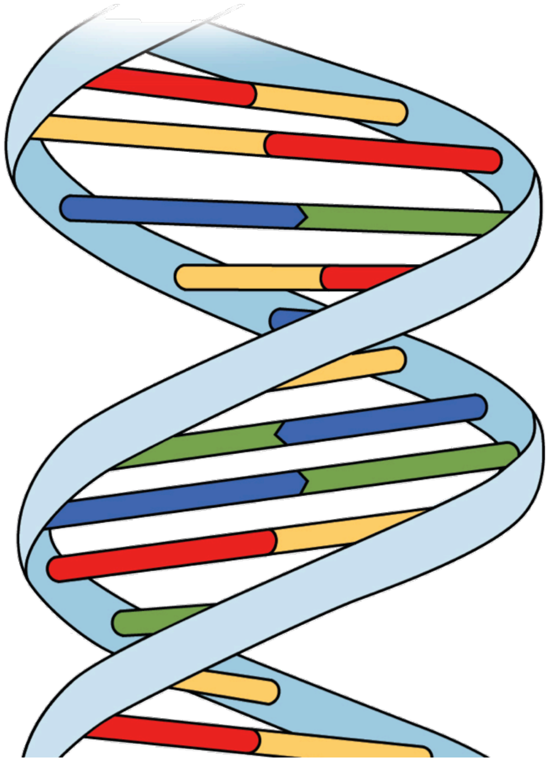
Opening the Black Box: Confronting Software-Based Evidence



Jeanna Matthews
Clarkson University/Data and Society

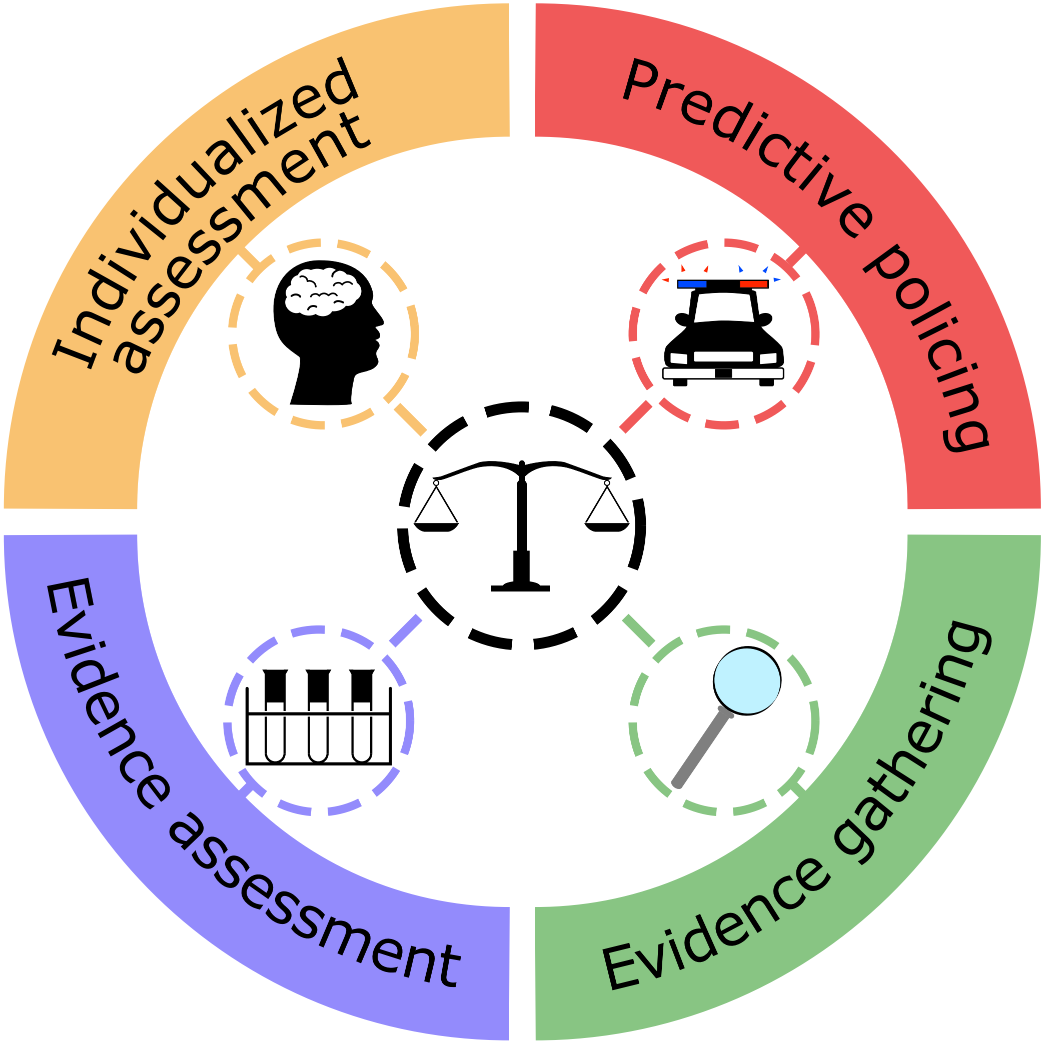
Questioning Forensics: Lawyers, Damn Lawyers, and Statistics

