Author: Annsophie Weber *Date: 07.07.2020*

Developed R scripts

Eight R scripts have been developed for the analyses. These can be divided in three groups on the basis of required input files.

The first group contains two scripts: one script to perform a differential gene expression analysis on not normalised transcript read counts ("1.1_differential_gene_expression _analysis.R") and a second script for preparing and normalising the data set of substances for the subsequent scripts ("1.2_normalisation_of_substances.R").

The output, meaning the results, of these both scripts is required for the five scripts of the second group. This group consists of scripts used for an in-depth analysis of the DE analysis for selected genes ("2.1_subsequent_differential_gene_expression_analysis.R"), creating Venn diagrams depending on the class of differential gene expression in two tissues ("2.2_venn_diagrams.R"), a PCA ("2.3_principal_component_analysis.R") to identify differences and similarities between growth stages, calculating correlations between the data sets ("2.4_correlation.R") and plotting a time series plot with substance concentrations and transcript counts combined in one plot ("2.5_substance_transcript_time _series_plots.R").

The third group consists of just one script that was created to offer a more detailed analysis of the previously calculated correlation and to gain an in-depth look into the behaviour of and relation between genes and substances of special interest ("3.1_subsequent_correlation_analysis.R"). In general, all produced output files are stored in a new generated directory, located in the same directory as the required input file. When using the scripts the user has to give some additional information besides the input files. All steps are well explained in each script through comments and are listed as well in the appendix of this thesis. In the following, each script is introduced and described individually and in detail.

Differential gene expression analysis with ImpulseDE2

To facilitate a DE analysis for a time series with ImpulseDE2, the script "1.1_differential_gene_expression_analysis.R" was implemented.

Each time series replicate with unnormalised transcript counts has to be supplied as individual input file in CSV format. Transcripts have to be recorded as rows with the timepoints as columns.

The differential gene expression analysis is carried out by using the "runImpulseDE2" function of the package "ImpulseDE2". All results and output files are generated extracting and combining information from the produced "objectImpulseDE2" object. In the script, a false-discovery rate corrected p-value threshold of 0.01 was set to identify differentially expressed genes with a reduced the number of false positives.

The output includes a report file in TXT format about the "runImpulseDE2" process and a TXT file which states how many genes where included in and excluded from the analysis and how many genes were associated with each differential expression class. Additionally, it creates a CSV file containing all genes which show zero transcripts at all timepoints in all replicates. Heatmaps are saved as PNG files, displaying raw as well as impulse model fitted z-scores of normalised transcript count means. Values displayed in the heatmaps are stored in corresponding CSV files. Besides, CSV files comprising normalised transcript counts, means and standard deviations in different combinations are generated. The ImpulseDE2 object is saved as RDS file. Optionally, time series plots for a number of top differentially expressed genes can be saved PNG format. The plots can contain the suggested impulse fit model, a line visualising the normalised transcript count means and points for all individual normalised means. A second additional option was added to enable the user to create CSV files containing only transcript counts above a selected threshold for at least one timepoint.

Normalisation of substances

The script "1.2_normalisation_of_substances.R" was developed to guaranty a consistent normalisation of substance concentrations. The input requires each time series replicate of substance measurements as individual XLSX file or sheet.

Each file has to comprise the column "Name" containing substance names and "RI" containing retention indices. Sometimes after a gas chromatography and a mass spectrometry a substance can not be clearly identified, so it will not be called by an IUPAC conform name, but by the retention index. The retention indices are needed to create a unique identifier for all substances, if some names are missing. Moreover, if the substance measurements were not obtained by a GC-MS workflow, instead of retention indices anything else can be put into the "RI" column, to make a unique identifier, if no name is stated. In the first step of the normalisation algorithm, each substance in each time series replicate is normalised based on the maximal substance measurement and then translated in percent. In case of a GC-MS workflow, the measured peak areas for each substance are transformed into relative values based on the maximal peak area are and then translated in percent.

