

Evaluation of the uncertainty model used by the Monte Carlo Like Approach

October/2018

This document explains how to run the tool developed to evaluate the uncertainty model used by the Monte Carlo Like Approach, when applied to perform branch flow security assessment. In this tool, the consistency between branch flows probabilistic forecasts and observations and the sharpness of the uncertainty forecasts of branch flows is checked by exploiting advanced metrics that traditionally are used in weather prediction.

The following metrics are computed:

- Univariate Analysis
 - Univariate Rank Histogram
 - Discrepancy or Reliability Index - Delta Index
 - Hypothesis tests to define the confidence level of accepting a uniform rank histogram (Chi square and Watson p-values for Rank histogram)
 - Mean distances between the SN and the ensemble
 - Number of times the SN is outside/inside the ensemble
 - Mean value of ensemble spread
 - Mean value of the Continuous Ranked Probability Score (CRPS)
 - Trade-off between sharpness and calibration
 - Sensitivity analysis of the used quantile values on metrics computation
 - Metrics time evolution
 - Boxplot visualization of the ensemble uncertainty for some timestamps
 - Univariate Analysis
- Multivariate Analysis
 - Multivariate Rank Histogram
 - Energy Score (ES)
 - Metrics time evolution

1 Script Code

The tool hereby detailed is a script written in the R programming language [1]. 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;
- **doParallel** (version 1.0.11) to perform parallel computation.

This tool consists of a main folder, called “**main_code**” in Figure 1, which contains the relevant R scripts and a configuration file:

- **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.

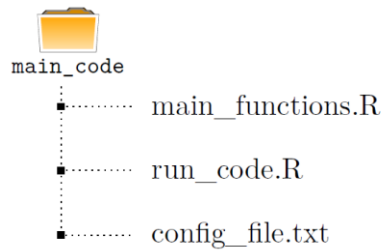


Figure 1 - Schematic representation of the tool

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 2) 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', 'gofest', 'gridExtra',
'lubridate'))
```

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

```

33 # -----
34 # 1. Libraries required
35 # -----
36
37 # Run the next comment line to install the packages:
38 # install.packages(c('fields','SpecsVerification','xtable',
39 #                    'reshape2','ggplot2','latex2exp',
40 #                    'gofest','gridExtra','lubridate','doParallel'))

```

Figure 2 - Schematic representation of the tool

2 Input data

2.1 Analyzed data

For each specific timestamp YYYY-mm-dd HH:MM, a “.csv” file named “**WF_YYYYmmdd_HHMM.csv**” is used (example: **WF_20170201_0030.csv**). This CSV file is obtained using a MATLAB script developed by INESC TEC and is organized as follows (see illustration in Figure 3):

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 loadflow results.
 - **The remaining states (state 1, state 2, ...)** identify the remaining ensemble members pre-contingency load-flow results [these 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 members provided from the MCLA, including the DACF base case.
2. The remaining columns identify the recorded steady-state 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 flow.

