

mrstudyr

Colton J. McCurdy mccurdyc@allegheny.edu

Installing mrstudyr Package from GitHub

This code will install the mrstudyr package from GitHub using the `install_github` function.

```
devtools::install_github("mccurdyc/mrstudyr")
```

Initialize the System

First, load in the libraries that are used in addition to those with the mrstudyr package (i.e., load all of the packages not used by mrstudyr but still used in this RMarkdown file). Note that right now the mrstudyr package will automatically load all of the packages that it needs to perform its various analyses. Now, we are ready to call the functions from the mrstudyr package and produce the appropriate summary tables and graphs.

```
suppressPackageStartupMessages(library(mrstudyr))
suppressPackageStartupMessages(library(knitr))
```

Comparing Mutant Reduction Techniques

Show the schemas used in this study

```
sqlite <- read_sqlite_data() %>% collect_normal_data()
schemas <- sqlite %>% select_all_schemas()
knitr::kable(schemas, format="latex", booktabs=TRUE)
```

Visualize summary graphs of data before performing reduction

The summary graphs before performing reduction include

- fractional operator costs (per dbms)
- fractional operator costs (per dbms, per schema)
- fractional operator frequencies (per dbms) (e.g., this operator accounts for 20% of total mutants)
- fractional operator frequencies (per dbms, per schema)
- original mutation score (per dbms)
- original mutation score (per dbms, per schema)

```
create_data_overview_graphs()
```

Visualize the correlation and cost reduction between reduced sets from performing random sampling

This will perform random sampling for all DBMSs where the technique outlined later will only display the data collected for SQLite.

```
create_random_sampling_graphs()
```

Visualize correlation between reduced and original mutation score generated by hill climbing

NOTE: Since the hill climbing technique takes long, I provide my data as a feather

To install and load the feather tool:

```
install.packages('feather')
library(feather)
```

To read the data from performing all reduction techniques

```
data <- read_feather("feathers/combined_technique_data.feather")
```

Now, you can create the random sampling, hill climbing, selective and selective random sampling plots using the data that I collected.

```
data %>% dplyr::filter(technique_group == 'RS') %>% visualize_plot_random_sampling_correlation()
data %>% dplyr::filter(technique_group == 'RS') %>% visualize_plot_random_sampling_cost_reduction()
data %>% dplyr::filter(technique_group == 'HC') %>% visualize_plot_hill_climbing_correlation()
data %>% dplyr::filter(technique_group == 'HC') %>% visualize_plot_hill_climbing_cost_reduction()
```

You will notice if you do this, you will only have the data from SQLite, this is because this is the DBMSs I used to generate the hill climbing model. In the future, we would like to also generate models from HyperSQL and PostgreSQL.

Apply hill climbing model to other DBMSs

First, you need to read in the generated model using the feather package again. For this example, I will only read in the small (generated from a more granular step size) model. Then, you will need to make sure you have the data for a DBMS read in. In this example, I use the HyperSQL DBMS because it is one of the DBMSs not used to generate the model.

```
small <- feather::read_feather("feathers/small-model.feather")
hypersql <- read_hypersql_data() %>% collect_normal_data()
apply_operator_model(hypersql, small)
```

Alternatively, you could use the data that I collected from applying the models to other DBMSs and step sizes by reading in the feather file.

```
all_dbms_joined_technique_data <- feather::read_feather("feathers/all_dbms_joined_technique_data.feather")
```

Now, you should be able to produce graphs comparing random sampling and hill climbing correlation and cost reduction using the following.

```
all_dbms_joined_technique_data %>% visualize_plot_correlation_all_reduction_techniques_box()
all_dbms_joined_technique_data %>% visualize_plot_cost_reduction_all_reduction_techniques_box()
```

If you are interested to compare the techniques at a higher level, you can compare the techniques by reduction technique group (e.g., random sampling versus hill climbing instead of at the configuration level).

```
all_dbms_joined_technique_data %>% visualize_plot_correlation_all_groups()
all_dbms_joined_technique_data %>% visualize_plot_cost_reduction_all_groups()
```

Perform a statistical analysis on the correlation reduced and original mutation scores

```
perform_wilcoxon_accurate(all_dbms_joined_technique_data, "correlation")
```

Perform a statistical analysis on the cost reduction

```
perform_wilcoxon_accurate(all_dbms_joined_technique_data, "cost reduction")
```

Perform a head-to-head effect size calculation comparing the correlation of the reduced and original mutation scores

```
perform_effectsize_accurate(all_dbms_joined_technique_data, "correlation")
```

Perform a head-to-head effect size calculation comparing the cost reduction

```
perform_effectsize_accurate(all_dbms_joined_technique_data, "cost reduction")
```