Mark K. Silver, Esq. (#019752000)
msilver@coughlinduffy.com
COUGHLIN DUFFY LLP
350 Mount Kemble Avenue
P.O. Box 1917
Morristown, New Jersey 07962-1917
Telephone: (973) 267-0058
Facsimile: (973) 267-6442

Dino L. LaVerghetta*
dlaverghetta@sidley.com
Matthew Hopkins (#230322017)
Matthew.hopkins@sidley.com
Iain C. Armstrong*
iarmstrong@sidey.com
SIDLEY AUSTIN LLP
1501 K Street, N.W.
Washington, D.C. 20005
Telephone: +1 202 736-8901
Facsimile: +1 202 736-8711

*Admitted *pro hac vice*

Attorneys for Amici Curiae

STATE OF NEW JERSEY,

Plaintiff-Respondent,

v.

COREY PICKETT,

Defendant-Appellant.

SUPERIOR COURT OF NEW JERSEY
APPELLATE DIVISION
Docket No. A-004207-19T2

Criminal Action

BRIEF OF *AMICI CURIAE* DRS. MATS HEIMDAHL AND JEANNA MATTHEWS

TO:

Joseph H. Orlando
Appellate Division Clerk's Office
P.O. Box 006
Trenton, New Jersey, 08625

Counsel for Respondent
Stephanie Davis Elson
Assistant Prosecutor
Hudson County Prosecutor's Office
595 Newark Avenue, 6th Floor
Jersey City, NJ 07306

TABLE OF CONTENTS

PAGE NOS.

TABLE OF AUTHORITIES

Page NOS.

Cases

Other Authorities

PRELIMINARY STATEMENT

TrueAllele is a black box. Information is entered, and out comes a result that could cause a defendant's life imprisonment or execution. Respondent does not want the Appellant, this Court, or anyone else to look inside the box. Acceding to that request would ignore sound science.

TrueAllele is a probabilistic genotyping ("PG") software program that purports to conduct extraordinarily complex mathematical computations to analyze DNA samples that cannot be analyzed by traditional methods. Put differently, TrueAllele is attempting to solve a problem that *cannot be verified manually*, meaning that lab users cannot check the system's accuracy. Troublingly, no one outside of Cybergenetics, TrueAllele's developer, knows how TrueAllele works. Software cannot be evaluated without full access to executable source code and related documentation. No one has been granted such access.

Even simple software programs are prone to flaws. A misplaced number, an incorrect assumption, or an unaccounted-for limitation can result in a failure. Problematically, such flaws are often latent and go undetected. The opportunities for and consequences of such flaws increase dramatically for complex software programs such as TrueAllele. Recent history is littered with examples of small flaws causing catastrophic failures.

It is virtually certain that there are flaws in the

TrueAllele software. On average, there will be six flaws for every 1,000 lines of code, and TrueAllele has 170,000 lines of code. Given its nature, TrueAllele is particularly likely to contain undetected flaws: users are unlikely to notice failures, the incentive structure makes reporting flaws less likely, and TrueAllele has not been subject to thorough, independent review.

Flaws have been discovered in other PG programs—including STRmix and Forensic Statistical Tool ("FST")—and in much simpler technologies such as breathalyzers. Those flaws—which called into question thousands of convictions—frequently went undiscovered until the source code was reviewed as part of the judicial process. Appellant's request to subject TrueAllele to such scrutiny is not only prudent but essential to determining whether TrueAllele operates as Cybergenetics claims.

INTEREST OF *AMICI CURIAE*[1]

Amici curiae are experts in engineering, testing, and validating computer systems, including forensic software. *Amici* submit this brief to explain why it is essential that Mr. Pickett have full access to the TrueAllele source code.

Dr. Mats Heimdahl is the Department Head of the Computer Science and Engineering Department of the University of Minnesota College of Science & Engineering. He has published on

[1] In addition to the *Amici*, eight other experts have expressed support for the filing of this brief. *See* Appendix A.

the engineering of safety critical software systems, including in top peer-reviewed journals. He has also served as an expert in numerous cases involving software engineering.

Dr. Jeanna Matthews is a Full Professor of Computer Science at Clarkson University and an affiliate at Data & Society. She is a member of the Association for Computing Machinery ("ACM") Council, founding co-chair of the ACM Technology Policy Subcommittee on Artificial Intelligence and Algorithm Accountability, and a member of the ACM Technology Policy Committee. She has particular expertise in evaluating the role of advanced technology in criminal justice. Dr. Matthews has authored or co-authored dozens of publications, including multiple articles focused on PG.