```
state;flow1_P;flow1_I;flow2_P;flow2_I;...;flowN_P;flowN_I
-2;1474.3216;2240;1579.6303;2400;...;1711.2662;2600
-1;482.98337;678.5826;-563.07263;810.9629;...;621.04254;885.30457
0;404.4225;567.527;-465.45856;668.78705;...;499.05396;706.25836
1;437.2018;613.6094;-526.8726;758.2563;...;574.36676;811.689
2;256.0917;359.30972;-449.7447;646.03766;...;384.2241;542.2157
...
m-1;245.29922;344.33472;-551.36334;794.31616;...;366.4391;517.8994
```

Figure 3 – Contents illustration of each “WF_YYYYmdd_HHMM.csv” file

Note 1: For each transmission line, the file should only comprise 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 i PST online platform.

Note 3: The maximum permanent limit for active power flows (“Smax”, in MVA) is assumed to be calculated from the maximum permanent limit for the electric current (“Imax”, in A) and by using the line nominal voltage (“Un”, in kV). Namely, by assuming that $S_{max} = \sqrt{3} \times U_n \times I_{max} / 1000$.

2.2 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.
- **do_parallel**¹: boolean value (TRUE or FALSE) which indicates if parallel computation is performed (use more than one CPU), typical value=TRUE.
- **Cores:** defines the number of CPU cores to use in parallel computation (ignored if do_parallel=FALSE), typical value=8. If Cores is greater than the available CPU cores then the maximum number of available cores is selected.
- **p-value:** p-value for marginal quantiles definition, **typical value=0.05**.

¹ Using an Intel processor i7-2600 CPU @ 3.40GHz × 8 cores, the execution time (for about 930 branch records and the time interval Feb-Apr 2017 = 1350 timestamps) was reduced from 30h to 6h30. The parallel performance was introduced in the time evolution plots, sensitivity analysis and spread plot computation.

- **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 plots are computed and saved, **typical value=FALSE**.
- **QQ Plot:** boolean value (TRUE or FALSE) which indicates if the QQ-plots are computed and saved, **typical value=FALSE**.
- **Sensitivity Analysis:** boolean value (TRUE or FALSE) which indicates if the sensitivity analysis is computed and saved, **typical value=FALSE**.
- **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 charts (like the one illustrated in Figure 4), **typical value ={0.025;0.05;0.1;0.15}**.
- **Boxplot.timestamp:** defines the chosen timestamp for boxplots computation (in format YYYY-mm-dd HH:MM:00), like the one illustrated in Figure 5.

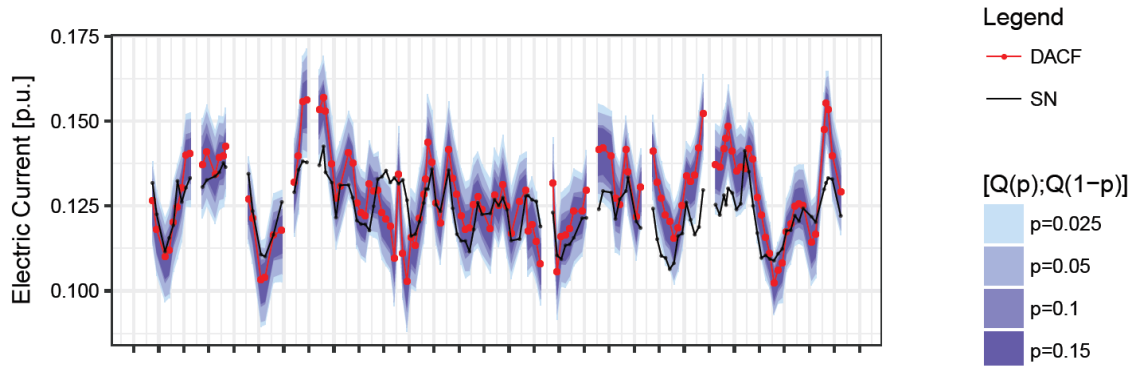


Figure 4 – Illustration of a computed fan chart (univariate analysis)

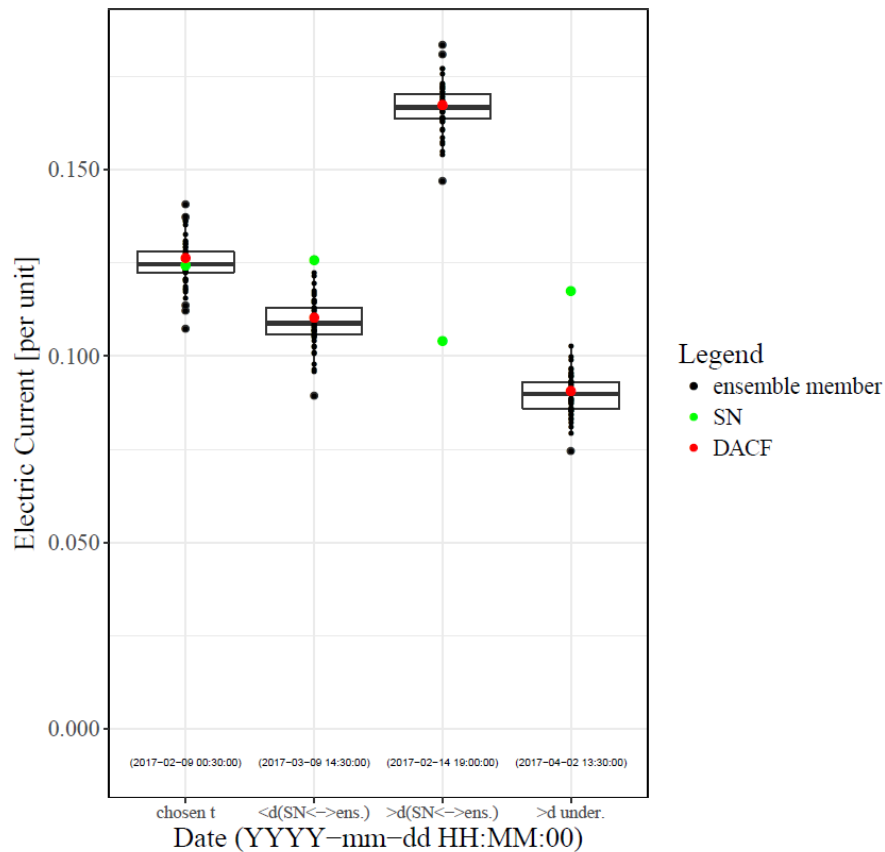


Figure 5 – Illustration of computed boxplots (univariate analysis)

Finally, if the user wants to **perform the analysis just for a specific set of flow in branches**, this can be done by adding the code name of the branch flow in the last lines of the **config_file.txt** (see Figure 6), **by default all the branch flows are analyzed**.

Example: If the user wants to analyze the branch flows whose code name in the input files is B.CARL72BIANC__TO__B.CARP7, COULAL71P.COR__TO__COULAP7, COULAL74CRUAS__TO__COULAP7 and REALTL71TAVEL__TO__REALTP7, then these code names must be included in the config_file.txt as illustrated in Figure 6.

