

# Package ‘mergingTools’

September 13, 2023

**Title** Tools to Merge Hardware Event Monitors (HEMs) Coming from Separate Subexperiments into One Single Dataframe

**Version** 1.0.1

**Description** Implementation of two tools to merge Hardware Event Monitors (HEMs) from different subexperiments. Hardware Reading and Merging (HRM), which uses order statistics to merge; and MULTI-Correlation HEM (MUCH) which merges using a multivariate normal distribution. The reference paper for HRM is: S. Vilardell, I. Serra, R. Santalla, E. Mezzetti, J. Abella and F. J. Cazorla, “HRM: Merging Hardware Event Monitors for Improved Timing Analysis of Complex MPSoCs,” in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 11, pp. 3662-3673, Nov. 2020, <doi:10.1109/TCAD.2020.3013051>. For MUCH: S. Vilardell, I. Serra, E. Mezzetti, J. Abella, and F. J. Cazorla. 2021. “MUCH: exploiting pairwise hardware event monitor correlations for improved timing analysis of complex MPSoCs”. In Proceedings of the 36th Annual ACM Symposium on Applied Computing (SAC '21). Association for Computing Machinery. <doi:10.1145/3412841.3441931>. This work has been supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 772773).

**License** GPL-3

**Encoding** UTF-8

**Depends** R (>= 4.2.0)

**LazyData** true

**RoxygenNote** 7.2.3

**Imports** dplyr, purrr, tidyr, stringr, rlang, MASS

**Suggests** DT, knitr, rmarkdown, devtools

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Sergi Vilardell [aut, cre]

**Maintainer** Sergi Vilardell <sergi.vilardell@bsc.es>

**Repository** CRAN

**Date/Publication** 2023-09-13 18:40:02 UTC

## R topics documented:

code2hem . . . . .	2
correlation_matrix . . . . .	3
data_hrm_raw_vignette . . . . .	3
data_much_raw_vignette . . . . .	4
generate_mvg_params . . . . .	4
get_independent_matrix . . . . .	5
HRM_merge . . . . .	6
MUCH_merge . . . . .	7
process_raw_experiments . . . . .	9
simulate_and_merge . . . . .	9
T2080_code2name . . . . .	10
<b>Index</b>	<b>12</b>

---

code2hem	<i>Code to HEM name</i>
----------	-------------------------

---

### Description

code2hem returns the name of the HEM corresponding to the code of the T2080 manual. When gathering experimental data the captured HEMs will be coded with a number. This functions transforms the number into the reference name in the T2080 manual.

### Usage

```
code2hem(cod = NULL)
```

### Arguments

cod                   HEM code on the T2080 manual.

### Value

Returns the name of the HEM that corresponds to the input code.

### Examples

```
code <- 1
hem <- code2hem(code)
hem
```

---

correlation\_matrix     *Compute correlation matrix*

---

**Description**

Compute correlation matrix

**Usage**

```
correlation_matrix(splitted_data = NULL)
```

**Arguments**

splitted\_data     Sample data.

**Value**

Returns a merged dataframe as stated in the description

**References**

S. Vilardell, I. Serra, R. Santalla, E. Mezzetti, J. Abella and F. J. Cazorla, "HRM: Merging Hardware Event Monitors for Improved Timing Analysis of Complex MPSoCs," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 11, pp. 3662-3673, Nov. 2020, <doi:10.1109/TCAD.2020.3013051>.

**Examples**

```
n_pmcs <- 6
data_much <- mergingTools::process_raw_experiments(data = data_much_raw_vignette,
                                                    n_pmcs = n_pmcs)
cor_matrix <- mergingTools::correlation_matrix(splitted_data = data_much)
```

---

data\_hrm\_raw\_vignette     *Data generated for HRM merging*

---

**Description**

Data from a hardware platform experiment

**Usage**

```
data_hrm_raw_vignette
```

**Format**

An object of class dataframe

data\_much\_raw\_vignette

*Data generated for MUCH merging*

---

### **Description**

Data from a hardware platform experiment

### **Usage**

data\_much\_raw\_vignette

### **Format**

An object of class dataframe

---

generate\_mvg\_params

*Generate multivariate Gaussian distribution parameters*

---

### **Description**

generate\_mvg\_params takes as input the experimental data along its correlation matrix to estimate the Sigma matrix and the vector of means to compute the multivariate Gaussian distribution

### **Usage**

```
generate_mvg_params(splitted_data = NULL, cor_matrix = NULL)
```

### **Arguments**

splitted\_data    Splitted experimental data.

cor\_matrix        Correlation matrix

### **Value**

Returns the parameters of the multivariate Gaussian distributions as well as a list of concatenated HEM readings

