The Right To Confront Your Accusers: Opening the Black Box of Forensic DNA Software

Jeanna Matthews
Marzieh Babaeianjelodar
Stephen Lorenz
Abigail Matthews
Clarkson University
Potsdam, New York
jnm@clarkson.edu
babaeim@clarkson.edu
lorenzsj@clarkson.edu
matthea@clarkson.edu

Mariama Njie
Iona College
New Rochelle, New York
mnjie1@gaels.iona.edu

Nathaniel Adams
Forensic Bioinformatic Services
Fairborn, Ohio
adams@bioforensics.com

Dan Krane
Wright State University
Dayton, Ohio
dan.krane@wright.edu

Jessica Goldthwaite
The Legal Aid Society
New York, New York
JWGoldthwaite@legal-aid.org

Clinton Hughes
Brooklyn Defender Services
Brooklyn, New York
chughes@bds.org

ABSTRACT

The results of forensic DNA software systems are regularly introduced as compelling evidence in criminal trials, but requests by defendants to evaluate how these results are generated are often denied. Furthermore, there is mounting evidence of problems such as failures to disclose substantial changes in methodology to oversight bodies and substantial differences in the results generated by different software systems. In a society that purports to guarantee defendants the right to face their accusers and confront the evidence against them, what then is the role of black-box forensic software systems in moral decision making in criminal justice? In this paper, we examine the case of the Forensic Statistical Tool (FST), a forensic DNA system developed in 2010 by New York City's Office of Chief Medical Examiner (OCME). For over 5 years, expert witness review requested by defense teams was denied, even under protective order, while the system was used in over 1300 criminal cases. When the first expert review was finally permitted in 2016, many problems were identified including an undisclosed function capable of dropping evidence that could be beneficial to the defense. Overall, the findings were so substantial that a motion to release the full source code of FST publicly was granted. In this paper, we quantify the impact of this undisclosed function on samples from OCME's own validation study and discuss the potential impact on individual defendants. Specifically, we find that 104 of the 439 samples (23.7%) triggered the undisclosed data-dropping behavior and that the change skewed results toward false inclusion for individuals whose DNA was not present in an evidence sample. Beyond this, we consider what changes in the criminal justice system could prevent problems like this from going unresolved in the future.

CCS CONCEPTS

• Software and its engineering → Correctness; Risk Management • Computing profession → Code of Ethics

KEYWORDS

algorithmic accountability; criminal justice software; probabilistic genotyping software; Forensic Statistical Tool (FST)

ACM Reference format:

Jeanna Matthews, Marzieh Babaeianjelodar, Stephen Lorenz, Abigail Matthews, Mariama Njie, Nathaniel Adams, Dan Krane, Jessica Goldthwaite and Clinton Hughes. 2019. The Right To Confront Your Accusers: Opening the Black Box of Forensic DNA Software. In Proceedings of S '19, January 27–28, 2019, Honolulu, HI, USA (AIES'19). ACM, New York, NY, USA.
https://doi.org/10.1145/3306618.3314279

1 Introduction

Increasingly big decisions about the lives of individuals are being made in a partnership between human decision makers and computer systems. In high stakes areas like hiring, housing, credit and criminal justice, societal principles negotiated collectively over time could be undermined in the process of automation. For example, automated systems are used throughout the criminal justice system from investigation/policing decisions to pretrial decisions to decisions about evidence at trial to sentencing decisions to parole decisions. In this context, it is reasonable to ask what the impact of these automated systems is on widely accepted principles of criminal justice decision making such as the right to a public trial, the rights of defendants to review and confront

the evidence against them and the right to equal justice under the law.

In this paper, we focus on the Forensic Statistical Tool (FST), a forensic DNA system developed in 2010 by New York City's Office of Chief Medical Examiner (OCME). We consider what types of oversight human decision makers were able to provide and what incentives were present for finding, fixing and disclosing bugs in the system. We propose a set of modifications to the holistic decision making process to encourage bugs in forensic DNA systems like FST to be found and fixed.

We conduct independent, third party testing of FST, a step that we argue should be regularly performed on any software used in the criminal justice system. Using a collection of over 400 mixed DNA samples of known composition from OCME's own validation study, we evaluate the impact of an undisclosed data-dropping routine discovered by defense experts during the first source code review ever permitted. Expert witnesses identified the potential for this function to drop data that could be helpful to the defense, but this paper is the first study of the quantitative impact of the change. Even using data from OCME's own validation study (i.e. no surprise test cases), we find that almost a quarter of the samples triggered the data-dropping behavior and that the change skewed results toward false inclusion for individuals whose DNA is not present in an evidence sample. Defense attorneys, judges and the scientific community should have been informed of this change, but were not. We argue that enabling independent, third party testing is essential to preventing this in the future. We discuss the current hurdles to independent testing and make concrete suggestions for reducing those hurdles in the interest of accountability, transparency, and justice.

