Analysis of Stress Resistance in Yarrowia lipolytica - Docker Setup Guide

Welcome in the workflow of analyzing growth data for Y. lipolytica Transcription Factors strains 😃
This guide helps you run the stress resistance analysis pipeline **using Docker Desktop** with **no command-line required**.

What You Need

- Docker Desktop installed
- Excel data files (your experimental results and strain layout)
- This Docker image:

mariagorczyca/stress-resistance-msc:latest

Folder Setup on Your Computer

Create the following folders on your computer (anywhere you like, for example in your Documents folder):

stress-resistance/
├── my_data/ ← put your Excel files here
├── stress_output/ ← output Excel files will be saved here
└── stress_figures/ ← plots and figures will be saved here

How to Run It in Docker Desktop

1. Open **Docker Desktop** and go to the **Images** tab.

2. Find the image:

mariagorczyca/stress-resistance-msc:latest

Click the **"Run"** button.

3. In the container setup screen:

- **Name**: (optional) set a name like `stress-pipeline`

- **Volumes**: click **"Add file system"** three times and configure:
  (this is unimportant how you name your folders locally, it is important thet they would refer to the exact folders in the Container Path)

Host Path	Container Path
C:\Users\yourname\stress-resistance\my_data	/stress-resistance-msc/data

Host Path	Container Path
`C:\Users\yourname\stress-resistance\stress_output`	`/stress-resistance-msc/output_data`
`C:\Users\yourname\stress-resistance\stress_visuals`	`/stress-resistance-msc/visualisations`

- Optionally, you can enter the DILUTION_FACTOR variable. By default, the script assumes DILUTION_FACTOR = 1.0, meaning that growth values should already be calculated by the user in the results_XXX.xlsx files. If a constant dilution was used throughout the entire experiment, you can optionally set a different DILUTION_FACTOR to have the script calculate growth by multiplying absorbance values accordingly.

4. Scroll down and click **Run Container**.

5. The analysis will automatically start. When it's done:

- Check your `stress_output/` folder for Excel results.
- Check your `stress_figures/` folder for generated plots.

File Requirements

Make sure to include the following files in `my_data/`:

1. **Strain layout** file
   Example: `test_strain_layout.xlsx`

- The script automatically recognises any file that ends with 'strain_layout.xlsx'. You can name the beginning of the file however you like.
- This file defines the mapping between your results and the strains used in your experiment.
- You should provide ECY numbers and the layout can follow any structure (e.g., same as the microplate reader's output) — whichever is most convenient.

2. **Results** files
   Example: `results_YPD_24.xlsx results_YPD-pH3-OA-_72.xlsx`

- The script recognizes any file that starts with 'results'.
- It extracts:
    - the experimental condition from the first part after the underscore, and
    - the time point from the second part.
    - For example, from the filename `results_pH3-OA+_48.xlsx`, the script sets: Condition: pH3-OA+ Time: 48 (in hours)
- Important! Use exactly two underscores in the filename: one after 'results', one between the condition name and the timepoint. You cannot add additional underscores elswere in the filename.

What do the scripts do?

1. `01_data_import.ipynb`

   Imports and organizes your experimental data.

   This script reads absorbance values from your result files and calculates growth (if raw OD values are used). It maps ECY numbers to strain names, types of genetic modification, and transcription factors. As output, it generates an Excel file containing all organized data in a single DataFrame.

2. `02_plot_conditions.ipynb`

   Visualizes growth under different environmental conditions.

   This script plots a line graph for each strain grouped by type of modification. Each subplot represents one environmental condition. It averages replicate values and is designed to provide a quick overview of your experiment outcomes 😃

3. `03_plot_TFs.ipynb`

   Compares transcription factor modifications across conditions.

   This script focuses on a single transcription factor and compares overexpression, knock-out, protein overexpression variants, and the control strain within a chosen condition. It plots means with standard deviation as an opaque band, giving you a visual basis to evaluate the impact of TF modifications on Yarrowia phenotype.

4. `04_ANOVA.ipynb`

   Performs statistical analysis.

   Runs ANOVA to determine which strain variants differ significantly from the control strain under specific conditions and timepoints. It outputs Excel files, where you can take a look what variants differ significantly.

5. `05_ANOVA_processing.ipynb`

   Extracts statistical significance with control strain.

   Extracts the relevant information from ANOVA and Tukey post-hoc results, concerning the significance of difference between modified TF and control strain. It alignes the statistical insights with the growth data to form a new statisctis data dataframe, to be used in the next step for preparing the fold change heatmaps.

6. `06_fold_change_heatmap.ipynb`

   Generates heatmaps of fold changes.

   Creates heatmaps that visualize fold changes (relative changes) between each studied strain and the control. Each timepoint has its own heatmap. Significant differences compared to control strain are annotated with fold change values.

7. `07_dendrogram.ipynb`

   Generates clustered dendrograms.

   Creates plots of clustered dendrograms showing the "relationship" between each TF modified strain and conditions. It clusters the growth data based on their means, and the metric used to cluser is mean distance. It standardizes the growth results in each column (condition) so that the condition-specific growth differences doesn't disturb the analysis. There is one dendrogram generated per each timepoint.

Re-running

To re-run the pipeline with new data:

1. Stop and delete the container (image stays safe).
2. Update your Excel files in `my_data/` on your computer.
3. Run the image again from Docker Desktop.

Need Help?

If anything goes wrong, contact me.

Have fun watching Your data being analyzed 😉