Yet another correlation analysis of mutation score, fault detection, and test suite size

CSE 599 course porject, Yiqun Chen[1,*]
Supervisor: René Just

1 University of Washington, Seattle

***yiqunc@uw.com**

Abstract

Finding best practice in test creation has been an active research area for software engineering domain for decades. A good test suite, by definition, should detect real faults. However the target of interest is usually unknown to the programmers a priori, which prompts the use of mutants, which are artifically generated faults. Mutation score based testing uses mutants as proxies for measuring strength of a test suite, and recently researchers have made multiple attempts to validate the merit of mutant score based testing strategy. This report aims to look at the relationship between fault detection, mutation scores, and size of test suites systematically and investigate a few claims made by Papadakis et al. **[citation]**. Specifically, they argued that while mutation score is highly correlated with fault detection, the correlation decreases after adjusting for test suite sizes. This observation leads to their conclusion of test suite size being a confounder, and the aim of this report is to demonstrate both conceputal and statistical analysis pitfalls in their work and offer alternative measures of correlation.

1 Introduction

This paper repeats and refutes the recent result from Papadakis et al. To answer the question of whether mutation scores are really indicative of real fault detection probability, they conducted a series of correlation based analysis and logistic regression modeling.

They report a drastic drop of correlation between fault detection and mutation scores after The major claims of their paper can be broken down as the following:

- Claim 1: Correlation between mutation score and real fault detection is induced by a confounding variable, test suite size.
- Claim 2: Both mutation score and test suite size are statistically singnificant predictors of real fault detection, but both are not very practically significant.
- Claim 3: Prioritizing test suites with higher mutation scores leads to an improved real fault detection compared to random selection.

For the remainder of the report, we will focus on investigating claim 1 and 2 and answer the following research questions (RQ):

- RQ 1: Are the results presented to support claim 1 in the original paper reproducible? If so, how sensitive are the results to choices of parameters such as sampling ratio, normalization of mutation scores, and etc.
- RQ 2: Controlling for test suite size explicitly is just a poor man's version of many statistical techniques such as multivariate regression or partial correlation. Given that the maximal Pearson correlation between a continuous variable and a binary variable is less than 1, would the same results hold for other measures of correlation?
- RQ 3: It's well-known that correlation doesn't imply causation; however, it's often neglected that a decrease of correlation after stratifying on a variable does not make the stratifying vairable a confounder. We instead propose the concept of coupling and a plausible causal pathway to formalize the idea of causality in this case.

1.1 Terminology

Before moving on to the constructive section of the paper, we first introduce a list of terms we will use for the rest of discussion, many of which will be illustrated using the kill matrix example in Table 1.

Test	Real Fault	Mutants			
	f	m_1	m_2	m_3	m_4
t_1	1	1	1	0	0
t_2	0	0	1	0	0
t_3	0	0	0	1	0

Table 1. Example Test-mutant matrix (kill matrix)

- Project: refers to the class of Java programs used in the study (Chart, Closure, Lang, Math, Time)
- Task: individual bugs studied ($n = 231$)
- (Real) fault: a single bug unique to each task.
- Fault detection (FD): we say a test suite detects the fault (equivalently FD=1) if at least one of the tests detects the fault. For instance, the suite $\{t_1, t_2, t_3\}$ detects the fault since $f = 1$ for t_1, but the subset $\{t_2, t_3\}$ does not detect the fault (FD=0).
- Mutation score (MS):
  - for a given suite of mutants $\mathcal{M}$, the un-normalized mutation score for a test suite T on $\mathcal{M}$ is total number of mutants detected by the individual tests. For instance, consider $\mathcal{M} = \{m_1, m_2, m_3, m_4\}$ and $T = \{t_1, t_2\}$, then the un-normalized mutation score would be 2 since m_1 and m_2 are detected.
  - for a given suite of mutants $\mathcal{M}$, the normalized mutation score for a test suite on $\mathcal{M}$ is the ratio of total number of mutants detected by the individual tests, divided by total number of detectable mutants by this test suite. Again, consider $\mathcal{M} = \{m_1, m_2, m_3, m_4\}$ and $T = \{t_1, t_2\}$, the normalized mutation score would be $\frac{2}{3}$ since m_4 cannot be detected by any tests in the entire test pool (t_1, t_2, t_3).

- Size of a test suite: number of tests in the suite.
- For a collection of tests T or mutants $\mathcal{M}$, we use $|T|$ or $|\mathcal{M}|$ to denote the cardinality of the set
- For a given real fault, tests in a suite can be categorized into either fault triggering or non-triggering. We use G to denote the subset of tests that detects the fault.
- Coupling probability: for a given test suite T, a fault f, and a mutant m, the coupling probability of a mutant to a real fault (we will suppress the dependence on T for notational convenience) is defined to be:

$$P_{m,f} = P(t \text{ detects } f | t \text{ detects } m), t \overset{\text{Uniform}}{\sim} T \tag{1}$$

  For instance, if we consider $T = \{t_1, t_2, t_3\}$ and fault f in Table 1, $P_{m_1,f} = 1$ and $P_{m_2,f} = \frac{1}{2}$. As for mutants that are not detected at all such as m_4, we use the convention that $P_{m_4,f} = 0$.