2 Background

Forensic DNA evidence has been an important part of criminal investigations for decades, but in recent years, probabilistic genotyping (PG) software has been introduced to interpret evidence that is too complex for manual human analysis. Substantial concerns about the accuracy and reliability of this software have been raised by scientists, journalists, and lawyers and there is substantial debate about its role in moral decision making in criminal justice.

Many factors can complicate forensic DNA interpretation, making automated software analysis an attractive alternative to human interpretation. Multiple contributors to an evidence sample make the results more difficult to interpret visually. Environmental factors can degrade DNA. Evidence samples may contain little DNA available for testing. When processing samples, DNA from the sample can fail to be detected in whole or in part (drop-out), and random fragments of DNA can be introduced (drop-in).

Models implemented in PG software can vary substantially in how they accommodate – or don't address – these issues. PG software also varies in how operator assumptions are taken into account, such as the number of contributors to a sample – a value that can only be estimated when evaluating the evidence sample (e.g. how could one conclusively determine how many people might have handled a gun) (Paoletti et al. 2005).

Additionally, most PG software assumes that all contributors to a sample are completely unrelated and from the same ethnic group; an assumption that is not valid in many actual cases. For example, investigations into an assault by a group Hasidic men included concerns about whether PG software that assumes no relationships between contributors can accurately distinguish among members of a more genetically insular population (Kirchner 2017).

There have also been substantial concerns for inappropriate bias in criminal justice software more broadly. For example, ProPublica found that the COMPAS software used widely throughout the United States to estimate a defendants risk of committing another crime was more likely to falsely flag black defendants as future criminals, while white defendants were mislabeled as low risk more often than black defendants (Angwin et al. 2016)(Chouldechova 2017). For facial recognition software which can at times be used criminal justice applications, Buolamwini and Gebru identify substantially higher error rates for dark-skinned women than for light-skinned men (Buolamwini and Gebru 2018).

All of these factors should add up to a need for healthy skepticism about the design, development, and use of complex software systems used in criminal justice, including PG software. The field of forensic DNA analysis should require robust independent review of PG systems prior to their use in casework, ongoing reporting of substantial modifications and ongoing independent review to promote investment in iterative improvement. Both in research and in casework, an emphasis should be placed on comparisons between multiple reasonable systems' evaluation of the same input data (Garofano et al. 2015)(NIST 2017). However, this is not the current state of the field.

Instead, these legitimate concerns have been further heightened by secrecy. Software vendors aggressively claim trade secret protection for their software. In many cases, developers have succeeded in resisting requests by defense attorneys to allow their own experts to review both the executable versions and source code of these systems, even under protective order (Tashea 2017). That is an extreme requirement of secrecy, especially in high-stakes criminal cases.

Legal scholars and defense attorneys have argued that defendant rights to confront the evidence against them and to a public trial should outweigh the intellectual property interests of software vendors (Wexler 2018). Furthermore, it has been argued that software vendors already enjoy substantial commercial protection from a first-mover advantage once the results of their product have been widely accepted in courtrooms and additional protection may do more to shield products from legitimate criticism of software quality, reliability, and accuracy than to protect intellectual property.

For PG software, peer-reviewed validation studies are typically conducted by the software developers. Internal validations

conducted by individual laboratories are usually unpublished, let alone independently reviewed. Rarely is there adversarial testing by a group incentivized to find problems and rarely are systems reviewed with an eye to how errors could impact a particular case or defendant. Defendants, particularly indigent defendants, rarely have access to resources to conduct adversarial testing in the context of their own case and even defense teams willing and able to do so may be denied access to the materials necessary to do so effectively.

For FST in particular, OCME refused any independent review of the source code, supporting software development materials, and executables, for years, even under a protective order. In a 2016 criminal case, a federal judge finally ordered OCME to provide FST's source code to the defense team under a protective order. The team of defense experts who reviewed the code identified a number of concerns, including a function, CheckFrequencyForRemoval, that they demonstrated was capable of dropping data that is helpful to the defense. This function runs counter to the methodology publicly described in previously sworn testimony and peer-reviewed publications and appears to have been introduced as a work around for other problems with the system. Between 2011 and 2017, FST was used in approximately 1,350 criminal investigations. This timeline of documented changes to FST suggest that the analyses in casework involved the version modified in this way.

In retrospect, we know that almost immediately after bringing FST online in April 2011, OCME had to take FST offline again for software maintenance. Based Freedom of Information requests and responses to litigation, we know that FST was modified in order to bring it back online in June 2011. It is not unusual to have bugs in software, but OCME's response to the problem is telling. Changes, including the CheckFrequencyForRemoval function, were made without any reporting the change to the NY State Commission on Forensic Science that approved FST for use in casework. In June 2017, Eugene Lien, OCME Assistant Director said in an affidavit, "Because this modification did not affect the methodology of the program, it did not require submission to the Commission on Forensic Science or the DNA Subcommittee." The results of our work strongly challenge this statement, as we will describe.

