise 9.1, the gene **BIRC5** is a member of this pathway. Now we need to find out where it is located. On the upper left hand side, you will find a search window. Search for **BIRC5**, and select the *human* variant that is located in the Nucleoplasm. That will open a small window underneath called *Details*.
 - (1) What protein complex is **BIRC5** a part of? What are the other proteins in that complex?
 - (2) Select the complex, then click on *Expression*. Name three tissues where **BIRC5** has a high gene expression.
 - (3) What reaction does **BIRC5** participate in here? Look at the inputs, outputs and catalysts for that reaction and describe what happens. *Hint: Click on the arrows that represent the reaction*
- (c) Another pathway **BIRC5** is annotated with contains proteins that are inhibited by **TP53**. We can use the webservice *STRING-db* to find more information about that protein.
 - (1) Visit <https://string-db.org/>, select *Search*, and write the gene symbol **TP53** into the field. Select *Homo sapiens* as the organism.
 - (2) Under *Settings*, select only *Experiments* and *Databases* as the active interaction sources, set the minimum required interaction score to *high confidence*, and press *Update*.
 - (3) Click on the Node of **TP53** to get a short summary. What role does it play in many tumor types and what is its main function?
 - (4) Expand the network around **TP53** to 20 proteins by pressing the *+More* button on the right twice.
 - (5) Select the *Analysis* tab. Here you can find gene annotations that are enriched in this subnetwork of 20 proteins. You can click on annotation to highlight the annotated proteins in the network above. Which of the 20 proteins are involved in small cell lung cancer? *Hint: Expand the list of KEGG Pathways*

Exercise 9.3: Jupyter Notebooks Recap

In exercises 2 and 3, you first learned about writing Python Code using Jupyter Notebooks. The website kaggle.com lets you create and run code in notebooks without installing anything on your local machine. You already learned about variables, loops and functions. This time, we will import additional python packages, and add an external dataset to the project. We will analyze a *Non-Small-Cell Lung Carcinoma* (NSCLC) dataset with the GEO accession [GSE74706](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE74706). It contains samples of cancer patients, and samples from healthy patients for reference.

- (a) Familiarize yourself again with the website. Log in to your account (or create a new one with your email or your Google account), and create a new notebook for this exercise.
- (b) Rename your notebook into "Tutorial 9", and delete the example code cell that already exists in the notebook. Create a new code cell, type "2+2", and execute the cell. If everything is working, you should see the result underneath.
- (c) Click on **+Add data** on the top right. Choose *Upload* in the top right of the window, next to the X. Select the second icon from the top on the left. Enter *lungcancer* as the title of your dataset. Copy the following URL and paste it under **Remote Files**: https://sbcbr.inf.ufrgs.br/data/cumida/Genes/Lung/GSE74706/Lung_GSE74706.csv
Click on **+Add Remote Files**, then on **Create**. Uploading the data will take about 1-2 minutes.

Alternatively, you can download the dataset through the link, and upload it to the notebook manually.

- (d) Now, we will load the dataset into a *DataFrame*, a data structure that is provided by the *pandas* package for Python. A *DataFrame* is essentially a table with rows and columns, similar to an Excel sheet or a SQL database. A reference can be found [here](#).

- (1) First, you have to find out the location of the document you just uploaded on the server. On the top right, there is a dropdown menu containing the input and output data. You will find the dataset under *input* → *lungcancer* → *Lung_GSE74706.csv*. Here, you can copy the path to the file.

- (2) This cell will read the *.csv* file into a *DataFrame* called *lungcancer_df*:

```
import pandas as pd
file_path = "../input/lungcancer3/Lung_GSE74706.csv"
lungcancer_df = pd.read_csv(file_path)
```

- (3) Create a new cell, simply write *lungcancer_df* into it, and execute the cell. Now you can see the table underneath. How many rows and columns does it have?

- (e) Next, we will clean up the data a bit. The first step is to save the labels (i.e. whether a sample is a normal- or a cancer sample) in a separate list. Next, any column that doesn't contain the gene expression of a gene is removed. That makes the data easier to work with. Create a cell that looks like this:

```
labels_df = lungcancer_df["type"]
features_df = lungcancer_df.drop(["type", "samples"], axis=1)
```

- (f) Finally, we calculate a few statistics on the data to get an overview.

- (1) Count how often each label occurs:

```
labels_df.value_counts()
```

How many *normal* and *NSCLC* (lung cancer) samples are present in the dataset?

- (2) Calculate statistics on the gene expression data:

```
features_df.T.describe()
```

A normalized dataset has the property that the mean values and standard deviations (std) of the samples are very close to each other. Is the dataset sufficiently normalized or is a normalization step necessary?

- (3) Create a histogram from the average expression of the genes:

```
features_df.mean().hist()
```

Shortly describe what you see on the plot.

Have fun!