- Estimating coupling probability: for any existing test suite $T_0 \subset T$, one can estimate this probability by its empirical counterpart:

$$\hat{P}_{m,f} = \frac{\hat{P}(t \text{ detects } m \text{ and } f)}{\hat{P}(t \text{ detects } m)} \tag{2}$$

- Perfect coupling: we say a mutant m is perfectly coupled to a fault f if $P_{m,f} = 1$. Specifically note that this coincides with the familiar concept of coupling in the literature if one consider the test suite T to be all the tests of interest. For instance, if we consider $T = \{t_1, t_2, t_3\}$, m_1 and f are perfectly coupled.
- Perfect de-coupling: we say a mutant m is perfectly de-coupled to a fault f if $P_{m,f} = 0$ but $P(m \text{ detected }) > 0$. For instance, if we consider $T = \{t_1, t_2, t_3\}$, m_3 and f are perfectly de-coupled but m_4 is not.
- Probabilistic coupling: we say a mutant m is probablistically coupled to a fault f if $P_{m,f} \in (0, 1)$.
- Pearson correlation: for any two given random variables X, Y, Pearson correlation is defined to be

$$Cor(X, Y) = \rho_{X,Y} = \frac{\mathbf{E}[(X - \mathbf{E}X)(Y - \mathbf{E}Y)]}{\sqrt{Var(X)Var(Y)}} \tag{3}$$

  where $\mathbf{E}$ denotes the expectation and $Var(\cdot)$ denotes the variance of a random variable. Specifically note that Pearson correlation is always bounded between [-1,1] and attains the upper/lower bound only if Y is a linear function of X.
- Empirical Pearson correlation: in practice we just take empirical average and variance to obtain the estimated correlation,

$$\hat{Cor}(X, Y) = \hat{\rho}_{X,Y} = \frac{\sum_{i=1}^{N}[(X_i - \bar{X})(Y_i - \bar{Y}]}{\sqrt{(\sum_{i=1}^{N}(X_i - \bar{X})^2)(\sum_{i=1}^{N}(Y_i - \bar{Y})^2)}} \tag{4}$$

  where $\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$.

- Partial correlation: we will only discuss partial correlation for 3 random variables in this report but in general, partial correlation (abbreviated as $PCor$) measures the association between two random variables with with the effect of a set of controlling random variables removed. It is similar to adding a controlling variable in multiple regression but provides a measure of the strength of relationship, instead of effect size.

  Consider three random variables X, Y, Z where Z is the control variable. Then partial correlation of X and Y after controlling for Z can be expressed numerically as:

$$PCor_Z(X,Y) = \rho_{X,Y|Z} = \frac{\rho_{X,Y} - \rho_{X,Z} \cdot \rho_{Y,Z}}{\sqrt{(1-\rho_{X,Z}^2)(1-\rho_{Y,Z}^2)}} \tag{5}$$

  The intuition for partial correlation is that we are looking at the correlation between X and Y after removing the linear effect of Z in both variables. Consider an extreme example where both X and Y are linear functions of Z, then they have Pearson correlation of ± 1. However, after removing the linear trend of Z, the resulting partial correlation will be 0.

- Odds ratio: consider the following logistic regression models

$$logit(P(FD = 1)) = \beta_0 + \beta_1 \cdot MS \tag{6}$$

  and

$$logit(P(FD = 1)) = \beta_0^* + \beta_1^* \cdot MS + \beta_2^* \cdot Size \tag{7}$$

  where $logit(x) = \log(\frac{x}{1-x}), x \in [0, 1]$.

  - Unadjusted odds ratio for MS: e^{β_1}
  - Adjusted odds ratio for MS: $e^{\beta_1^*}$

2 Method

2.1 Sampling mechanisms

In this section, we introduce the following two sampling mechanisms for generating datasets. While the original paper used uniform 0-20% sampling for what they call "random" dataset and used 0.025-50%, with 2.5% spacing for generating "fixed" datasets, we decide to change the limit for the latter to 20% as well. The original paper does not mention any difficulties encountered during sampling but we note that when the sampling ratio is high ($> 10 - 20\%$ for most tasks), every sampled suite will almost always have at least one test that detects the fault and makes it impossible to compute statistics of interest.

In practice, we compute and record fault detection, mutation score, and size in each random samples for downstream analysis.

We provide below two examples where we listed only 3 out of 10,000 rows in a returned data frame.

2.2 Statistical methods

We consider the following methods, some of which are used in the original paper while the rest are proposals from our own.

Algorithm 1 Generating random size datasets

procedure RANDOMSIZE(Test suite T, Lower bound (LB) = 0,Upper bound (UB) = 0.2, $N = 10,000$)
    ResultList = []
    for $i = 1, \cdots, N$ **do**
        $R_i \sim Uniform(LB, UB)$
        $N_i = \lceil R_i \cdot |T| \rceil$
        Sample N_i numbers without replacement from $\{1, 2, \cdots, |T|\}$
        Append to ResultList the sampled subset of rows T_{N_i}.
    return ResultList

Algorithm 2 Generating fixed size datasets

procedure RANDOMSIZE(Test suite T, Sampling ratio R, $N = 10,000$)
    ResultList = []
    for $i = 1, \cdots, N$ **do**
        $N_i = \lceil R \cdot |T| \rceil$
        Sample N_i numbers without replacement from $\{1, 2, \cdots, |T|\}$
        Append to ResultList the sampled subset of rows T_{N_i}.
    return ResultList

Task	Fault detection	Mutation score	Size
Chart-1	1	300	893
Chart-1	1	312	1085
Chart-1	1	286	943

Table 2. An example for test suites generated from a random size approach

Task	Fault detection	Mutation score	Size
Chart-1	1	299	930
Chart-1	1	300	930
Chart-1	1	327	930

Table 3. An example for test suites generated from a fixed size approach (with sampling ratio = 15%)

2.2.1 (Partial) correlation

Consider a sampled dataset with random size (e.g. Table 2), we compute the following correlation quantities:

- $\hat{\rho}_{MS,FD}$ and $\hat{\rho}_{MS,FD|Size}$
- $\hat{\rho}_{Size,FD}$ and $\hat{\rho}_{Size,FD|MS}$

Since for each fixed size samples, there is no variation in the size variable, we can only compute $\hat{\rho}_{MS,FD}$ and $\hat{\rho}_{MS,FD|Size} = \hat{\rho}_{MS,FD}$ in this case.

