

Data

get_data is used to fill the database (sqlite) with bug data from defects4j and associated gzoltar spectra.

To load data for a defects4j-bug (e.g. Time 1) into the db use ipython:

```
from getdata.get_data import GetData
d = GetData()
d.get_data(project='Time', bug=1)
```

If no bug is given, it will get all bugs of the project, if no project is given as well, all bugs of all projects are processed.

Model

model is used to build decision trees with Weka and to parse and analyze its result. It also calculates the suspiciousness for each method.

To generate the model and get the methods ordered by suspiciousness use ipython:

```
from model import Model
m = Model('Time', 1)      ### takes a while, builds the decision tree and parses it
results, methods, mid_2_list_id = m.get_suspicious_methods()
```

=> **results** contains the **suspiciousness** values for each **mid** (method id) and **methods** contains the **method names** (= Java-File name and line number of method) and their **lengths** in the same order (by highest suspiciousness).

mid_2_list_id is a mapping from each mid to its place in the list *methods*.

Evaluation

In evaluateresult.py in `__main__`:

```
bugs = [ {'project': 'Time', 'bug': 1} ]
```

add those defects4j bugs you want to evaluate to the list and call:

```
print_results(bugs, 'test')
```

where 'test' is a file to which the output is written (it's also printed on the console). If the needed **data** is **not found in the database** (get_data.db) the data for that bug **will be generated** (defects4j, gzoltar, weka, ...).

OR in `__main__`:

```
evaluate_project('Mockito', 'test')
```

to evaluate the **whole project** (here Mockito). With that command **only** those **bugs** of the given project that are **already in the database** are evaluated.

Then call:

```
python evaluateresult.py
```

and you should get something like:

Results for bug Mockito 12

=====

Buggy method: org/mockito/internal/util/reflection/GenericMaster.java, line 16

Rank: 1.0

Best rank: 1

Worst rank: 2

Wasted effort: 3

10 most suspicious methods ordered by suspiciousness:

0.568182, org/mockito/internal/util/reflection/GenericMaster.java, line 16

0.568182, org/mockito/internal/util/reflection/GenericMaster.java, line 8

0.535354, org/mockito/ArgumentCaptor.java, line 70

0.535354, org/mockito/ArgumentMatcher.java, line 57

0.535354, org/mockito/internal/matchers/CapturingMatcher.java, line 16

0.535354, org/mockito/internal/progress/HandyReturnValues.java, line 17

0.535354, org/mockito/ArgumentCaptor.java, line 122

0.535354, org/mockito/internal/configuration/DefaultAnnotationEngine.java, line 69

0.489899, org/mockito/internal/util/reflection/AccessibilityChanger.java, line 16

0.489899, org/mockito/internal/util/reflection/FieldSetter.java, line 19

Installation:

Requirements:

Linux with

- Oracle JDK 7
- <https://bitbucket.org/rjust/fault-localization-data/overview> (for gzoltar and defects4j)
- Weka <https://www.cs.waikato.ac.nz/ml/weka/>

Either you download the program from moodle or clone the project from github:

```
git clone https://github.com/Florian1990/ml4swt  
git checkout final
```

To install the required packages for python:

```
pip install -r requirements.txt
```

If you want to use a prefilled database you can use get_data.db from:

<https://web.tresorit.com/l/#H9wKJ5AWg18m9078-1FLKA> (this URL is accessible until 20th of February)