3 Script Language and Data Format

The tool hereby detailed is a script written in the R programming language. The R version used in the development was 3.4 and the following R packages are required:

- `fields` (version 1.1-1) to compute the Euclidean distance matrices;
- `SpecsVerification` (version 0.5-2) to perform statistical tests related to the deviation from flatness of a rank histogram;
- `xtable` (version 1.7-4) and `latex2exp` (version 0.4.0) to export results into the latex document format;
- `reshape2` (version 1.4) for conversion between long and wide data format;
- `ggplot2` (version 2.2.1) and `gridExtra` (version 2.3) for graphics rendering;
- `goftest` (version 1.1-1) to perform Watson hypothesis test;
- `lubridate` (version 1.7.1) to deal with dates.

This tool consists of a main folder, called "`main_code`", which contains the relevant R scripts and a configuration file (Figure 23):

- `main_functions.R` is a script which contains all the functions to compute the proposed metrics, developed to evaluate the quality of the MCLA model.
- `run_code.R` is an auxiliar R script which runs `main_functions.R` and saves the results according to the definitions in `config_file.txt`.
- `config_file.txt` is a configuration file which may be edited to change such definitions as the name of the output folder and metrics to be computed.

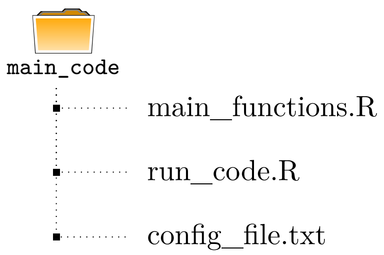


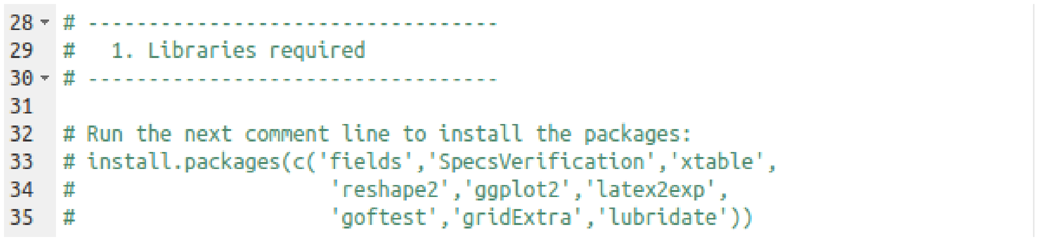
Figure 23: Schematic representation of the tool

Remarks:

- The required packages mentioned above are automatically installed on the first run, if not installed already. In any case, you might want to run the following described command to ensure the latest versions of the packages are installed:

– `main_functions.R` contains three commented lines (see Figure 24) which should be uncommented in the first run, in order to install the necessary R packages. Alternatively, the user can copy and run the following command:

```
install.packages(c('fields','SpecsVerification','xtable',
'reshape2','ggplot2','latex2exp','goftest','gridExtra',
'lubridate'))
```



```
# ---------------------------------
#   1. Libraries required
# ---------------------------------

# Run the next comment line to install the packages:
# install.packages(c('fields','SpecsVerification','xtable',
#                      'reshape2','ggplot2','latex2exp',
#                      'goftest','gridExtra','lubridate'))
```

Figure 24: Commands to uncomment in the first run.

- In order to avoid format errors, the `config_file.txt` should be edited and saved using a text editor.

3.1 Input data

3.1.a) Analyzed data. For each specific timestamp YYYY-mm-dd HH:MM, a "`.csv`" file called `WF_YYYYmmdd_HHMM.csv` is used. This CSV file is obtained using the MATLAB script developed by INESC TEC and is organized as follows:

1. The first column identifies the state ID where:

    - **State -2** identifies the line in the data with the permanent maximum limit values of each recorded pre-contingency operating condition ("Imax", in A, for electric currents and "Smax", in MVA, for active power flows).
    - **State -1** identifies the line in the data with the SN pre-contingency load-flow results.
    - **State 0** identifies the line in the data with the DACF pre-contingency load-flow results.
    - **The remaining states** (state 1, state 2, ...) identify the ensemble members pre-contingency load-flow results [the ensemble members are the MCLA states created for the DACF base case]. Note that this states goes from State 1 to State m-1, with m being the number of ensemble forecasts provided from the MCLA, including the DACF base case.

2. The remaining columns identify the recorded pre-contingency operating conditions, which can be the active power flow (P in MW) or the electric current (I in A), for each pre-selected transmission line.