The intuition is that if partial correlation drops significantly compared to the marginal correlation, we have some confidence that the third variable size (or MS) plays a role in the effect of MS (or size) to real fault detection. Note that this is an extension to the correlation presented by the authors since they only compared the correlation computed with random size versus correlation computed with fixed size, which, as we will see in a bit, is a potentially misleading effect size measure.

2.2.2 Linear regression

We fit the following linear regression models to assess the effect size of mutation score on fault detection:

- Unadjusted: $P(FD = 1) = \beta_0 + \beta_1 \cdot MS$, with random size sampling
- Adjusted: $P(FD = 1) = \beta_0^* + \beta_1^* \cdot MS + \beta_2^* \cdot Size$,with random size sampling
- Fixed size: $P(FD = 1) = \tilde{\beta}_0 + \tilde{\beta}_1 \cdot MS$, with fixed size sampling

A natural comparison in this case will just be comparing the value of β_1, β_1^*, and $\tilde{\beta}_1$. Since the adjusted regression analysis is essentially stratifying on size, we expect β_1^* to be close to $\tilde{\beta}_1$ for a range of sampling ratio. If the unadjusted regression coefficient is much larger than the adjusted one, intuitively one would expect that a lot of association between MS and FD can be attributed to size.

2.2.3 Logistic regression

We fit the following logistic regression models to assess the effect size of mutation score on fault detection:

- Unadjusted: $logit(P(FD = 1)) = \beta_0 + \beta_1 \cdot MS$, with random size sampling
- Adjusted: $logit(P(FD = 1)) = \beta_0^* + \beta_1^* \cdot MS + \beta_2^* \cdot Size$,with random size sampling
- Fixed size: $logit(P(FD = 1)) = \tilde{\beta}_0 + \tilde{\beta}_1 \cdot MS$, with fixed size sampling

Similar to the linear regression case, a natural comparison in this case will just be comparing the value of $e^{\beta_1}, e^{\beta_1^*}$, and $e^{\tilde{\beta}_1}$ (the exponential of regression coefficient is also referred to as estimated odds ratios and is a popular way to summarize the association between a continuous/binary variable and a binary outcome). Again, since the adjusted regression analysis is essentially stratifying on size, we expect $e^{\beta_1^*}$ to be close to $e^{\tilde{\beta}_1}$ for a range of sampling ratio.

3 Results

3.1 Replicating the original paper

In this final report, due to space and time constraint, we did not put together a separate replication study. This, however, will be one of the to-do items for next steps.

We provide a "replication" of Table 3 in the original paper with un-normalized mutation scores instead of normalized ones (See Figure 1). we see that the same trend indeed holds, i.e., the random size sampling produces much higher correlation than the aggregated fixed size correlation. This piece of evidence underpins the essence of their claim that size is a confounder and mutation score is in fact not as correlated with real fault detection as previous scholars thought.

3.2 Extended correlation analysis

While controlling for size explicitly using an equally spaced sampling approach, we argue that one should consider the concept of partial correlation directly, which accounts for the linear effect of a potentially confounding variable (e.g. size in the

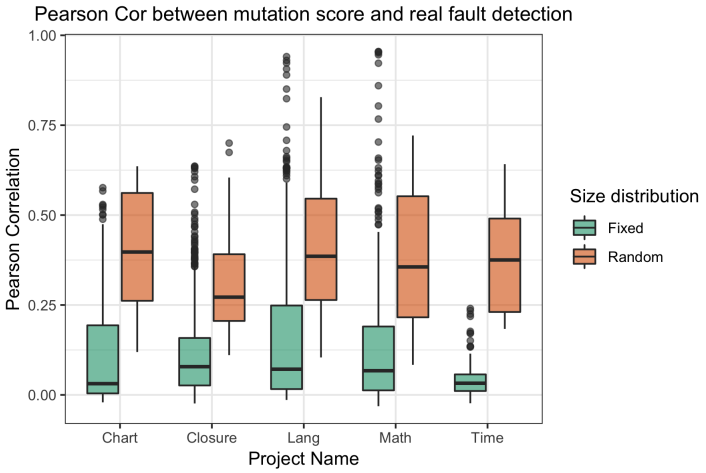

Figure 1. Comparing correlations between mutation score and real fault detection for random and fixed size sampling

original study). In addition to the partial correlation of MS and FD controlling for size, we also consider the partial correlation of size and FD controlling for MS.

Note that for some of the datasets due to the imbalance in the outcome (virtually all samples will detect the fault), one cannot compute correlation between fault detection and mutation score and we excluded these in our analyses below.

The motivation for the latter analysis is to quantify the effect of mutant score when considering the correlation of size and fault detection, which is neglected and would be revealing: due to the lack of causal framework, in the previous studies, a confounder is established using the phenomenon of reduced correlation. However, if the same effect is shown in size as well, by the same logic, we would arrive at the conclusion that mutation score is a confounding variable in the causal path from test suite size to fault detection as well. However, these two claims cannot be simultaneously true and reveal a fundamental shortcoming in the 2015 paper: causation is not equal to correlation and a causal pathway should come from reasoning with domain knowledge instead of being generated with data.

