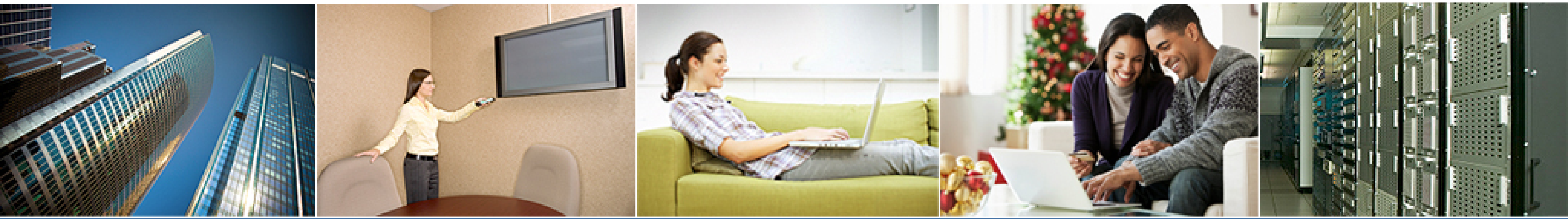


Dynamic Regression Suite Generation Using Coverage-Based Clustering

Shahid Ikram, Jim Ellis

Outline

- Objective
  - AKA “Problem we are trying to solve”
- Motivation
  - Aha, insight
- Background
- Our Methodology
- Case Studies
- Results and Summary
- References

OBJECTIVES

Verification Environment

- DUT, configurations, stimuli, checkers, coverage.
- A test checks a design for different configurations using different stimuli.
- A regression test-suite consists of tests, needed to be run for any changes in design.
  - Categorized like soft, hard, publish, nightly, golden etc.
  - Created using the designer and verifier's insight.
  - Mostly a static entity e.g. whole publish suite is run to publish any changes in design.

Quality of Verification

- In the absence of tools for complete verification, the quality of a verification environment is measured by
  - Coverage of the design features
    - Code (line, toggle etc.) , functional coverage.
  - Checkers
    - Assertions, protocol checkers, coherence checkers etc.
  - Fault simulations
    - Faults insertion, activation, detection.

An Optimal Regression(1)

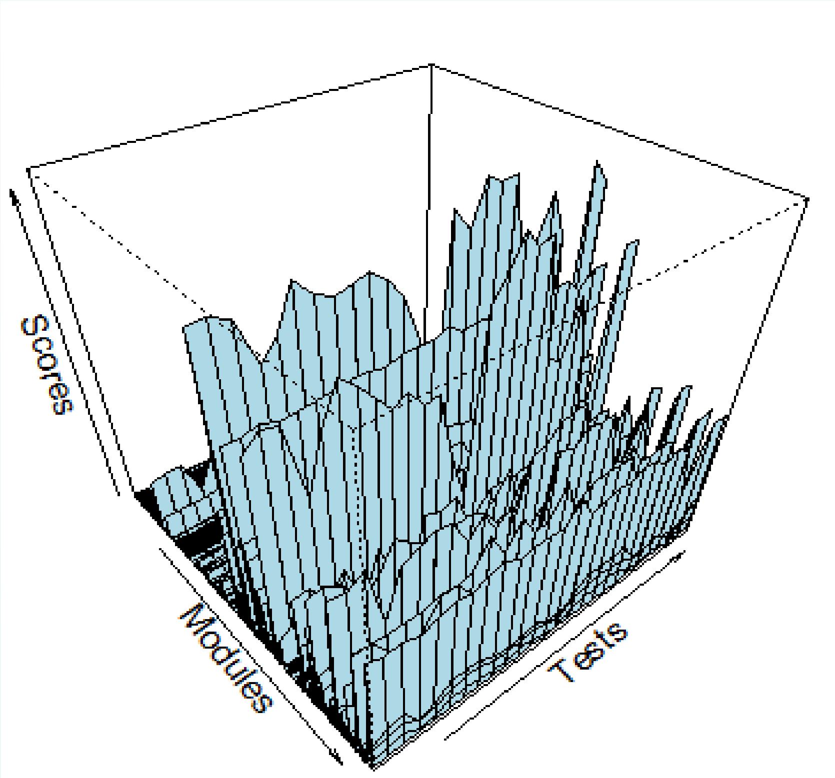
- An optimal regression-suite consists of only the needed tests and nothing less or more.
  - It has to be dynamic because changes in design occur randomly throughout the design.
- Can we create a sub-list of the tests related to the changes being submitted?
  - Hence save time and other resources like machines, licenses.
  - Effectively, reduce the time to market?