As a second step, means for each percentage substance concentration depending on the timpepoint will be calculated. To ensure a convincing mean, each timepoint has to comprise at least three nonzero values for a substance, from which a mean can be calculated, otherwise all values will be set to zero for the substance and the depending timepoint. As a third and final step, these calculated means are again normalised based on the maximal percentage for each substance, similar to the first step. This ensures that the maximal amount of a substance is always 100%.

Consequently, a CSV file containing these percentage normalised means for each substance and each timepoint is generated as output. Optionally, it is possible to generate a CSV file containing a subset of selected timepoints.

In-depth evaluation of selected differentially expressed genes

The DE analysis performed by the script "1.1_differential_gene _expression_analysis.R" gives a general view of all differentially expressed genes grouping them into four classes.

Unfortunately, it is not directly possible to have a look into single genes or groups of specific genes the user might be interested in. That is why the script "2.1_subsequent_differential_gene _expression_analysis.R" was developed for a subsequent DE analysis, giving the user the chance to ease generating time series plots and small heatmaps for one or more individual genes.

As input file serves the RDS file containing the ImpulseDE2 object generated by the script "1.1_differential_gene_expression_analysis.R". In addition, names of selected genes that should be evaluated have to be provided by the user.

First of all, if none of the selected genes shows any transcript counts at any timepoint, meaning they are all zero, a TXT file will be generated stating so, resulting in no other output files. If single selected genes show no transcript counts at all, they will be excluded from the analysis and no output files will be generated for these genes. The analysis for all other non-zero state selected genes will be performed regularly. Similar to the script for the over-all differential gene expression analysis, this script outputs two heatmaps in PNG format visualising the raw as well as the impulse model fitted z-score values from normalised transcript count means for the selected genes and as a matter of course corresponding CSV files containing the values displayed in the heatmaps. Furthermore, this script generates analogously to the over-all analysis CSV files comprising normalised transcript counts, means and standard deviations in different combinations for the selected genes. The same time series plots that can be produced optional with the over-all differential gene expression analysis script for the top differentially expressed genes are generated obligatory for the selected genes by this script and saved as PNG files.

Comparison of differentially expressed genes between tissues by using Venn diagrams

In transcriptomic studies, two or more samples, conditions or tissues are prevalently analysed and compared, mostly focussing on differentially expressed genes. Subsequent to the DE analysis, which has to be performed for each sample, condition or tissue individually by the corresponding script, an additional script ("2.2_venn_diagrams.R") was implemented to visualise similarities and differences between the differentially expressed genes of two transcriptomic data sets with Venn diagrams.

As input serve two CSV files, more precisely one of the z-score CSV files for each sample, condition or tissue, generated by the script for differential gene expression analysis. These files do not only contain the z-scores as numbers, plotted in the heatmap, but also the classes of differential expression for each listed gene.

Since ImpulseDE2 performs a classification into four classes of differential expression, 16 pairwise comparisons between two data sets can be evaluated plus one comparison of all differentially expressed genes not taking the classification into account.

These up to 17 comparisons are automatically performed by this script and corresponding Venn diagrams are saved in PNG format. If in one or both of the data sets one or more classes of differential expression do not occur, there will be no Venn diagrams generated as visualising an empty set does not make sense. When in doubt about the set sizes, information on the size of the classes can be found in a file generated by the differential gene expression analysis script. For a quick overview on all pairwise comparisons, a CSV file comprising a table with the corresponding amounts of overlapping genes is generated. Moreover, a second CSV file is produced containing the names of overlapping genes for each pairwise comparison.

PCA, Friedman and Wilcoxon-Nemenyi-McDonald-Thompson test

The PCA and Friedman test R script ("2.3_principal_component_analysis.R") can be used with either a time series data set of substance concentrations in percent or transcript counts.

It requires a CSV file as input for the analysis. The timepoints should be defined as columns and the substances or genes as rows.

