Coverage Is Not Strongly Correlated with Test Suite Effectiveness

Laura Inozemtseva and Reid Holmes
University of Waterloo

High coverage does not guarantee high quality.

:(

Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you. (0% complete)

If you'd like to know more, you can search online later for this error: HAL_INITIALIZATION_FAILED

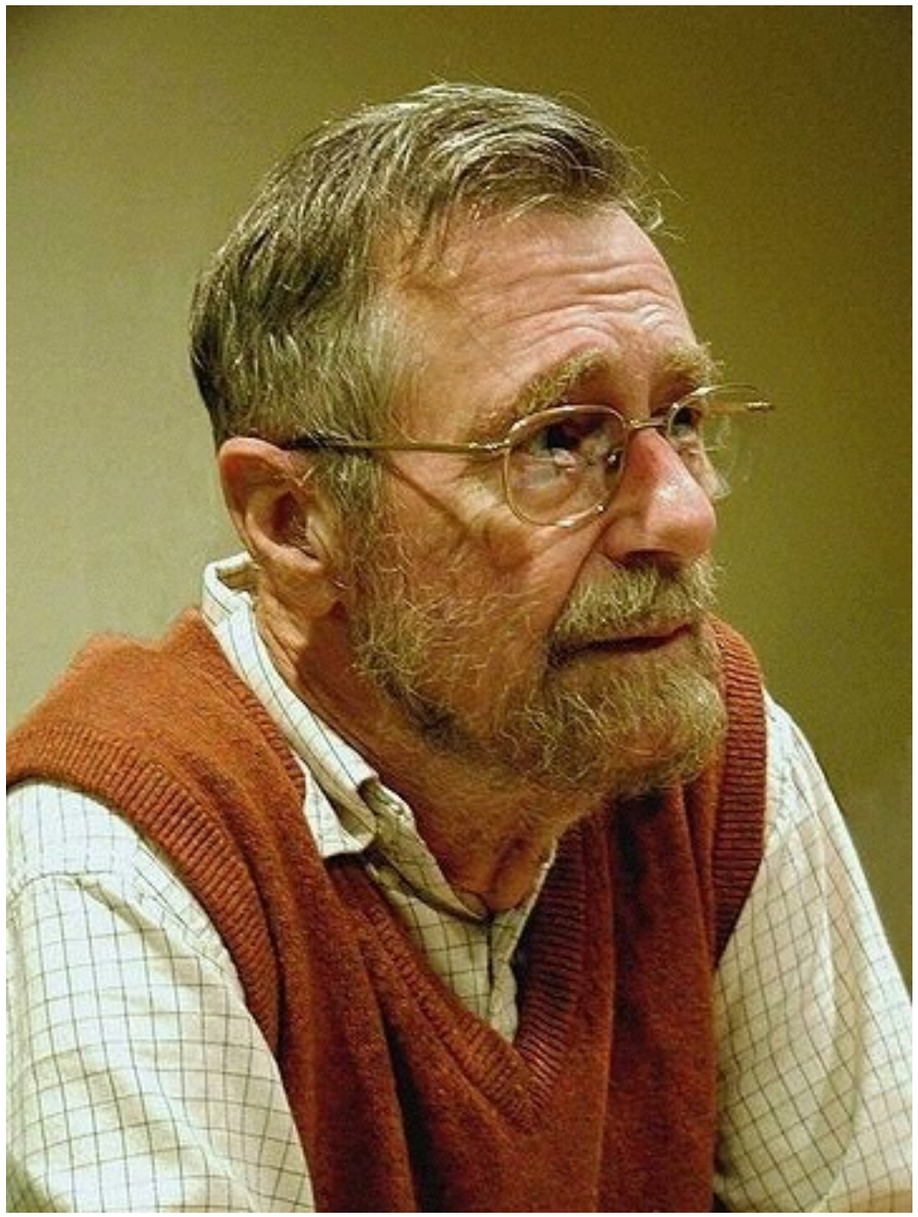


Image: Hamilton Richards

“Program testing can be used to show the presence of bugs, but never to show their absence!”

How can we estimate the fault detection ability of a test suite?

Coverage: the percentage of “structures” in the code executed by the test suite.

Structures can be statements, branches, paths, etc.

"…the resulting metric should be usable for predicting the effectiveness of the test process."

–William Perry, *Effective Methods for Software Testing*

Year
1993
1994
1994
1997
1998
1999
2005
2006
2009
2013
2014

Year	Corr?
1993	~
1994	✔
1994	✔
1997	~
1998	~
1999	✖
2005	~
2006	✔
2009	~
2013	✔
2014	✔

Year	Corr?	Large Programs
1993	~	✖
1994	✔	✖
1994	✔	✖
1997	~	✖
1998	~	✖
1999	✖	✖
2005	~	✖
2006	✔	✖
2009	~	✖
2013	✔	✔
2014	✔	✔

Year	Corr?	Large Programs	Realistic Suites
1993	~	✗	✗
1994	✓	✗	✗
1994	✓	✗	~
1997	~	✗	✗
1998	~	✗	✗
1999	✗	✗	✓
2005	~	✗	~
2006	✓	✗	~
2009	~	✗	~
2013	✓	✓	~
2014	✓	✓	✓

Year	Corr?	Large Programs	Realistic Suites	# of Tests Controlled
1993	~	✗	✗	✓
1994	✓	✗	✗	✓
1994	✓	✗	~	✓
1997	~	✗	✗	✓
1998	~	✗	✗	✓
1999	✗	✗	✓	✓
2005	~	✗	~	✗
2006	✓	✗	~	✓
2009	~	✗	~	✓
2013	✓	✓	~	✗
2014	✓	✓	✓	✗

Year	Corr?	Large Programs	Realistic Suites	# of Tests Controlled
1993	~	✗	✗	✓
1994	✓	✗	✗	✓
1994	✓	✗	~	✓
1997	~	✗	✗	✓
1998	~	✗	✗	✓
1999	✗	✗	✓	✓
2005	~	✗	~	✗
2006	✓	✗	~	✓
2009	~	✗	~	✓
2013	✓	✓	~	✗
2014	✓	✓	✓	✗

Contribution: a study using large programs and developer-written test suites that controls for suite size (number of test cases)

Method

Method

1. Select programs to study

Method

1. Select programs to study
   - Five large programs: *O*(100 KSLOC)

Method

1. Select programs to study
   - Five large programs: *O*(100 KSLOC)
   - Developer-written test suites

Method

1. Select programs to study
2. Make test suites

Method

1. Select programs to study
2. Make test suites
    - Random selection

Method

1. Select programs to study
2. Make test suites
    - Random selection
    - Fixed size: 3, 10, 30, 100, 300, 1000, 3000

Method

1. Select programs to study
2. Make test suites
    - Random selection
    - Fixed size: 3, 10, 30, 100, 300, 1000, 3000
    - 1000 suites of each size

Method

1. Select programs to study
2. Make test suites
    - Random selection
    - Fixed size: 3, 10, 30, 100, 300, 1000, 3000
    - 1000 suites of each size
    - 31,000 suites total

Method

1. Select programs to study
2. Make test suites
3. Measure suite coverage

Method

1. Select programs to study
2. Make test suites
3. Measure suite coverage
   - CodeCover

Method

1. Select programs to study
2. Make test suites
3. Measure suite coverage
   - CodeCover
   - Statement, decision, MCC

Method

1. Select programs to study
2. Make test suites
3. Measure suite coverage
4. Measure suite effectiveness

Method

1. Select programs to study
2. Make test suites
3. Measure suite coverage
4. Measure suite effectiveness
   - % mutants detected