November 2, 2018

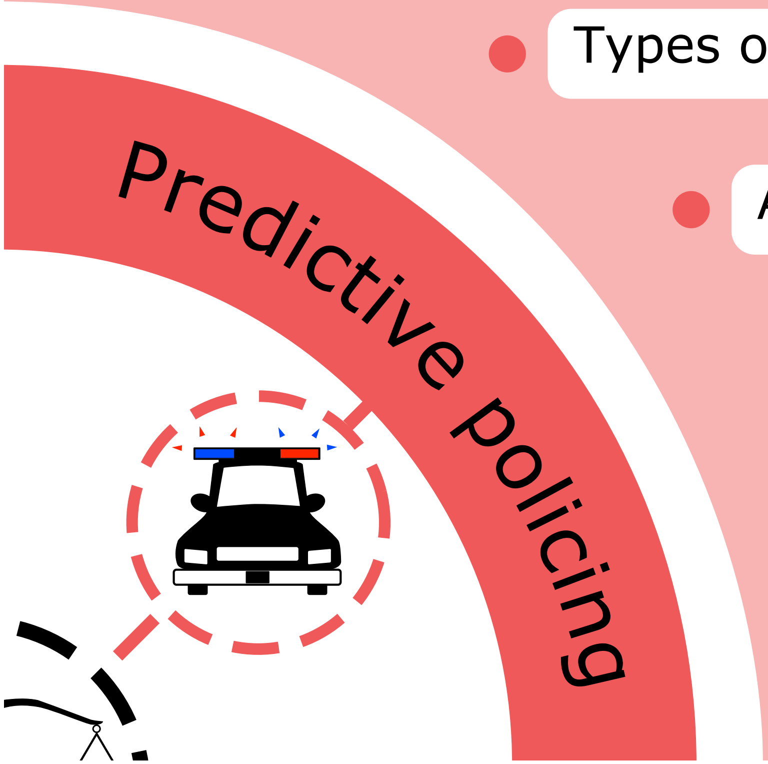
Software in the Criminal Justice System

- Software/algorithmic decision making used increasingly throughout the criminal justice system
  - Investigation/policing decisions, pretrial decisions, evidence at trial, sentencing decisions, parole decisions

- Often black boxes for which trade secret protection is claimed to be more important than rights of individual defendants or citizens to understand the decisions