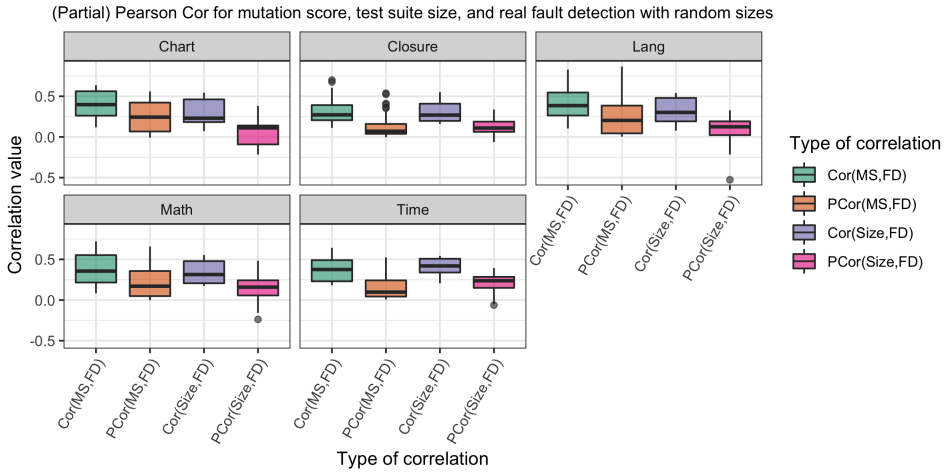

Figure 2. Comparing all four (partial) correlations for (mutant score,fault detection) and (test suite size, fault detection)

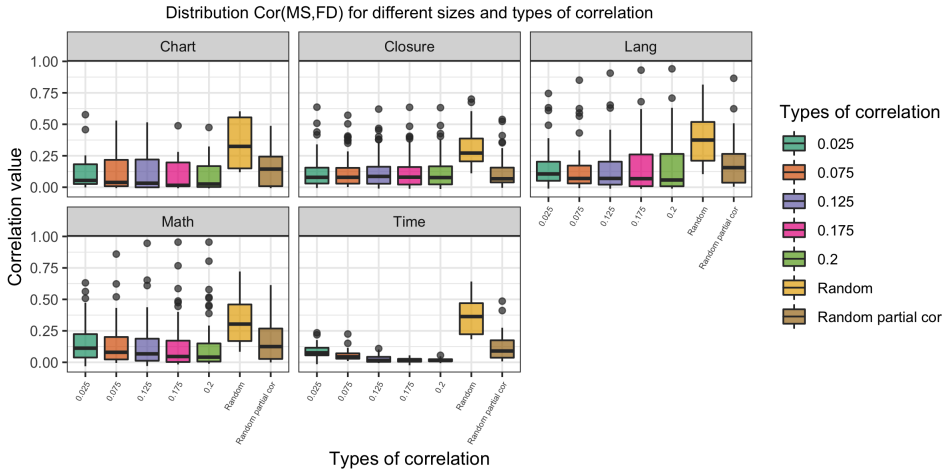

Figure 3. Comparing all four (partial) correlations for (mutant score,fault detection) and (test suite size, fault detection) with random and fixed size sampling

We see from Figure 2 that across all projects, partial correlation adjusted for the other variable (MS or size) in general decreases. However, we see that the maginitude of decrease seems to be more pronounced for the correlation between size and fault detection adjusting for mutation score! If anything, one could argue that mutation score is a confounder for the pathway of $size \to FD$.

In Figure 15, we only look at the correlation between mutation score and fault detection with different sampling schemes. Note that partial correlation in principle achieves what the fixed size sampling aims to do, except that we are looking at some discretized value of uniform distribution instead of a conitnuous density on $(0, 0.2)$. Toward this end, we argue that if one is really interested in using correlation as an effect size measure, which in the next section we will demonstrate a few reasons not to, computing the partial correlation directly would be a preferred alternative.

3.3 Investigating the behavior of maximal correlation

3.3.1 Pearson correlation is bounded

While it's well-known that a Pearson correlation is bounded between -1 and 1, the same limit does not apply to the correlation of a dichotomous variable and a continuous variable. **[cite the paper]** Specifically, the maximal correlation between a normally distributed random variable X and a Bernoulli random variable Y with success probability p is given by

$$\rho_{X,Y;\max} = \frac{1}{\sqrt{2\pi p(1-p)}} \exp(-\frac{1}{2} z_{1-p}^2) \tag{8}$$

where z_{1-p} is the $1-p$ quantile of a standard normal distribution.

Therefore, it is not relevant to refer to a 0.8 Pearson correlation being strong and 0.4 correlation being weak; one need to evaluate the strengthen in view of the upper limit. This statistical fact will be relevant after we introduce the class imbalance problem below: note that if the probability of detecting a fault fails into either extremely high (≥ 0.99) or extremely low (≤ 0.01), the maximal achievable correlation would be bounded by 0.26, which is conventionally considered as a low correlation coefficient.

3.3.2 Two-stage sampling model

In this section, we use a two-stage sampling model to characterize the limit of the Monte Carlo simulation used both in method section and in the original paper. Consider the entire test suite T_f for a fault f and a fixed sampling ratio r, the Monte Carlo sampling without replacement really is to approximate the underlying distribution of all $\binom{|T_f|}{\lceil r \cdot |T_f| \rceil}$ subsets of size $\lceil r \cdot |T_f| \rceil$. Now for given r and T_f, the real fault detection outcome is either 0 or 1, which is a Bernoulli random variable with parameter $p_{r,T_f} = 1 - \frac{\binom{|T_f|-|G|}{\lceil r \cdot |T_f| \rceil}}{\binom{|T_f|}{\lceil r \cdot |T_f| \rceil}}$ if $|T_f| - |G| \geq r \cdot |T_f|$ and 1 otherwise.