Subsequent to the findings of the defense experts in 2016 and in response to filings by ProPublica and Yale's Media Freedom and Information Access Clinic, the judge unsealed both the experts' findings (produced originally under a protective order) and the entire FST source code that had been so closely guarded by OCME for years. ProPublica then published the findings and the source code on GitHub (FST 2017). Appendix 1 contains the source code for the data-dropping function, CheckFrequencyForRemoval, from this GitHub repository.

This paper represents the first quantitative study of the impact of the CheckFrequencyForRemoval function. We will describe in detail how it is possible for this function to drop data that is helpful to the defense. Then, using over 400 samples of known origin from OCME's own validation study, we quantify the impact of the function on the results both for individuals known to have contributed to a sample and individuals who did not contribute to the sample. While we find no evidence of a deliberate attempt to disadvantage the defense, we do see a willingness to put in sloppy fixes when problems with the software were identified and a failure to consider how those fixes could impact defendants. The system as a whole failed to put in the necessary provisions for accountability and transparency in order to incentivize disclosure and true repair.

3 FST and the OCME Validation Study

The NY State Commission on Forensic Science approved FST for use in casework based on a validation study designed and conducted by OCME. The validation study underlying FST consisted of 439 two- and three-person mixtures of varying quantities of DNA and contributor proportions, genotyped using both High Copy Number (HCN) and Low Copy Number (LCN) protocols. Since these mixtures were created in a controlled laboratory setting, their true contributors and known non-contributors are known.

The 439 mixtures were generated to serve as test evidence samples for which the "correct" answers are known. These samples were constructed based on single-source blood and cheek swab samples of known origin as well as from items handled by multiple individuals, such as a computer mouse or a pen. Some, but not all, of the touched items were cleaned with bleach and ethanol prior to handling. Despite this pre-cleaning step, it is interesting to note that some samples still contained DNA that did not belong to any of the deliberate contributors.

OCME evaluated all 439 mixtures in comparison to their known contributors and a set of 1,246 non-contributors. The non-contributor set consists of genotypes developed from OCME morgue bodies and a national data set (Butler et al 2003). Allele frequency rates were established for NYC by OCME through genotyping morgue bodies. OCME developed a subset of these genotypes at only thirteen of the fifteen loci used by FST, simulating genotypes for the remaining two loci. Subpopulations were grouped by self- or OCME-reporting into African-American, Asian, Caucasian, and Hispanic categories. The lab removed information on the races of the donors to the mixtures, though in publications they do claim that the mixtures represent the diversity of New York City (Mitchell et al. 2012) (People v. Collins 2013).

OCME originally wanted to validate FST for four-person mixtures and additional four-person mixtures were generated during the study, but ultimately FST was not validated for the evaluation of four-person mixtures (Mitchell et al. 2012). OCME never published the validation data set but did produce it in 2012 pursuant to an agreement reached after litigation in the case People v. Collins. It was produced in printed form, then scanned and partly transcribed by the defense team.

Since individuals share alleles and the genotypes of all individuals are not known, one normally cannot conclude that a specific individual is the sole possible source of DNA recovered from an evidence sample. Case law in the United States requires that a statistical weight of evidence be provided when an individual cannot be excluded as a possible contributor to a casework sample, in order to assist them in determining the strength of that evidence. For example, if one

in three individuals could not be excluded as possible contributors to a particular sample, then the strength of that conclusion is minimal, while if only one in a billion individuals could not be excluded, the strength would be high.

Statistical weights calculated by PG systems are presented as likelihood ratios (LR), composed of the probability of observing the data generated during the course of testing evidentiary samples, E, given two competing hypotheses. These hypotheses are typically constructed as H1 or the prosecutor's hypothesis, Hp, which includes the defendant as a contributor, and an alternative hypothesis H2, or the defense hypothesis, Hd, which does not include the defendant as a contributor. For samples containing DNA from multiple individuals, both Hp and Hd will include additional contributors, either assumed contributors whose genotypes are known or contributors whose identities are unknown. The common formula, where E is the observed data is: $LR = \Pr(E|Hp) \,/\, \Pr(E|Hd)$ Consequently, a likelihood ratio of 1 is deemed "inconclusive" while an LR >1 is inclusionary or inculpatory (suggestive of guilt) and an LR <1 is exclusionary or exculpatory (suggestive of innocence).

Likelihood ratios are calculated for each locus and multiplied using the product rule, assuming linkage equilibrium between loci. Due to the complexity of forensic DNA mixture data and measurement uncertainty, it's generally held that there is no "ground truth" for LRs, even for samples of known composition, against which PG results can be compared for accuracy (Steele and Balding 2014). Consequently, confidence in PG systems is based in the appropriateness of the models underlying their algorithms and the quality of their software development processes and resulting executables.

Some labs, including OCME, provide "verbal equivalency" for LR values. LRs of 1-10 are described by OCME as "limited support" for the numerator hypothesis, generally Hp. Similarly, LRs of 10-100, 100-1000, and 1000+ are described as "moderate," "strong," and "very strong," respectively (OCME 2016).