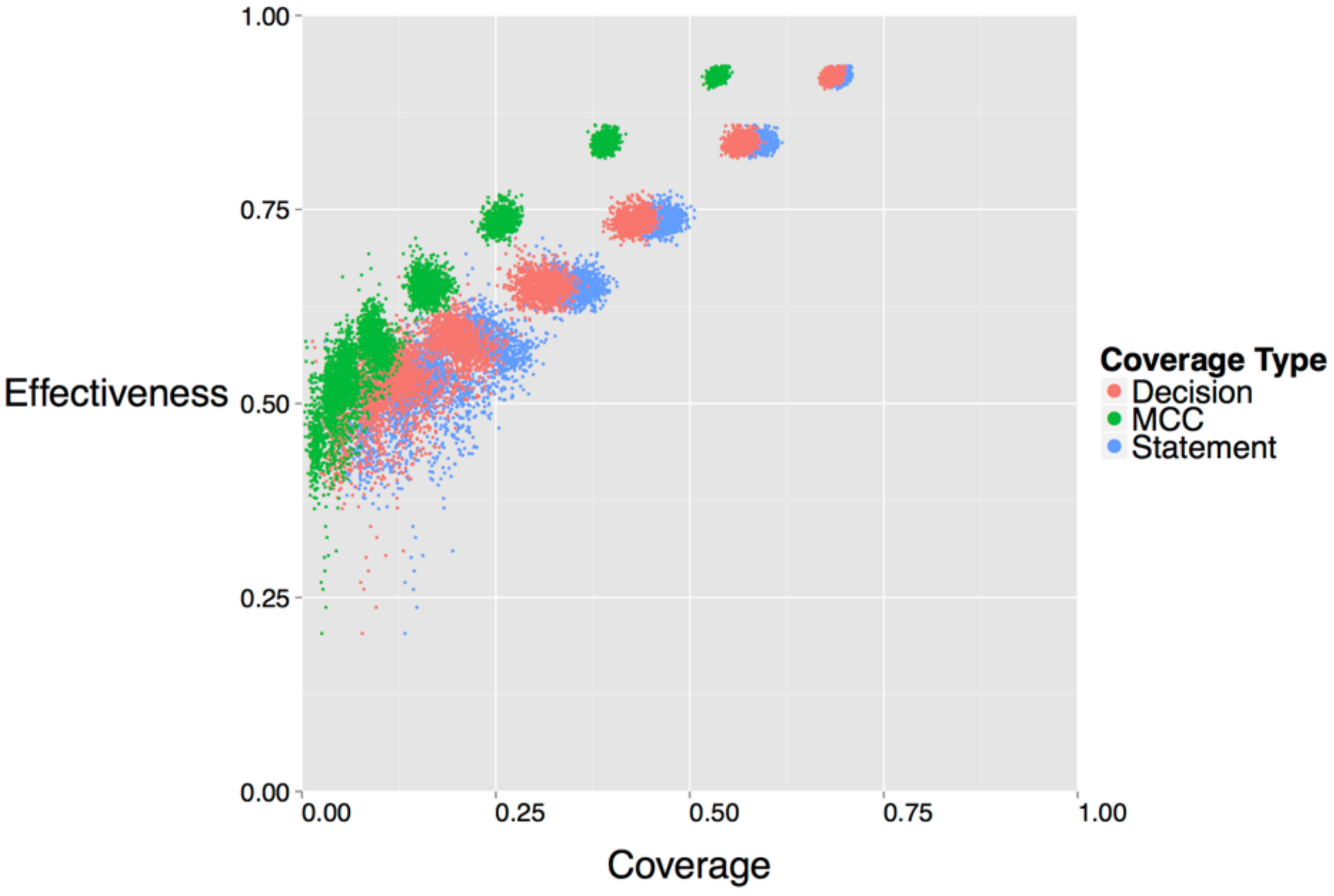
Method

1. Select programs to study
2. Make test suites
3. Measure suite coverage
4. Measure suite effectiveness
   - % mutants detected
   - Representative of fault detection effectiveness