Note that what we calculated above is the exact distribution of fault triggering, without any Monte Carlo approximation. If we further approximate the distribution of mutation score using a normal, we can compute the maximal correlation for all different Java programs as a function of sampling ratio. We include two programs as a proof of concept analysis in Figure 4. Now we can see that Math-6f validates our hypothesis in the last section: when the sampling ratio is above about 5%, the maximal correlation drops down to negligible values.

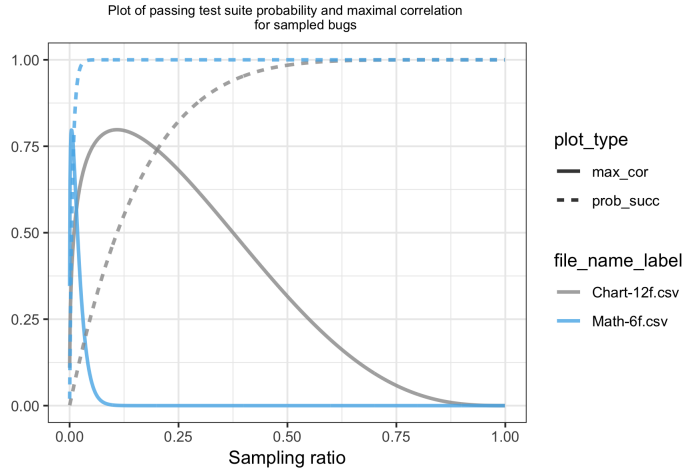

Figure 4. Maximal correlation for two Java programs as a function of sampling ratio

When we proceed to compute the maximal correlation for all 231 programs (See Figure 5), we see that there is a huge heterogeneity in terms of the trend of maximal correlation; therefore aggregating over these sampling ratios and programs within a project might be misleading and not representative of underlying relationship. To sum up, in this section we demonstrate a few overlooked shortcomings of using correlation to characterize the effect size of mutation score onto fault detection.

3.4 Linear regression

While linear regression does not guarantee the fitted outcome for a binary variable will stay between 0 and 1, it is a straightforward and valid measure of effect size, if we consider the probability of detecting a real fault is linear in both mutation score and size. We fitted models as described in the method section and compared $\beta_1, \beta_1^*, \tilde{\beta}_1$ (i.e. the effect size of mutation score in three different cases) below (See Figure 6). We note that when using this alternative effect size measure, the fixed size sampling regression model and random size sampling model adjusting for size give us really similar results. It is not surprising since adding size into the regression equation just implicitly stratifies on size. In addition we see that while not adjusting for size gives us a slightly larger

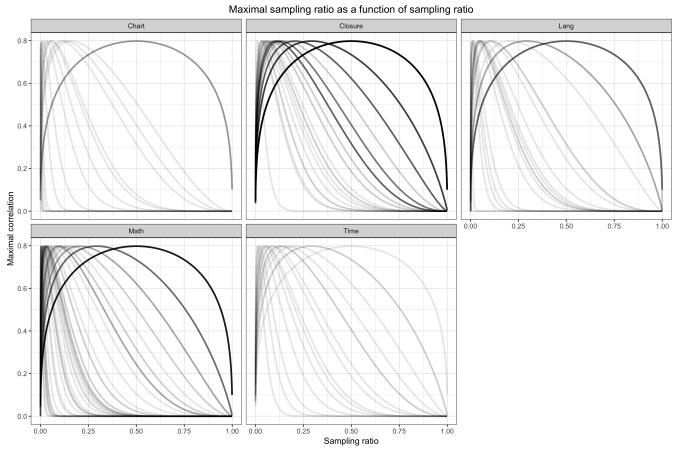

Figure 5. Histogram representation of the posterior distribution

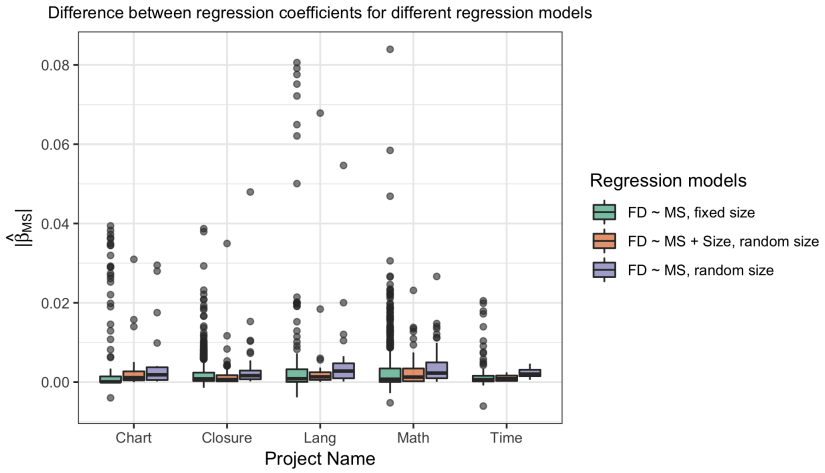

Figure 6. Histogram representation of the posterior distribution

coefficient for mutant score in the random size case, the difference is not nearly as striking as that computed from correlation; further validating the point that correlation alone is an incomprehensive measure of effect size.

We also include Figure 7 to investigate the difference within fixed sampling scheme, among different sampling ratios. There appears to be no significant difference visually.

3.5 Logistic regression

Logistic regression is a far more popular choice for modeling a binary outcome and we include below the analogous results as in the last section, except we are reporting the estimated odds ratio, i.e., the exponential of regression coefficients β. For some of the datasets due to the imbalance in the outcome (virtually all samples will detect the fault), the `glm` call in `R` did not converge and we excluded these in our analyses below.