For each sample, FST calculates four LR using allele frequencies from each of OCME's four default reference subpopulations (Asian, Black, Caucasian and Hispanic). As a conservative measure, only the lowest of these four LRs is included in the final written report.

FST originally included all 15 loci in its calculations, as one would expect. However, CheckFrequencyForRemoval removes data for loci where frequencies of observed alleles across all replicate amplifications summed to ≥97% in any of FST's four reference subpopulations. Logically, frequencies over 100% should not occur, but in practice, they do. There are many things that could lead to this inaccuracy (e.g. errors introduced by multiple rounds of amplifications (drop-out errors), contamination errors (drop-in errors) or issues with the use of minimum allele frequencies. However, rather than deal with the inaccuracies in a transparent way, OCME chose to deal with the errors by simply dropping the contribution of any loci that approaches the 100% boundary. This is done even if the in-formation at that locus is exculpatory information that would have helped the defense or whether it is inculpatory evidence that would help the prosecution. Nothing was done to report when this occurred and it is completely possible that the set of dropped data would have altered the LR values reported or even the verbal equivalence category of the result. This clearly seems relevant to defense teams, the NY State Commission on Forensic Science and the public, but no such disclosure was ever made by OCME. In the next section, we quantify how often on OCME's own validation study this type of change occurred in the reported LR or the verbal equivalence category.

4 Independent Comparison Testing

In this section, we describe our independent testing of FST. We describe both how we automated testing of FST and the results of comparing FST output with and without the CheckFrequencyForRemoval function. We examine the impact of that change on both known contributors to samples in the OCME validation study as well as a set of non-contributors.

We acquired FST v2.5 from the ProPublica's GitHub repository (commit 5b353500d) (FST 2017). FST is a C# ASP.NET application using an MS SQL database. Its interface is browser-based. We ran it on a QEMU virtual machine with Windows 10 Pro 64-bit using 32 GB of RAM and an Intel Xeon processor.

FST's installation is non-trivial, requiring database connection configuration and a custom Windows service per installation. Two versions of FST v2.5 were used for all analyses – one with the CheckFrequencyForRemoval function enabled and one with it disabled. The disabled version is intended to emulate the pre-modification version of FST, though the source code for that version has not been publicly released. FST does not provide an external API or command line interface, so we developed several noninvasive wrapper scripts for automation.

4.1 With and Without CheckFrequencyForRemoval

LRs were generated for all 439 two and three person mixtures from FST's validation study and compared to their known contributors for a total of 1,245 evaluations. We found that 104 of the 439 mixtures (23.7%) were subject to the locus-dropping behavior. It is worth emphasizing that this is for OCME's own validation data, not a new test set to which they did not have access. It is hard to see how they could conclude that the CheckFrequencyForRemoval change was a minor one that did not need to be disclosed.

The change in LRs between FST versions is shown in Figure 1. In Figure 1, we report all four of the LRs reported for each sample, not just the lowest one as would be used for mapping onto verbal-scale equivalent labels when reporting results. The y-axis value for each point is the LR reported with CheckFrequencyForRemoval enabled and the x-axis value is the LR for the sample without CheckFrequencyForRemoval. Figure 1 explores the impact on the results for known contributors to a sample (a simulation of guilty parties).

Figure 2 explores the impact on the results for non-contributors (a simulation of innocent individuals). Forty

samples exhibiting locus-dropping behavior during the known contributor analysis were selected for comparison against 700 non-contributors from the national data set using both versions of FST. A range of samples were selected, from 15-575pg of template DNA as well as on the basis of deducible (20) vs. non-deducible (20); 2-person mixtures (12) vs. 3-person mixture (28); and HCN (17) vs. LCN (23). As in Figure 1, we report all four of the LRs reported for each sample, not just the lowest one. The y-axis value for each point is the LR reported with CheckFrequencyForRemoval enabled and the x-axis value is the LR for the sample without CheckFrequencyForRemoval.

In both Figure 1 and 2, points above the lightly dotted center line (y=x) represent where the modified version of FST reports an LR value that is more inclusionary and points below the line represent more exclusionary result. By dropping data, the results skew towards inaccuracy – they skew incorrectly towards more exclusionary (below the line) for real contributors and towards inclusionary (above the line) for non-contributors. For the non-contributors or simulated innocent individuals who did not actually contribute to the sample, the addition of CheckFrequencyForRemoval does skew towards inclusionary and thus is discarding information that could be helpful to the defense.