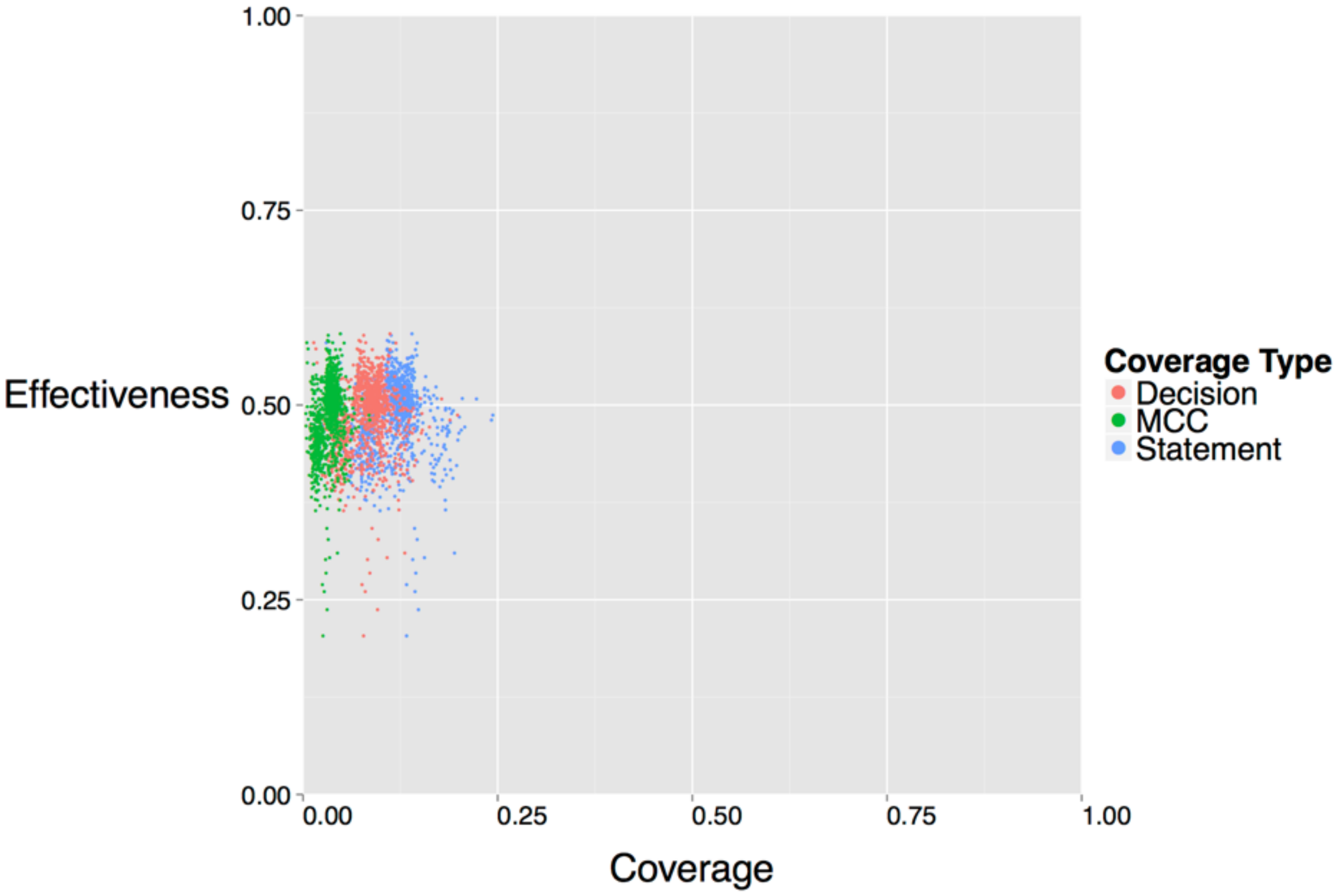
Method

1. Select programs to study
2. Make test suites
3. Measure suite coverage
4. Measure suite effectiveness

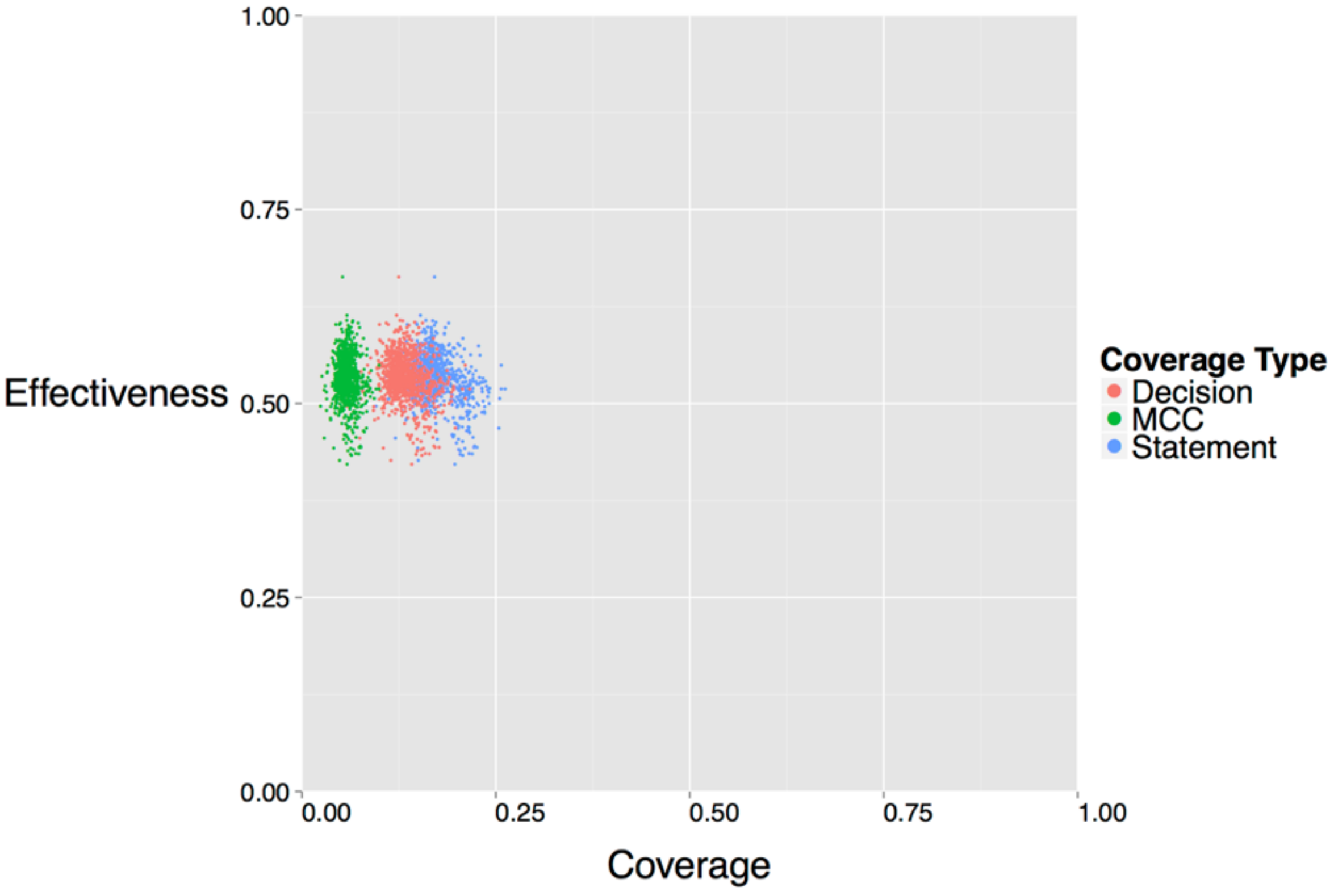
Closure: All Sizes