In this script, the "prcomp" function from the "stats" package, R's base package for statistical functions, is used, calculating the principal component analysis by a singular value decomposition. The Friedman test and the subsequent post hoc test are calculated by using a function Galili (2010) provided as a source, with the slight modification that no parallel coordinate plots and boxplots will be generated as default. This function performs a pairwise comparison, called the Wilcoxon-Nemenyi-McDonald-Thompson test (Hollander and Wolfe, 1999), to test in this case for significant differences between the timepoints with correction of multiplicity, comparable to a Wilcoxon signed-rank test (Galili, 2010).

The output of this script consists of a PCA plot (PNG format), a scree plot (PNG format), a selectable number of variables that have the highest effect on principal component 1 and 2 (CSV file), the result of the Friedman test (TXT file), and the results of Wilcoxon-Nemenyi-McDonald-Thompson post hoc test with all results of the pairwise comparison, meaning the test for significant differences between the samples (CSV file).

Correlation coefficients

To identify potential coherences within a transcriptome, volatilome or metabolome and also between genes and substances, the script "2.4_correlation.R" was developed.

Two CSV input files are required to perform the calculation of a correlation matrix. It is possible to compare a time series data set of substances with a time series data set of transcripts, as long as they share the same amount of timepoints. Furthermore, it is possible to use the same file twice as input, so the correlations within the data set can be studied. As input should serve the CSV file containing normalised transcript count means generated by the script for differential gene expression and/or the CSV file containing normalised substance means generated by the script for normalisation of substances.

The user is able to choose the method, either Pearson, Kendall or Spearman, by which the correlation coefficients should be calculated. These calculated correlation coefficients are stored in a correlation matrix, saved as CSV file. Additionally, a heatmap will be generated visualising the correlation matrix. The script optionally offers the possibility to extract the x highest correlations, where x is a number the user is allowed to set.

Plotting simultaneously substance concentrations and transcript counts

The script "2.5_substance_transcript_time_series_plots.R" was implemented to create time series plots which contain a percentage substance concentration and transcript counts of one or two data sets. Dependent on the input files, this script appertains to the second group of scripts. Nevertheless, if one divides the scripts into groups by a logical sequence of execution, it fits better to the third group and might rather be executed after the subsequent correlation analysis. That is because after the subsequent correlation analysis the user might have more insight into coherences between selected substances and genes, which can then be plotted with this script in a common time series plot.

To create the time series plots, one CSV file containing the normalised percentage substance concentrations and one CSV file containing normalised transcript count means is required, both necessarily representing the same timepoints. In addition, a second CSV file containing another set of normalised transcript count means can be read into the environment and plotted within the same plot. This second transcript time series does not have to represent the same amount of timepoints represented by the substances and first set of transcript counts and can be a subset of these timepoints.

The user then has to select substances and genes which should be plotted together. One substance is always plotted together with one gene of one or two transcriptome data sets, respectively. When defining more than one substance and more than one gene, time series plots are produced for all combinations.