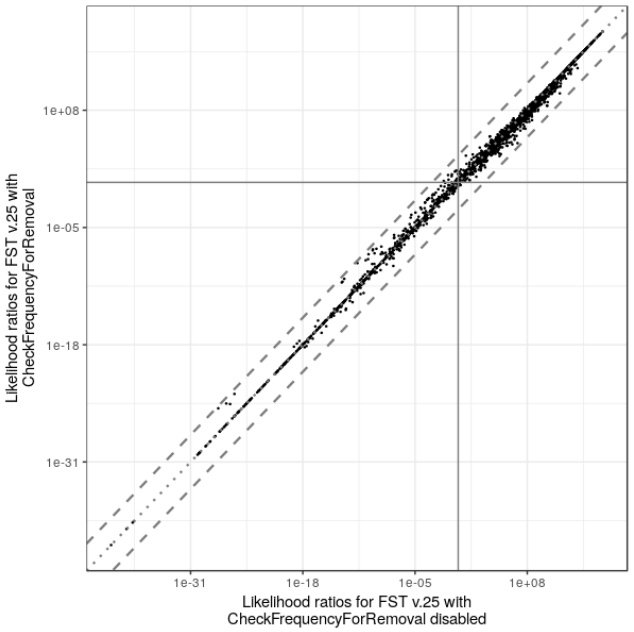

Figure 1: Known contributor likelihood ratios for FST v2.5 with the data-dropping function CheckFrequencyForRemoval vs. FST v2.5 with the function disabled. A log10 scale is used for both axes.

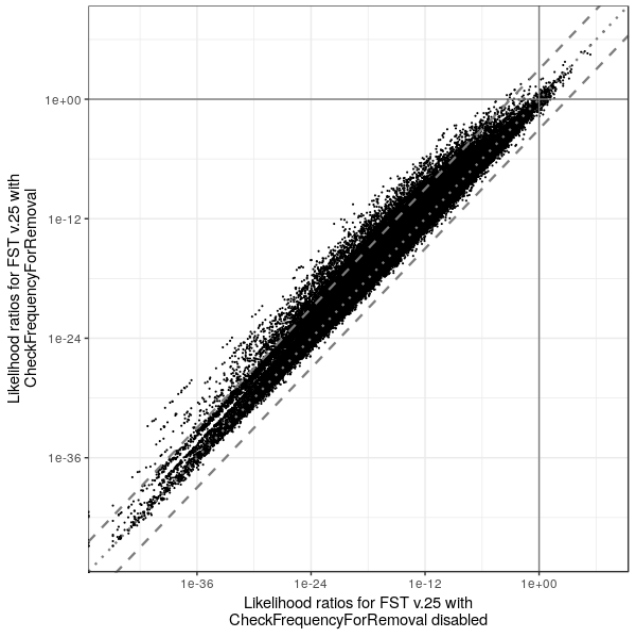

Figure 2: Non-contributor likelihood ratios for FST v2.5 with the data-dropping function vs. FST v2.5 with this function disabled. A log10 scale is used for both axes.

4.2 Changes In Verbal-Scale Equivalent Labels

Changes to LR verbal-scale equivalents are especially likely to impact the perceived weights of evidence. To examine this we used only the lowest LR reported for each sample rather than all four reported LRs as shown in Figures 1 and 2.

For the 1,245 known contributor tests described above, changes in verbal equivalencies were observed for thirty-six comparisons (2.9%). Eleven false-exclusions (0.9%) became more exclusionary when locus-dropping was enabled while only two became more inclusionary while still remaining below LR=1. Three true-inclusion LRs became falsely exclusionary when locus-dropping was enabled, and one false-exclusion LR changed to a true inclusion. For the non-contributor tests described above (40 samples compared to 700 non-contributing individuals for a total of 28,000 comparisons), a false inclusion rate of 0.08% is observed, higher than the 0.03% reported for all analyses conducted in the FST validation study.

In addition, five results changed from true-exclusion LRs when locus-dropping was disabled to false-inclusion LRs when locus-dropping was enabled. This is highly relevant for defense teams as it is represents four individuals who were in fact not contributors to an evidence sample who would have been incorrectly implicated by FST results. We also saw four LRs changed from false inclusions without locus-dropping to true exclusions when locus-dropping was enabled.

There were 115 (0.4%) LRs that varied in verbal-scale equivalency labels between FST versions. Of the 294 LRs reported between 0.001-1,000 by one or both FST versions, those 115 constituted 39.1% of LRs near 1, suggesting that non-contributor LRs near 1 are similarly susceptible to differences in verbal-scale equivalency labels between versions of FST.

Appendix 2 contains tables tracking changes in verbal-scale equivalents for both known contributors and non-contributors. For this set of samples for OCME's own validation study, the number of impacted samples is modest. However, it still makes clear that the impact does occur and could impact the fate of individuals in court. Individuals impacted would have no way of knowing that data helpful to their case was dropped and without disclosure of the FST source, there would have been little to no-incentive to ever repair, or even acknowledge, the problem.