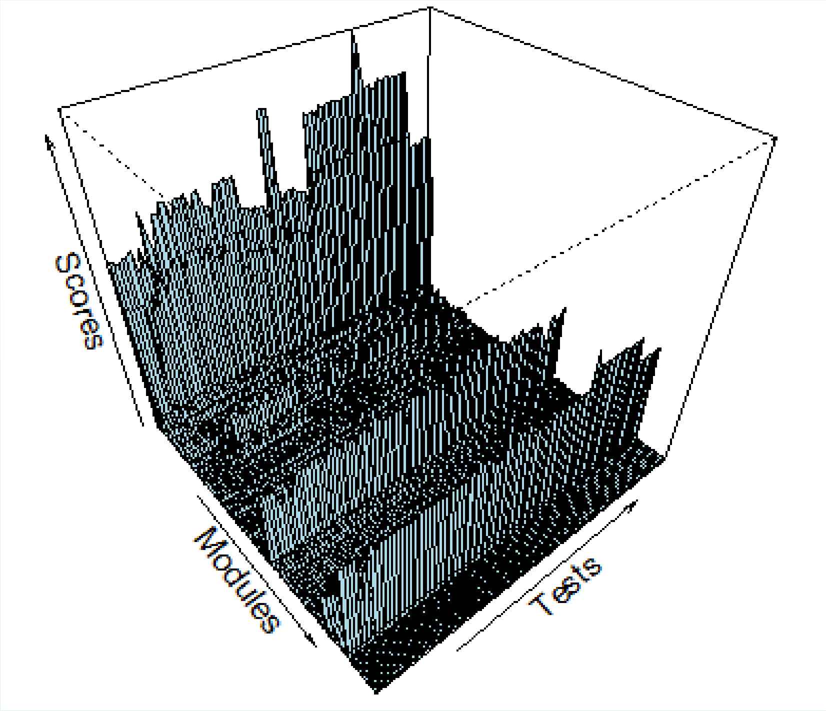
An Optimal Regression(2)

- When RTL is modified, we want to select a subset of the available test-suite that:
  - Ensure the coverage of the changes made.
  - Ensure the functionality related to the modified RTL is not broken.
  - And take minimal resource in terms of run time, licenses, debug time etc.

MOTIVATION

Aha Moment!





Observations/Questions

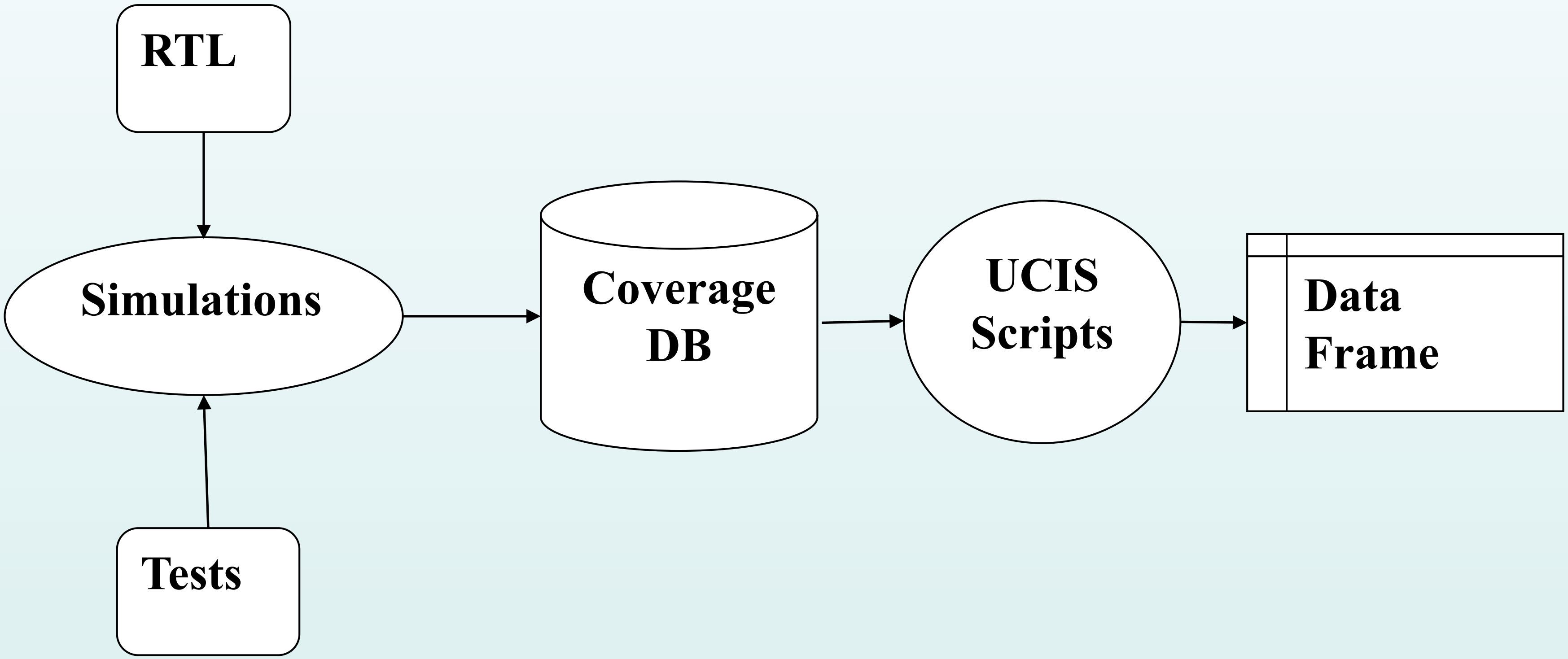
- Some modules showing similar coverage behavior for the given tests suites.
  - Are these modules functionally related?
- But to qualify this observation we need to answer two questions:
  - Are these patterns a statistical coincidence or these modules are correlated?
    - Needs empirical case studies.
  - If these modules are indeed correlated, can we algorithmically identify them?
    - Machine learning