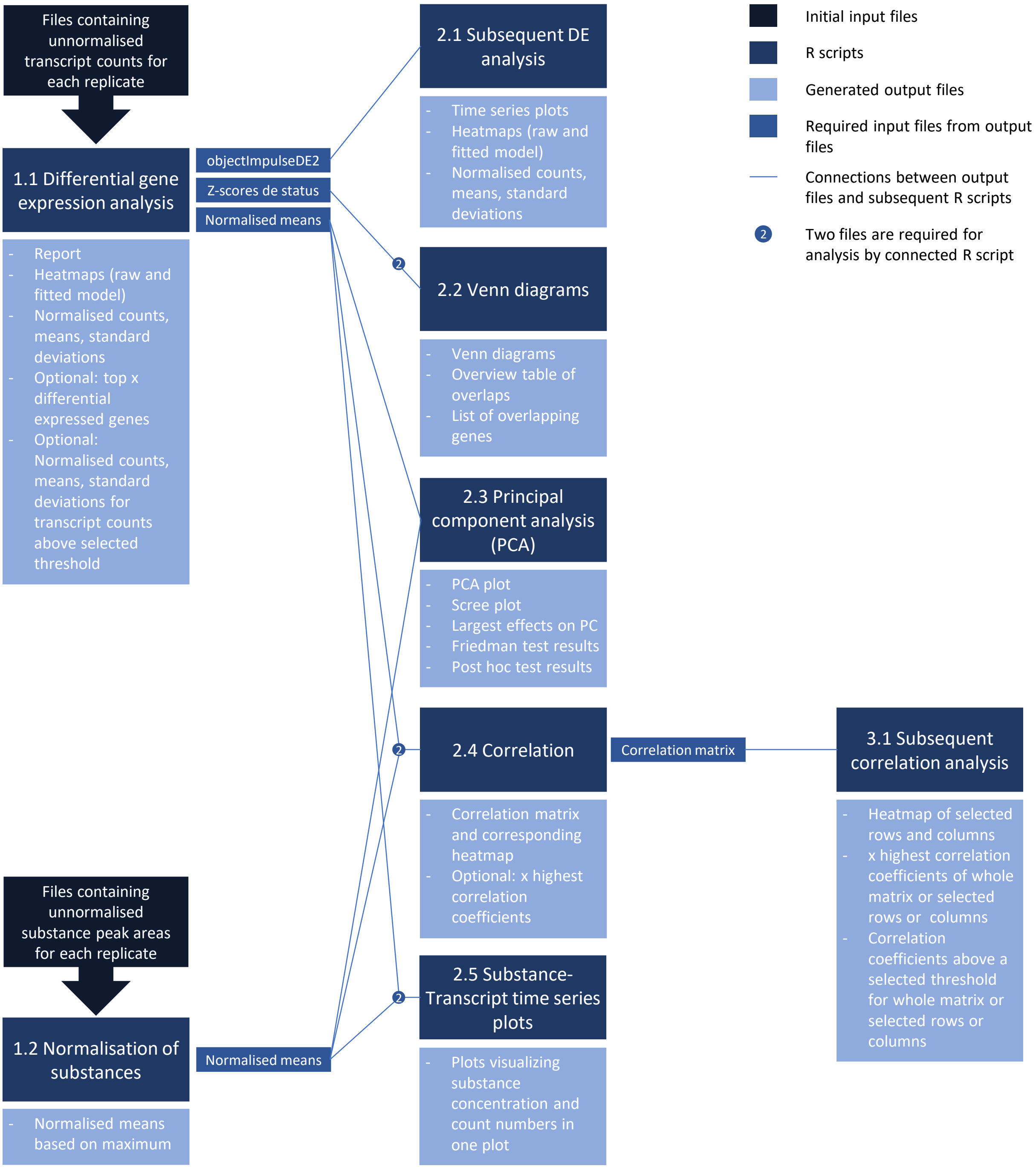
The generated time series plots are saved in PNG files. The values displayed in each plot are saved in corresponding CSV files. If a selected gene does not show any transcripts at any timepoint, a TXT file instead of a time series plot will be generated, stating that the selected gene shows no transcripts.

Subsequent analysis of coherences between genes and substances

Whereas script "2.4_correlation.R" provides the opportunity to have a look at the correlations of entire data sets, the script "3.1_subsequent_correlation_analysis.R" provides the opportunity to look at selected correlations which are of interest for the user.

The CSV file correlation matrix generated by the correlation script serves as input for this script.