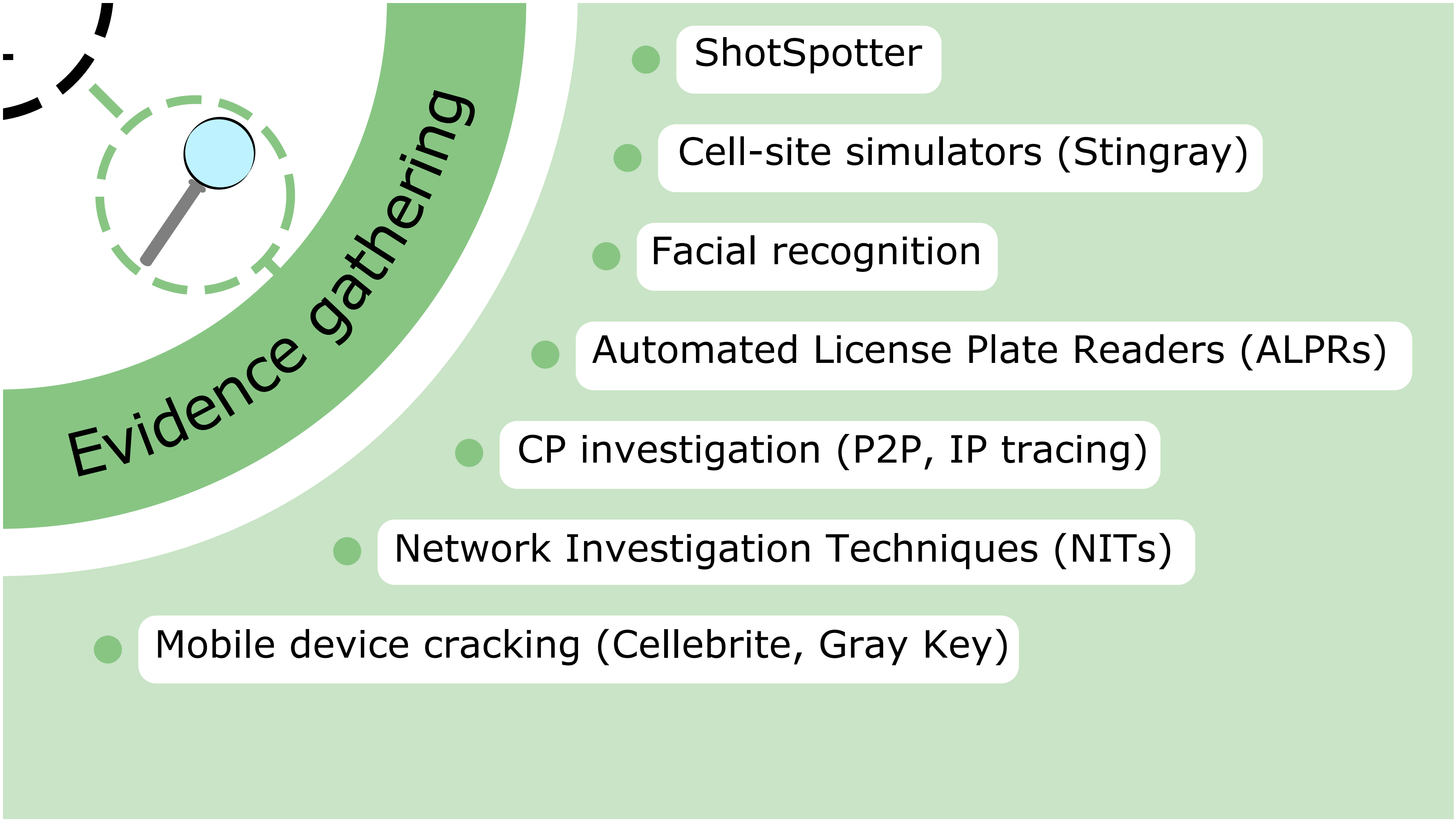


Individualized assessment
Predictive policing
Evidence assessment
Evidence gathering



