Hitchhikers Need Free Vehicles! Shared Repositories for Statistical Analysis in SBST

Gregory M. Kapfhammer
Allegheny College, USA

Phil McMinn
University of Sheffield, UK

Chris J. Wright
University of Sheffield, UK

ABSTRACT

As a means for improving the maturity of the data analysis methods used in the search-based software testing field, this paper presents the need for shared repositories of well-documented statistical analysis code and replication data. In addition to explaining the benefits associated with using these repositories, the paper gives suggestions (e.g., the testing of analysis code) for improving the study of data arising from experiments with randomized algorithms.

1. INTRODUCTION

The field of search-based software testing (SBST) often involves the implementation and experimental evaluation of algorithms that employ randomization. For instance, automated test data generation (ATDG) with the alternating variable method, or *AVM*, employs randomness both when started and when it restarts after not finding data that meets the testing objectives [3]. Or, a genetic algorithm performing automated test suite prioritisation (ATSP) that reorders tests during regression testing will randomly mutate portions of a candidate test suite to aid in finding the best ordering [6].

Scientists must carefully design and conduct the experiments evaluating these algorithms to ensure that they account for any inherent randomness. It is additionally important that these individuals employ the right methods to analyze the results from these experiments. In the year 2011, Arcuri and Briand published a conference paper outlining some practical guidelines for using statistical methods to analyze randomized algorithms [1], like those often used in SBST. The journal version of this paper, entitled "A Hitchhiker's Guide to Statistical Tests for Assessing Randomized Algorithms in Software Engineering" [2], expands on the earlier paper by further explaining how to rigorously analyze empirical results.

It is hard to underestimate the ways in which these two papers have benefited the SBST community. For instance, many SBST researchers now correctly use the non-parametric Wilcoxon rank-sum test to perform hypothesis testing. To complement these significance tests, many individuals in the field use the nonparametric $\hat{A}_{12}$ statistic of Vargha and Delaney [5] to compute effect sizes, thereby determining the average probability that one approach outperforms another. While these two papers have achieved laudable ends, we argue that the subtleties of various statistical analyses might cause well-intentioned SBST researchers to make mistakes that compromise the validity of their results. To this end, we advocate for the enhancement of methodological maturity in the SBST field through the development and use of shared repositories of well-documented statistical code. That is, we note that SBST community members — "hitchhikers" that we are — need code "vehicles" to ensure that we attain to greater levels of correctness in our statistical analyses.

SBST16, May 16-17, 2016, Austin, TX, USA
© 2016 ACM. ISBN 978-1-4503-4166-0/16/05...$15.00
DOI: http://dx.doi.org/10.1145/2897010.2897017

2. HITCHHIKERS' DILEMMAS

To demonstrate the need that the SBST community has for shared repositories of statistical analysis code, we develop two fictional examples loosely based on our past experiences with the implementation and evaluation of algorithms that use randomization.

First, an SBST researcher named Tom wants to calculate $\hat{A}_{12}$ and remembers that there is some R code in the "Hitchhiker's Guide" paper. Yet, to his dismay, the paper does not include a self-contained function for computing an effect size and furthermore, the necessary code and equations are on separate pages of the paper. After puzzling over curiosities such as the purpose of the `seq_along` function, Tom decides to see if there is an R package that already provides a function for computing $\hat{A}_{12}$ and finds "effsize". Much to his chagrin, Tom realises that the package's code for $\hat{A}_{12}$ is different from what Arcuri and Briand recommend. Wanting to ensure that he calculates the effect size correctly, Tom writes tests to confirm that the code he thinks the "Guide" recommends is numerically equivalent to that which is provided by the effsize package.

The completion of Tom's analysis is further delayed when his colleague suggests that he read a recent article, by Neumann et al. [4], suggesting that Vargha-Delaney effect sizes be "transformed" to ensure that the correct conclusions are reached. Since Tom is computing effect sizes for the execution timings of an ATDG algorithm, he realises that, following Neumann et al.'s advice, he must transform his large data set. Since Tom is not an expert R programmer, his attempt at a transformation is error-prone and his solution is slow. Although Tom has heard of the "dplyr" R package and the benefits it brings to data analysis, he faces a deadline and decides to submit his paper with $\hat{A}_{12}$ values based on un-transformed timings.