The script for subsequent correlation analyses offers a high flexibility aiming to provide the user with the results that best suit the needs. It is possible to extract the x highest correlation coefficients and/or correlation coefficients above a certain chosen threshold not only from the whole matrix, but also from selected rows and/or columns. Although Cohen (1988) classified $r = 0.50$ as large effect, it might be wiser to start with a threshold of $r = 0.70$ or even $r = 0.80$ in large data sets. Depending on the structure of the correlation matrix, the user defines the names of selected rows and columns, reflecting selected substances and/or genes. The produced data frames are saved in CSV files. If both selected rows and columns are provided by the user, a heatmap visualising the correlation coefficients of this subset is additionally generated, saved in PNG format and the correlation coefficients of this subset are saved in a CSV file.


Files containing unnormalised transcript counts for each replicate
1.1 Differential gene expression analysis
- Report
- Heatmaps (raw and fitted model)
- Normalised counts, means, standard deviations
- Optional: top x differential expressed genes
- Optional: Normalised counts, means, standard deviations for transcript counts above selected threshold
objectImpulseDE2
Z-scores de status
Normalised means
Files containing unnormalised substance peak areas for each replicate
1.2 Normalisation of substances
- Normalised means based on maximum
Normalised means
2.1 Subsequent DE analysis
- Time series plots
- Heatmaps (raw and fitted model)
- Normalised counts, means, standard deviations
2
2.2 Venn diagrams
- Venn diagrams
- Overview table of overlaps
- List of overlapping genes
2.3 Principal component analysis (PCA)
- PCA plot
- Scree plot
- Largest effects on PC
- Friedman test results
- Post hoc test results
2
2.4 Correlation
- Correlation matrix and corresponding heatmap
- Optional: x highest correlation coefficients
Correlation matrix
2
2.5 Substance-Transcript time series plots
- Plots visualizing substance concentration and count numbers in one plot
3.1 Subsequent correlation analysis
- Heatmap of selected rows and columns
- x highest correlation coefficients of whole matrix or selected rows or columns
- Correlation coefficients above a selected threshold for whole matrix or selected rows or columns
Initial input files
R scripts
Generated output files
Required input files from output files
Connections between output files and subsequent R scripts
2
Two files are required for analysis by connected R script

Flowchart diagram for the developed R scripts. The diagram shows the initial input data (dark blue with arrow), the R scripts (dark blue), the files generated by each script (light blue) and the generated files that serve as input and are required for subsequent R scripts (medium blue). The dependencies are visualised by lines between the required output files and the subsequent script. Scripts that require two input files from the output of preceding scripts are marked with a small circle and a "2".

Input for developed scripts

In the following the user inputs are listed in tabular form for each script.

Table A1: User input required for "1.1_differential_gene_expression_analysis.R"

Variable	User input
replicate_file_x	Import CSV files for each replicate containing time series data.
data	List all replicate_file_x, required for automated processing.
TimeCateg	Specify the names of each timepoint in a character vector.
Time	Specify the names of each timepoint in a numeric vector.
number_of_top_genes	OPTIONAL: Write a number of how many top differentially expressed genes time series plots are requested.
show_fitted_model_in_plot	OPTIONAL: Boolean value required. FALSE for not showing the line of the impulse fitted model, TRUE for showing the line.
colour_fitted_model	OPTIONAL: Write the name of a colour for the line of the impulse fitted model or NULL, if you chose FALSE for "show_fitted_model_in_plot".
show_normalised_count_means	OPTIONAL: Boolean value required. FALSE for not showing the line of the normalised count means, TRUE for showing the line.
colour_normalised_count _means	OPTIONAL: Write the name of a colour for the line of the normalised count means or NULL, if you chose FALSE for "show_normalised_count_means".
show_normalised_counts	OPTIONAL: Boolean value required. FALSE for not showing points for the normalised counts, TRUE for showing the points.