```

1 "Parameter", "Value"
2 "Results_path", "~/Desktop/RTE/results1"
3 "Data_path", "~/Desktop/RTE/data_"
4 "p-value", 0.05
5 "Minimum timestamps", 100
6 "Spread Plot", TRUE
7 "QQ Plot", TRUE
8 "Sensitivity Analysis", TRUE
9 "Initial timestamp", "2017-01-01 00:00:00"
10 "Final timestamp", "2017-06-19 23:00:00"
11 "Fan.chart.p.value1", 0.025
12 "Fan.chart.p.value2", 0.05
13 "Fan.chart.p.value3", 0.1
14 "Fan.chart.p.value4", 0.15
15 "boxplot.timestamp", "2017-03-12 00:00:00"

1 "Parameter", "Value"
2 "Results_path", "~/Desktop/RTE/results1"
3 "Data_path", "~/Desktop/RTE/data_"
4 "p-value", 0.05
5 "Minimum timestamps", 100
6 "Spread Plot", TRUE
7 "QQ Plot", TRUE
8 "Sensitivity Analysis", TRUE
9 "Initial timestamp", "2017-01-01 00:00:00"
10 "Final timestamp", "2017-06-19 23:00:00"
11 "Fan.chart.p.value1", 0.025
12 "Fan.chart.p.value2", 0.05
13 "Fan.chart.p.value3", 0.1
14 "Fan.chart.p.value4", 0.15
15 "boxplot.timestamp", "2017-03-12 00:00:00"
