Introduction to Python for Bioinformatics and the Life Sciences

Instructions for final project

Preparation: Download gene expression and phenotype data

- **Download normalized bulk gene expression** data from breast cancer patients here: https://gdc-hub.s3.us-east-1.amazonaws.com/download/TCGA-BRCA.htseq_fpkm.tsv.gz
- **Download corresponding phenotype data** here: https://gdc-hub.s3.us-east-1.amazonaws.com/download/TCGA-BRCA.GDC_phenotype.tsv.gz

After having downloaded the data, create a Jupyter notebook satisfying the following criteria:

- The notebook should carry out all of the steps detailed below, assuming that the two datasets are contained in the same directory as the notebook.
- Explicitly state all dependencies in a Markdown cell at the very top of the notebook and then import all of them together in the next cell.
- Clearly explain all steps you are carrying out, e.g., using Markdown cells.
- The notebook should run smoothly when executed from top to bottom without throwing any errors.
- You will be graded based on the following criteria: syntactic correctness of the code (20 points); logical correctness of the code (20 points); readability of the code (15 points); documentation and signposting text (10 points); and finally, readability (5 points), quality (15 points) and correctness (15 points) of the plots.

Submit your notebook through StudOn by May 15, 2025, 23:59 CET.

Q & A session for project submissions

- May 5, 2025, 16:15 – 17:45
- Room 3.17, Werner-von-Siemens-Str. 61 (third floor)

Step 1: Load and filter data

- **Load both datasets into pandas DataFrame objects.** Use sample IDs as indices and gene IDs as columns. If the data is not provided in this format, transpose it if necessary.
- **Partition the gene expression data frame into two data frames** such that one data frame only contains data for primary tumor samples and the other one only contains data for healthy tissue samples. You find the information necessary to split the data in the phenotype data. To identify the variable which contains the necessary information on the

sample types, you can have a look at this paper: https://doi.org/10.1093/bib/bbad413 (especially the Methods section).

Step 2: Identify differentially expressed genes

- **Find genes that are differentially expressed** between primary tumor and healthy tissue samples using the Mann-Whitney U test provided by SciPy: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.mannwhitneyu.html
- **Correct for multiple testing** by using an appropriate function provided by the statsmodels package: https://www.statsmodels.org/stable/index.html.
- For each gene, **compute the log2 fold change** between primary tumor and healthy tissue samples.
- **Generate a pandas DataFrame with the results** which, for each gene, contains the *P*-value (from the Mann-Whitney U test), the adjusted *P*-value (from the multiple testing correction) and the log2 fold change.
- **Write the results to a csv file.**

Step 3: Generate a volcano plot

- **Generate a volcano plot** visualizing the results generated in Step 2, using appropriate seaborn and matplotlib functions.
- **Save the plot in a PDF.**
- **Highlight the 10 genes with the smallest *P*-values** by plotting them in a different color and annotating the dots with the gene IDs.

Step 4: Carry out gene set enrichment analysis

- **Carry out gene set enrichment analysis** with GSEApy's Enrichr API to find KEGG, GO-MF, GP-CC, and GO-BP pathways that are significantly enriched with the 10 genes with the smallest *P*-values identified in Step 3.
- **Plot the result** using one of GSEApy's in-built plotting functions.
- **Hint:** TCGA uses Ensemble gene IDs (with version numbers, the trailing numbers after the dots in the IDs). GSEApy expects gene names. So you have to map the Ensemble IDs to gene names. There are many Python packages you can use for this. One option is GSEApy's Biomart API. With this, you can map Ensemble gene IDs without version numbers to gene names. So you have to remove the version number from the IDs used by TCGA before feeding them into GSEApy's Biomart API.

Step 5: Same workflow for another phenotype variable

- **Identify another biologically or clinically interesting variable** provided in the phenotype data. To find such a variable, you can (but do not have to) again take inspiration from this paper: https://doi.org/10.1093/bib/bbad413. Justify your choice in a Markdown cell.
- **Carry out Steps 1 to 4 for the variable you have selected** instead of the sample type variable used before.