colour_normalised_counts	OPTIONAL: Write the name of a colour for the points of the normalised counts or NULL, if you chose FALSE for "show_normalised_counts".
time_series_plot_axis_name_label_size	OPTIONAL: Set the size for axis names and labels in time series plots. Recommended: numbers between 10 and 20.
time_series_plot_legend_label_size	OPTIONAL: Set the size for legend labels in time series plots. Recommended: numbers between 10 and 20.
show_TimeCateg_on_x_axis	OPTIONAL: Boolean value required. FALSE for showing numbers on the x-axis of time series plots, TRUE for showing the names provided with TimeCateg.
transcript_threshold	OPTIONAL: Set a number as threshold for transcript counts. Additional CSV files containing only transcript counts above the threshold will be generated.
output_name	Define an output name. The name should be as short as possible, all relevant information about each result is automatically stored in the output name.

Table A2: User input required for "1.2_normalisation_of_substances.R"

Variable	User input
replicate_x	Import XLSX files for each replicate containing time series data.
data	List all replicate_x, required for automated processing.
TimeCateg	Specify the names of each timepoint in a character vector.
Time	Specify the names of each timepoint in a numeric vector.
output_name	Define an output name. The name should be as short as possible, all relevant information about each result is automatically stored in the output name.
selected_Time	OPTIONAL: Means of selected timepoints can be saved in an additional CSV file. Requires a numeric vector with selected timepoints.
output_name_selected_timepoints	OPTIONAL: Define an output name for the additional file containing the selected timepoints. The name should be as short as possible, all relevant information about each result is automatically stored in the output name.

Table A3: User input required for "2.1_subsequent_differential_gene_expression_analysis.R"

Variable	User input
objectImpulseDE2	Import RDS files generated by the differential gene expression analysis script.
TimeCateg	Specify the names of each timepoint in a character vector.
selected_genes	OPTIONAL: Requires a character vector containing names of genes that should be visualised.
heatmap_row_title_font_size	OPTIONAL: Set the size for the row title in the heat map. Recommended: a number between 10 and 30.
heatmap_row_column_name_font_size	OPTIONAL: Set the size for row and column names in the heat map. Recommended: a number between 10 and 30.
legend_title_font_size	OPTIONAL: Set the size for legend title in the heat map. Recommended: a number between 10 and 30.
legend_label_font_size	OPTIONAL: Set the size for legend lable in the heat map. Recommended: a number between 10 and 30.
legend_height	OPTIONAL: Set the size of the legend bar. Recommended: a number between 100 and 200.
show_fitted_model_in_plot	OPTIONAL: Boolean value required. FALSE for not showing the line of the impulse fitted model, TRUE for showing the line.
colour_fitted_model	OPTIONAL: Write the name of a colour for the line of the impulse fitted model or NULL, if you chose FALSE for "show_fitted_model_in_plot".

show_normalised_count_means	OPTIONAL: Boolean value required. FALSE for not showing the line of the normalised count means, TRUE for showing the line.
colour_normalised_count _means	OPTIONAL: Write the name of a colour for the line of the normalised count means or NULL, if you chose FALSE for "show_normalised_count_means".
show_normalised_counts	OPTIONAL: Boolean value required. FALSE for not showing points for the normalised counts, TRUE for showing the points.
colour_normalised_counts	OPTIONAL: Write the name of a colour for the points of the normalised counts or NULL, if you chose FALSE for "show_normalised_counts".
time_series_plot_axis_name_label_size	OPTIONAL: Set the size for axis names and labels in time series plots. Recommended: numbers between 10 and 20.
time_series_plot_legend_label_size	OPTIONAL: Set the size for legend labels in time series plots. Recommended: numbers between 10 and 20.
show_TimeCateg_on_x_axis	OPTIONAL: Boolean value required. FALSE for showing numbers on the x-axis of time series plots, TRUE for showing the names provided with TimeCateg.
output_name	Define an output name. The name should be as short as possible, all relevant information about each result is automatically stored in the output name.