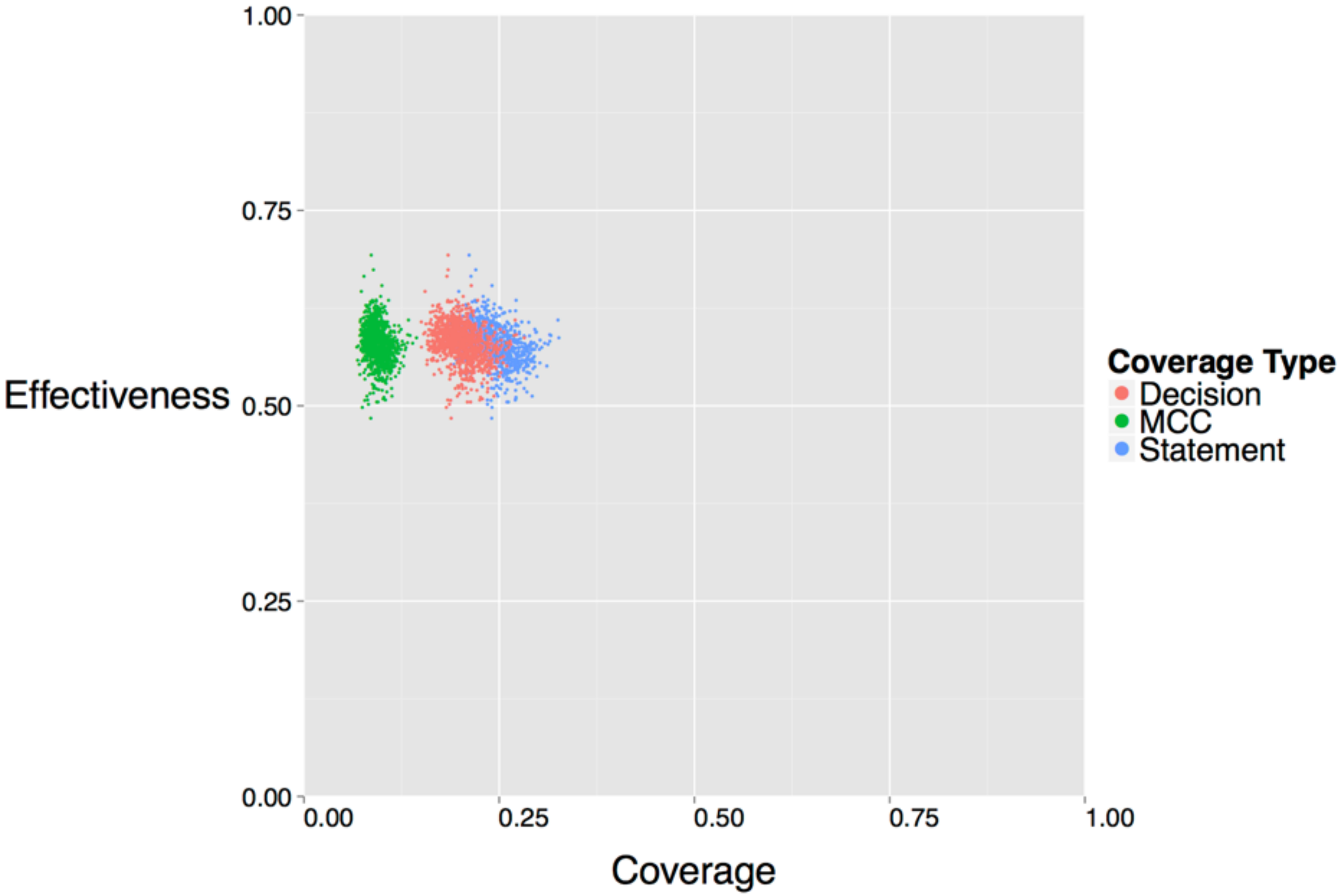
Closure: Size 3 Suites



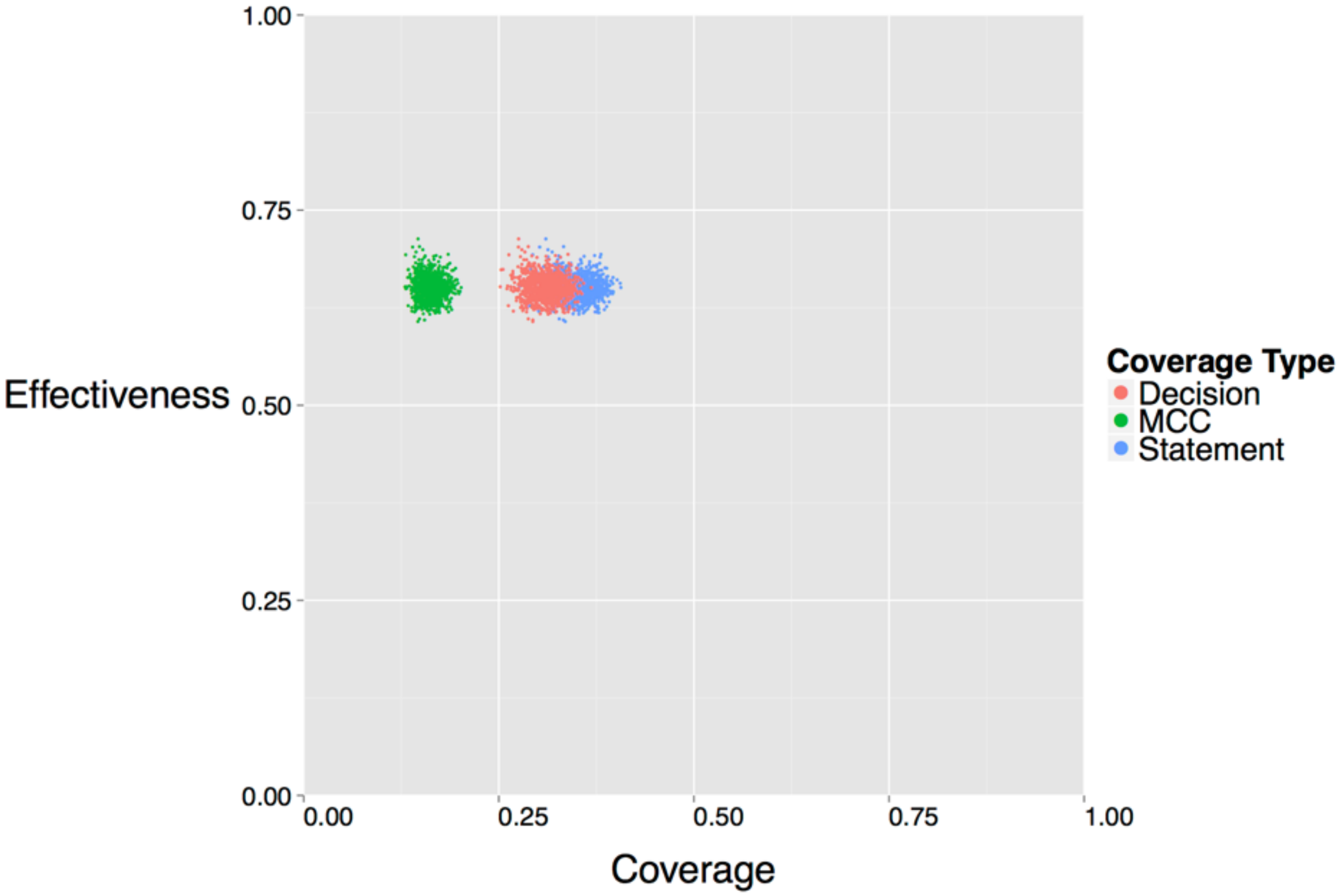
Closure: Size 10 Suites



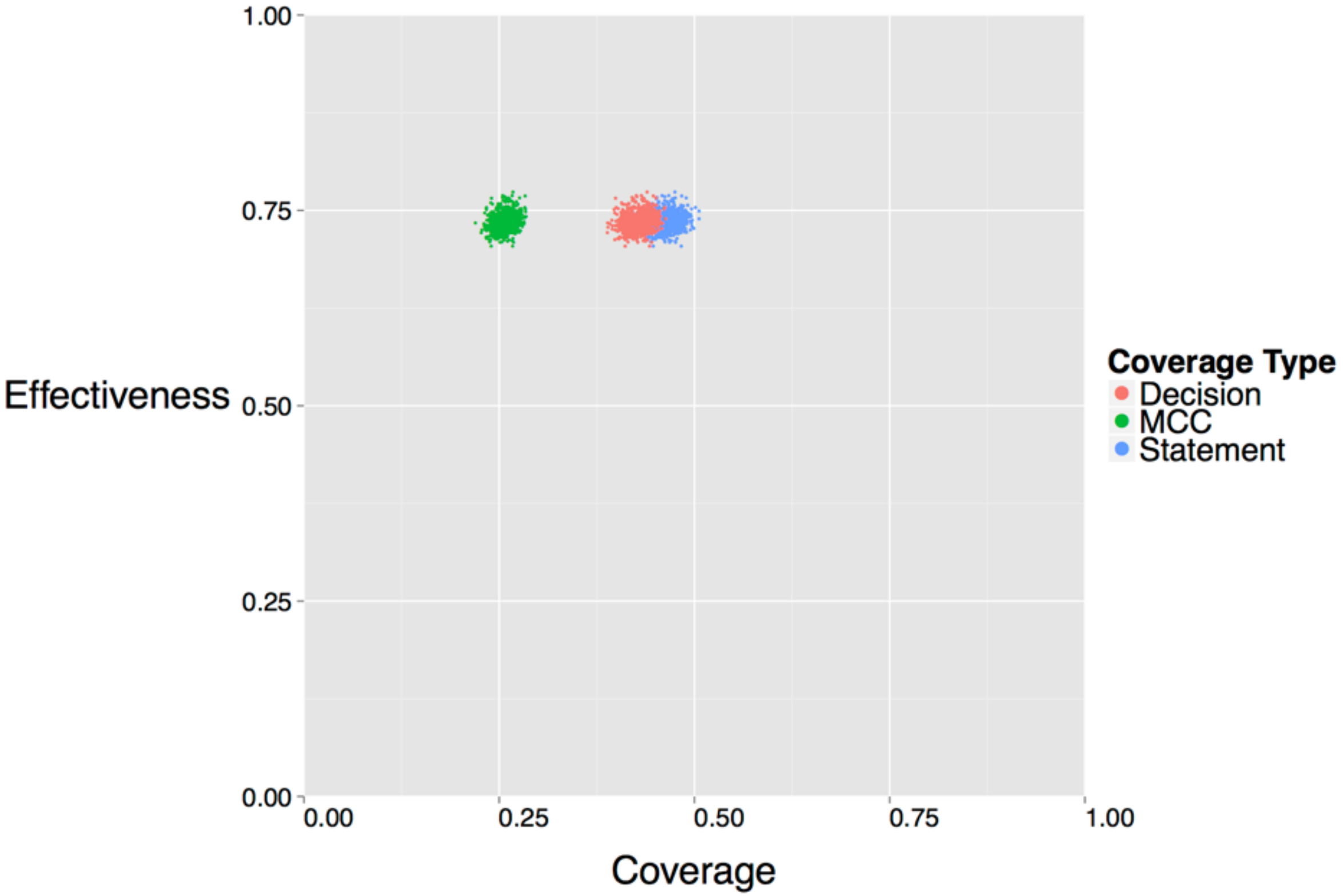
Closure: Size 30 Suites