5 Criminal Justice Decision Making

Individual defendants and the public often will not have this opportunity to look under the hood of software used in criminal justice decision making. So, it is important to use this case as a lens through which to consider the flaws in our criminal justice system. What incentives exist for debugging black-box software systems used in the criminal justice system in general? Would it have been possible for defense teams to find this issue without source code access? Would it be possible to know whether this bug is impacting a particular defendant? What would have been the incentives for disclosure or improvement if OCME had been allowed to deny defense expert review and adversarial testing indefinitely? What incentives exist for protecting our long treasured decision making principles such as the right to a public trial, the rights of defendants to review and confront the evidence against them and the right to equal justice under the law?

Here we propose a set of modifications to the holistic decision making process to encourage bugs in forensic DNA systems like FST to be found and fixed.

One key lesson is the importance of adversarial review (another cornerstone of the judicial process). If validation studies are designed by the developers, they are likely to focus on demonstrating the effectiveness of the system rather than on aggressively identifying problems. In an environment where any bug report is answered with "you are just complaining because you are guilty" what incentive will there be to even investigate reports of errors? Also, what will counteract the tendency to sweep errors under the rug when they are found or to put in an inappropriate fix to make the problem go away as we saw in the FST case? In this case, we note that it was only the persistence of defense teams that provided the last stop-gap measure for forcing debugging of these systems.

We recommend targeting the procurement phase of software. When labs use public money to purchase, validate, and train on PG software, procurement policies should require or at least give substantial credit for products that include pro-transparency factors. Such factors could include open-source software, access to software engineering artifacts including bug tracking/change log databases, internal testing plans and results, software requirements and specifications, hazard and risk assessments, design documents, etc. Ideally, developers or third parties would offer bug bounties or other funding streams to incentivize third party testing.

There are actually many PG software systems that all purport to do the same task. Each of these systems is claimed to be a reasonable model for evaluating mixed DNA, though their underlying mechanisms for calculating LRs vary. An incriminating result from any one PG systems can be damning evidence in court. Easier access for comparison testing would be advantageous if their outputs could be compared more directly. One key advance would be surfacing important parameters like drop-in rates, drop-out rates and the population frequency files used, rather than burying some values inside the source code. Establishing common granularities of variation (e.g. different drop-out rates per loci vs. one overall drop-out rate) would be an important advance. Common file formats for input data would be a further improvement, decreasing time costs and transcription risks (e.g. typos) when comparing results of different systems.

Most PG systems are designed with casework in mind. Many systems have no native ability to batch-process multiple evidence samples or compare a single evidence item to multiple reference profiles (e.g. a set of non-contributors or an offender DNA database). While it is possible to modify source-available systems to enable batch-processing, modifications risk introducing defects and require further software validation. APIs, or at least CLIs, could allow for easier batch-processing tasks in both casework and research settings. Requiring these during the procurement phase would be an important advance.

Open-source systems are attractive for obvious reasons of transparency, accountability, and traceability. It cannot be overemphasized that the post-validation modification made to FST was only publicly acknowledged by OCME after FST's source code was examined in conjunction with independent testing. Seemingly minor changes to source code can have substantial impacts in criminal casework.

Terms-of-service contracts for software in the criminal justice space should not have clauses preventing third-party review or publishing of results. Non-disclosure agreements and protective orders covering commercial systems, complicate reviews and prevent dissemination of results – regardless of how essential those findings are to defendants or to the criminal justice system as a while.

ACKNOWLEDGMENTS

The Brown Institute has generously supported this work through a 2018-2019 Magic Grant. Thank you to Anthony Mangiacapra and Graham Northup from Clarkson University and Richard Torres from The Legal Aid Society for their contributions and support. Thank you to the wonderful researchers and staff at Data and Society.

REFERENCES

[1] Alessandrini, F., Cecati, M., Pesaresi, M., Turchi, C., Carle, F., and Tagliabracci, A. 2003. Fingerprints as evidence for a genetic profile: morphological study on fingerprints and analysis of exogenous and individual factors affecting DNA typing. J. Forensic Sci. 48, 3 (2003), 586–592. DOI:https://doi.org/10.1520/JFS2002260

[2] Angwin, J., Larson, J., Mattu, S., and Kirchner, L. 2016. Machine Bias. ProPublica. Retrieved from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[3] Buolamwini, J. and Gebru, T. 2018. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. Proceedings of Machine Learning Research 81:1–15.

[4] Butler, J., Schoske, R., Vallone, P., Redman, J., Kline, M. et al. 2003. Allele frequencies for 15 autosomal STR loci on US Caucasian, African American, and Hispanic populations. J. Forensic Sci. 48, 4 (2003), 908–911.

[5] Chouldechova, A. 2017. Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. Big Data 5(2):153–163. DOI:https://doi.org/10.1089/big.2016.0047

[6] Forensic Statistical Tool Source Code (FST). 2017. https://github.com/propublica/nyc-dna-software

[7] Garofano, P., Caneparo, D., D'Amico, G., Vincenti, M., and Alladio, E. 2015. An alternative application of the consensus method to DNA typing interpretation for Low Template-DNA mixtures. Forensic Sci. Int. Genet. Suppl. Ser. 5, (2015), e422–e424. DOI:https://doi.org/10.1016/j.fsigss.2015.09.168

[8] Haned, H., Slooten, K., and Gill, P. 2012. Exploratory data analysis for the interpretation of low template DNA mixtures. Forensic Sci. Int. Genet. 6, 6 (2012), 762–774.

