# TreMSuc

*Release 1.0*

**Gabor Balogh**

**Mar 22, 2024**

# CONTENTS:

"TreMSuc" a tool to choose, harvest and analyse expression and methylation data of the TCGA-projects

CHAPTER

ONE

# BUILD AND ACTIVATE THE PROVIDED CONDA ENV:

```
$ conda env create -f deseq_env.yaml
$ conda activate deseq_pipeline
```

call the script without any options to enter the interactive mode and set each option step by step:

```
$ python main_deseq.py
```

print help page:

```
$ python main_deseq.py --help
```

CHAPTER

TWO

# DOCUMENTATION OF MODULES, CLASSES AND FUNCTIONS:

## 2.1 TreMSuc

TreMSuc, a tool to choose, harvest and analyse methylation and rna count data of the TCGA-projects with help of the package metilene and DEseq2.

Build and activate the provided conda env: bash

$ conda env create -f metilene_env.yaml

$ conda activate metilene_pipeline

call the script without any options to enter the interactive mode and set each option step by step:

$ python main_metilene.py

print help page:

$ python main_metilene.py --help

```
TreMSuc [OPTIONS]
```

### Options

**-o, --out_path** <out_path>

path to save the result files

**Default**
/homes/biertruck/gabor/TCGA-pipelines

**-p, --project** <project>

TCGA project to be applied. Any TCGA project can be chosen, like: -p TCGA-CESC -p TCGA-HNSC ...

**-d, --drugs** <drugs>

drug(s), like: -d drug1 -d drug2 or drugcombination(s), like: -d drug1,drug2

**-c, --cores** <cores>

number of cores provided to snakemake

**Default**
1

**-C, --cutoff** <cutoff>

Cut-off parameter

**Default**
0

**-t, --threshold** <threshold>
threshold parameter

**Default**
0

**-e, --execute** <execute>
choose which pipeline shall be executed

**Default**
DESeq2, metilene

**-D, --dryrun**
snakemake dryrun

**Default**
False

**-r, --report**
just create a report

**Default**
False

**-v, --version**
printing out version information: Version 1.0

CHAPTER

# THREE

# SHORT TUTORIAL:

## 3.1 Performing an example analysis:

The easiest way of applying a run is entering the interactive mode (it is supposed that you cloned the deseq_pipeline git repository and cd into that dir):

```
$ python main_deseq.py
```

With it, every needed parameter is offered for further analyses. First, the available projects are presented, based on that selection, available drugs or drug combinations can be chosen.

In contrast to that, the parameter needed could be applied via command line. An example terminal call for the projects TCGA-CESC and TCGA-HNSC together with the drug cisplatin and the combination carboplatin,paclitaxel would be:

```
$ python main_deseq.py -p TCGA-CESC -p TCGA-HNSC
-d cisplatin -d carboplatin,paclitaxel -o /OUTPUT_path -D -A
```

First of all, the needed data for the selected projects is loaded via the TCGA API and stored in:

- /OUTPUT_path/TCGA-CESC/TCGA-CESC_data_files/ and
- /OUTPUT_path/TCGA-HNSC/TCGA-HNSC_data_files/

Intermediate merged tables and additional meta_data tables are stored in the project directories:

- /OUTPUT_path/TCGA-CESC/
- /OUTPUT_path/TCGA-HNSC/

First, single project analyses are performed. The actual analysis is determined by the project, and by the drugs combination. The directory for the drugs combination is created out of the applied drugs, so here, the DRUGS_title is 'carboplatin,paclitaxel_cisplatin'.

Everything below that drugs directory, is restricted to the chosen drugs s.t. the results of both single project analyses are placed in:

- /OUTPUT_path/TCGA-CESC/carboplatin,paclitaxel_cisplatin/
- /OUTPUT_path/TCGA-HNSC/carboplatin,paclitaxel_cisplatin/

After the single project analysis, the projects are combined. Those results are stored in an additional directory, composed out of the applied projects, so here, the PROJECT_title is: 'TCGA-CESC_TCGA-HNSC', those results are saved in the directory:

- /OUTPUT_path/TCGA-CESC_TCGA-HNSC/carboplatin,paclitaxel_cisplatin/

Since the analysis is determined by the project and drug combination, results for 3 different approaches are created, two for the single projects and one for the aggregation of the two projects. For all of them, a respective REPORT.pdf is

created, containing a summarized representation of the most important results and plots, along with some explanations to them. They are stored at:

- /OUTPUT_path/TCGA-CESC/carboplatin,paclitaxel_cisplatin/REPORT.pdf
- /OUTPUT_path/TCGA-HNSC/carboplatin,paclitaxel_cisplatin/REPORT.pdf
- /OUTPUT_path/TCGA-CESC_TCGA-HNSC/carboplatin,paclitaxel_cisplatin/REPORT.pdf

## 3.2 Recreate the performed analysis:

To rerun the analysis and reproduce all the outputs and results created with it, a single Snakemake configuration file is created. It is stored in the cloned repository location under the 'Snakes' subdir. Since the analysis is determined by the composition of projects and drugs, the unique filename of this configuration file is composed out of it. For the example with CESC and HNSC, together with cisplatin and carboplatin,paclitaxel, that would be:

- SCRIPT_path/Snakes/snakemake_config_TCGA-CESC_TCGA-HNSC_carboplatin,paclitaxel_cisplatin.yaml

The Snakefile needed is also hold available at:

- SCRIPT_path/Snakes/Snakefile

This file must be edited and the path to the config yaml file, the OUTPUT_path and the SCRIPT_path must be inserted.

With that, the Snakefile is configured to run the analyses again. Change the directory into the SCRIPT_path/Snakes/ path and run for example:

```
$ snakemake --cores 7
```

This would use 7 cores of your machine if available and make use of parallelisation of steps where it is feasible.

# INDEX

## Symbols



## T