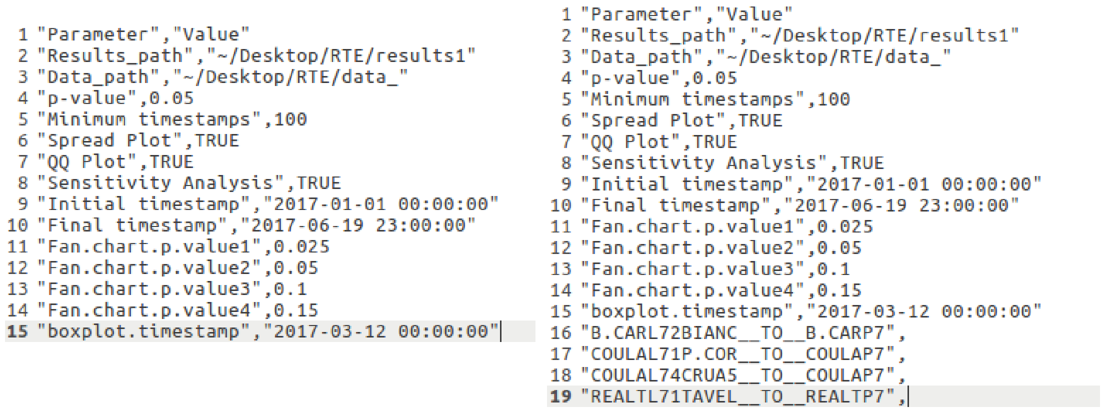
Note 1: For each transmission line, the file only comprises the active power flow and the electric current for one substation bus of the transmission line. This will avoid performing a statistical analyzing for operating conditions that usually have similar values.
Note 2: The code name used for these operating conditions is the one obtained from the files generated by the iPST online platform.
Note 3: The maximum permanent limit for active power flows ("Smax", in MVA) is calculated from the maximum permanent limit for the electric current ("Imax", in A) and by assuming the line nominal voltage ("Un", in kV). Namely, by assuming that Smax = sqrt(3)×Un×Imax/1000.

3.1.b) Configuration data. The relevant configurations are defined in the `config_file.txt` file. In this file, the following parameters may be changed:

- `results_path:` path to save all the results.
- `data_path:` path with the input data to be analyzed.
- `p-value:` quantiles definition of section 2.2.4, **typical value=0.05.**
- `Minimum timestamps:` minimum number of measures to include each variable in the analysis, **typical value=100.**
- `Spread Plot:` boolean value (`TRUE` or `FALSE`) which indicates if the spread plot described in section 2.2.6 is computed and saved, **typical value=`TRUE`.**
- `QQ Plot:` boolean value (`TRUE` or `FALSE`) which indicates if the QQ-plot described in section 2.2.6 is computed and saved, **typical value=`TRUE`.**
- `Sensitivity Analysis:` boolean value (`TRUE` or `FALSE`) which indicates if the sensitivity analysis described in section 2.2.9 is computed and saved, **typical value=`TRUE`.**
- `Initial timestamp:` initial timestamp of the analysis (in format YYYY-mm-dd HH:MM:00) which can include or not the input data timestamps.
- `Final timestamp:` final timestamp of the analysis (in format YYYY-mm-dd HH:MM:00) which can include or not the input data timestamps.
- `Fan.chart.p.valuei:` defines the four values to use for the Fan chart (first plot ilustrated from Figure 16 to Figure 19, (in section 2.2.10), **typical value ={0.025;0.05;0.1;0.15}.**
- `Boxplot.timestamp:` defines the chosen timestamp for the boxplots described in section 2.2.11 (in format YYYY-mm-dd HH:MM:00).

Finally, if the user wants to **perform the analysis just for a specific set of branches**, this can be performed by adding the code name of the branches in the last lines of the `config_file.txt` (see Figure 25), **by default all the branches are analyzed.**

Example: If the user wants to analize the branches whose code name in the input files is `COULAL71P.COR__TO__COULAP7_I` and `COULAL74CRUA5__TO__COULAP7_I` then these code names must be included in the `config_file.txt` as ilustrated in Figure 25.



```
"Parameter","Value"
"Results_path","~/Desktop/RTE/results1"
"Data_path","~/Desktop/RTE/data_"
"p-value",0.05
"Minimum timestamps",100
"Spread Plot",TRUE
"QQ Plot",TRUE
"Sensitivity Analysis",TRUE
"Initial timestamp","2017-01-01 00:00:00"
"Final timestamp","2017-06-19 23:00:00"
"Fan.chart.p.value1",0.025
"Fan.chart.p.value2",0.05
"Fan.chart.p.value3",0.1
"Fan.chart.p.value4",0.15
"boxplot.timestamp","2017-03-12 00:00:00"
```