In Figure 8, we see a similar trend as in linear regression: adjusted odds ratio is very similar to fixed sampling; both are attenuated compared to the unadjusted odds ratio.

We also include Figure 9 to investigate the difference within fixed sampling scheme, among different sampling ratios. There appears to be little difference across different sampling ratio visually. Therefore we would recommend using adjusted odds ratio (or other transformations of logistic regression output such as predicted success probablity)

Figure 7. Histogram representation of the posterior distribution

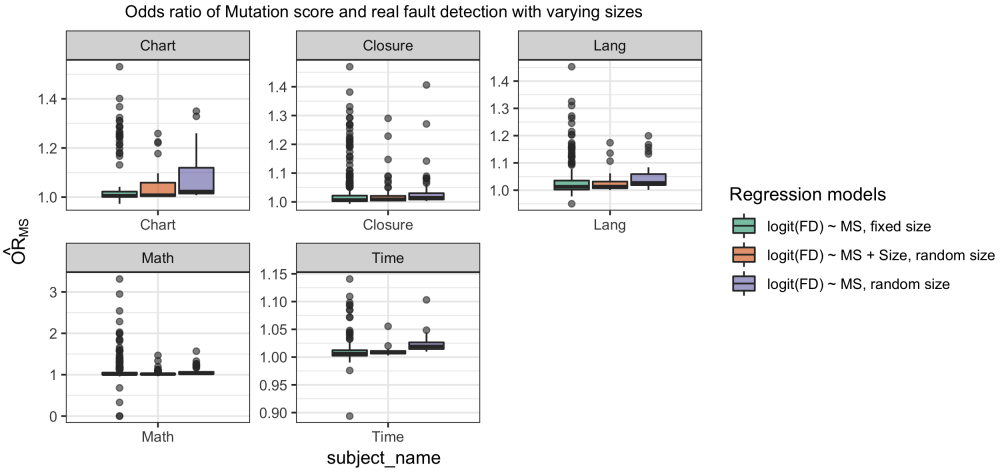

Figure 8. Histogram representation of the posterior distribution

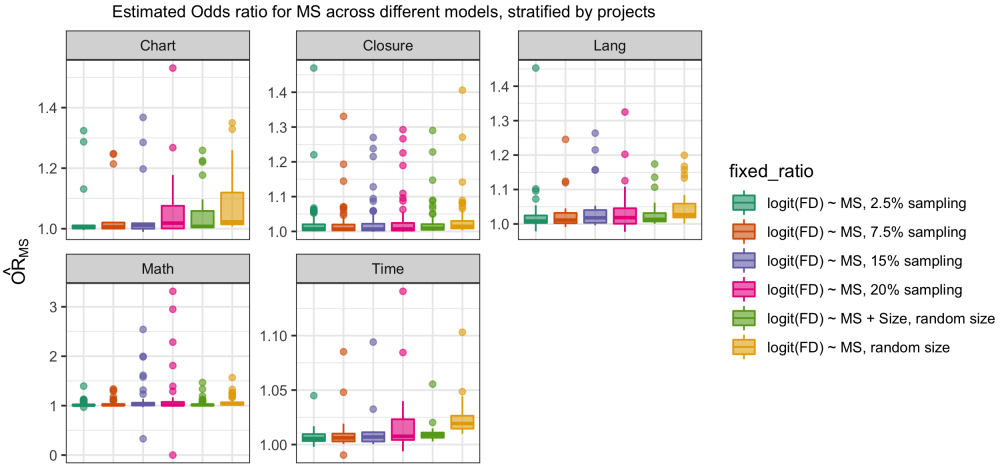

Figure 9. Histogram representation of the posterior distribution

instead of manual stratification.

3.6 Coupling

We now revisit the idea of probabilistic coupling between a mutant and a fault. Recall that coupling is one of the underlying explanations for mutation score based testing: highly coupled mutants lead to more predictive mutation score for real fault detection. For the scope of our project, we looked at the proportion of perfectly coupled mutants across projects and the distribution of coupling probabilities (where perfect couple has probability 1).

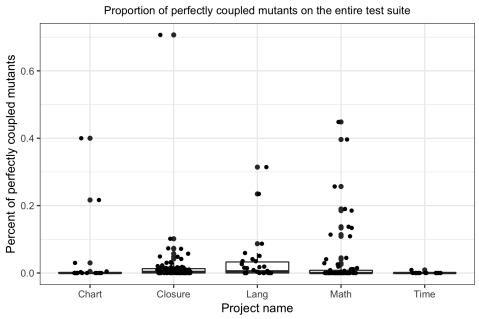

Figure 10. Histogram representation of the posterior distribution

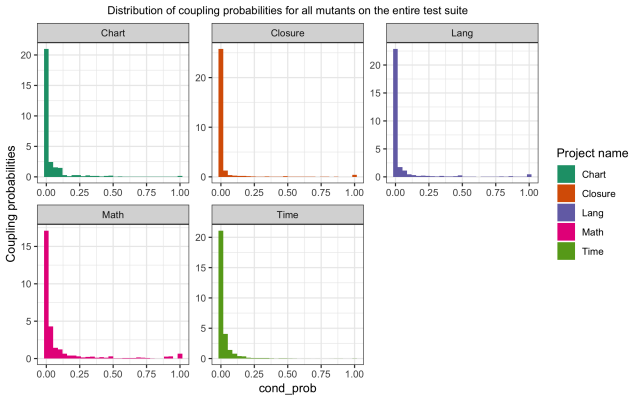

Figure 11. Histogram representation of the posterior distribution

We see in Figure 10 that most of the Java programs have a low proportion of perfectly coupled mutants and there is a decent amount of variation across different projects. The distributions of estimated coupling probability are visually similar across different projects: most of the mass is concentrated between 0 and 0.1, with some heavy tails close to 0.9-1 for Math and Lang.

