

**Exercise for
Constraint-based Modeling of Cellular Networks
20 October 2022**

**Homework should be sent to Anika Küken (ankueken@uni-potsdam.de)
by 27.10.2022 before the exercise**

Exercise

We get familiar with creation and handling of numbers, vectors, matrixes and tables in Matlab.

Homework

The file "GC_MS_Data_exercise_1.txt" (adapted from Jozefczuk et al. 2010) contains tab-separated raw data on metabolite peak height from GC-MS analysis for 97 metabolites. Metabolite abundance was measured under control conditions and four stress conditions in 3 biological replicates at 9 time points, indicated by a suffix to the sample_ID and in the column 'BIOLOGICAL_replicate' and 'timePoint', respectively.

- a. Use function *readtable()* to read the data from the file into a Matlab table. Use the first row as variable names!
- b. Create a numeric (double) matrix from the metabolite abundance data of the table.
- c. Use function *histogram()* to plot a histogram summarizing the distribution of log-transformed peak height across all metabolites under control conditions at time point 9 with 30 bins. Next, overlay the histogram created before with a histogram showing the distribution of log-transformed peak height across all metabolites under cold stress at time point 9 using the same bins (i.e. bin edges) as before. Add axis labels and a legend to the figure.

*Hint: functions intersect() and strcmp() can help finding indices that match condition and time point
check command >> hold on*

save histogram as variable >> h1 = histogram() to get access to used bin edges

- d. Calculate a new (double) matrix in which the replicates of each metabolite are pooled by calculating the mean across the 3 replicates, respectively.

Hint: use nested for loops that iterate over conditions and time points

- e. Normalize the new matrix by dividing each value by the row sum.
- f. Create a new table from the normalized values, using the metabolite names as variable names. Add columns for time point and condition. *Hint: functions repmat(), array2table(), cell2table() can be helpful*
- g. Create a new figure using function *plot()* showing the change in normalized abundance of Pyruvic acid over time for the different conditions. Do not forget to add axis labels and a legend.

Hand in: **Commented** source code (.m file) for running the above calculations, plotting the figures (do not forget to add axis labels) and creating the new table.

NOTES:

- Use long, descriptive variable names. For example, use "matrixnormalized" instead of "mn".
- Add a comment line to at least each two lines of code.
- Pay attention that variables do not have the name of an existing function, such as "table".