METHODOLOGY

Plan of Action

- Designs selection
  – Designs with matured test suites.
- Simulations and data collection
  – UCIS to collect coverage data.
- Data analysis
  – R(a data modeling/processing/mining language).
- Data comparison
  – Fault simulations, to compare the quality of dynamically generated suites with original suites.

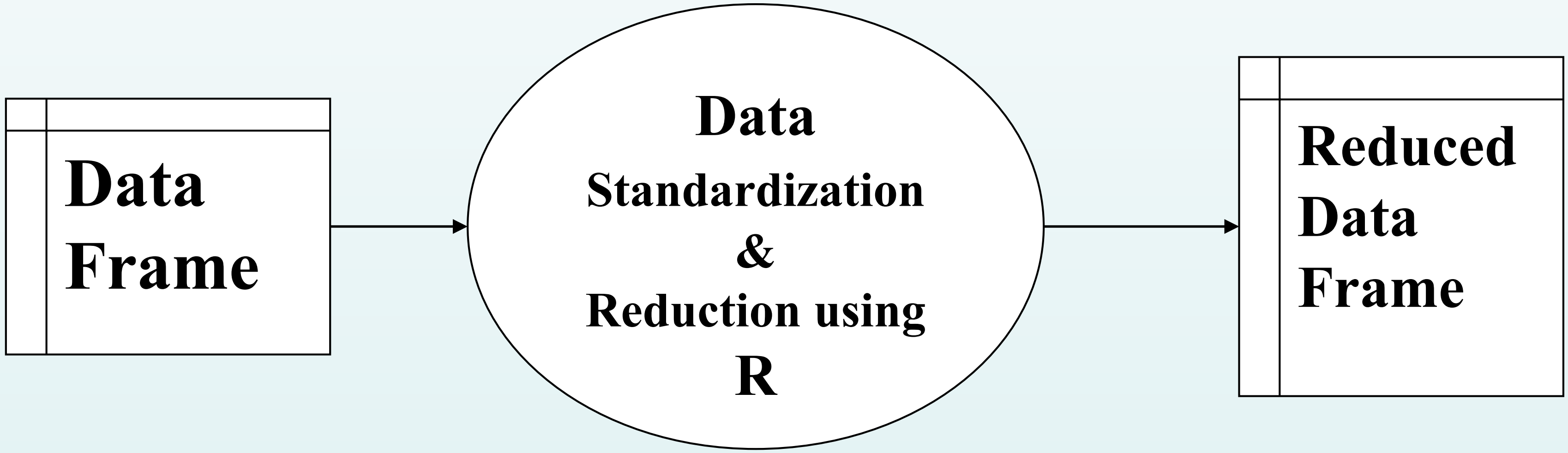
Data Collection



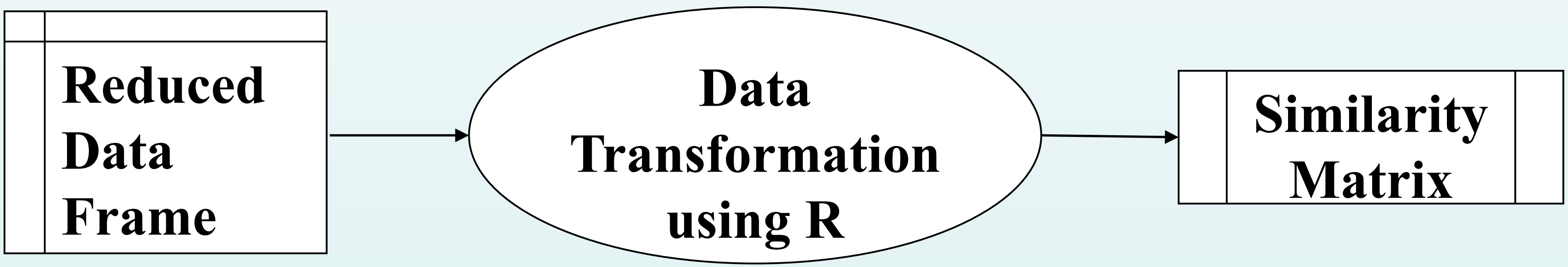
Coverage DataFrame

Module	Test	Time	Line	Toggle	Branch
Bisr_ctl	Bar0_d	987	28	14	28
Bisr_dp	Bar0_d	987	13	914	4
Arb_rot	Bar0_d	987	5	3	2