PROCEDURAL HISTORY AND STATEMENT OF FACTS

Amici rely on the history and facts stated by the parties.

ARGUMENT

I. Software Faults Are Ubiquitous

Source code is the human-readable formal plan for software that provides the instructions for how the computer will function. Simple programs can require thousands of lines of code, and complex programs, such as TrueAllele, can require hundreds of thousands of lines of code. Source code faults are ubiquitous and difficult to detect. As software complexity increases, so does the risk of faults.

A. Even simple software is prone to failure

Software engineering entails three primary "domains" or phases illustrated in the "Simple Example" below. An error in any of these domains makes the system untrustworthy.

Simple Example: the three domains of software engineering		
Problem Identification	**Algorithm Development**	**Software Implementation**
The need to compute the sum of the first *n* whole numbers (*i.e.*, 1+2+3+4+…+*n*) based on a user inputting the value of *n*.	Option 1: add each number until the first *n* numbers are added. Option 2: compute *n**(*n*+1)/2. Both algorithms arrive at the same result: 1+2+3+4=10 4*(4+1)/2=10	An engineer creates a program to implement the algorithm, requiring two inputs from the user: the number *n* and the first name of the user making the request. The program also gets the date and time from the computer on which it is running.

The Simple Example illustrates the development of a basic and verifiable software program. The problem is easy to understand and the algorithm can be verified by a mathematical proof. Still, there are many opportunities for error. The software might work for users named "Bob," but give the wrong result for users named "Mohamed," because the software was inadvertently designed to handle names of three characters or fewer. More troublingly, the software might print the wrong sum for certain numbers, giving the correct answer when *n* is 92,672, but the wrong answer when *n* is 92,689 because the highest number that the program can process is 4,294,967,295.

These types of failures occur, in part, because software is

non-continuous and does not behave like traditional engineered systems. Traditional systems follow the laws of physics, which makes them relatively straightforward. Consider a simple crane used to lift boats. The crane has been designed to handle x kilograms before failing. Once the crane has been shown to lift boats weighing x kilograms, one can be confident that it will lift boats weighing 0.4 x kilograms. The result would not differ depending upon the day of the week, the color of the boat, or the name of the boat's owner. The same cannot be said for the software used to process the handling of the boats. If the software has not been programmed to handle certain cases (such as weekend transactions, red boats, or long names), it might fail entirely or produce erroneous results.

Because software is non-continuous, test results cannot be interpolated; any input could cause a failure. One cannot assume that because a software program gives the correct result when a user inputs 3,000 or 5,000 the software will work when the user enters 4,000. Because the number of potential inputs to a program is astronomical and testing them all is impossible, the software might fail entirely for any given input, or, worse, produce erroneous results that appear plausible.

B. Opportunities for error are higher in complex programs

Most software flaws result from simple mistakes. The more complex the program, the greater the risk of flaws. "A research

study has shown that professional programmers average six software defects for every 1000 lines of code (LOC) written." Hoang Pham, *Software Reliability*, *in* WILEY ENCYCLOPEDIA OF ELECTRICAL AND ELECTRONICS ENGINEERING 565, 565 (John G. Webster ed., 1999).

In many situations, software faults are not discovered until they result in obvious and catastrophic failures. For example, NASA's Mars Climate Orbiter exploded because the software controlling its thrust was written to use English units instead of metric units. *See* Andrew Pollack, *Missing What Didn't Add Up, NASA Subtracted an Orbiter*, N.Y. Times (Oct. 1, 1999).[2] Two Boeing 737 MAX aircraft crashed because a software modification made the aircraft vulnerable to nosedives. *See* Niraj Chokshi, *House Report Condemns Boeing and F.A.A. in 737 Max Disasters*, N.Y. Times (Sept. 16, 2020).[3] An Ariane 5 rocket exploded because a fault in a program tried to "stuff a 64-bit number into a 16-bit space." James Gleick, *Little Bug, Big Bang*, N.Y. Times (Dec. 1, 1996).[4] A "software glitch" in Therac-25 radiation therapy machines resulted in cancer patients receiving excessive radiation when a certain group of commands was entered. *Fatal Radiation Dose in Therapy Attributed to Computer Mistake*, N.Y. Times (June 21, 1986).[5]

---

[2] https://nyti.ms/34GaQCR.
[3] https://nyti.ms/2GL0E3u.
[4] https://nyti.ms/2GRp0sA.
[5] https://nyti.ms/34wZpx6.

