pytest-flakefighters

A Suite of Tools for Detecting and Classifying Flaky Test Failures

PROBLEM AND APPROACH

Flaky Tests non-deterministically pass and fail against the same version of code.

TRADITIONAL FLAKY TEST DETECTION

Re-run Tests to Detect Flakiness

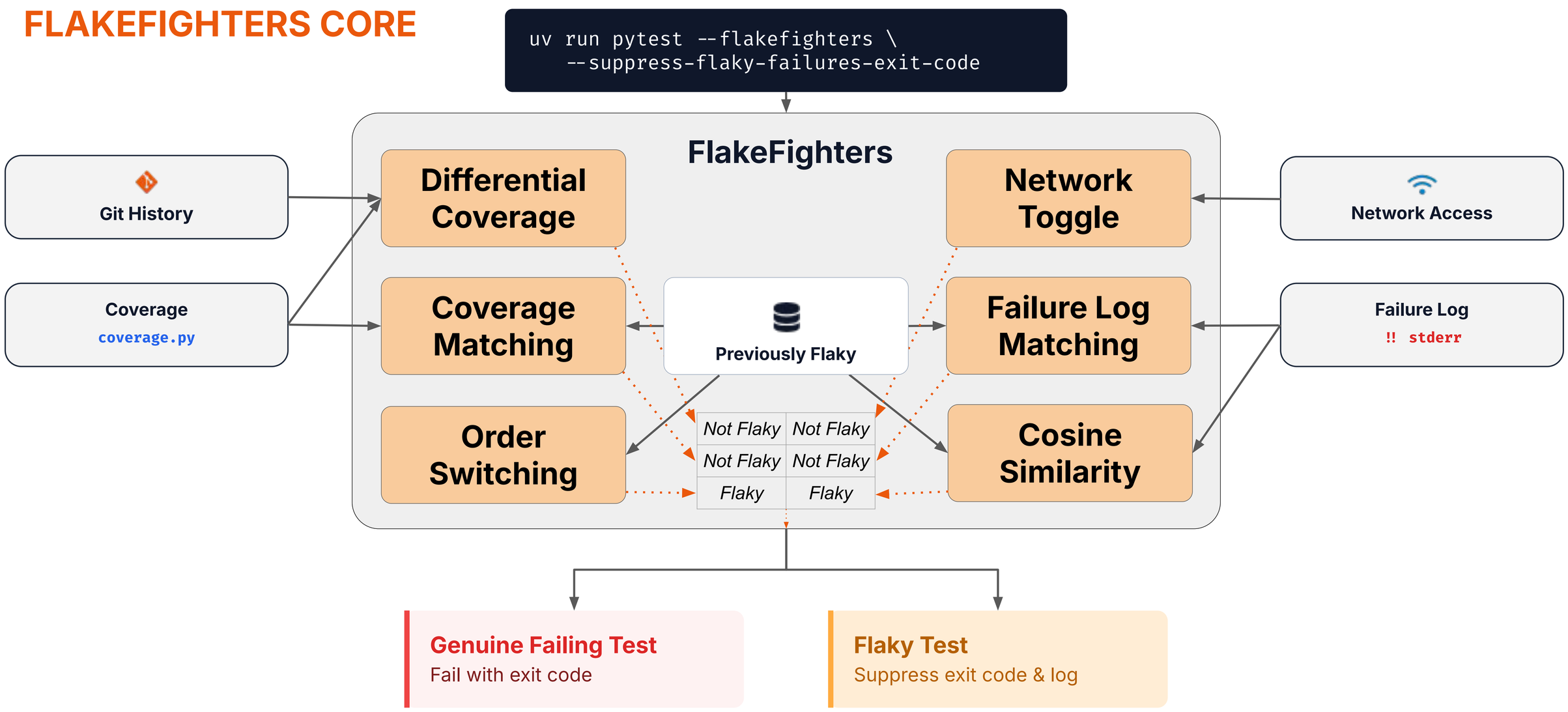
- Wastes resources
- Does not tell you why its flaky
- Erodes trust in tests

FLAKEFIGHTERS

Detect and Classify Flakiness

- Swiss army knife of techniques
- Classifies the underlying cause
- Avoids time-consuming debugging

FLAKEFIGHTERS CORE



USING FLAKEFIGHTERS

Pytest

```
> pytest
═══ short test summary info ═══
FAILED concord232/test_alarm_control_panel.py::
test_update_connection_error - AssertionError:
assert 'Unable to connect' in 'INFO SyncWorker_0
homeassistant.loa ...
```

Pytest + FlakeFighters

```
> pytest --flakefighters
═══ short test summary info ═══
FLAKY concord232/test_alarm_control_panel.py::
test_update_connection_error - AssertionError:
assert 'Unable to connect' in 'INFO SyncWorker_0
homeassistant.loa ...
```

CONFIGURING FLAKEFIGHTERS

Use any combination of the FlakeFighters to suit your project

Set thresholds to suit your project

```
[ ... traceback_matching.CosineSimilarity]
threshold=0.8
# Cosine similarity ⩾ 0.8 is classed as a match
```

Pick your FlakeFighters

```
--active-flakefighters DiffCov CosineSimilarity
```

E.g. use only differential coverage and cosine simularity.

Customize storage & report paths

```
--database-url=flakefighters.db # default local
--html # pytest html report
--json-report # pytest json report
```


University of Sheffield

Contact Megan Maton:
m.e.maton@sheffield.ac.uk

MIT
Open Source
License

github.com/test-flare