4 Discussion

4.1 Takeway and recommendations

4.2 Threats to validity

4.3 Future directions

Supplementary Information

Proof

Miscellaneous

- Github repo (mostly codes and documentation):`https://github.com/yiqunchen/EM_single_neuron_mixture`
-
-
-

Supplementary figures

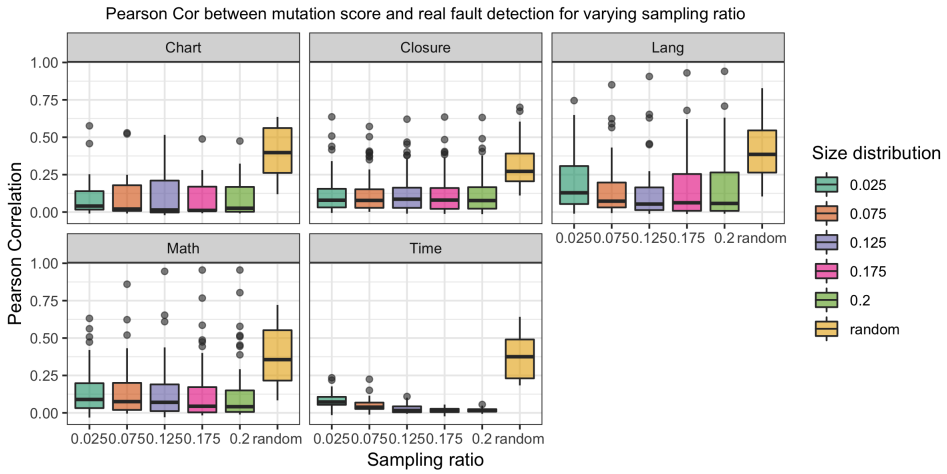

Figure 12. Histogram representation of the posterior distribution

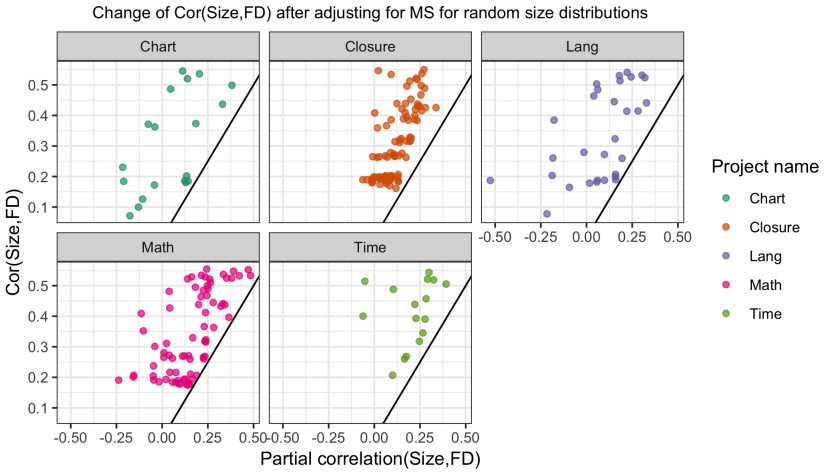

Figure 13. Histogram representation of the posterior distribution

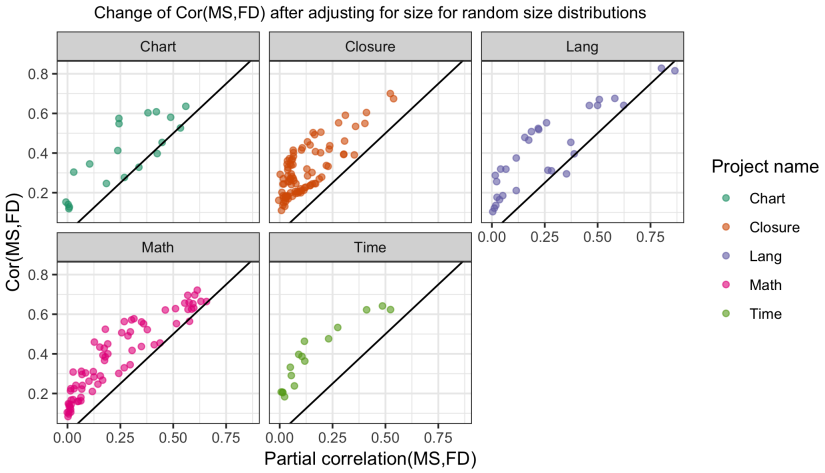