Software can also have errors that cause subtle yet serious failures. For example, software faults caused a jury selection program to exclude zip codes where most African-Americans lived. Nina W. Chernoff, *No Records, No Right: Discovery & the Fair Cross-Section Guarantee*, 101 Iowa L. Rev. 1719, 1723-24, 1731-32 (2016). As the Sixth Circuit recognized, "[t]he glitch was a mistyped parameter in the software, buried in a mountain of computer code, that was only discovered after a broad statistical analysis led to an extensive internal investigation." *Ambrose v. Booker*, 684 F.3d 638, 645 (6th Cir. 2012).[6] Such latent faults are the hardest errors to find and fix, leading to undetected harm over many years.[7]

II. TrueAllele's Source Code Likely Contains Undetected Flaws

Numerous factors suggest that TrueAllele is likely to contain undetected flaws, including that: (1) flaws have been discovered in other PG programs and less complex forensic tools, often only after source code was produced pursuant to judicial orders; (2)

[6] Another example of a latent fault is a software bug that mistakenly caused convicts in Washington to be released early from prison for more than a decade. See Michelle Shephard, *More Than 3,200 US Prisoners Have Been Released Early Because of a Software Glitch*, BBC News (Dec. 23, 2015), https://bbc.in/2FfsDIh.

[7] While most software faults are unintentional, complex source code also provides bad actors with an opportunity to do harm in subtle ways. A high-profile example is Volkswagen's use of buried source code on 11 million cars to cheat state emissions tests. Mike Spector & Mike Colias, *Volkswagen Pleads Guilty to Criminal Charges in Emissions-Cheating Scandal*, Wall St. J., (Mar. 10, 2017), https://on.wsj.com/30MqxXR.

flaws are unlikely to be noticed because forensic lab users cannot check the system's accuracy; (3) unlike commercial software, the incentive structure for forensic software does not encourage reporting flaws; (4) TrueAllele has not been subject to thorough, independent review; and (5) experts have raised concerns about TrueAllele's lack of reproducibility.

A. Other forensic programs, including PG programs, have been found to contain faults

Recent history is littered with examples of latent flaws in forensic software being discovered after the software was used in numerous arrests and convictions. These examples include not only flaws in other PG software programs, but also flaws in much simpler programs.

In total, at least thirteen "coding faults" have been found in STRmix, TrueAllele's chief competitor.[8] In one notable example, the miscode impacted 60 criminal cases, requiring new likelihood ratios to be issued in 24 cases. David Murray, *Queensland Authorities Confirm 'Miscode' Affects DNA Evidence in Criminal Cases*, Courier-Mail (Mar. 20, 2015).[9]

Another PG program, FST, avoided independent review for years until a federal judge ordered source code disclosure. Stephanie J. Lacambra et al., *Opening the Black Box: Defendants'*

[8] *See* STRmix, *Summary of Miscodes* https://bit.ly/36ILKWi (last updated Sept. 15, 2020).
[9] https://bit.ly/34DBlZy.

Rights to Confront Forensic Software, The Champion, May 2018, 28, 32. The defendant's expert discovered that "[a] secret function . . . was present in the software, tending to overestimate the likelihood of guilt," and that "[t]he actual functioning of the software, revealed upon inspection of the source code, did not use the methodology publicly described in sworn testimony and peer-reviewed publications." *Id.*[10] Once these flaws were discovered, a high-profile conviction based on FST analysis was overturned. *See* Alan Feuer, *Hasidic Man Convicted of Beating Black Student Gets Verdict Overturned*, N.Y. Times (Oct. 10, 2018).[11]

In recent years, thousands of faults have been discovered in the source code of top breathalyzer systems. According to a 2019 *New York Times* investigation, breathalyzers "generate skewed results with alarming frequency," and "Judges in Massachusetts and New Jersey have thrown out more than 30,000 breath tests in the past 12 months alone." Stacey Cowley & Jessica Silver-Greenberg, *These Machines Can Put You in Jail.*

---

[10] *See also* Jeanna Neefe Matthews et al., *When Trusted Black Boxes Don't Agree: Incentivizing Iterative Improvement and Accountability in Critical Software Systems*, 2020 Proc. AAAI/ACM Conf. on AI, Ethics, & Soc'y 102, 103 (FST developers "aggressively resisted expert witness review that could have exposed the problem for 5 years while using the output of the system as evidence in over 1000 serious criminal cases.").