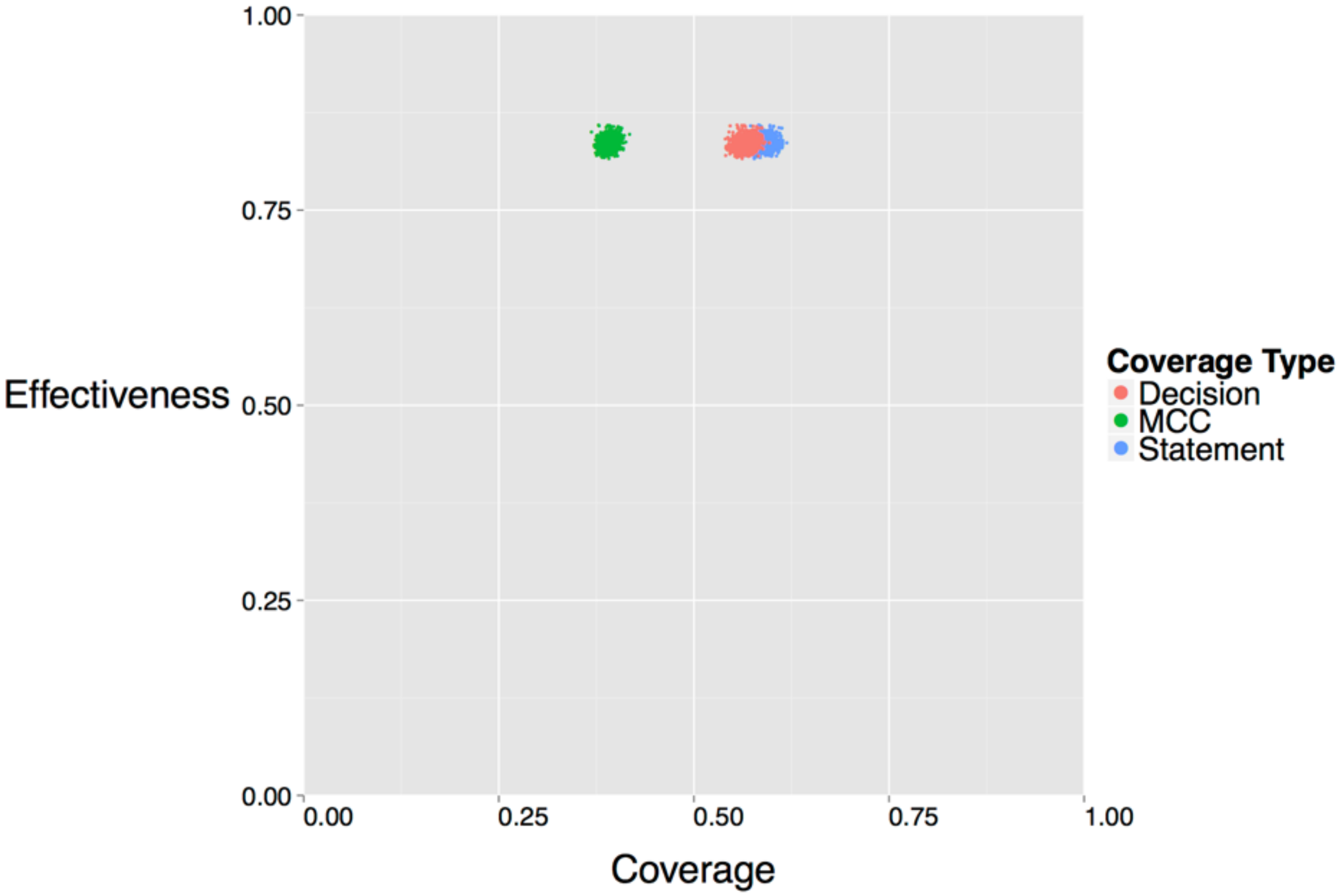
Closure: Size 100 Suites



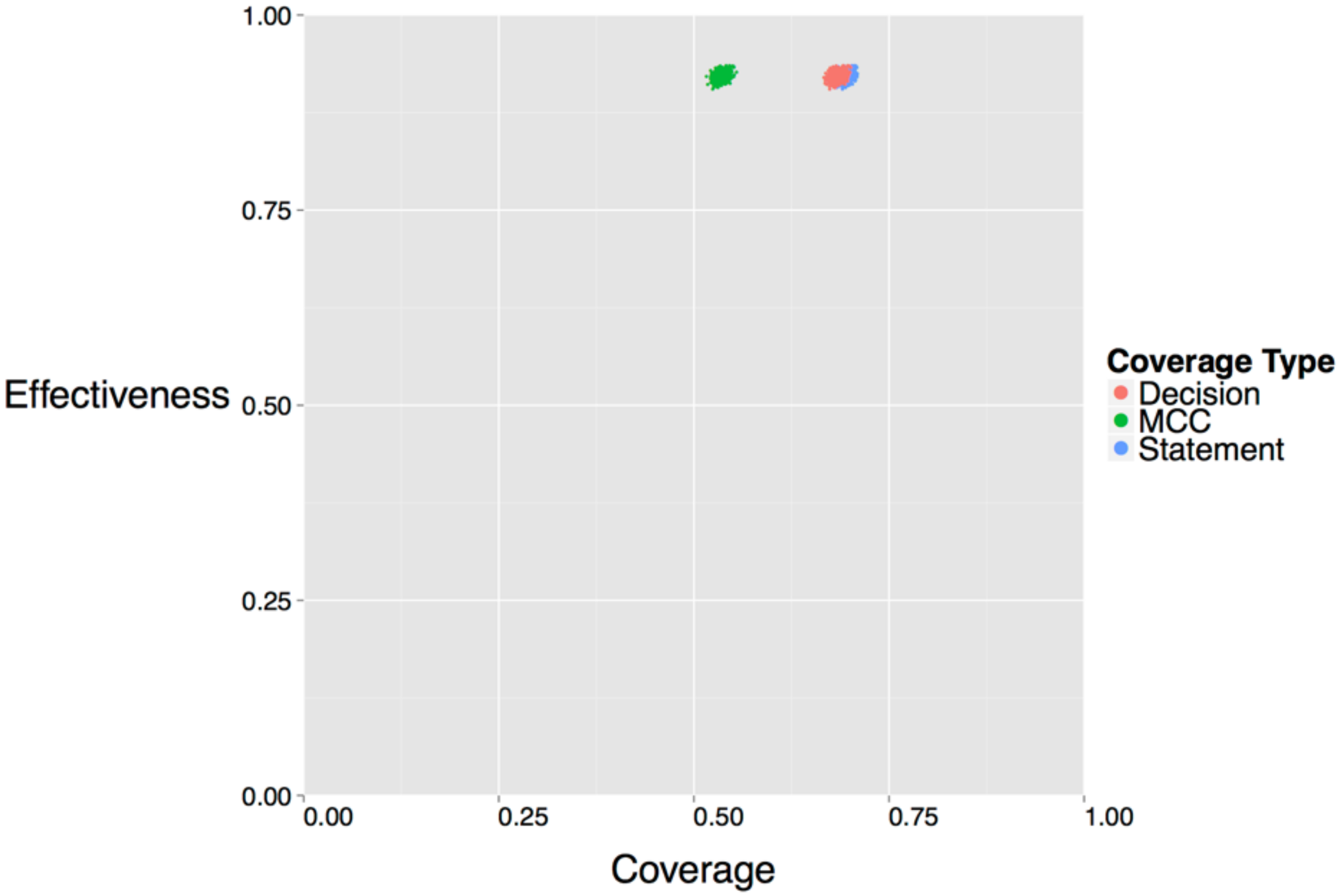
Closure: Size 300 Suites

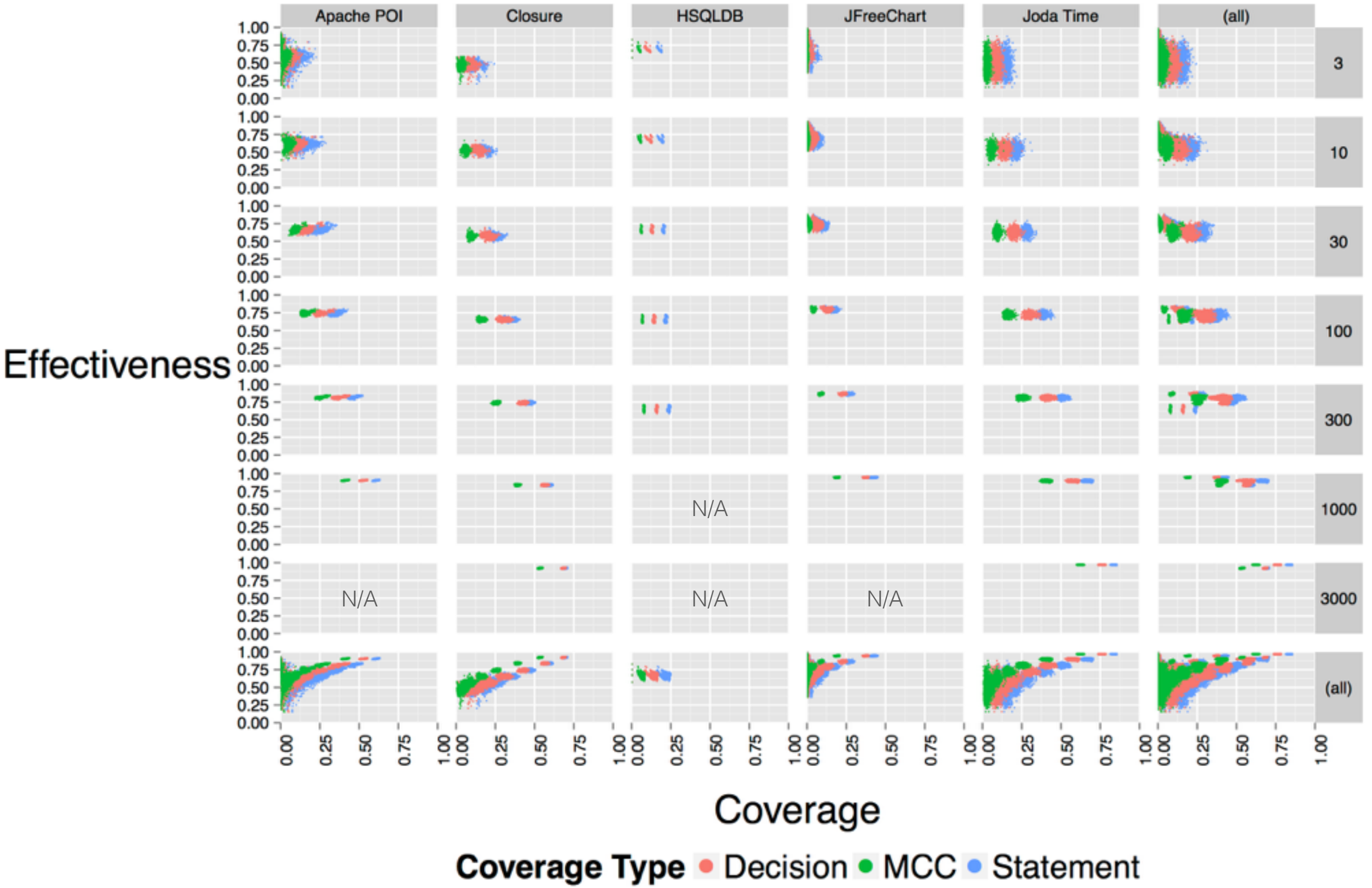


Closure: Size 1000 Suites