Figure 14. Histogram representation of the posterior distribution

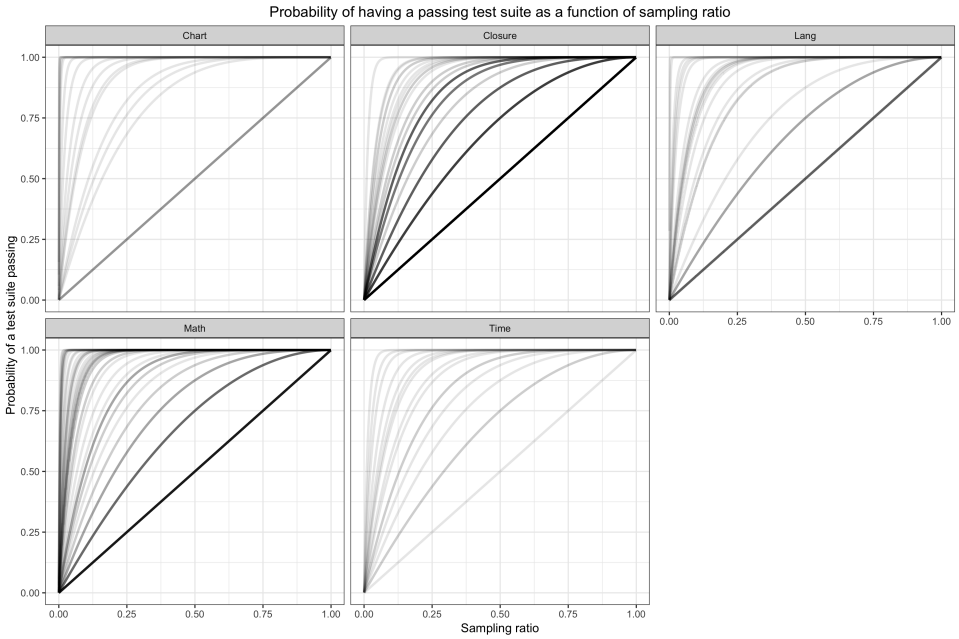

Figure 15. Histogram representation of the posterior distribution

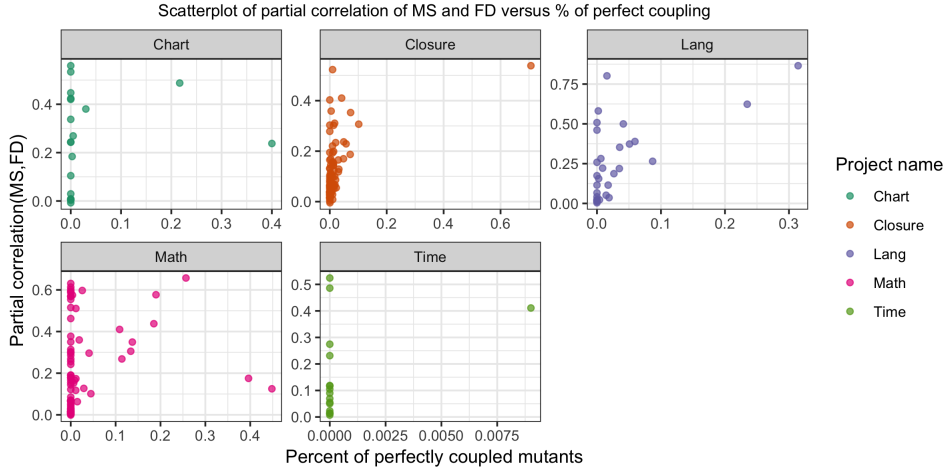

Figure 16. Histogram representation of the posterior distribution

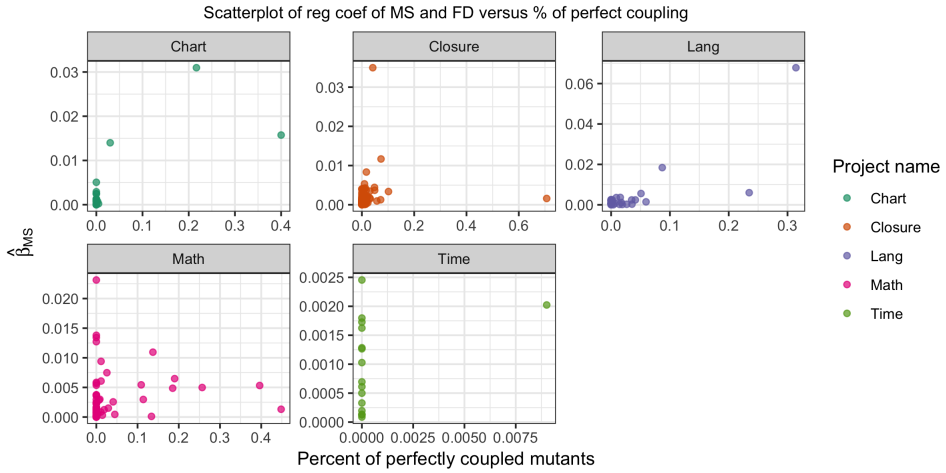

Figure 17. Histogram representation of the posterior distribution

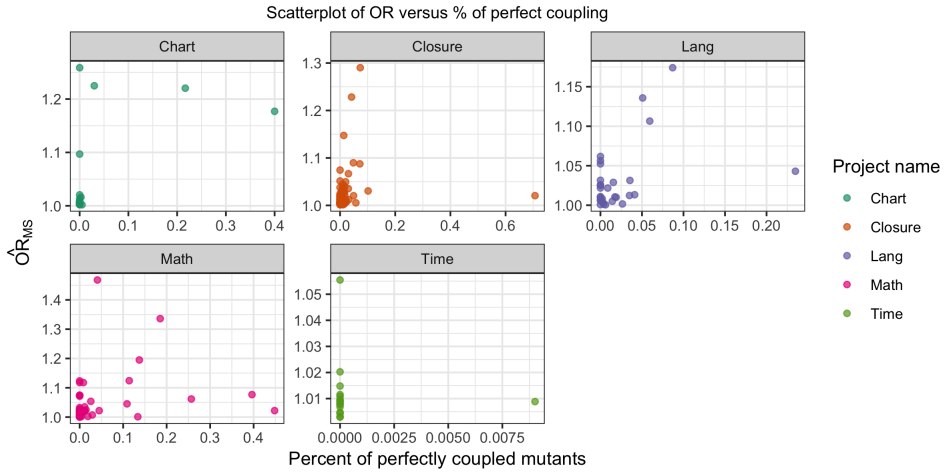

Figure 18. Histogram representation of the posterior distribution