Package ‘benchmarkmeData’

October 12, 2022

Type Package
Title Data Set for the ‘benchmarkme’ Package
Version 1.0.4
Maintainer Colin Gillespie <csgillespie@gmail.com>
Description Crowd sourced benchmarks from running the ‘benchmarkme’ package.
License GPL-2 | GPL-3
URL https://github.com/csgillespie/benchmarkme-data
BugReports https://github.com/csgillespie/benchmarkme-data/issues
Depends R (>= 3.5.0)
Imports dplyr, graphics, tibble, utils
Suggests benchmarkme, covr, DT, testthat
Encoding UTF-8
LazyData TRUE
RoxygenNote 7.1.0
NeedsCompilation no
Author Colin Gillespie [aut, cre] (<https://orcid.org/0000-0003-1787-0275>)
Repository CRAN
Date/Publication 2020-04-23 15:10:02 UTC

R topics documented:

benchmarkmeData-package ........................................ 2
get_datatable_past ............................................. 2
is_blas_optimize .............................................. 3
make_data_set .................................................. 3
past_results .................................................... 4
past_results_v2 ............................................... 4
plot_past ....................................................... 4
summarise_results ............................................. 5
The benchmarkmeData package

Description
This package contains the results from users running the benchmarkme package. The key function is plot_past().

Author(s)
<csgillespie@gmail.com>

See Also
https://github.com/csgillespie/benchmarkme-data

Examples
plot_past("prog")

get_datatable_past  Interactive table of results

Description
A summary of past results

Usage
get_datatable_past(test_group, blas_optimize = NULL, cores = 0)

Arguments
test_group One of "prog","matrix_fun","matrix_cal","read5","read50","read200","write5","write50" or "write200". Default value prog.
blas_optimize Default NULL. The default behaviour is to plot all results. To plot only the BLAS optimized results, set to TRUE, otherwise FALSE.
cores Default 0, i.e. no parallel.

Examples
## Need the DT package
## View all results for prog test
get_datatable_past("prog")
**is_blas_optimize**

**Description**
Try to determine parallel BLAS, which implies non-standard R! Compare user with elapsed time. If user > elapsed, then parallel BLAS

**Usage**
```r
is_blas_optimize(results)
```

**Arguments**
- `results` The output from a benchmark_* call.

**make_data_set**

*Functions for manipulating uploaded results*

**Description**
Functions used for moving and creating the past_results_v2 data set from uploaded data. The move_files function is used to moved files from the server to another location, whilst removing any empty data sets.

**Usage**
```r
make_data_set(from)
```
```r
move_files(from, to)
```

**Arguments**
- `from` A directory containing the uploaded results.
- `to` Destination directory

**Note**
One of the unit tests uploads an empty results file. Files where the results are NULL are moved to a sub-directory (called) empty in the to directory. If the empty directory doesn’t exist, it is created.
Currently these functions are specific to my set-up.
past_results  

*Benchmarking results*

**Description**  
A summary of past benchmarks.

**Format**  
A data frame

---

past_results_v2  

*Benchmarking results*

**Description**  
A summary of past benchmarks.

**Format**  
A data frame

---

plot_past  

*Scatter plot of past benchmarks*

**Description**  
Plot the previous benchmarks. This function creates two figures.  
- Figure 1: Total benchmark time over all benchmarks (in seconds) on the y-axis.  
- Figure 2: Relative time (compared to the smallest benchmark).

The data set used is data(past_results_v2).

**Usage**  

```
plot_past(test_group, blas_optimize = NULL, cores = 0, log = "y")
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>test_group</td>
<td>One of &quot;prog&quot;, &quot;matrix_fun&quot;, &quot;matrix_cal&quot;, &quot;read5&quot;, &quot;read50&quot;, &quot;read200&quot;, &quot;write5&quot;, &quot;write50&quot; or &quot;write200&quot;. Default value prog.</td>
</tr>
<tr>
<td>blas_optimize</td>
<td>Default NULL. The default behaviour is to plot all results. To plot only the BLAS optimized results, set to TRUE, otherwise FALSE.</td>
</tr>
<tr>
<td>cores</td>
<td>Default 0, i.e. no parallel.</td>
</tr>
<tr>
<td>log</td>
<td>By default the y axis is plotted on the log scale. To change, set the the argument equal to the empty parameter string, &quot;&quot;.</td>
</tr>
</tbody>
</table>
**summarise_results**

## Examples

```r
## Plot all past results for the `prog` benchmark
plot_past("prog", blas_optimize = NULL)
```

---

### Description

Selects and aggregates over the `past_results_v2` data set or the `results` input data set.

### Usage

```r
summarise_results(res)
select_results(test_group, results = NULL, blas_optimize = NULL, cores = 0)
```

### Arguments

- **res**: A list containing benchmark results and system information.
- **test_group**: One of "prog", "matrix_fun", "matrix_cal", "read5", "read50", "read200", "write5", "write50" or "write200". Default value `prog`.
- **results**: Default `NULL`. If `NULL` the `past_results_v2` data set is used. Otherwise, the input data set.
- **blas_optimize**: Default `NULL`. The default behaviour is to plot all results. To plot only the BLAS optimized results, set to `TRUE`, otherwise `FALSE`.
- **cores**: Default `0`, i.e. no parallel.

### Value

A data frame

### Examples

```r
select_results("prog", blas_optimize = NULL)
```
Index

* package
  benchmarkmeData-package, 2

benchmarkmeData
  (benchmarkmeData-package), 2
benchmarkmeData-package, 2

get_datatable_past, 2

is_blas_optimize, 3

make_data_set, 3
move_files (make_data_set), 3

past_results, 4
past_results_v2, 4
plot_past, 4

select_results (summarise_results), 5
summarise_results, 5