

IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF PENNSYLVANIA

UNITED STATES OF AMERICA)
)
) Criminal No. 19-369
)
LAFON ELLIS)

Declaration of Nathaniel Adams and Jeanna Matthews

We, Nathaniel Adams and Jeanna Matthews, Ph.D., declare that we have personal knowledge of the following, and if called upon to do so, could and would testify competently to the matters contained herein.

I. Qualifications

1. Nathan Adams has a Bachelor of Science in Computer Science from Wright State University and has worked as a Systems Engineer at Forensic Bioinformatic Services, Inc. for the past eight years. He has served as an expert witness in cases involving probabilistic genotyping software in both federal and state court around the country and internationally. Mr. Adams has inspected the source code of other probabilistic genotyping software including STRmix and FST under protective orders in federal cases.

2. Dr. Jeanna Matthews is a Full Professor of Computer Science at Clarkson University and an affiliate at Data & Society. She holds a Ph.D. in Computer Science from the University of California, Berkeley, and one of her current research projects focuses on the analysis and comparison of probabilistic genotyping software.

3. We have each written a prior declaration in this case. Our qualifications are listed in more detail therein.

II. Software Validation and Verification Processes Are Distinct from Laboratory and Forensic Science Validation Processes

4. We have been asked by attorney Khasha Attaran to address the need for access to

software materials, including source code, for inspecting the TrueAllele probabilistic genotyping software system used in the case of *United States v. Lafon Ellis*. We have requested numerous TrueAllele software materials, which are listed in the subpoena filed April 29, 2020, but have received very few of these requested materials.

5. Probabilistic genotyping systems like TrueAllele are software programs that conduct complex statistical and biological analysis, when an individual analyst would have difficulty drawing conclusions about a sample. Because the analysis is performed by the computer program, software engineering principles are a key component of assessing whether any results TrueAllele produces are reliable.

6. We emphasize that forensic DNA and computing disciplines both recognize and emphasize the need for validated products and systems, but that the label of “validated” is achieved through different processes in each discipline. We do not intend to diminish the significance or relevance of guidance and standards published by SWGDAM and FBI, ISFG, or ASB, but suggest that **practices common to software development and described in software standards, guidance, articles, and texts are also relevant considerations**. Stated differently, SWGDAM and FBI, ISFG, or ASB do not satisfactorily address issues of software engineering.

7. Professional practice and research in software engineering aims at preventing, detecting, and removing flaws.^{1,2,3,4} These efforts can be costly and time-consuming, but when market forces incentivize the mitigation of flaws, software developers invest in quality improvement processes.

¹ I. Sommerville, *Software Engineering*, 10th ed. (2015)

² F. Brooks, *The Mythical Man-Month: Essays on Software Engineering*, Anniv. ed. (1995)

³ Ass’n for Computing Mach., *ACM Transactions on Software Engineering and Methodology* <https://dl.acm.org/journal/tosem> (last accessed February 2, 2021)

⁴ Inst. of Elec. & Electronics Eng’rs, *IEEE Transactions on Software Engineering*, <https://www.computer.org/product/journals/tse> (last accessed February 2, 2021)

8. As described in paragraph 25 of the July 16, 2020 declaration by Nathan Adams, IEEE Std 1012-2012 describes verification and validation tasks by integrity level [from 1 (lowest) to 4 (highest)]. Integrity levels 3 and 4 include generating and reviewing documentation such as stakeholder requirements, software design documents, and software qualification test reports.⁵

9. The source code and executable versions of software programs are highly relevant to any review of verification and validation. Source code is the human readable list of commands (e.g., “if the user enters X, then do Y,” “repeat trying possible solutions until one works”) written by programmers and used to generate the executable version itself. It is like being able to see the blueprints for the house instead of walking through it. The normal user-facing executable version is produced from the source code but is designed to be interpreted by the computer rather than read by a human. When a user runs the executable version of software, it often generates graphics and an interface with which the user can interact. Examining both the source code and the executable version are both important and give different windows into potential flaws, including flaws that could affect the likelihood ratio reported by the system.⁶

10. The other types of documentation described in IEEE Std 1012-2012 are also highly relevant to a review of verification and validation. The existence and completeness of such documentation provides a means to evaluate a level of quality, inherent to the software, which is not directly measurable from the code or executable versions of the software alone. The non-existence of such documentation is also informative. It can indicate that the developers are not following practices common to the software engineering field.

⁵ Inst. of Elec. & Electronics Eng’rs, *IEEE Std. 1012-2012: IEEE Standard for System and Software Verification and Validation* (2012).