Table A4: User input required for "2.2_venn_diagrams.R"

Variable	User input
input_file_1	Import first z-score CSV file generated by the differential gene expression analysis script.
category_name_1	Specify a name for the first data set.
category_name_1_colour	Write the name of a for the first data set displayed in the Venn diagrams.
input_file_2	Import second z-score CSV file generated by the differential gene expression analysis script.
category_name_2	Specify a name for the second data set.
category_name_2_colour	Write the name of a for the second data set displayed in the Venn diagrams.
print_mode	Choose between "raw" for absolute numbers or "percent" for percent displayed in the Venn diagrams.
sclaing	Boolean value required. Write TRUE, if the circles displayed in the Venn diagram should be scaled corresponding to the numbers they represent, write FALSE if they should all have the same size.
category_name_font_size	OPTIONAL: Set the size for the names displayed in the Venn diagrams. Recommended: a number between 1 and 2.
numbers_font_size	OPTIONAL: Set the size for the numbers displayed in the Venn diagrams. Recommended: a number between 1 and 3.
category_name_fontface	OPTIONAL: Choose between "bold" or "plain" for the font face of the names displayed in the Venn diagrams.

numbers_fontface	OPTIONAL: Choose between "bold" or "plain" for the font face of the numbers displayed in the Venn diagrams.
output_name	Define an output name. The name should be as short as possible, all relevant information about each result is automatically stored in the output name.

Table A5: User input required for "2.3_principal_component_analysis.R"

Variable	User input
input_file	Import CSV file containing a time series, e.g. a file with transcriptome means generated by the differential gene expression analysis script or a file with substance means generated by the normalisation of substances script.
largest_effect_on_PC	Write a number of how many genes/substances with the largest effect on the principal components should be stored in a CSV file.
scree_plot_axis_tick_size	OPTIONAL: Set the size for the axis labels in the scree plot or NULL, if you want to use the default. Recommended: a number between 1 and 2.
scree_plot_axis_name_size	OPTIONAL: Set the size for the axis names in the scree plot or NULL, if you want to use the default. Recommended: a number between 1 and 2.
scree_plot_bar_colour	OPTIONAL: Write the name of a colour for the bar plot or NULL, if you want to use the default.
pca_plot_axis_name_label_size	OPTIONAL:Set the size for the axis names and axis labels in the PCA plot or NULL, if you want to use the default. Recommended: a number between 10 and 20.
pca_plot_stages _label_size	OPTIONAL:Set the size for the names of the stages displayed in the PCA plot or NULL, if you want to use the default. Recommended: start with 4 and adjust the number depending on your names.
output_name	Define an output name. The name should be as short as possible, all relevant information about each result is automatically stored in the output name.

Table A6: User input required for "2.4_correlation.R"

Variable	User input
input_file_1	Import a first CSV file containing a time series, e.g. a file with transcriptome means generated by the differential gene expression analysis script or a file with substance means generated by the normalisation of substances script.
input_file_1_data_type	Write down a name for the first data set. This will be the row title displayed in the heat map.
input_file_2	Import a second CSV file containing a time series, e.g. a file with transcriptome means generated by the differential gene expression analysis script or a file with substance means generated by the normalisation of substances script. It can be the same as input_file_1
input_file_2_data_type	Write down a name for the second data set. This will be the column title displayed in the heat map.
method	Decide on a correlation method for calculating th correlation coefficients. Choose between "pearson", "kendall" and "spearman". For non normally distributed data, which is often the case for time series data, "spearman" is recommended.
number_of_highest_correlation _coefficients	OPTIONAL: Write a number of how many highest correlation coefficients should be stored in a CSV file.
heatmap_row_title_font_size	OPTIONAL: Set the size for the row title in the heat map. Recommended: a number between 10 and 30.

legend_title_font_size	OPTIONAL: Set the size for legend title in the heat map. Recommended: a number between 10 and 30.
legend_label_font_size	OPTIONAL: Set the size for legend lable in the heat map. Recommended: a number between 10 and 30.
legend_height	OPTIONAL: Set the size of the legend bar. Recommended: a number between 100 and 200.
output_name	Define an output name. The name should be as short as possible, all relevant information about each result is automatically stored in the output name.