[11] https://nyti.ms/33GHuoD.

Don't Trust Them, N.Y. Times (Nov. 3, 2019).[12] After the New Jersey Supreme Court granted source code access, defense experts found that the code was "littered with 'thousands of programming errors.'" *Id.* Similar errors were found across the country. *Id.*

It is unlikely that TrueAllele is somehow free of flaws. Rather—as was the case with the forensic tools discussed above—TrueAllele's flaws will simply go undetected until defendants are granted meaningful access to the source code.[13]

B. Flaws in TrueAllele are unlikely to be noticed

Although some faults may generate an error message or crash a program entirely, other faults operate more stealthily. A fault in TrueAllele's source code would likely not prevent the system from generating a match statistic; it would just prevent the system from generating an *accurate* match statistic. And in the vast majority of cases, the technician operating the machine would never be able to tell that the result was incorrect.

Such dormant flaws are a bigger problem for complex programs such as TrueAllele. In simpler systems, the user can evaluate the program's results based on other forms of analysis. For example, if a computer's calculator application indicates that two plus two equals five, the user can check the math by

[12] https://nyti.ms/3jHAyNt.

[13] *See* Natalie Ram, *Innovating Criminal Justice*, 112 Nw. U. L. Rev. 659, 682 (2018) ("In the few cases in which courts have compelled disclosure of private source code . . . reviewers identified significant errors in almost every instance.").

performing the addition mentally. Not so with TrueAllele.

With PG, the software is attempting to solve a problem that *cannot be verified manually*. Technicians cannot "check the math" on the results, because it is not humanly possible to perform TrueAllele's calculations. The only indication of what the right answer "should be" is the result from TrueAllele. Even if a match statistic was inflated by a factor of a million, the lab would probably not be able to tell that a failure had occurred.

C. Cybergenetics and law enforcement have incentives to not identify or report software flaws

In case after case, including Mr. Pickett's, software companies and governments have demonstrated a determination to avoid subjecting PG programs to meaningful scrutiny. This is likely due, in part, to the unique incentive structure that applies to forensic software. In traditional commercial software, failures are often discovered and reported by customers. The manufacturer has a financial incentive to quickly correct the flaws. With criminal justice software, however, this incentive structure does not necessarily exist. The customers are prosecutors, but those harmed by software flaws are criminal defendants. *See* Matthews et al., supra note 10, at 103.[14]

[14] There are incentives to dismiss defendants' claims that results in their cases could not be accurate. *See* Jeanna Matthews et al., *You're Just Complaining Because You're Guilty: A DEF CON Guide to Adversarial Testing of Software Used in the Criminal Justice System* (Aug. 11, 2018), https://youtu.be/4cscBvDYP-Q.

Even if state employees were able to detect failures in TrueAllele's source code, they may not have an incentive to report those flaws. TrueAllele allows prosecutors to achieve convictions where they otherwise would not. If the results support conviction, there is a risk that the government will not reliably report failures.[15] Similarly, Cybergenetics may not have an incentive to identify flaws in TrueAllele. A flaw's discovery could cast doubt on the product's reliability, undermine prior convictions, and threaten Cybergenetics' financial livelihood.

This incentive structure likely impacted how the government handled the case of Florencio Jose Dominguez. Due largely to results from a PG program, Dominguez was convicted of murder and sentenced to 50 years to life. Greg Moran, *Murder Case that Highlighted DNA-Analysis Controversy Ends with Plea to Reduced Charge, Release*, San Diego Union Trib. (Dec. 6, 2019).[16] When Dominguez's counsel moved to reopen the case based on suspicions about the DNA analysis, a California judge ordered the disclosure of the program's source code. *Id.* But instead of giving Dominguez a chance to conduct an evaluation of the software—during which flaws could be discovered—the prosecutors

[15] For example, significant lapses and misconduct at the FBI Crime Lab went unreported and/or uninvestigated for years. *See* David Johnston, *Report Criticizes Scientific Testing at F.B.I. Crime Lab*, N.Y. Times (Apr. 16, 1997) (quotation marks omitted), https://nyti.ms/3nPWWXc.