In the second scenario, an SBST researcher named Elaine wants to perform hypothesis testing for the data that she has collected about a ATSP algorithm. In this data set, which was curated with assistance from her industrial partners, it is possible to discern when one test suite ordering is better than another even though the difference between suite scores is not meaningful. Elaine consults the "Hitchhiker's Guide" and decides to use the `wilcox.test` function mentioned in Section 11 of the paper. After scanning R's documentation for this function and surmising that it is suitable for her purposes, Elaine performs the statistical analysis. Owing, at least in part, to the fact that Section 11 of the "Guide" uses the phrase "interval-scale results", Elaine questions whether it is valid to apply the chosen test to her ordinal data. This doubt, and the page limits of the conference to which she submits her paper, lead her to only report a few of the *p*-values from the hypothesis tests.

3. SHARED REPOSITORIES

It is important to underscore the fact that this paper's goal is not to call into question the ways in which the "Hitchhiker's Guide" has benefited the SBST community. Rather, we propose that it is time

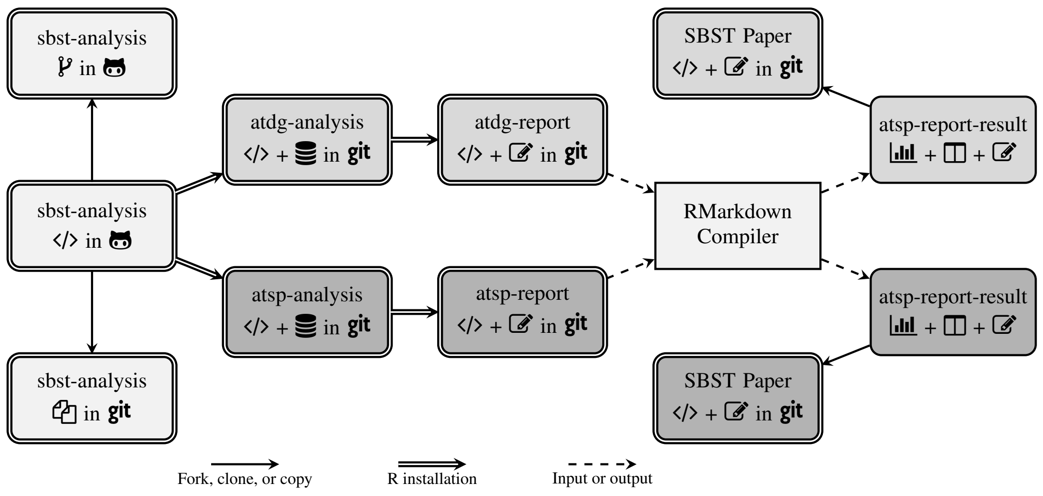

Figure 1: Shared code repositories to support the analysis of data arising from experiments with search-based software testing tools.

to build on the noteworthy foundation set by Arcuri and Briand by encapsulating their theory and practical suggestions into a free and open-source statistical software package shared through GitHub.

Figure 1 lays out our vision for the "free vehicles" that will improve the methodological maturity of the "hitchhikers" in the SBST community. The core of our proposal is the "sbst-analysis" R package that is implemented with the R "devtools" package and hosted on GitHub. R packages, as described and produced by Wickham, support the disciplined creation and delivery of self-contained R source code [8]. This repository will provide well-documented functions implementing all of the best practices for the statistical analysis of the randomized algorithms used by SBST researchers. When proven functions for statistical analysis already exist in R, then this repository could link to them and, additionally, provide supporting guidelines that explain how they should best be used.