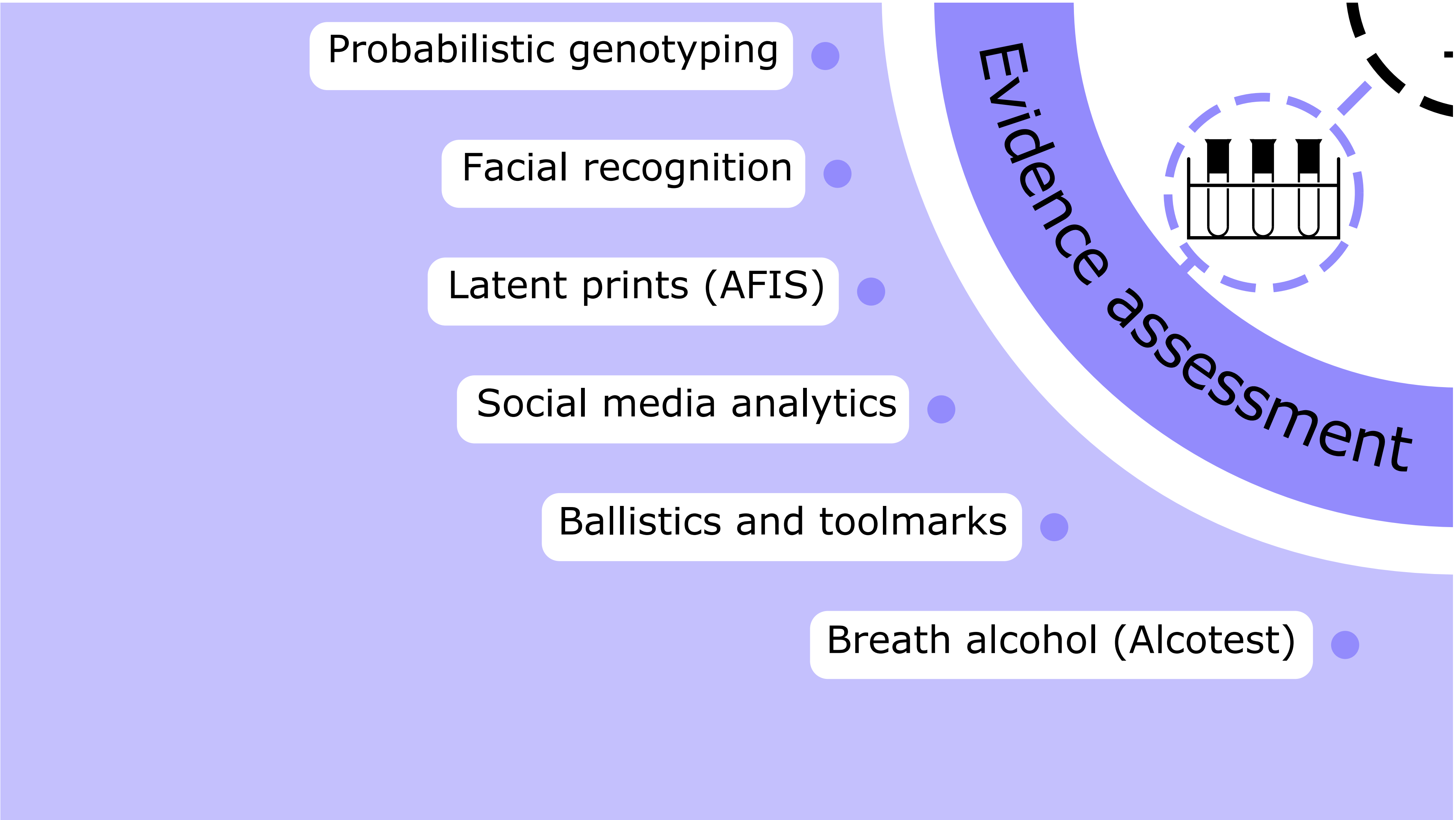
Predictive policing

- Caution needed during a stop
- Types of policing needed
- Areas to police
- People to stop
- Gang and affiliation databases
- People likely to become victims



Evidence gathering

- ShotSpotter
- Cell-site simulators (Stingray)
- Facial recognition
- Automated License Plate Readers (ALPRs)
- CP investigation (P2P, IP tracing)
- Network Investigation Techniques (NITs)
- Mobile device cracking (Cellebrite, Gray Key)