[16] https://bit.ly/3nszGOZ.

allowed him to walk free. *Id.* The government's actions suggest that it would go to great lengths to prevent the discovery of flaws in the software.[17]

D. Existing testing of TrueAllele is incomplete and unreliable

1. The validation studies were not independent

One serious flaw in the TrueAllele "validation" work cited by the Respondent is that those studies originated almost exclusively from within Cybergenetics' orbit. *See* Appellant's Br. 29 n.12. Of the 36 validation studies cited by the Respondent, 35 came from Cybergenetics, Mark Perlin (who runs Cybergenetics), or law enforcement agencies. *Id.*[18]

The lack of independent review raises serious concerns about the reliability of the studies, and was the chief criticism of PG software, including TrueAllele, in a report by the President's Council of Advisors on Science and Technology (the "PCAST Report").[19] The PCAST Report called for more testing that "should be performed by or should include independent

---

[17] *See* Matthews et al., *supra* note 10, at 102 ("[D]evelopers may be tempted to avoid costly debugging by claiming intellectual property protection in order to keep knowledge of known problems away from defendants . . . ."); Lacambra et al., *supra*, at 38 ("[P]rosecutors consistently urge courts to . . . deprive criminal defendants of access to forensic software.").

[18] Only a single PowerPoint presentation came from another source. *Id.*

[19] President's Council of Advisors on Sci. & Tech., Exec. Office of the President, *Forensic Science in Criminal Courts: Ensuring Scientific Validity of Feature-Comparison Method* 78-81 (Sept. 2016), https://bit.ly/34D6L1X.

research groups not connected with the developers of the methods and with no stake in the outcome." *Id.* at 81.[20]

2. The validation studies did not have access to source code

None of the validation studies cited by the Respondent had access to the TrueAllele source code. *See* Appellant's Br. 28. Thus, even if the studies support the soundness of the TrueAllele algorithm, it is impossible for them to establish that the algorithm is correctly implemented in the TrueAllele software. Without access to the program source code, researchers can say no more than that the results generated by the program are plausible. But, the software could appear to produce plausible results while still concealing latent errors.

Quite simply, studies that do not have access to source code cannot verify that the program is operating correctly:

> *[R]eliance on validation studies in place of source code access, rather than alongside it, is likely insufficient to verify that software has performed as its designer claims. In part, this stems from the limited verification that can be gleaned from "black-box testing"—testing that "considers only the inputs and outputs of a system or component." As technologists have explained, "[c]omputer scientists . . . have shown that black-box evaluation of systems is the least powerful of a set of available methods for understanding and verifying system behavior." More powerful and effective is "white-box testing," in which "the person doing a test can see the system's code and uses that knowledge to more effectively search for bugs."*

[20] A cornerstone of the gate-keeping test for expert opinions in civil cases is independent validation and reproducibility. *See Daubert v. Merrell Dow Pharms., Inc.*, 509 U.S. 579, 593 (1993). Surely, a lesser standard should not apply in criminal cases.

> *Accordingly, researchers have concluded that, to enable effective scientific inquiry, "anything less than the release of source programs is intolerable . . . ."*

Ram, *supra*, at 688-89 (citations omitted).

3. The validation studies were incomplete

Another major shortcoming "of all the published TrueAllele® validation studies is that the number of samples tested was relatively small." William C. Thompson et al., *Forensic DNA Statistics: Still Controversial in Some Cases*, The Champion, Dec. 2012, 12, 20.[21] Because of the non-continuous nature of software, the results from a small set of inputs cannot be reliably interpolated into cases involving different sets of inputs.[22] Unless the DNA profiles and contribution proportions analyzed in this case are similar to the limited samples used in the validation studies, those studies are of little value here.

Experts who have evaluated the validation studies have doubts about applying those studies to complex cases:

> *[E]xisting validation is insufficient to prove that TrueAllele® can consistently make correct genotype inferences in challenging, problematic cases such as*

[21] PG software is generally not validated for samples involving a large number (e.g., four or more) contributors, but, in practice, PG *is* used on such samples or where a large number of contributors cannot be ruled out (e.g., Touch DNA on a gun).

[22] Although some of the TrueAllele validation studies were published in peer-reviewed journals, it is important not to overstate what that means. A journal's decision to publish a study is based merely on whether the results should be seen by the scientific community. Publication of a software validation study is not—and is not intended to be—a stamp of approval endorsing the program's use in the criminal justice system.