⁶ S. Lacambra, J. Matthews, & K. Walsh, *Opening the Black Box: Defendants’ Rights to Confront Forensic Software*, *The Champion*, 28-39 (May 2018).

11. Probabilistic genotyping systems exhibit a number of significant risk factors for latent defects: (1) their nature as complex scientific software, (2) the small user base of forensic DNA practitioners in comparison to more widely used software and (3) the difficulty in manually verifying the output of the software. That is, latent defects are more likely with complex scientific software because multiple factors complicate the testing of (and consequently, detection of bugs or flaws in) it.⁷ Furthermore, more widely used software will have faults identified at a greater quantity and/or rate than lesser-used software,⁸ so flaws are likely to go undetected for longer when it comes to probabilistic genotyping software. Finally, the software is frequently used in cases where forensic analysts would have difficulty verifying the output of the software manually.

12. We advocate for a limited, independent evaluation of probabilistic genotyping software used in criminal proceedings, such as TrueAllele, for two primary purposes. The first purpose is to assist relevant domain experts (e.g. biologists and statisticians) to investigate the actual software operation of its published models. The second purpose is to evaluate any substantive record of software-domain verification and validation efforts and, lacking that, assess the system's *capacity and readiness* for undergoing common verification and validation tasks. In order to accomplish these two goals, we cannot overstate the need for meaningful access to relevant software development materials, including technical documentation and source code.

13. In many other instances, systems claiming compliance with SWGDAM, FBI QAS, ISFG, and ASB guidance and standards have experienced post-validation discovery of defects and coding flaws, affecting likelihood ratio calculations in casework. Likelihood ratios for affected casework

⁷ U. Kanewala & J. Bieman, *Testing scientific software: A systematic literature review*, 56 Information and Software Technology 1219-1232 (2014).

⁸ T. Dey & A. Mockus, *Modeling Relationship between Post-Release Faults and Usage in Mobile Software*, PROMISE'18: Proceedings of the 14th International Conference on Predictive Models and Data Analytics in Software Engineering, 56-65 (2018).

are not always recalculated. And in some instances, defects are not publicly disclosed by the developer for years after discovery.⁹

14. While Dr. Perlin asserts that access to source code is not necessary to assess the reliability of TrueAllele's results, we simply disagree. The field of software engineering recognizes the importance of source code access and review for common verification and validation tasks. Complex software frequently has latent errors that are difficult to detect and which could impact reported results. Meaningful access to the source code can help uncover those latent defects.

III. Meaningful access to source code

15. In addition to sufficient time, access to relevant tools enables thorough software inspections. Thorough software inspections of systems as complex as TrueAllele do not involve straight-through reading source code from the first line to the last line. As with inspections of physical devices or systems, source code inspections of probabilistic genotyping software should involve the use of tools with which the program and its development processes can be measured, tested, and evaluated from a variety of perspectives. These inspections fall into two major categories: static and dynamic.

16. Static inspections do not involve executing the code of the software, itself. A human developer simply reading code is a major source of the developer's understanding of the program behavior, though it is very time-consuming with effective rates at approximately 200 lines of code per hour.¹⁰ Myriad static analysis tools exist to make this process faster and more effective but require that these tools to be installed on a computer that can be used for analysis. These are

⁹ STRmix, *Investigation into the effect on the LR due to miscode present in STRmix™ V2.3.07*, <https://strmix.com/assets/STRmix/STRmix-PDFs/STRmix-V2.3.07-miscode-investigation-June-2020.pdf> (last accessed February 2, 2021)

¹⁰ C. Kemerer & M. Paulk, *The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP Data*, 35 (4) IEEE Trans. on Soft. Eng. 534-550 (2009).

commonly used for detecting certain types of errors, evaluating program complexity, measuring conformance to coding standards, and measuring test and requirements coverage.^{11,12}

17. Dynamic inspections can include manual debugging. Debugging is a term used for both the task of removing software bugs or defects and also the process of directly inspecting the program as instructions are executed. By executing the program one module - or even one line of instruction – at a time, a developer can “trace” the state of the program as various calculations are performed and program tasks are executed.¹³ Debugging is therefore also an invaluable tool for learning the low-level, step-by-step, “live” operation of the system. Dynamic inspections can also include the execution of automated tests.¹⁴

18. An integrated development environment (IDE) is a software suite of programs used by developers to write and inspect software. They ease navigation of thousands of lines of code by automatically linking related-but-separate parts of code and also enable the use of automated static and dynamic tools. IDE’s are ubiquitous for constructing and reviewing complex software.