[9] Kirchner, L. 2017. Thousands of Criminal Cases in New York Relied on Disputed DNA Testing Techniques. ProPublica. Retrieved from https://www.propublica.org/article/thousands-of-criminal-cases-in-new-york-relied-on-disputed-dna-testing-techniques

[10] Mitchell, A., Tamariz, J., O'Connell, K., Ducasse, N., Budimlija, Z., Prinz, M., and Caragine, T. 2012. Validation of a DNA mixture statistics tool incorporating allelic drop-out and drop-in. Forensic Sci. Int. Genet. 6, 6 (2012), 749–761.

[11] People v. Collins, 2013. 15 N.Y.S.3d 564 (Sup.Ct. Kings Co. 2015), 6/17/2013.

[12] Mitchell, A. 2013. Testimony of Adele Mitchell in admissibility hearing in People v. Collins, 15 N.Y.S.3d 564 (Sup.Ct. Kings Co. 2015), 5/1/2013; 5/2/2013; 5/21/2013.

[13] National Research Council (NRC). 1996. The Evaluation of Forensic DNA Evidence. National Academies Press, Washington, D.C. DOI:https://doi.org/10.17226/5141

[14] New York City Office of the Chief Medical Examiner (OCME). 2016. Forensic Biology Protocols for Forensic STR Analysis: Forensic Statistical Tool (FST). Retrieved from https://www1.nyc.gov/assets/ocme/downloads/pdf/technical-manuals/protocols-for-forensic-str-analysis/forensic-statistical-tool-fst.pdf

[15] New York City Office of the Chief Medical Examiner. Technical Manuals (OCME). 2018. Retrieved November 5 2018 from https://www1.nyc.gov/site/ocme/services/technical-manuals.page

[16] NIST. 2017. NIST to Assess the Reliability of Forensic Methods for Analyzing DNA Mixtures. Retrieved August 8, 2018 from https://www.nist.gov/news-events/news/2017/10/nist-assess-reliability-forensic-methods-analyzing-dna-mixtures

[17] Paoletti, D., Doom, T., Krane, C., Raymer, M., and Krane, D. 2005. Empirical analysis of the STR profiles resulting from conceptual mixtures. J. Forensic Sci. 50(6): 1361–6.

[18] Steele, C. and Balding, D. 2014. Statistical evaluation of forensic DNA profile evidence. Annu. Rev. Stat. Its Appl. 1, (2014), 361–384.

[19] Tashea, J. 2017. Defense lawyers want to peek behind the curtain of probabilistic genotyping. ABA Journal. Retrieved from http://www.abajournal.com/magazine/article/code_of_science_defense_lawyers_want_to_peek_behind_the_curtain_of_probabil

[20] Wexler, R. 2018. Life, Liberty, and Trade Secrets: Intellectual Property in the Criminal Justice System. Stanford Law Rev. 70, 5 (2018), 1343. DOI:https://doi.org/10.2139/ssrn.2920883

[21] Worth, K. 2018. Framed for Murder By His Own DNA. PBS Frontline. Retrieved from https://www.pbs.org/wgbh/frontline/article/framed-for-murder-by-his-own-dna/