> *mixture cases with unequal contributions from the contributors, limited quantities of DNA, degradation due to environmental insult, etc.*

Id.

E. Third-parties have expressed concerns about the reproducibility of TrueAllele's results

Experts have criticized TrueAllele for its inability to reproduce results. Reproducibility "is a central requirement of the scientific process." Peter Ivie & Douglas Thain, *Reproducibility in Scientific Computing*, 51 ACM Comput. Surv. 3, 63:1 (July 2018). However, TrueAllele regularly produces dissimilar likelihood ratios from multiple analyses of the same sample. Thompson et al., *supra*, at 20.

For example, in one case, Cybergenetics analyzed a single sample four times and produced four different likelihood ratios: 389 million, 1.9 billion, 6.03 billion, and 17.8 billion. *Id.* Mark Perlin chose to report the 6.03 billion number, reasoning that "it was the center of the range of values." *Id.* Such a degree of unexplained variation is troubling.

III. Full Access to the TrueAllele Source Code and Supporting Materials Is Necessary

Review of TrueAllele's source code is necessary to identify the existence and import of any flaws in the program. But that review must be meaningful, and cannot consist of merely allowing an expert to review TrueAllele's 170,000 lines of code with a pad and paper. The restrictions that Cybergenetics has imposed

on Appellant's review of the source code, Appellant's Br. 5-9, would render expert review essentially meaningless. TrueAllele's code cannot be meaningfully reviewed without full access to the executable source code and software development documentation. *See* Lydia Pallas Loren & Andy Johnson-Laird, *Computer Software-Related Litigation: Discovery and the Overly-Protective Order*, 6 Fed. Cts. L. Rev. 1, 14-18 (2012).

Three of the restrictions, in particular, render meaningful review impossible: (1) the inability to compile and execute the source code; (2) the inability to access software development documentation; and (3) the inability to communicate with subject-matter experts.

A. Access to executable software is necessary

Preventing Appellant's expert from compiling and executing the code makes it impossible for the expert to test the software or understand how it operates. *See id.* at 47 ("A clause permitting only handwritten notes is burdensome in the extreme."); *id.* at 53-54 ("[T]he prohibition on actually compiling the source code is mystifying."). Without the ability to actually execute the source code, Mr. Pickett's expert would be restricted to just theorizing about the program.

Quite simply, other than by running the program, Mr. Pickett's expert cannot evaluate how the software actually operates, whether the results yielded by TrueAllele are

reproducible, or whether it works the way that Cybergenetics claims. *See* Lacambra et al., *supra*, at 29. As experts in the field, *Amici* know that a complex system cannot be debugged using a pen and paper. Any legitimate search for flaws requires, at a minimum, running the program and observing how the system responds to various inputs. A meticulous paper review of source code might spot some obvious faults, but, in general, an expert would not be able tell the difference between an error and an unusual coding choice without actually running the software. *Id.*

B. Access to supporting documentation is necessary

In order to evaluate the reliability of TrueAllele, Appellant's expert would need access to TrueAllele's software development documentation, including testing, software design, bug reporting, change logs, and program requirements. *See* Loren & Johnson-Laird, *supra*, at 17-18.

Such documentation not only acts as a "road map" for the expert to understand the source code, *id.* at 17, but also allows the expert to determine whether Cybergenetics followed industry standards in developing TrueAllele. Because software is error prone, there are industry standards for software verification and validation. *See, e.g.*, *IEEE Standard for System and Software Verification and Validation*, IEEE Std 1012-2012; Sci. Working Grp. on DNA Analysis Methods, *Guidelines for the Validation of*

Probabilistic Genotyping Systems (June 2015).[23] Access to TrueAllele's software development documentation would allow an expert to determine whether those standards were followed.[24]

The software development documentation would also allow the expert to determine whether Cybergenetics followed the heightened protocols that should be applied to safety-critical systems, which are systems in which failures could cause loss of life, significant property damage, or environmental damage. *Amici* believe that TrueAllele is a safety-critical system, because a malfunction can result in the wrongful execution or incarceration of an innocent individual. An international standard known as IEC 61508 addresses the standards applicable to the development of safety-critical software systems.[25]

Access to development documentation also allows a reviewer to focus on areas where material faults might exist. *See* Loren & Johnson-Laird, *supra*, at 17-18. For example, such access would

---

[23] https://bit.ly/3lrI13D.