## Examples

```
n_pmcs <- 6
data_much <- mergingTools::process_raw_experiments(data = data_much_raw_vignette,
                                                  n_pmcs = n_pmcs)
cor_matrix <- mergingTools::correlation_matrix(splitted_data = data_much)
dep_lvl <- 0.85
# Remove the HEMs which are linearly dependant on other HEMs
cor_matrix_independent <- mergingTools::get_independent_matrix(cor_matrix = cor_matrix,
                                                            dep_lvl = dep_lvl)
mvg_params <- mergingTools::generate_mvg_params(splitted_data = data_much,
                                                cor_matrix = cor_matrix_independent)
```

---

```
get_independent_matrix
```

*Generate independent HEMs*

---

## Description

There may be some HEMs which are linear dependent on another HEM. This produces conflicts when generating the multivariate Gaussian distribution as the Sigma matrix will not be invertible, causing an error. To avoid this, `get_independent_matrix` tells the user which HEMs are independent so they can choose which HEMs to use. To do so the user must input the maximum correlation between HEMs by `dep_lvl`. If `dep_lvl` is too high (>0.95) the user may leave very dependent HEMs and thus the Sigma matrix will not be invertible. We advise the user to lower the `dep_lvl` until the Sigma matrix is invertible.

## Usage

```
get_independent_matrix(cor_matrix = NULL, dep_lvl = NULL)
```

## Arguments

<code>cor_matrix</code>	Correlation matrix
<code>dep_lvl</code>	Dependency level allowed. A number between 0 and 1 indicating the maximum correlation allowed between HEMs.

## Value

Returns a list of independent HEMs

## Examples

```
n_pmcs <- 6
data_much <- mergingTools::process_raw_experiments(data = data_much_raw_vignette,
                                                  n_pmcs = n_pmcs)
cor_matrix <- mergingTools::correlation_matrix(splitted_data = data_much)
dep_lvl <- 0.85
# Remove the HEMs which are linearly dependant on other HEMs
```

```
cor_matrix_independent <- mergingTools::get_independent_matrix(cor_matrix = cor_matrix,
                                                              dep_lvl = dep_lvl)
```

---

 HRM\_merge

*HRM merge*


---

## Description

HRM\_merge merges HEMs coming from separate subexperiments into one single dataset. A Subexperiment is a measurement of a group of HEMs the size of the allowed PMCs by the machine. For HRM there is a designated anchor HEM that will be included on each subexperiment. For more details check the reference paper below. The input data must be in a specific format:

<b>1</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>4</b>	<b>5</b>
30	15	20	40	10	12
35	16	25	25	13	15
32	14	30	21	11	17

where the numbers on top are the codes for the HEMs on the T2080. In this case we have two subexperiments: subexp1 = (1, 2, 3), subexp2 = (1, 4, 5); and H1 is the anchor HEM that should be included in each subexperiment. The data will be processed into:

<b>H1</b>	<b>H2</b>	<b>H3</b>	<b>H1</b>	<b>H4</b>	<b>H5</b>
30	15	20	40	10	12
35	16	25	25	13	15
32	14	30	21	11	17

The processing transforms the code of the HEMs to the reference name on the T2080 manual. merge\_hems sorts each subexperiment by the anchor HEM, and the substitutes the anchor HEM by its quantiles. If one inputs the example dataset, merge\_hems will return:

<b>H1</b>	<b>H2</b>	<b>H3</b>	<b>H4</b>	<b>H5</b>
21	15	20	11	17
31	14	30	13	15
40	16	25	10	12

## Usage

```
HRM_merge(data = NULL, anchor_hem = NULL)
```

## Arguments

data                    Sample data.

anchor\_hem      Reference HEM for merging

**Value**

Returns a merged dataframe as stated in the description

**References**

S. Vilardell, I. Serra, R. Santalla, E. Mezzetti, J. Abella and F. J. Cazorla, "HRM: Merging Hardware Event Monitors for Improved Timing Analysis of Complex MPSoCs," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 11, pp. 3662-3673, Nov. 2020, <doi:10.1109/TCAD.2020.3013051>.

**Examples**

```
library(mergingTools)
n_pmcs <- 6
data_hrm <- mergingTools::process_raw_experiments(data = data_hrm_raw_vignette, n_pmcs = n_pmcs)
merged_data <- mergingTools::HRM_merge(data = data_hrm, anchor_hem = "PROCESSOR_CYCLES")
```

---

MUCH_merge	<i>MUCH Merge</i>
------------	-------------------

---

**Description**

MUCH\_merge merges HEMs coming from separate subexperiments into one single dataset. A Subexperiment is a measurement of a group of HEMs the size of the allowed PMCs by the machine. For MUCH you need to measure each HEM against the rest of them. For instance, let's say that one has 2 PMCs and 3 HEMs to measure. The input data coming from the experiments should look like this:

<b>1</b>	<b>2</b>	<b>1</b>	<b>3</b>	<b>2</b>	<b>3</b>
30	15	20	54	15	24
35	16	25	32	10	29
32	14	30	45	9	32

where the numbers on top are the codes for the HEMs on the T2080. In this case we have three subexperiments: subexp1 = (1, 2), subexp2 = (1, 3, 6), subexp3 = (2, 3); The data will be processed into:

<b>H1</b>	<b>H2</b>	<b>H1</b>	<b>H3</b>	<b>H2</b>	<b>H3</b>
30	15	20	54	15	24
35	16	25	32	10	29
32	14	30	45	9	32





---

 process\_raw\_experiments

*Process raw experimental data*


---

### Description

process\_raw\_experiments splits the data into separate dataframes. The number of different dataframes is the number of subexperiments and the number of variables on each one is the number of PMCs. The validation data is on the form of several subexperiments that don't need to be merged together. The intent of the data is to provide information on how each HEM behaves when put against another HEM. The validation data them is a collection of subexperiments where each HEM is put against each other.