Appendix 1: CheckFrequencyForRemoval

This appendix contains the source code for the function CheckFrequencyForRemoval, available on GitHub at the following link: https://github.com/propublica/nyc-dna-software/blob/master/FST.Common/Comparison.cs.

```
/// This function checks for the total frequencies according to races
and removes the allelles from calculation

        /// if the sum of frequencies are greater than 0.97.

        /// </summary>

        public void CheckFrequencyForRemoval(DataTable dtFrequencies)

        {

            // if our db connection isn't initialized, do it. then,
get all the ethnicities (races)

            myDb = myDb ?? new Database();

            DataTable raceTable = myDb.getAllEthnics();

            int intsr = 0;

            string[] srem = new string[comparisonLoci.Count];



            // we go through all the comparison loci and check
whether the sum of the frequencies for that locus is greater than
0.97.

            // if it is, we remove the locus. frequencies are only
used for the alleles in the evidence replicates.

            for (int i = 0; i < comparisonLoci.Count; i++)

            {

                bool blRemove = false;

                // get a CSV list of alleles for all the replicates
at a locus

                IEnumerable<string> unknownPair =
EvidenceAllelesAtLocus(evidenceAlleles[comparisonLoci[i]]);

                // check if the frequency is greater than 0.97 for
any of the races. frequencies are values for an allele at a locus for
a certain race

                foreach (DataRow eachRow in raceTable.Rows)

                {
```

```
                    string raceName =
eachRow.Field<string>("EthnicName");
                    float freqSum = GetFrenquencySum(unknownPair,
comparisonLoci[i], raceName, dtFrequencies);

                    if (freqSum >= 0.97)
                    {
                        blRemove = true;
                        break;
                    }
                }
                if (blRemove)
                {
                    srem[intsr] = comparisonLoci[i];
                    intsr++;
                }
            }

            // now we iterate through all the loci and remove them
from the list of comparison loci, the evidence, and known and
comparison profiles
            for (int i = 0; i < srem.Length; i++)
            {
                if (srem[i] != null)
                {
                    string locus = srem[i];

                    // remove the locus from the comparisons
                    for (int j = 1; j <=
comparisonData.NumeratorProfiles.ComparisonCount; j++)
                    {
                        if(comparisonAlleles[j].ContainsKey(locus))
                            comparisonAlleles[j].Remove(locus);
                    }

                    // remove the locus from the knowns
                    int knownCount =
(comparisonData.NumeratorProfiles.KnownCount >
comparisonData.DenominatorProfiles.KnownCount)
                                        ?
comparisonData.NumeratorProfiles.KnownCount
                                        :
comparisonData.DenominatorProfiles.KnownCount;
                    for (int j = 1; j <= knownCount; j++)
                    {
                        if(knownAlleles[j].ContainsKey(locus))
                            knownAlleles[j].Remove(locus);
                    }

                    // remove the locus from the evidence replicates
                    for (int j = 1; j <= replicates; j++)
                    {
                        if(evidenceAlleles.ContainsKey(locus))
                            evidenceAlleles.Remove(locus);
                    }

                    // remove the locus from the list of comparison
loci
                    comparisonLoci.Remove(locus);
                }
            }
        }
```

Appendix 2: Tables of Changes in Verbal Scale Equivalent Labels

			FST v2.5 with CheckFrequencyForRemoval											
			Support for Hd				LR = 1	Support for Hp						
			Very strong	Strong	Moderate	Limited	Inconclusive	Limited	Moderate	Strong	Very strong			
FST v2.5 with CheckFrequencyForRemoval disabled	Support for Hd	Very strong	*186*	0	0	0	0	0	0	0	0	186	14.9%	20.5%
		Strong	3	*20*	1	0	0	0	0	0	0	24	1.9%	
		Moderate	2	4	*19*	1	0	0	0	0	0	26	2.1%	
		Limited	0	0	2	*16*	0	1	0	0	0	19	1.5%	
	LR = 1	Inconclusive	0	0	0	0	*0*	0	0	0	0	0	0.0%	0.0%
	Support for Hp	Limited	0	0	1	2	0	*24*	4	0	0	31	2.5%	79.5%
		Moderate	0	0	0	0	0	1	*34*	3	0	38	3.1%	
		Strong	0	0	0	0	0	0	6	*57*	2	65	5.2%	
		Very strong	0	0	0	0	0	0	1	2	*853*	856	68.8%	
			191	24	23	19	0	26	45	62	855	1,245		
			15.3%	1.9%	1.8%	1.5%	0.0%	2.1%	3.6%	5.0%	68.7%			
			20.6%					79.4%						

Table 1: Known-Contributor: Changes in verbal-scale equivalent labels for known contributor LRs between versions of FST with and without locus-dropping behavior. Italicized values on the diagonal indicate no change in label between versions.

			FST v2.5 with CheckFrequencyForRemoval											
			Support for Hd				LR = 1	Support for Hp						
			Very strong	Strong	Moderate	Limited	Inconclusive	Limited	Moderate	Strong	Very strong			
FST v2.5 with CheckFrequencyForRemoval disabled	Support for Hd	Very strong	*27,705*	13	1	0	0	0	0	0	0	27,719	99.0%	99.9 %
		Strong	42	*100*	9	0	0	0	0	0	0	151	0.5%	
		Moderate	9	6	*49*	9	0	1	0	0	0	74	0.3%	
		Limited	0	1	8	*20*	0	3	0	0	0	32	0.1%	
	LR = 1	Inconclusive	0	0	0	0	*0*	0	1	0	0	1	0.0%	0.0%
	Support for Hp	Limited	2	0	1	2	0	*8*	5	0	0	18	0.1%	0.1%
		Moderate	0	0	0	0	0	2	*1*	1	0	4	0.0%	
		Strong	0	0	0	0	0	0	0	*0*	0	0	0.0%	
		Very strong	0	0	0	0	0	0	0	0	*1*	1	0.0%	
			27,758	120	68	31	0	14	7	1	1	28,000		
			99.1%	0.4 %	0.2 %	0.1 %	0.0 %	0.1 %	0.0 %	0.0 %	0.0 %			
			99.9%					0.1%						

Table 2: Non-Contributor: Changes in verbal-scale equivalent labels for non-contributor LRs between versions of FST with and without locus-dropping behavior. Italicized values on the diagonal indicate no change in label between versions.