Documentation: `process_data.sh`

`process_data.sh` is a shell script that processes subsystem-level data obtained from Superfocus. It supports two data processing methods: counts (raw counts normalized) and relab (relative abundance). The script accepts user inputs for subsystem levels, data type, sample type, and dataset name. Based on these inputs, it processes the data and outputs cleaned and formatted results.

1. **Usage**

./process_data.sh <subsystem_level_X.xls> <counts|relab> <sample_type> <dataset_name>

2. **Arguments**

<subsystem_level_X.xls>: The input file containing subsystem information, output from Superfocus. The file name should include 'level_1', 'level_2', or 'level_3'.

<counts|relab>: The method for processing the data:

counts: Processes the data as raw counts. Counts based on the number of hits of each input file.

relab: Processes the data as relative abundance.

<sample_type>: The type of sample (e.g., 'Mags', 'Genes', 'miRNA').

<dataset_name>: The name of the dataset or the condition being analyzed (e.g., 'Human', 'Ruminants', 'decrease_RME', 'increase_RME').

3. **Example Command**

./process_data.sh subsystem_level_1.xls counts Mags Ruminants

This command processes a file named `subsystem_level_1.xls` with the counts method, for sample type `Mags` and dataset `Ruminants`.

4. **Options**

--help: To display the usage instructions:

./process_data.sh --help

5. **Script Flow**

a. **File and Parameter Checks:**

The script checks if the input file exists.

It extracts the subsystem level from the file name (level_1, level_2, or level_3). If the level is not found in the file name, the script exits with an error.

b. **Temporary Directory Creation:**

A temporary directory is created for intermediate files, named `Temp/${level}_${sample_type}_${dataset_name}`.

c. Processing by Method:

Relative Abundance (`relab`):

Copies transposed data and filters for columns with percentage data.
Cleans the output by removing percentage symbols.
Transposes the cleaned data and saves the final output.

Counts (`counts`):

Copies transposed data and filters for non-percentage columns.
Removes unnecessary rows and transposes the remaining data.
Saves the final output as counts.

d. Output Files:

For `relab` method: `${level}_${sample_type}_${dataset_name}_relab.txt`.

Example Output

A file named `level_1_Mags_Ruminants_relab.txt` is created containing the processed relative abundance data.

For `counts` method: `${level}_${sample_type}_${dataset_name}_counts.txt`.

Example Output

A file named `level_1_Mags_Ruminants_counts.txt` is created containing the processed raw counts data.

e. Clean Up (optional):

The script can optionally remove intermediate files stored in the `Temp` directory.

6. Error Handling

If the input file is not found, an error message is displayed.

If the file name does not contain 'level_1', 'level_2', or 'level_3', an error message is displayed.

If an invalid processing method (`counts` or `relab`) is specified, the script exits with an error.

7. Conclusion

The `process_data.sh` script is a robust tool for processing subsystem-level data in either counts or relative abundance formats, enabling streamlined data handling for various types of biological samples and datasets.