(a) Analysis performed for all branches.

```
"Parameter","Value"
"Results_path","~/Desktop/RTE/results1"
"Data_path","~/Desktop/RTE/data_"
"p-value",0.05
"Minimum timestamps",100
"Spread Plot",TRUE
"QQ Plot",TRUE
"Sensitivity Analysis",TRUE
"Initial timestamp","2017-01-01 00:00:00"
"Final timestamp","2017-06-19 23:00:00"
"Fan.chart.p.value1",0.025
"Fan.chart.p.value2",0.05
"Fan.chart.p.value3",0.1
"Fan.chart.p.value4",0.15
"boxplot.timestamp","2017-03-12 00:00:00"
"B.CARL72BIANC__TO__B.CARP7",
"COULAL71P.COR__TO__COULAP7",
"COULAL74CRUA5__TO__COULAP7",
"REALTL71TAVEL__TO__REALTP7",
```

(b) Analysis performed for a selected set of branches.

Figure 25: `config_file.txt` ilustration with and without variable selection.

About `data_path`. The "`.csv`" named `WF_YYYYmmdd_HHMM.csv` must be in a common folder and the code must receive the path for this common folder. **The R script considers all the "`.csv`" files inside the `data_path` folder as input data.** Figure 26 illustrates situations supported by the code.

3.2 Output data

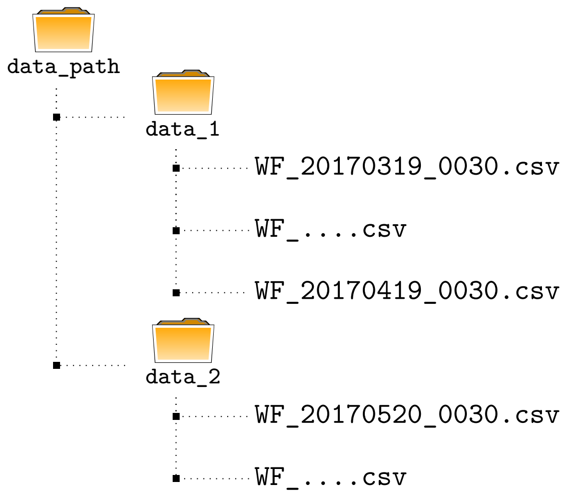
As mentioned before, the output of the script is saved in the `results_path` defined by the user. Figure 27 illustrates the structure of this folder and the possible outputs, according to the configuration file.

3.3 First Use of the Code

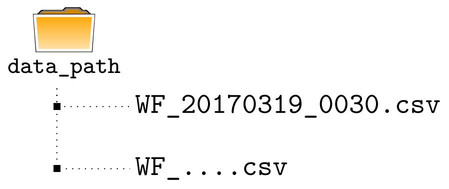
In order to use this tool it is essential to have the `main_functions.R`, `run_code.R` and `config_file.txt` files in the same folder, named `path_tool`.

On the first use, the user needs to follow the following steps:

1. change the `path_tool` in `run_code.R`
2. change the `data_path` and `results_path` in `config_file.txt`
3. upgrade the required packages if there is an error related to the packages versions (see the above remarks in section 3).



(a) Data organized by sub-folders.