[24] For example, such documentation would show if TrueAllele was audited through independent verification and validation (IV&V). IV&V is "[v]erification and validation (V&V) performed by an organization that is technically, managerially, and financially independent..." Ron Ross et al., Nat'l Inst. Of Standards & Tech., Special Pub. 800-160, *Systems Security Engineering: Considerations for a Multidisciplinary Approach in the Engineering of Trustworthy Secure Systems* 168 (2016).

[25] Systems are categorized by "safety integrity level," ranging from SIL1 to SIL4. Each safety level has additional requirements, with SIL4 being the most demanding. *See Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems*, IEC 61508, 2010.

allow an expert to evaluate why Cybergenetics revised TrueAllele's source code more than twenty-five times. *See* Andrea Roth, *Trial by Machine*, 104 Geo. L.J. 1245, 1273 (2016). "[W]ith no published documentation as to what has been revised or why," *id.*, and without access to the source code, "it is impossible to know whether those changes corrected undisclosed errors or inadvertently introduced new ones." Ram, *supra*, at 681.

C. Communication with subject-matter experts is necessary

The restrictions preventing Mr. Pickett's expert from communicating with others while reviewing TrueAllele's source code are particularly problematic, given that TrueAllele is a cross-disciplinary program. An expert reviewing the source code would need to consult with subject-matter experts, including statisticians and biologists, in order to determine whether the problem identification, algorithm development, and software implementation domains were properly defined and executed.

CONCLUSION

Mr. Pickett's very freedom hinges upon the results yielded by a black-box software program. Mr. Pickett and the judicial system more generally deserve to understand how that program works, and the only means of doing so is by providing full access to the executable source code and supporting documentation. The lower court refused to order such access.

Accordingly, this Court should reverse.

Respectfully submitted,

By: */s/ Mark K. Silver*

Mark K. Silver, Esq.
COUGHLIN DUFFY LLP
350 Mount Kemble Avenue
P.O. Box 1917
Morristown, New Jersey 07962-1917
Telephone: (973) 267-0058
Facsimile: (973) 267-6442
msilver@coughlinduffy.com

Dino L. LaVerghetta*
Matthew Hopkins
Iain C. Armstrong*
SIDLEY AUSTIN LLP
1501 K Street, N.W.
Washington, D.C. 20005
Telephone: +1 202 736-8901
Facsimile: +1 202 736-8711

Attorneys for Amici Curiae
Dated: October 14, 2020

*Admitted *pro hac vice*

Mark K. Silver, Esq. (#019752000)
msilver@coughlinduffy.com
COUGHLIN DUFFY LLP
350 Mount Kemble Avenue
P.O. Box 1917
Morristown, New Jersey 07962-1917
Telephone: (973) 267-0058
Facsimile: (973) 267-6442

Dino L. LaVerghetta*
dlaverghetta@sidley.com
Matthew Hopkins (#230322017)
Matthew.hopkins@sidley.com
Iain C. Armstrong*
iarmstrong@sidey.com
SIDLEY AUSTIN LLP
1501 K Street, N.W.
Washington, D.C. 20005
Telephone: +1 202 736-8901
Facsimile: +1 202 736-8711

*Admitted *pro hac vice*

Attorneys for Amici Curiae

STATE OF NEW JERSEY,

Plaintiff-Respondent,

v.

COREY PICKETT,

Defendant-Appellant.

SUPERIOR COURT OF NEW JERSEY
APPELLATE DIVISION
Docket No. A-004207-19T2

Criminal Action

APPENDIX TO BRIEF OF *AMICI CURIAE* DRS. MATS HEIMDAHL AND JEANNA MATTHEWS

TO:

Joseph H. Orlando
Appellate Division Clerk's Office
P.O. Box 006
Trenton, New Jersey, 08625

Counsel for Respondent
Stephanie Davis Elson
Assistant Prosecutor
Hudson County Prosecutor's Office
595 Newark Avenue, 6th Floor
Jersey City, NJ 07306

Appendix A:
Experts Who Have Expressed Support for the Filing of the Amicus Brief Submitted by Drs. Mats Heimdahl and Jeanna Matthews[1]