16 "B.CARL72BIANC__TO__B.CARP7",
17 "COULAL71P.COR__TO__COULAP7",
18 "COULAL74CRUA5__TO__COULAP7",
19 "REALTL71TAVEL__TO__REALTP7",

```

Figure 6 – config_file.txt illustration with and without variable selection

About data_path: The “.csv” files 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 7 illustrates situations supported by the code.

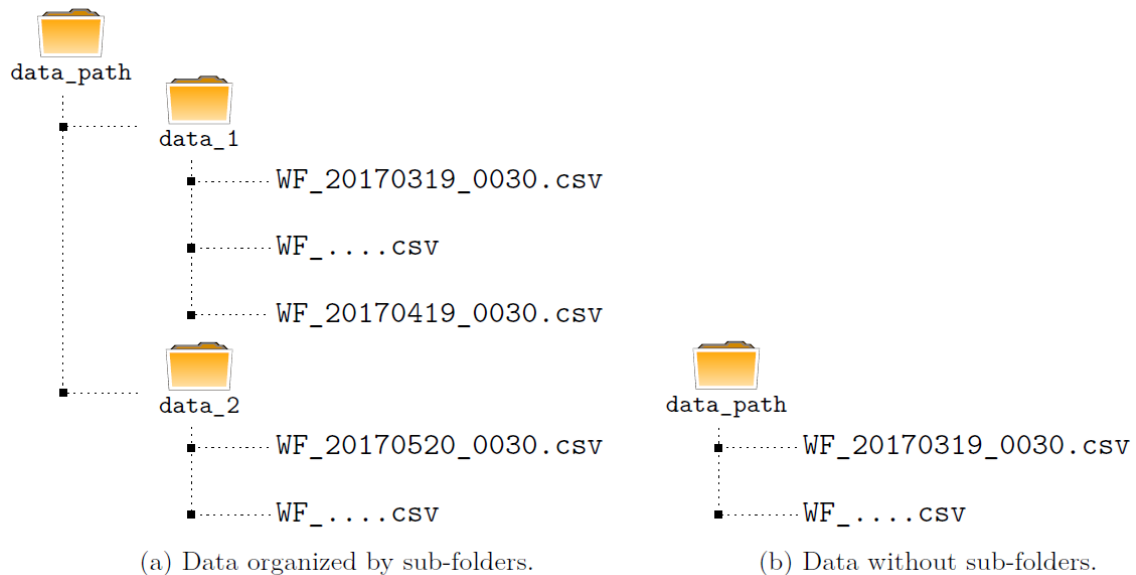


Figure 7 – Input data folder illustration

3 Output data

As mentioned before, the output of the script is saved in the **results_path** defined by the user. Figure 8 illustrates the structure of this folder and the possible outputs, according to the configuration file.

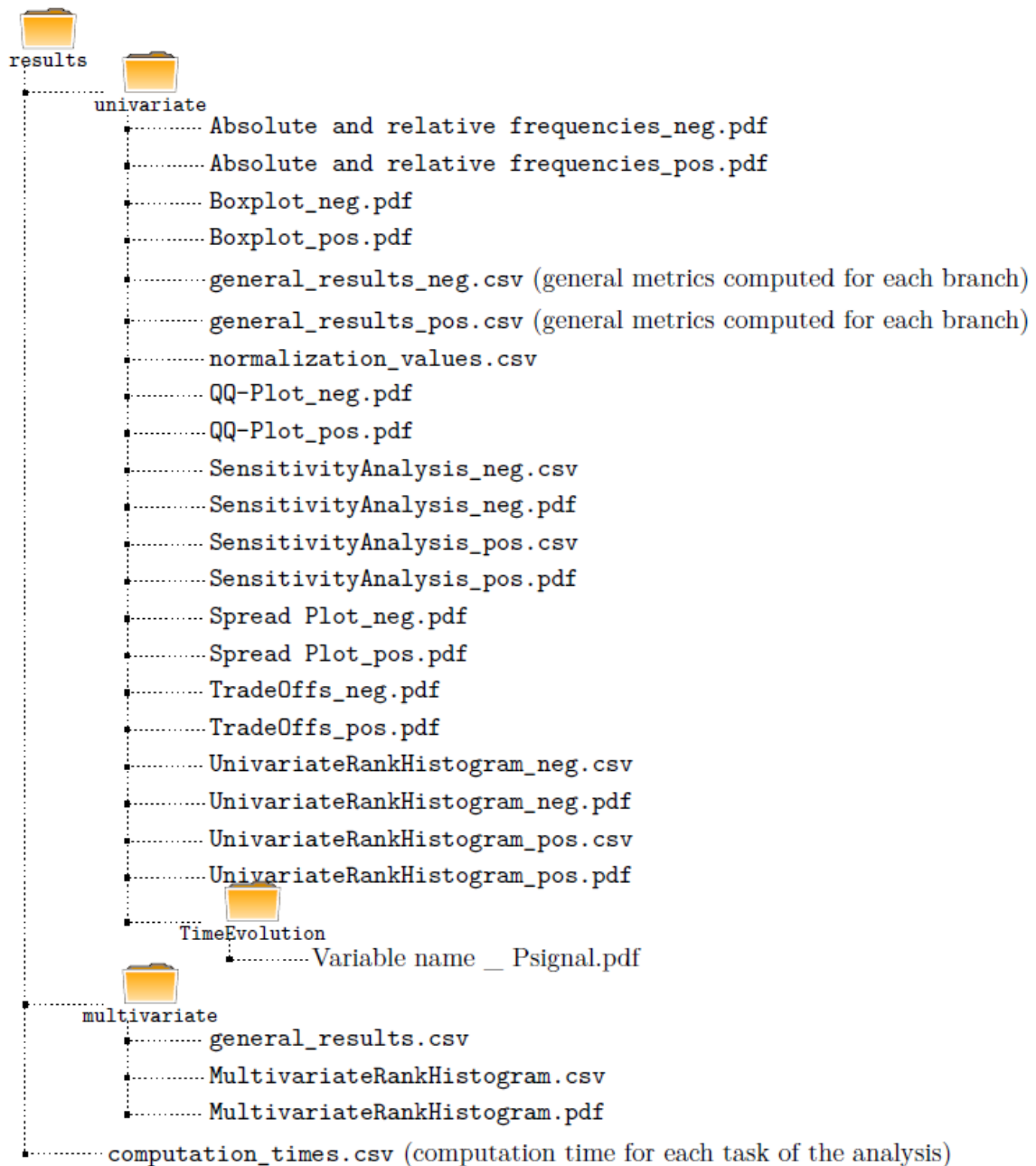


Figure 8 – Schematic representation of the results folder

4 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 1).

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 `/`.

5 References

- [1] R Core Team (2018). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.