

# Automatic Detection of Pseudo-tested Methods using Python and Pytest

Nicholas Tocci

Gregory M. Kapfhammer

## Introduction

Python programs are often complex and difficult to test. But test coverage does not show that a method was adequately tested. A method  $m$  is psuedo-tested if a test passes even when  $m$  is not run.

Since it may be time-consuming and error-prone to manually detect psuedo-tested methods, Function-Fiasco automatically discovers them for engineers.

## Implementation

Function-Fiasco uses technologies like:

- Pytest
- Coverage
- Decorators
- Instrumentation

Function-Fiasco performs these steps:

1. Instrument all program methods
2. Elide execution of chosen method
3. Run the tests and observe behavior
4. Run steps (2) and (3) for all methods
5. Report the psuedo-tested methods

Steps are optimized, ensuring that the tool scales to large Python programs.



ALLEGHENY  
COLLEGE

# Function-Fiasco is an automated tool that detects pseudo-tested methods in real Python programs

## Does the status of $T_i$ change when Function-Fiasco does not run $m$ ?



Scan the QR Code to visit our GitHub project

## Preliminary Results

Program	Coverage	Total	Modified	Pseudo-Tested
Hashids-Python	97%	16	10	8
Bleach	48%	368	8	2
Pycco	77%	22	6	5
Howdoi	78%	20	2	0
Flashtext	81%	42	7	4
Honcho	85%	58	7	5
Maya	90%	88	13	3
GatorGrader	99%	92	54	30
Hatch	100%	134	14	6
Nikola	67%	732	16	9

Function-Fiasco detects pseudo-tested methods in real Python programs, suggesting the need for improved testing.

## Future Work

Add new features to Function-Fiasco:

- Handle more kinds of methods
- Improve type fuzzing capability
- Better observe parameterized tests
- Report more types of test coverage

Use improved Function-Fiasco to detect and improve pseudo-tested methods.

## Conclusion

Pseudo-tested methods exist in many real-world Python programs. Function-Fiasco automatically detects these methods, saving time that testers can instead devote to improving test suites. Available on GitHub, Function-Fiasco aids the implementation of high-quality Pytest test suites and Python programs.

## Get Involved

If you would like support the development of Function-Fiasco, please raise an issue on the tracker or create a pull request to add a new feature or bug fix.

## Acknowledgements

Poster creation aided by Cory Wiard.  
Feedback provided by Aravind Mohan.