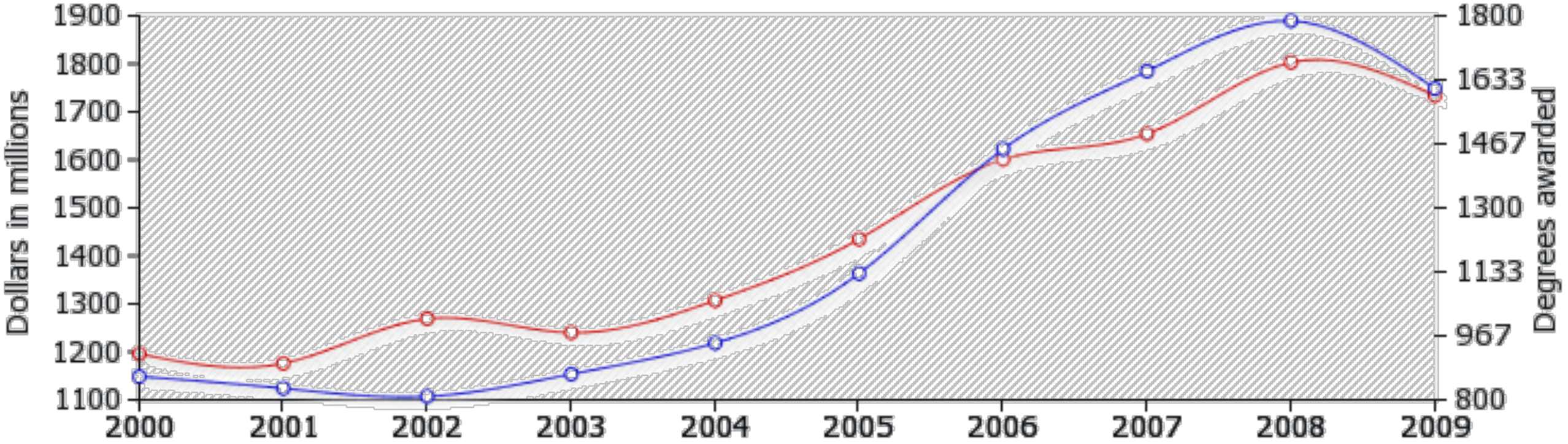


Closure: Size 3000 Suites




Apache POI
Closure
HSQLDB
JFreeChart
Joda Time
(all)
3
10
30
100
300
1000
3000
(all)
N/A
N/A
N/A
N/A
Effectiveness
Coverage
Coverage Type
Decision
MCC
Statement