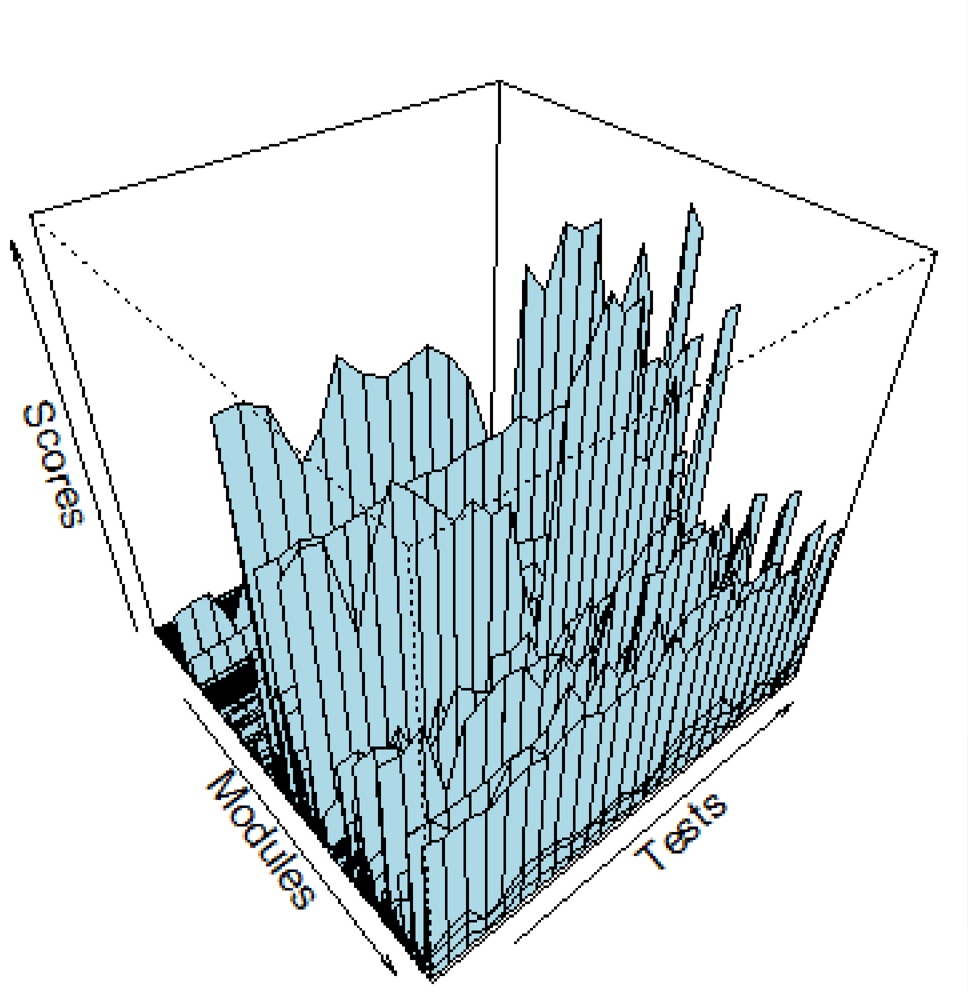
Data Standardization



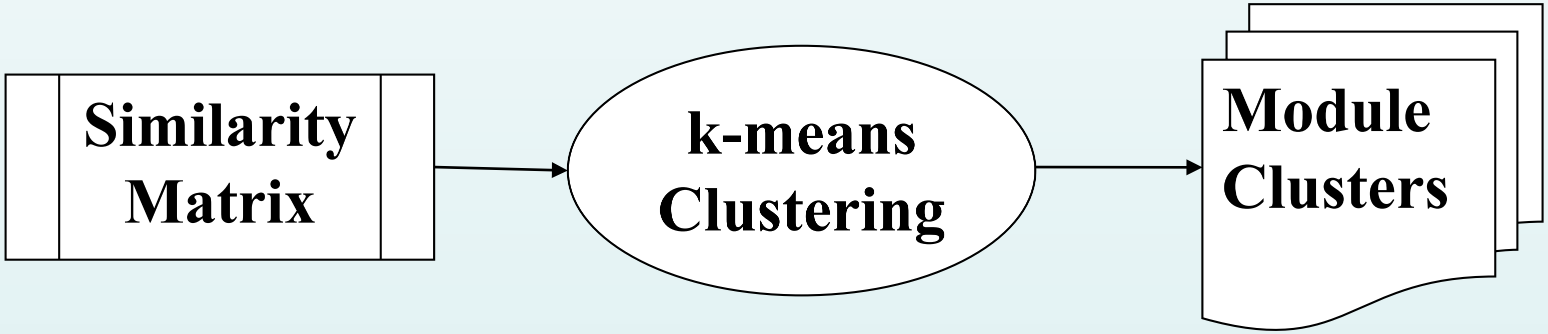
Similarity Matrix



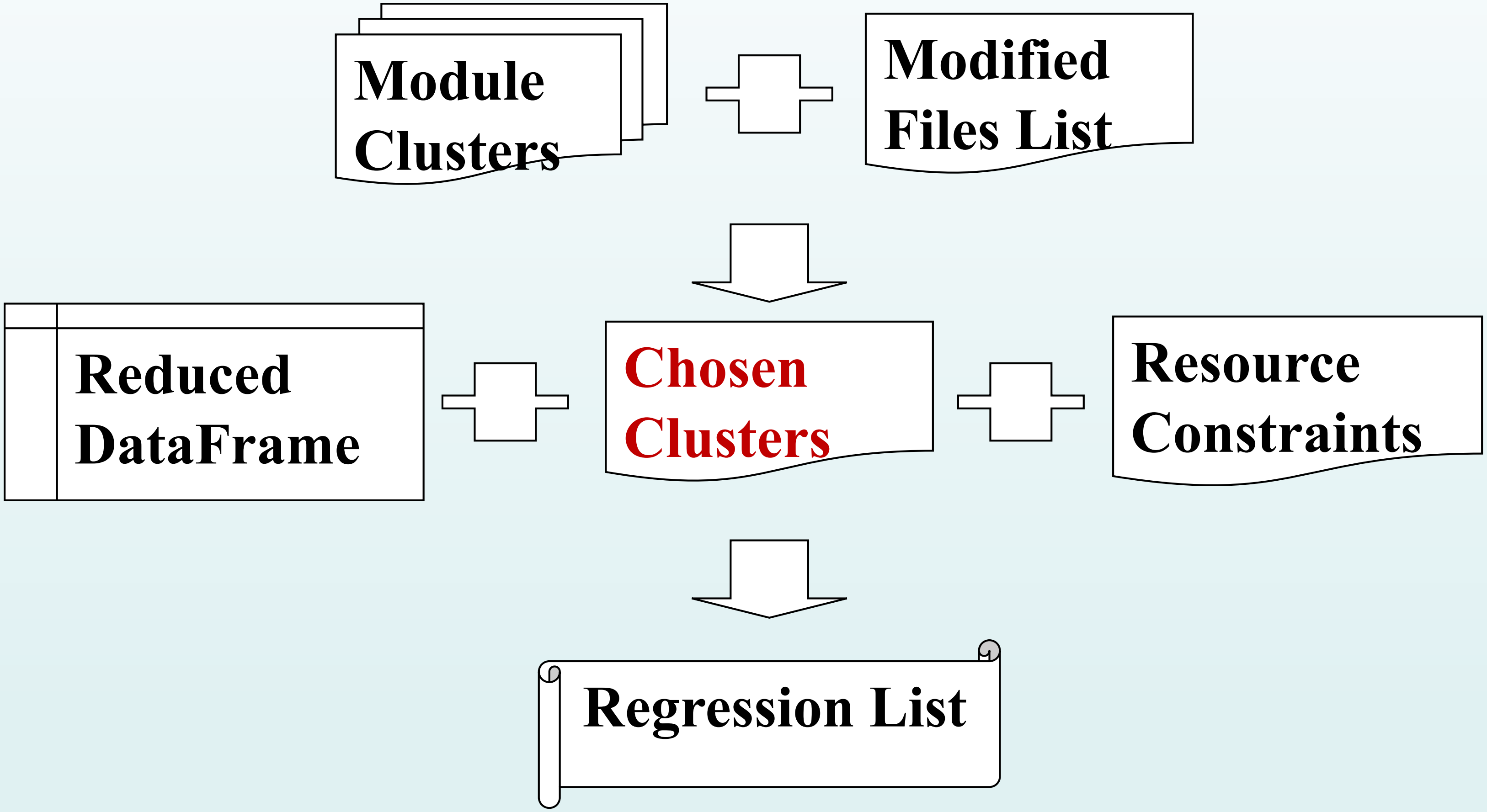
A Sample Similarity Matrix



Module Clustering



Regression List Generation



Tests List for a Cluster

Tests	aScore	sd	time	CV
basic_msix	2482	269	1005	75.99
basic_swi_rsp_err	2482	269	1476	75.99
basic_prp_sml_qsize	2479	268	1239	75.87
perf_bw_basic_1_1028	2478	268	2039	75.88
basic_multi	2475	267	2323	75.75