Table A7: User input required for "2.5_substance_transcript_time_series_plots.R"

Variable	User input
substances	Import a CSV file containing time series data of substances. The concentrations should be stated in percent.
transcriptome_file_1	Import a first CSV file containing time series data of a transcriptome. It has to show the same amount of timepoints as the substance time series data.
transcriptome_file_2	OPTIONAL: Import a second CSV file containing time series data of a transcriptome. This data set is allowed to show a differing amount of timepoints as the substance time series data.
transcriptome_name_1	Write down a name for the first transcriptome data set. This will be displayed in the legend of the time series plot.
transcriptome_name_2	Write down a name for the second transcriptome data set. This will be displayed in the legend of the time series plot.
TimeCateg	Specify the names of each timepoint in a character vector.
show_TimeCateg_on_x_axis	OPTIONAL: Boolean value required. FALSE for showing numbers on the x-axis of time series plots, TRUE for showing the names provided with TimeCateg.
selected_substances	Requires a character vector containing names of substances that should be plotted.
substance_plot_colour	Write the name of a for the line of the substance concentration displayed in the time series plot.

selected_genes	Requires a character vector containing names of genes that should be plotted.
gene_plot _colour	Write the name of a for the line of the transcript counts displayed in the time series plot.
output_name	Define an output name. The name should be as short as possible, all relevant information about each result is automatically stored in the output name.

Table A8: User input required for "3.1_subsequent_correlation_analysis.R"

Variable	User input
input_file	Import a CSV file containing a correlation matrix calculated by the correlation script.
data_type_row	Write down a name for the data comprised in the rows of the correlation matrix. This will be the row title displayed in the heat map.
data_type_col	Write down a name for the data comprised in the columns of the correlation matrix. This will be the column title displayed in the heat map.
selected_rows	OPTIONAL: Requires a character vector containing names of selected rows that should be further evaluated.
selected_cols	OPTIONAL: Requires a character vector containing names of selected columns that should be further evaluated.
heatmap_row_column_title_font_size	OPTIONAL: Set the size for the row title in the heat map. Recommended: a number between 10 and 30.
heatmap_row_column_name_font_size	OPTIONAL: Set the size for row and column names in the heat map. Recommended: a number between 10 and 30.
legend_title_font_size	OPTIONAL: Set the size for legend title in the heat map. Recommended: a number between 10 and 30.
legend_label_font_size	OPTIONAL: Set the size for legend lable in the heat map. Recommended: a number between 10 and 30.
legend_height	OPTIONAL: Set the size of the legend bar. Recommended: a number between 100 and 200.

number_of_highest_correlation_coefficients	OPTIONAL: Write a number of how many highest correlation coefficients of the whole matrix should be stored in a CSV file.
number_of_highest_correlation_coefficients_selected	OPTIONAL: Write a number of how many highest correlation coefficients of the selected rows and/or columns should be stored in a CSV file.
threshold	OPTIONAL: Write a number of where to set a threshold for correlation coefficients of the whole matrix which should be stored in a CSV file.
threshold_selected	OPTIONAL: Write a number of where to set a threshold for correlation coefficients of the selected rows and/or columns which should be stored in a CSV file.
output_name	Define an output name. The name should be as short as possible, all relevant information about each result is automatically stored in the output name.

Bibliography

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed). Hillsdale, N.J: L. Erlbaum Associates.

Galili, T. (2010). Post hoc analysis for Friedman's Test (r code). Blog. Retrieved May 28, 2019, from https://www.r-statistics.com/2010/02/post-hoc-analysis-for-friedmans-test-r-code/

Hollander, M. & Wolfe, D. A. (1999). *Nonparametric statistical methods* (2nd ed). Wiley series in probability and statistics. New York: Wiley.