### Usage

```
process_raw_experiments(data = NULL, n_pmcs = NULL)
```

### Arguments

data	Validation data.
n_pmcs	Number of PMCS used on the experiment

### Value

Returns a list of dataframes as stated in the description.

### Examples

```
n_pmcs <- 6
data_much <- mergingTools::process_raw_experiments(data = data_much_raw_vignette,
                                                    n_pmcs = n_pmcs)
```

---

 simulate\_and\_merge

*Simulate the MVG and merge the HEMs*


---

### Description

simulate\_and\_merge takes as input the parameters of the multivariate Gaussian distribution and simulates multiple samples. The number of samples is determined by n\_sims. On each simulation, each HEM of the experimental data is sorted from lowest to highest independently. Then, the simulated MGD is used as the model to arrange the experimental data. For instance, if on the MGD the 10th highest value of HEM 1 is paired with the 2nd highest and the 4th highest values of HEMs 2 and 3 respectively, Then, we modify the experimental data to copy this arrangement. After merging and arranging, the 10th highest value of HEM 1 on the experimental data will be paired with the 2nd highest and the 4th highest values of HEMs 2 and 3 respectively. The algorithm does multiple simulations of the MGD and keeps the sample that gives the lowest error on the correlation matrix when the data is merged w.r.t. the correlation matrix of the experimental data.

**Usage**

```
simulate_and_merge(
  mvg_params = NULL,
  n_runs = NULL,
  n_sims = NULL,
  cor_matrix = NULL
)
```

**Arguments**

<code>mvg_params</code>	parameters from the multivariate Gaussian distribution
<code>n_runs</code>	Number of rows for the output merged data
<code>n_sims</code>	Number of simulations for the multivariate Gaussian distribution to find the optimal merge
<code>cor_matrix</code>	Correlation matrix

**Value**

Returns a merged dataframe as stated in the description

**Examples**

```
n_pmcs <- 6
data_much <- mergingTools::process_raw_experiments(data = data_much_raw_vignette,
                                                  n_pmcs = n_pmcs)
cor_matrix <- mergingTools::correlation_matrix(splitted_data = data_much)
dep_lvl <- 0.85
# Remove the HEMs which are linearly dependant on other HEMs
cor_matrix_independent <- mergingTools::get_independent_matrix(cor_matrix = cor_matrix,
                                                             dep_lvl = dep_lvl)
mvg_params <- mergingTools::generate_mvg_params(splitted_data = data_much,
                                                cor_matrix = cor_matrix_independent)

n_sims <- 10
n_runs <- 1000
merged_data <- mergingTools::simulate_and_merge(mvg_params = mvg_params,
                                              n_runs = n_runs,
                                              n_sims = n_sims,
                                              cor_matrix = cor_matrix_independent)
```

---

T2080\_code2name

*T2080 HEM codes to HEM names*


---

**Description**

Dataset used to transform the codes from the T2080 manual into readable HEM names.

*T2080\_code2name*

11

**Usage**

`T2080_code2name`

**Format**

An object of class `data.frame` with 262 rows and 2 columns.

**Examples**

```
data(T2080_code2name)
```

# Index

- \* **HEM**
  - code2hem, [2](#)
- \* **HRM**
  - correlation\_matrix, [3](#)
  - HRM\_merge, [6](#)
  - process\_raw\_experiments, [9](#)
- \* **MUCH**
  - generate\_mvg\_params, [4](#)
  - get\_independent\_matrix, [5](#)
  - MUCH\_merge, [7](#)
  - process\_raw\_experiments, [9](#)
  - simulate\_and\_merge, [9](#)
- \* **datasets**
  - data\_hrm\_raw\_vignette, [3](#)
  - data\_much\_raw\_vignette, [4](#)
  - T2080\_code2name, [10](#)

code2hem, [2](#)

correlation\_matrix, [3](#)

data\_hrm\_raw\_vignette, [3](#)

data\_much\_raw\_vignette, [4](#)

generate\_mvg\_params, [4](#)

get\_independent\_matrix, [5](#)

HRM\_merge, [6](#)

MUCH\_merge, [7](#)

process\_raw\_experiments, [9](#)

simulate\_and\_merge, [9](#)

T2080\_code2name, [10](#)