Script Documentation: `analyse_by_dataset.R`

This script performs a functional analysis of biological datasets and generates a plot of the top most abundant functions based on user-specified thresholds. The analysis filters functions by abundance and prevalence, and the resulting plot visualizes the top expressed functions in the dataset. The script is designed to be run from the command line with user input for customizable analysis.

1. **Usage**

Rscript analyse_by_dataset.R <input_file> <dataset_name> <sample_type> <abundance_threshold> <prevalence_threshold> <top_functions> <colour_graph>

2. **Parameters**

<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 script process_data.ssh.

<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.

<abundance_threshold>: A numeric value representing the minimum abundance required for a function to be considered "expressed." This value must be greater than or equal to zero.

<prevalence_threshold>: A numeric value between 0 and 1 that sets the minimum proportion of samples in which a function must be expressed to be included in the analysis. This threshold determines function prevalence.

<top_functions>: The number of top functions to display in the plot, based on their average abundance across samples.

<colour_graph>: A string representing the color to be used for the bars in the plot. This can be either a named color (e.g., "blue") or a hex code (e.g., `'#FF5733'`).

3. **Help**

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

Rscript analyse_by_dataset.R --help

4. **How it works**

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

b. Argument Parsing: The script checks that all required arguments are provided. If the number of arguments is incorrect or any thresholds are out of bounds (e.g., prevalence not between 0 and 1), it stops execution and outputs an error message.

c. 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. The last column of the input file is excluded.

d. Filtering by Thresholds:

Functions with abundance values above the specified threshold are marked as "expressed."

The percentage of samples in which each function is expressed is calculated.

Functions that meet or exceed the prevalence threshold are selected for further analysis.

e. 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.

f. 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.

g. 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).

h. Completion Message: Once the analysis and plotting are complete, a message is displayed, indicating the analysis is finished.

5. Example

Rscript analyse_by_dataset.R input_data.txt Ruminants Mags 0.05 0.6 20 "#FF5733"

This example processes the dataset `input_data.txt`, analyzing "Ruminants" for the sample type "Mags." It filters functions with an abundance threshold of 0.05 and a prevalence threshold of 0.6, displaying the top 20 functions in a plot with a color specified by the hex code `#FF5733`.

6. 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")
```

7. **Error Handling**

Incorrect number of arguments: The script will stop and display a message if the required arguments are not provided.

Invalid abundance threshold: If the abundance threshold is negative, the script stops and prompts the user for a valid threshold.

Invalid prevalence threshold: The script ensures that the prevalence threshold is between 0 and 1. If it falls outside this range, execution is halted, and the user is asked to provide a valid value.

Too few functions: If the number of functions meeting the criteria is less than the specified number of top functions, the script will stop and alert the user to choose a smaller number of top functions.

8. **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>_functions_<dataset_name>_<sample_type>_<level>.png