CASE STUDIES

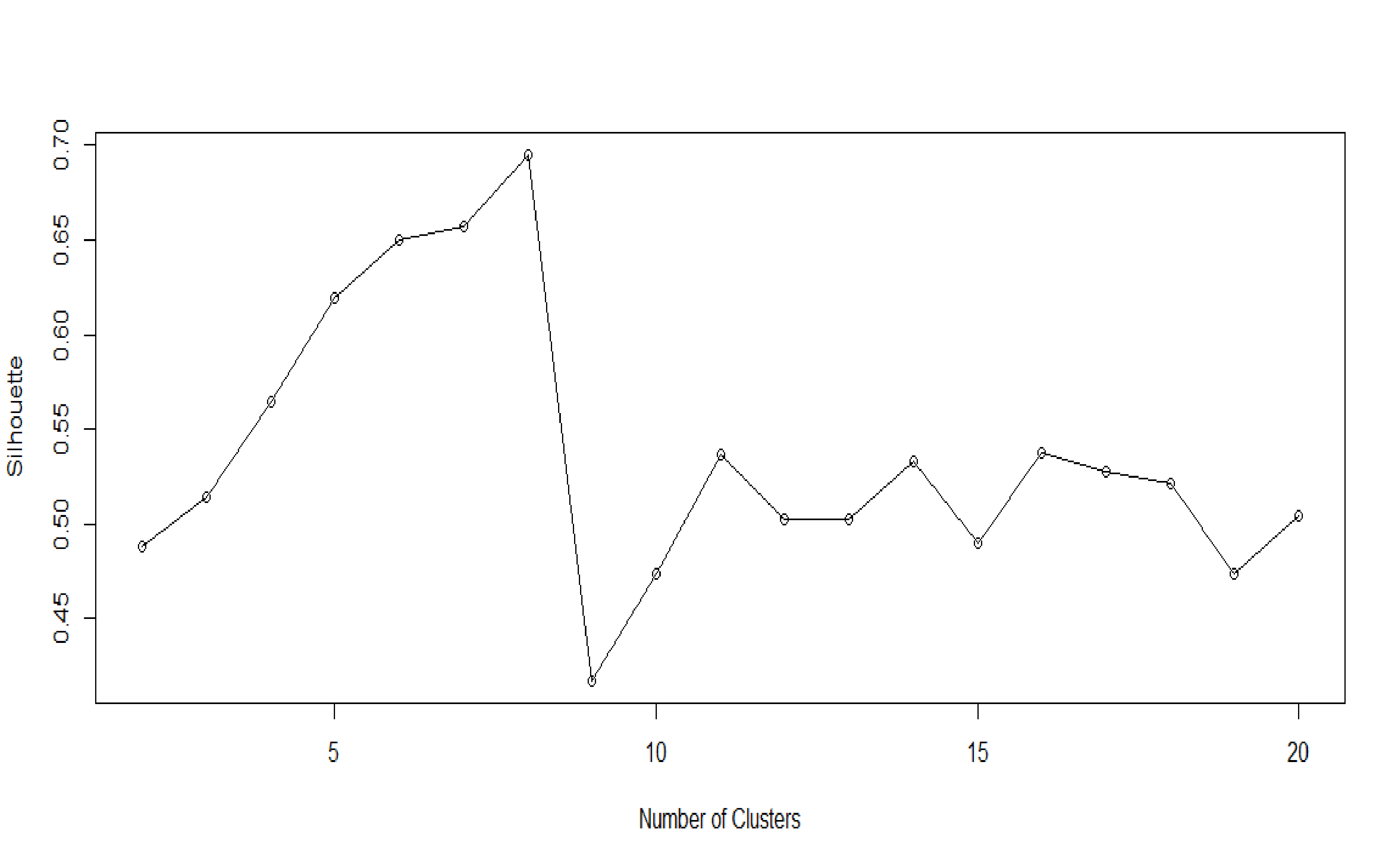
Experimental Setup

- How to compare the quality of dynamically generated tests list with the statically generated?
  - Verification environment is kept the same.
  - Selection of tests is different.
- A fault simulator called Certitude.
  - Artificially induced faults and a statistical quality measure called ComputeMetric.
    - A sample of faults to get the statistical quality measure within predefine error limits.

Case Study A

- A queue management and submission block.
- 231 modules with 13811 instances.
- Verification environment consists of 45 tests.
- After data reduction phase, we were left with 34 modules that were related to Design A feature set.