Result 1: coverage is not strongly correlated with effectiveness when suite size is controlled



[http://www.tylervigen.com/]

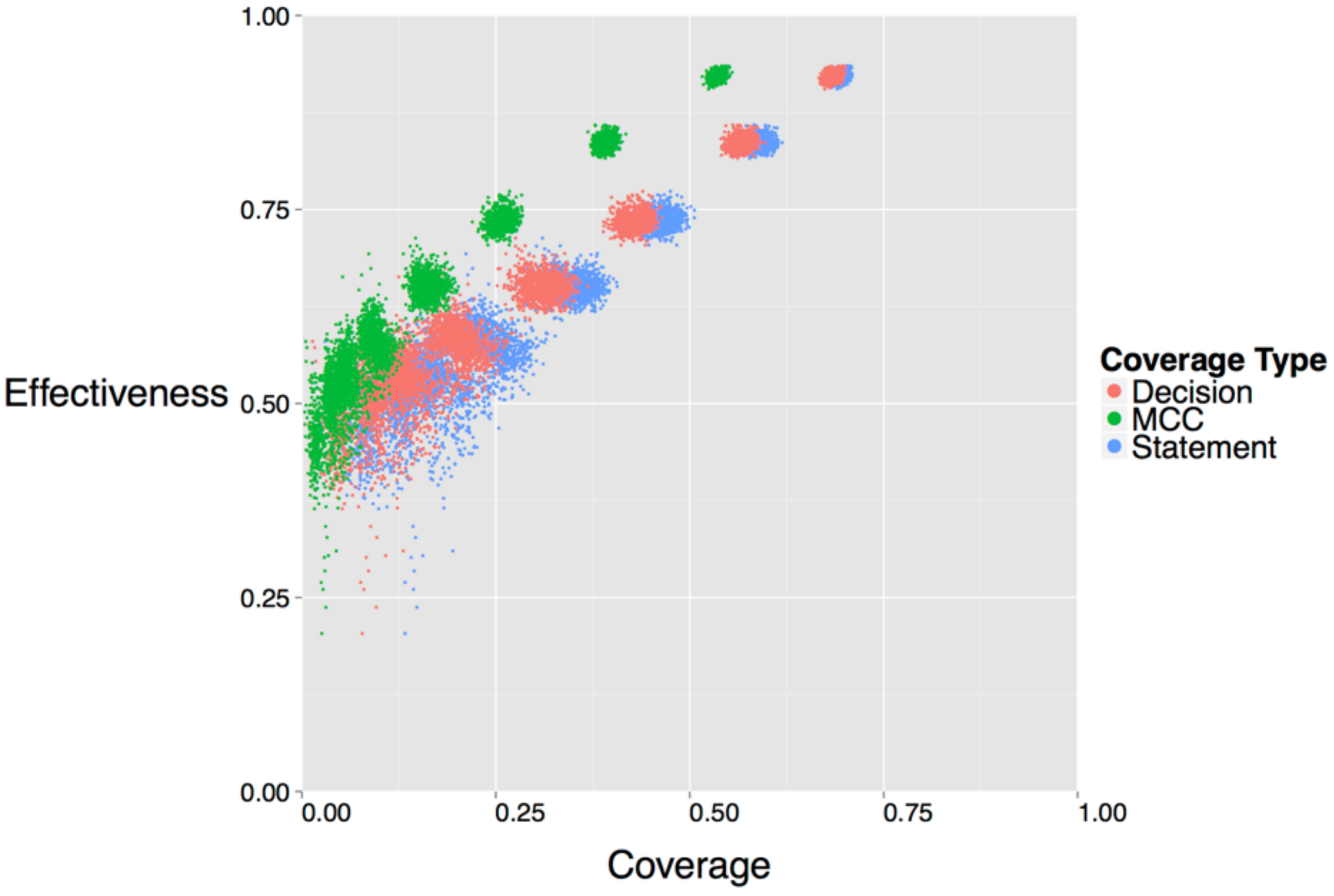
- Total revenue generated by arcades (US)
- Computer science doctorates awarded (US)

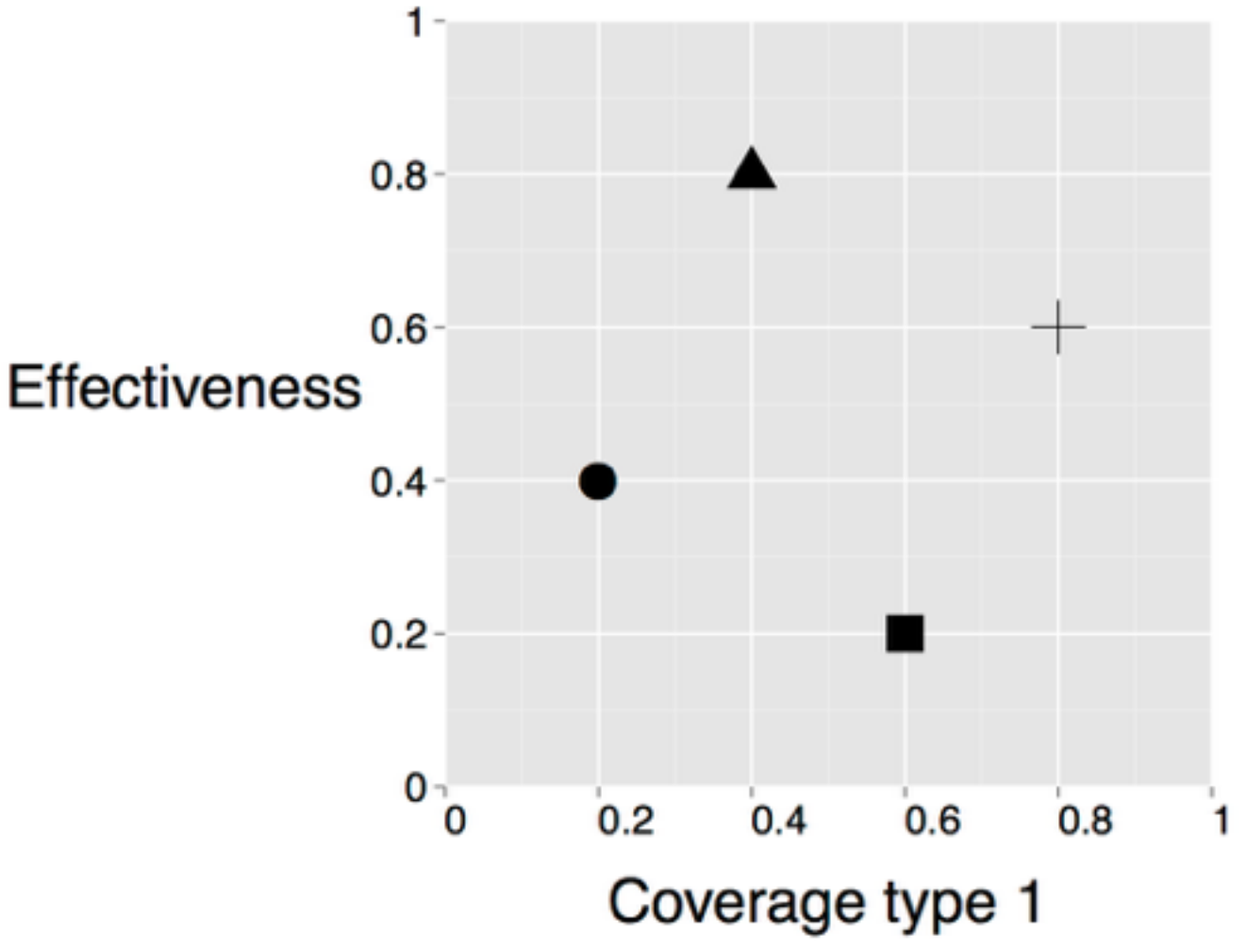
“Our test suite has 70% coverage, so it will catch a lot of bugs.”