Since this repository is publicly available, inquisitive researchers can easily clone it (as indicated by the ⧉ in git annotation in Figure 1) and then study the functions to better understand their operation and assumption(s). When new analysis ideas emerge (e.g., the transformations proposed by Neumann et al. [4]), then the developers of these new methods can fork sbst-analysis (as shown by the ⑂ in 🐙 label), add new functions or modify the existing ones, and then submit a merge request back to the maintainers of sbst-analysis; once approved the new code will be available to the community, thereby speeding the uptake of this new analysis idea.

Now, let's assume that sbst-analysis exists and Tom decides to use it to help him calculate effect sizes. Tom follows the easy-to-understand recommendations by Wickham and creates his own Git repository for an R package called "atdg-analysis" that installs sbst-analysis as its main provider of functionality and contains all of the needed code and data, as shown by the </> + 🛢 in git label. Since sbst-analysis uses R's dplyr package to efficiently manipulate large data sets, the code in Tom's package correctly completes the advocated data transformations. Tom can now see that the untransformed data leads to an effect size supporting the opposite conclusion of the $\hat{A}_{12}$ that was computed with the transformed data.

Moreover, Elaine might leverage the thorough documentation in sbst-analysis to discover that `wilcox.test` is a correct analysis function for her ordinal data. At this point, both Elaine and Tom can use a standard like RMarkdown to write a report that calls the functions from their R packages and, additionally (as indicated by the </> + 📝 in git label), includes textual content explaining the assumptions and analysing the results. These reports would allow both Elaine and Tom to best follow the advice of Arcuri and Briand to "provide full statistics for the collected data" [2]. When these reports are input into the RMarkdown compiler, the results of running the embedded R code (e.g., graphs, summarized data tables, and statistical output) are available to Elaine and Tom when they want to submit their next paper to an SBST-related venue.

It is worth noting that Figure 1 uses the git symbol to suggest that Tom and Elaine will store all of their deliverables in a Git repository, which is either private or publicly hosted by GitHub. Even though the proposed approach only requires the sbst-analysis package to be publicly available on GitHub, the SBST community will further benefit if the researchers using it agree to also make their analysis packages, reports, results, and papers accessible to others. For instance, if Elaine makes her atsp-analysis and atsp-report repositories publicly available, this will better enable other researchers to both replicate her analyses and build on her results.

Along with arguing that "hitchikers need free vehicles" — or, in other words, that SBST researchers need publicly available software and documentation as a way to improve their statistical analyses — this paper also puts forth practical suggestions for improving the functions in repositories like sbst-analysis and atsp-analysis.

As already mentioned, Tom and Elaine should use the expressive and efficient functions in dplyr to summarise and transform the data provided with their R packages. In support of their use of dplyr's functions, they should also ensure that their data sets are organized in a "tidy" fashion where "each variable is a column, each observation is a row, and each type of observational unit is a table" [7]. Finally, SBST researchers should realise that defects in their analysis functions are a threat to the validity of their results and, as such, use R packages like "testthat" to implement and run test cases.

4. REFERENCES

[1] A. Arcuri and L. Briand. A practical guide for using statistical tests to assess randomized algorithms in software engineering. In *Proc. of 33rd ICSE*, 2011.

[2] A. Arcuri and L. Briand. A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering. *STVR*, 24(3), 2014.

[3] P. McMinn, C. J. Wright, and G. M. Kapfhammer. The effectiveness of test coverage criteria for relational database schema integrity constraints. *TOSEM*, 25(1), 2015.

[4] G. Neumann, M. Harman, and S. Poulding. Transformed Vargha-Delaney effect size. In *Proc. of 7th SBSE*, 2015.

[5] A. Vargha and H. D. Delaney. A critique and improvement of the CL common language effect size statistics of McGraw and Wong. *EBS*, 25(2), 2000.

[6] K. R. Walcott, M. L. Soffa, G. M. Kapfhammer, and R. S. Roos. Time-aware test suite prioritization. In *Proc. of ISSTA*, 2006.

[7] H. Wickham. Tidy data. *JSS*, 59(1), 2014.

[8] H. Wickham. *R Packages*. O'Reilly, 2015.