19. At a minimum, meaningful access to examine TrueAllele’s source code requires the code to be on a personal computer, with a full keyboard and mouse. The code needs to be “buildable” into a working executable from the MATLAB IDE with all necessary libraries and working build instructions provided. In addition, the ability to install code analysis tools is necessary to evaluate

¹¹ Mathworks, *checkcode*, <https://www.mathworks.com/help/matlab/ref/checkcode.html> (last accessed February 2, 2021)

¹² J. Abraham, *Improving Software Quality with Static Code Analysis*, <https://www.mathworks.com/company/newsletters/articles/improving-software-quality-with-static-code-analysis.html> (last accessed February 2, 2021)

¹³ Mathworks, *Debug a MATLAB Program*, https://www.mathworks.com/help/matlab/matlab_prog/debugging-process-and-features.html (last accessed February 2, 2021)

¹⁴ Mathworks, *Testing Frameworks*, <https://www.mathworks.com/help/matlab/matlab-unit-test-framework.html> (last accessed February 2, 2021)

the code.

20. Providing these materials in these formats will not change the performance of Cybergenetics's copies of the software, nor will our analysis interfere with Cybergenetics's ability to continue using TrueAllele in the interim.

IV. Source code flaws in probabilistic genotyping software can have real-world impacts on the results produced by the programs.

21. In his July 16, 2020 statement in this case, Nathan Adams described his experience with identifying a previously undisclosed modification to the Forensic Statistical Tool (FST) software from the New York City Office of the Chief Medical Examiner software (OCME). This modification was made in response to a defect not detected by OCME until after it had concluded its FST validation efforts, though neither the defect nor the modification were mentioned in the peer-reviewed article submitted for publication following the modification.¹⁵ These undisclosed software behaviors affected the reported likelihood ratios under certain circumstances.

22. Without public disclosure, let alone access to the source code, these software behaviors were not known outside the laboratory, including independent scientists, defense teams, and the general public. OCME had not disclosed them in previous cases, nor were they disclosed to the publications that had vetted the earlier validation studies for FST and upon which OCME relied in arguing FST was reliable.

23. FST is not alone in probabilistic genotyping programs that have been modified after the post-validation detection of defects. As of January 29, 2021, developers of the chief commercial competitor of TrueAllele, STRmix™, have disclosed fourteen "coding faults" affecting likelihood

¹⁵ A. Mitchell, et al. *Validation of a DNA mixture statistics tool incorporating allelic drop-out and drop-in*, 6 Forensic Sci. Int'l: Genetics 749-761 (2012).

ratio calculations in casework versions of STRmix™.¹⁶

24. Additionally, in *Nebraska v. Charles Simmer*, Nathan Adams wrote in a December 7, 2017 declaration to the Court:

A 46-page document entitled “VUIer Updates – Cybergenetics” constitutes a change log for the VUIer component of the TrueAllele system. A word search for the term “bug” indicates 101 uses of the term in this document. While not all of these bugs appear to substantially affect the quality of the VUIer component, there are no formal description of how the defects were detected or how they were removed.

25. It is concerning that no similar document has been provided in this case. Stated again, TrueAllele disclosed a document discussing the identification of 101 “bugs” and consequent modifications to the TrueAllele software, in another case, but did not provide the same or updated document in this case. While we do not expect all of these “bugs” detected by TrueAllele developers to have made their way to casework systems, no description of the source, effects, or confirmation of removal of these bugs is publicly available.

V. Conclusion

26. For the results reported by probabilistic genotyping systems like TrueAllele to be reliable, the software needs to run correctly. If there are errors in the code, the likelihood ratios could be skewed. Practices common to the software engineering field provide guidance on how to assess software, including by analyzing the source code. Such analysis requires meaningful access to the source code.

27. In several criminal cases, various degrees of meaningful access to closed-source probabilistic genotyping systems has been provided. This access has enabled informed evaluations of program behaviors by biology and computing experts. It has allowed informed commentary to

¹⁶ STRmix, *Summary of miscodes*, <https://strmix.com/news/summary-of-miscodes/> (last accessed January 29, 2021)

courts on software development practices of the probabilistic genotyping systems reviewed. It also enabled identification of a previously undisclosed post-validation modification to a probabilistic genotyping system.

28. We are requesting materials relevant to the development and operation of TrueAllele be provided to us in a manner allowing meaningful access in order for us to provide our opinions to the court as to whether TrueAllele has any clear and present defects and whether it has a record of completed, meaningful verification and validation tasks or if such tasks can even be completed.

February 16, 2021



Nathaniel Adams, Dayton, Ohio



Jeanna Matthews, Ph.D., Potsdam, New York