

## Script Documentation: **exclusive\_functions.R**

This script analyzes exclusive functions from a specified input file, calculates their average abundance, and generates a bar plot of the top expressed functions. It is designed to be run from the command line using Rscript.

### 1. Usage

```
Rscript exclusive_functions.R <input_file> <dataset_name> <sample_type> <top_functions>
                                <colour_graph>
```

### 2. Arguments

**<input\_file>**: The file contains the functional data for analysis. The file should be in a table format, with the first column containing function names and the remaining columns containing quantification values. This input file is the output of the `comparative_statistical_analysis.R`.

**<dataset\_name>**: A descriptive name for the dataset being analyzed. This can be the name of the organism (e.g., "Human," "Ruminants") or experimental conditions (e.g., "increase\_RME," "decrease\_RME").

**<sample\_type>**: The type of biological sample (e.g., "Mags," "Genes," "miRNA") that the data represents.

**<top\_functions>**: An integer indicating the number of top functions to display in the results. The value must be less than or equal to the number of functions available in the input file.

**<colour\_graph>**: The color to be used for the graph, which can be specified by name (e.g., "blue") or hex code (e.g., "#FF5733").

### 3. Help

To display the help message, run the script with the `--help` argument:

```
Rscript exclusive_functions.R --help
```

### 4. Requirements

R Packages: The script requires the following R libraries:

- **ggplot2**: for plotting.

```
install.packages("ggplot2")
```

- **scales**: for scaling options in the plot.

```
install.packages("scales")
```

- **stringr**: for string manipulation in the plot.

```
install.packages("stringr")
```

## 5. How it works

**a. Command-line Help:** If the script is run with the `--help` argument, it will display usage instructions and exit.

**b. Data Input:** The input file is read into a data frame. The first column is treated as function names, and the subsequent columns represent quantifications of those functions across samples.

**c. Mean Abundance Calculation:** The average abundance of each function across samples is computed, and the top `n` functions (specified by `<top_functions>`) are selected based on their mean abundance.

**e. Plot Creation:**

The script uses `ggplot2` to create a horizontal bar plot of the top functions.

The plot title includes the dataset name, sample type, and the number of top functions.

The x-axis labels the functions, and the y-axis represents their average abundance.

Functions are displayed in decreasing order of abundance.

**f. Plot Saving:** The plot is saved as a `.png` file. The filename includes information from the input file (dataset name, sample type, and function classification level).

**g. Completion Message:** Once the analysis is complete, a message is printed to the console indicating that the functional analysis of exclusive pathways for the specified organism or experimental conditions and sample type is finished.

## 6. Error Handling

The script checks that exactly five arguments are provided. If not, an error message is displayed indicating the correct usage.

If the input file contains no rows, the script stops and notifies that there are no exclusive pathways for the specified organism.

If the requested number of top functions exceeds the number available in the data, an error message is shown.

## 7. Output

The output of the script includes:

A `.png` file containing the bar plot of the top functions, named in the format:

`Top_<top_functions>_exclusive_functions_<dataset_name>_<sample_type>_<level>.png`