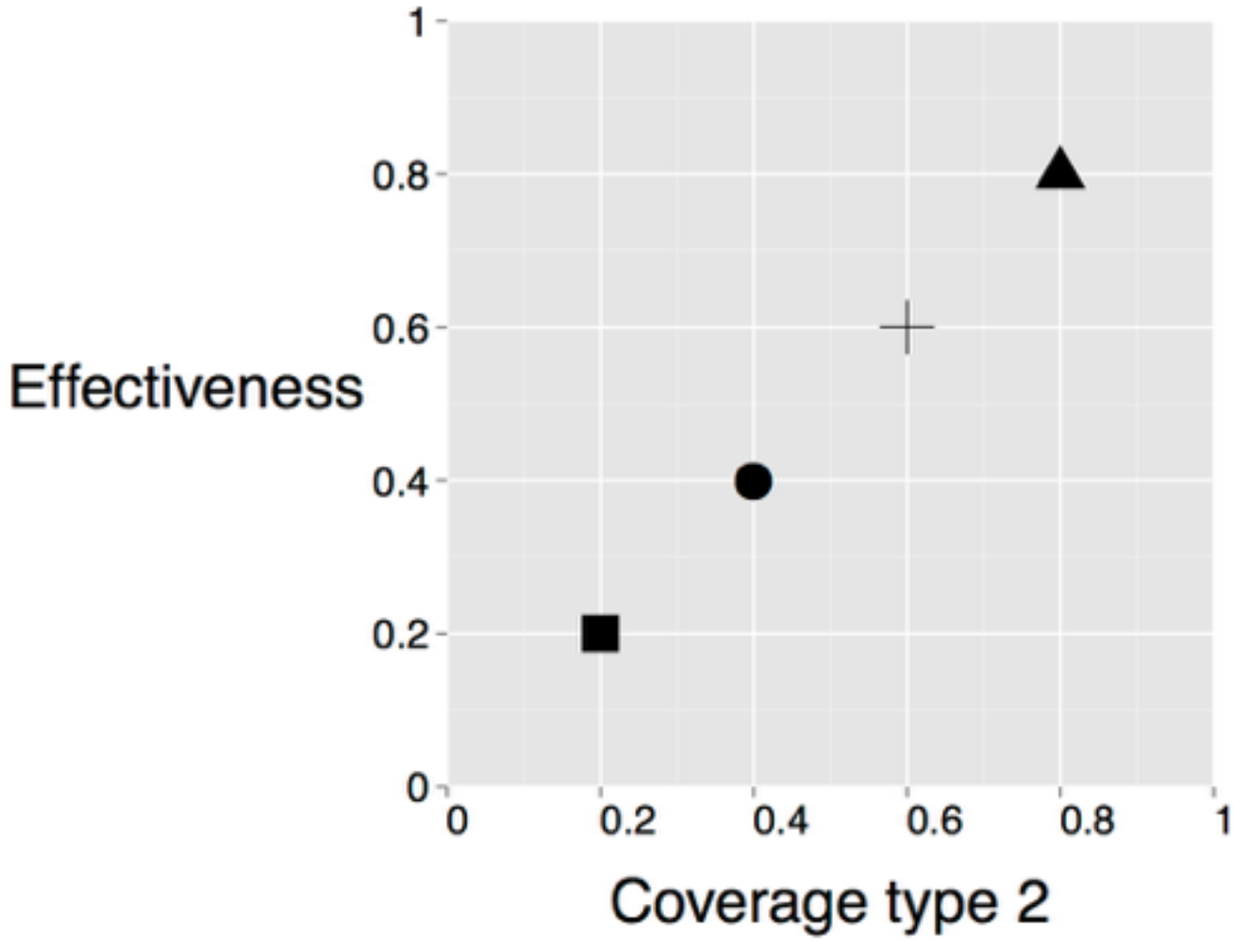
≡

“Arcades made $2B last year, so a lot of PhD students will graduate.”

Closure: All Sizes




Effectiveness
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
Coverage type 1


Effectiveness
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
Coverage type 2

Coverage Types	Correlation (Kendall Tau)
Statement/Decision	0.92
Decision/MCC	0.91
Statement/MCC	0.92

Result 2: stronger coverage types provide little extra information about non-adequate suites

Are Mutants a Valid Substitute for Real Faults in Software Testing?

René Just, Darioush Jalali, Laura Inozemtseva, Michael D. Ernst, Reid Holmes and Gordon Fraser.

[FSE 2014]

Using Fault History to Improve Mutation Reduction

Laura Inozemtseva, Hadi Hemmati, and Reid Holmes.

[FSE New Ideas 2013]

High coverage does not guarantee high quality.

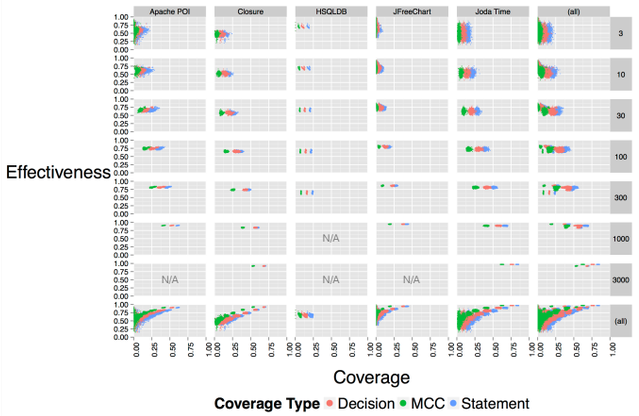
lorg.org

Year	Corr?	Large Programs	Realistic Suites	Suite Size Controlled
1993	~	✖	✖	✔
1994	✔	✖	✖	✔
1994	✔	✖	~	✔
1997	~	✖	✖	✔
1998	~	✖	✖	✔
1999	✖	✖	✔	✔
2005	~	✖	~	✖
2006	✔	✖	~	✔
2009	~	✖	~	✔
2013	✔	✔	~	✖
2014	✔	✔	✔	✖

Method

1. Select programs to study
2. Make test suites
3. Measure suite coverage
4. Measure suite effectiveness

16



25

Are Mutants a Valid Substitute for Real Faults in Software Testing?

Rene Just, Darioush Jalali, Laura Inozemtseva, Michael D. Ernst, Reid Holmes and Gordon Fraser.

TR; FSE submission.

30

Raw kill score: number of mutants killed/number of mutants generated

Normalized kill score: number of mutants killed/number of mutants covered

Condition	MCC	MCDC
Every point of entry and exit in the program has been invoked at least once	Yes	Yes
Every decision in the program has taken all possible outcomes at least once	Yes	Yes
Every condition in a decision in the program has taken all possible outcomes at least once	Yes	Yes
Every combination of condition outcomes within a decision has been invoked at least once	Yes	No
Every condition in a decision has been shown to independently affect that decision's outcome	No	Yes

But Aerospace!

- 100% MCDC coverage may be correlated with faults detected, but it is not necessarily causal
- Aerospace has different timelines, budgets, development processes, hiring standards, …