1. **JAYADEV ATHREYA, Ph.D.,** is a Professor of Mathematics and the Comparative History of Ideas at the University of Washington; the founder of the Washington Experimental Mathematics Lab; and the managing editor for the journal Experimental Mathematics. He is also Special Advisor to the Director for the Pacific Institute of Mathematical Sciences. He obtained his Ph.D. in mathematics in 2006 from the University of Chicago, and has held positions at Yale, Princeton, and the University of Illinois.
2. **RICARDO BAEZA-YATES, Ph.D.,** is Director of Data Science Programs at Northeastern University, Silicon Valley campus, since 2017. Before, he was VP of Research at Yahoo Labs from 2006 to 2016. He is co-author of the best-seller Modern Information Retrieval textbook published by Addison-Wesley in 1999 and 2011 (2nd ed.), that won the Association for Information Science and Technology 2012 Book of the Year award. In 2009 he was named Association for Computing Machinery ("ACM") Fellow and in 2011 Institute of Electrical and Electronics Engineers ("IEEE") Fellow, among other awards and distinctions. His areas of

[1] All of the listed experts have expressed support in their individual capacities.

expertise are web search and data mining, information retrieval, data science and algorithms in general.

3. **MARC CANELLAS, Ph.D.,** is the Vice-Chair of the IEEE-USA Artificial Intelligence and Autonomous Systems Policy Committee and a third-year law student at the New York University School of Law. He has a Ph.D. in aerospace and cognitive engineering from the Georgia Institute of Technology. He is an expert in human-machine interaction in complex, safety-critical systems; and the governance of advanced technology, particularly in the criminal legal system.

4. **JAMES HENDLER, Ph.D.,** is the Director of the Institute for Data Exploration and Applications and the Tetherless World Professor of Computer, Web, and Cognitive Sciences at Rensselaer Polytechnic Institute. He has authored over 400 books, technical papers, and articles in the areas of Semantic Web, artificial intelligence, agent-based computing, and high-performance processing. He is the chair of the ACM's US technology policy committee and a Fellow of the Association for the Advancement of Artificial Intelligence, the British Computer Society, the IEEE, the American Association for the Advancement of Science, the ACM, and the National Academy of Public Administration. He

is also the former Chief Scientist of the Information Systems Office at the US Defense Advanced Research Projects Agency ("DARPA").

5. **REBECCA MERCURI, Ph.D.,** is the founding President and lead digital forensic investigator of Notable Software, Inc. Her doctoral dissertation, other writings in peer-reviewed publications (including for the ACM and IEEE) regarding the inherent insecurity and fallibility of black box software in electronic voting equipment, and decades of personal advocacy, have been acclaimed for spearheading the trend toward global adoption of voter verified paper ballots in public elections. Earlier, in her Computer Science career, she performed line-by-line reviews of source code deployed in aircraft collision avoidance systems. Now, as an expert witness, she has testified in State and Federal U.S. Courts regarding the need to perform such reviews of black box forensic tools used remotely by law enforcement to collect data on unsuspecting citizens.

6. **FALCON DARKSTAR MOMOT** is a general information security analyst with 6 years of experience, and provides technical leadership on projects to test systems for security bugs and unexpected functionality at a variety of companies in different industries, comprising hardware devices,

software solutions, and networks. He serves on the program committee for two information security industry conferences. His work on a DARPA project to detect insider threats using fundamental principles of computing resulted in his being named in 7 patents. Falcon is studying toward an M.Sc. Information Systems, and holds a B.Sc. Computer Science from the University of Lethbridge and a CISSP.

7. **DAVID MUSSINGTON, Ph.D.,** is Professor of the Practice and Director of the Center for Public Policy and Private Enterprise at the University of Maryland College Park's School for Public Policy. He is an internationally published expert in critical infrastructure cybersecurity. Dr. Mussington was Senior Advisor for Cyber Policy at the Office of the Secretary of Defense, and is a former member of the White House National Security Council Staff. He is a member of the Advisory Board of Verified Voting, the ACM US Technology Policy Committee, and an elected Board member of the International Information System Security Certification Consortium ("ISC^2").

8. **DAVID WAGNER, Ph.D.,** is Professor of Computer Science at the University of California at Berkeley, with expertise in the areas of computer security, computer science, and electronic voting. He has published over 100 peer-reviewed

papers in the scientific literature and has co-authored two books. His research has analyzed and contributed to the security of cellular networks, 802.11 wireless networks, electronic voting systems, and other widely deployed systems. He has testified before Congress on the importance of access to source code for electronic voting machines.