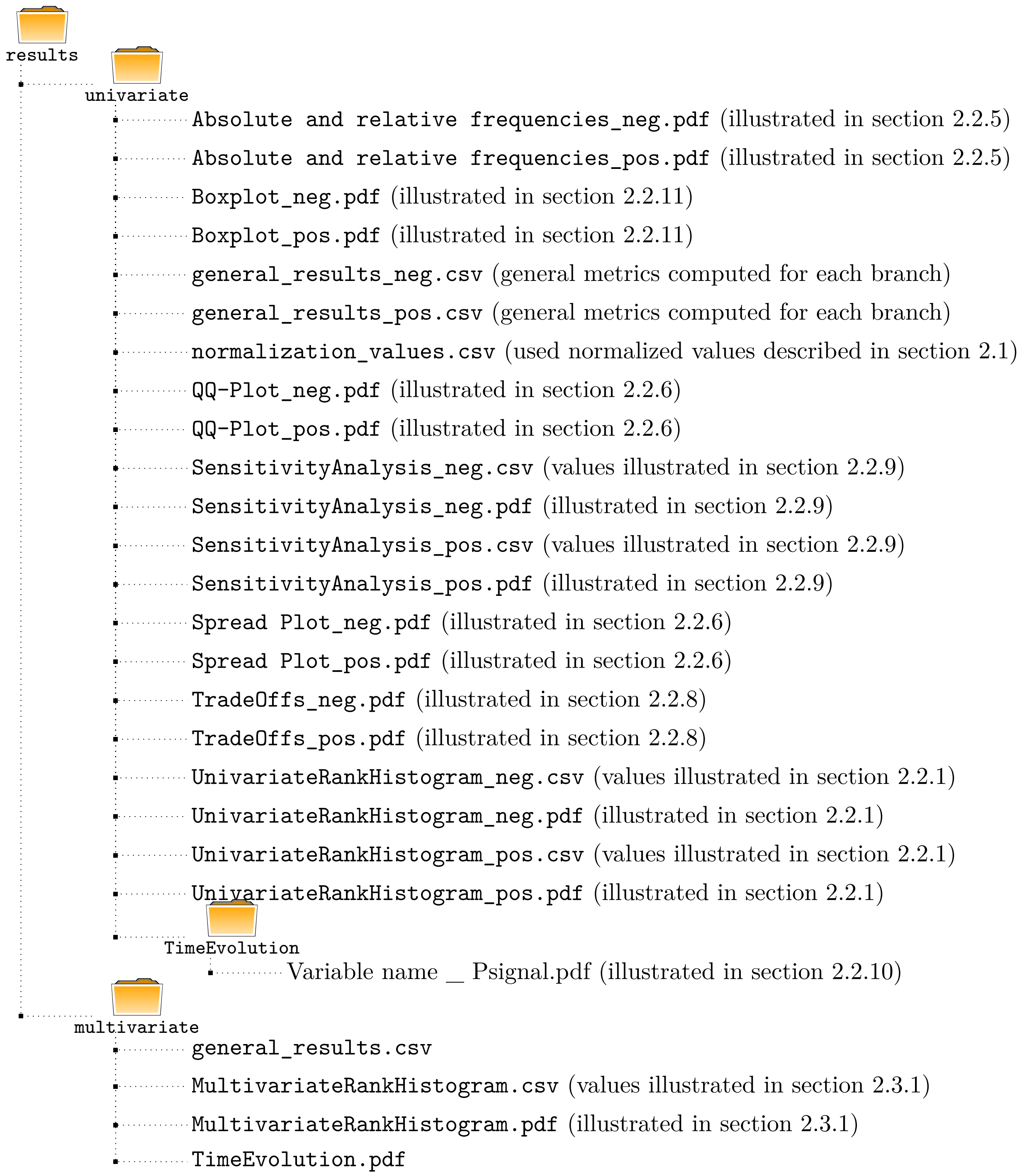


(b) Data without sub-folders.

Figure 26: Input data folder ilustration.

After these procedures, the user just needs to execute the file `run_code.R` and all the results will be generated and saved, according to the specified parameters.

The tool runs in Linux or Windows operating systems. However, if Windows operating system is being used, in path definition it's mandatory to replace '\' by '/'.



```
results
├── univariate
│   ├── Absolute and relative frequencies_neg.pdf (illustrated in section 2.2.5)
│   ├── Absolute and relative frequencies_pos.pdf (illustrated in section 2.2.5)
│   ├── Boxplot_neg.pdf (illustrated in section 2.2.11)
│   ├── Boxplot_pos.pdf (illustrated in section 2.2.11)
│   ├── general_results_neg.csv (general metrics computed for each branch)
│   ├── general_results_pos.csv (general metrics computed for each branch)
│   ├── normalization_values.csv (used normalized values described in section 2.1)
│   ├── QQ-Plot_neg.pdf (illustrated in section 2.2.6)
│   ├── QQ-Plot_pos.pdf (illustrated in section 2.2.6)
│   ├── SensitivityAnalysis_neg.csv (values illustrated in section 2.2.9)
│   ├── SensitivityAnalysis_neg.pdf (illustrated in section 2.2.9)
│   ├── SensitivityAnalysis_pos.csv (values illustrated in section 2.2.9)
│   ├── SensitivityAnalysis_pos.pdf (illustrated in section 2.2.9)
│   ├── Spread Plot_neg.pdf (illustrated in section 2.2.6)
│   ├── Spread Plot_pos.pdf (illustrated in section 2.2.6)
│   ├── TradeOffs_neg.pdf (illustrated in section 2.2.8)
│   ├── TradeOffs_pos.pdf (illustrated in section 2.2.8)
│   ├── UnivariateRankHistogram_neg.csv (values illustrated in section 2.2.1)
│   ├── UnivariateRankHistogram_neg.pdf (illustrated in section 2.2.1)
│   ├── UnivariateRankHistogram_pos.csv (values illustrated in section 2.2.1)
│   ├── UnivariateRankHistogram_pos.pdf (illustrated in section 2.2.1)
│   └── TimeEvolution
│       └── Variable name __ Psignal.pdf (illustrated in section 2.2.10)
└── multivariate
    ├── general_results.csv
    ├── MultivariateRankHistogram.csv (values illustrated in section 2.3.1)
    ├── MultivariateRankHistogram.pdf (illustrated in section 2.3.1)
    └── TimeEvolution.pdf
```

Figure 27: Schematic representation of the results folder.