Finding Optimal Clustering



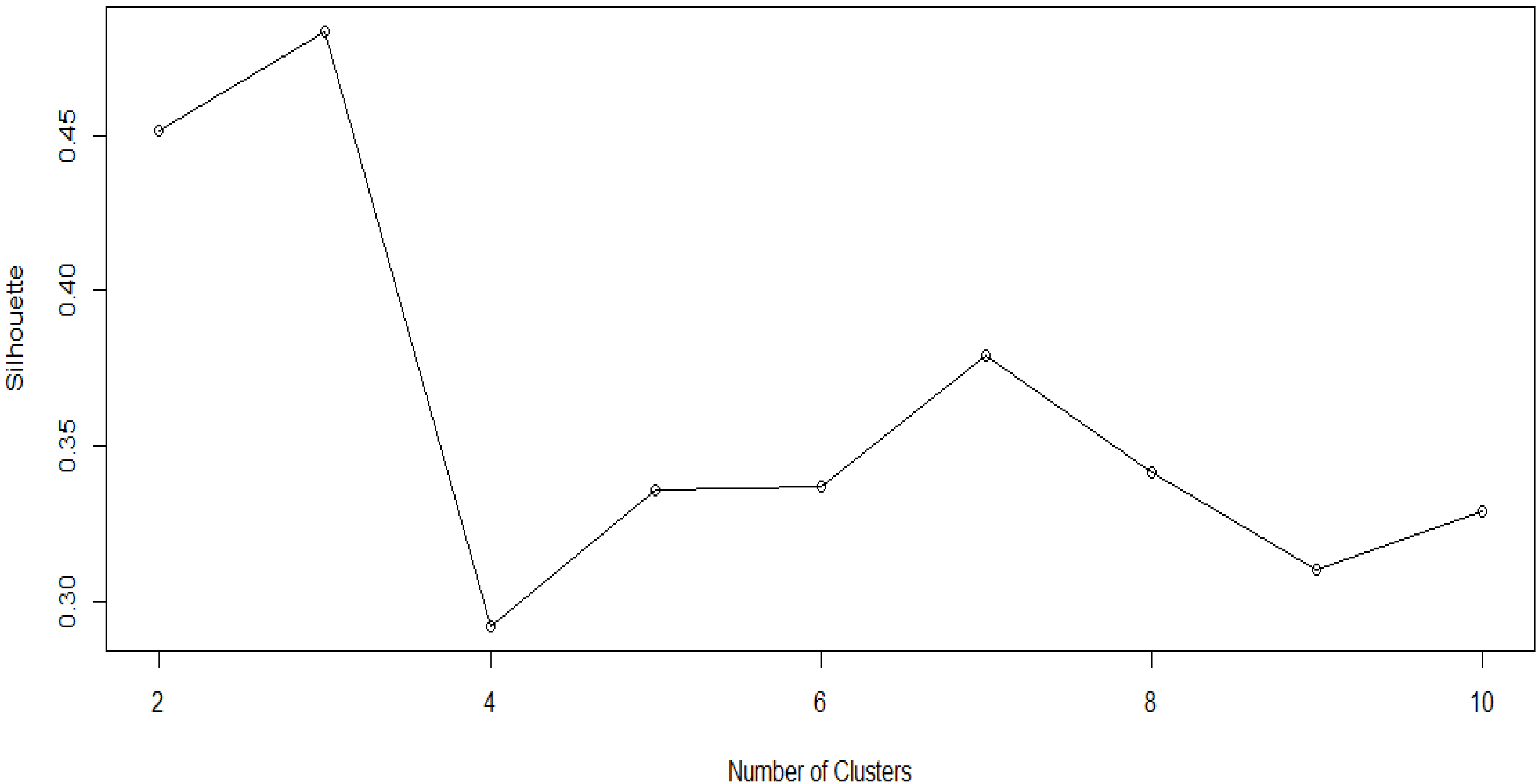
Study Results

	Number of Modules	Number of Tests	Activation Score	Propagation Score
Full-Block	34	45	94.60%	76.02%+-4.54
Cluster-1	7	10	94.33%	74.32%+-4.44
Cluster-2	11	10	96.45%	73.55%+- 4.75
Cluster-3	7	10	94.33%	74.28%+-4.32

Case Study B

- A serial line interface.
- 296 modules.
- Verification environment consist of 112 tests.
- After the data reduction phase, the number of modules left were 92.

Finding Optimal Clustering



Study Results

	Number of Modules	Number of Tests	Activation Score	Propagation Score
Full-Block	92	112	92.31%	86.43%+-4.72
Cluster-1	18	20	97.30%	85.43%+-4.44
Cluster-2	8	20	96.45%	83.55%+-4.75
Cluster-3	11	20	95.43	82.78%+-4.56

RESULTS AND SUMMARY

Results

- Got statistically equivalent results using dynamic clustering methods with
  - Fewer tests,
  - Less cumulative run time,
  - Fewer licenses.
- Most importantly,
  - Saving in terms of human resources as fewer tests to look at.

Summary

- Regression are key to success of a project.
- Regressions' evolve with the project and may include redundant/overlapped tests that results in wastage of resources like time, money and humans.
- Faster and precise regression creation methods can be helpful.
- We have a shown one method to improve the process. Much more can be done.

Acknowledgements

- Monica Farkash’s pioneering work.
- R language developers and resources.

References

- Yehia, Ahmed. "*UCIS Applications: Improving Verification Productivity, Simulation Throughput, and Cover Closure Process*" DVCON 2014, San Jose, 2014.
- Synopsys, Inc., "*Coverage Technology User Guide*," Synopsys, Inc., Mountain View, CA, 2012
- Farkash, Monica etc., "*Mining Coverage Data for Test Set Coverage Efficiency*", DVCON 2015, Santa Clara, CA.
- Farkash, Monica etc., "*Regression Optimization using Hierarchical Jaccard Similarity and Machine Learning*", DAC 2013, Austin, TX.
- Arafeen, M.J., and Do, Hyunsook. "*Test Case Prioritization Using Requirements-Based Clustering*", IEEE Sixth International Conference on Software Testing, Verification, and Validation (ICST).
- Synopsys, Inc., "*Certitude User Manual*," Synopsys, Inc., Mountain View, CA, 2013