Evidence assessment

Probabilistic genotyping

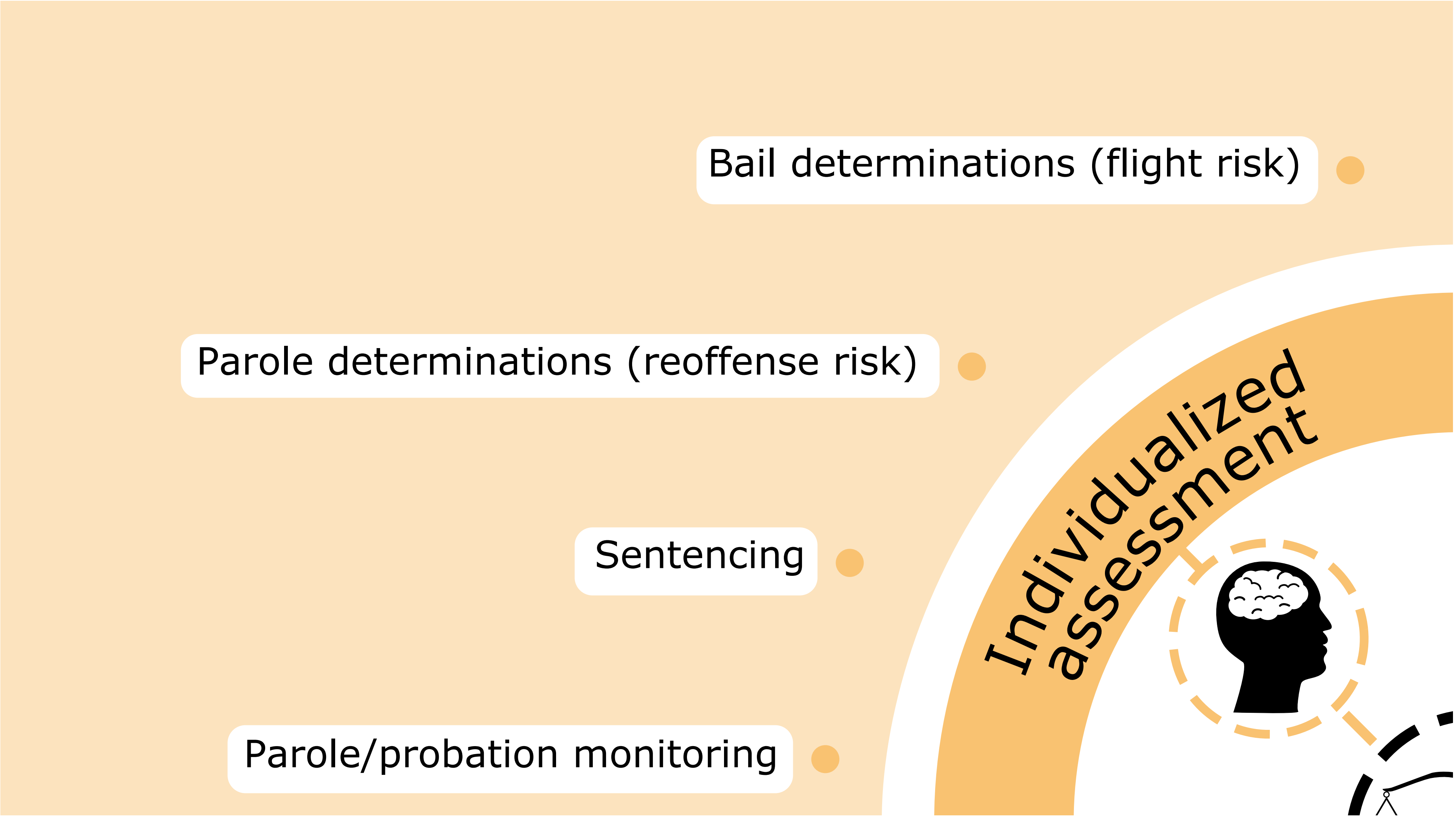
Facial recognition

Latent prints (AFIS)

Social media analytics

Ballistics and toolmarks

Breath alcohol (Alcotest)



Bail determinations (flight risk)

Parole determinations (reoffense risk)

Sentencing

Parole/probation monitoring

Individualized assessment

Software Bugs

- We have all used software with bugs.
  - Company did a lot of testing before release, generally works but still had problems for you
  - Perhaps a problem others had difficulty reproducing but that is clearly happening in your unique system

- Forensic software has no special immunity from the bugs and mistakes that plague software in other fields.

- Software and complex systems need an iterative process of debugging and improvement!
  - Anyone who has used technology knows that there are glitches and bugs and unintended consequences!
  - Anyone who builds technology knows how easy it is for there to be substantial bugs you did not find!

- How can we find bugs and fix problems if the answer is always “you can’t question” and “you are just complaining because you are guilty”?

Validation studies?

- What they are and what they are not
- Developers of the system do their own testing and publish results
- They have vested interest to demonstrate the system is working NOT to find bugs.
- What would our daily computing would look like with only this?
- What is the forcing function for debugging? Especially when interests of customers are different than interests of those being decided about?

Debugging?

- Huge advantages to independent, third-party testing aimed at finding bugs!
- If only those with interests in the success of software see the details, we have a huge problem and a recipe for injustice!
- Shifting to individual defense teams?
- People report errors and the system gets better, but what if all “bug reports” dismissed as just guilty people complaining?
- Doomed to run our criminal justice system on buggy software?

The 5 Stages of Debugging

At some point in each of our lives, we must face errors in our code. Debugging is a natural healing process to help us through these times. It is important to recognize these common stages and realize that debugging will eventually come to an end.

Denial

This stage is often characterized by such phrases as "What? That's impossible," or "I know this is right." A strong sign of denial is recompiling without changing any code, "just in case."

Bargaining/Self-Blame

Several programming errors are uncovered and the programmer feels stupid and guilty for having made them. Bargaining is common: "If I fix this, will you please compile?" Also, "I only have 14 errors to go!"

Anger

Cryptic error messages send the programmer into a rage. This stage is accompanied by an hours-long and profanity-filled diatribe about the limitations of the language directed at whomever will listen.

Depression

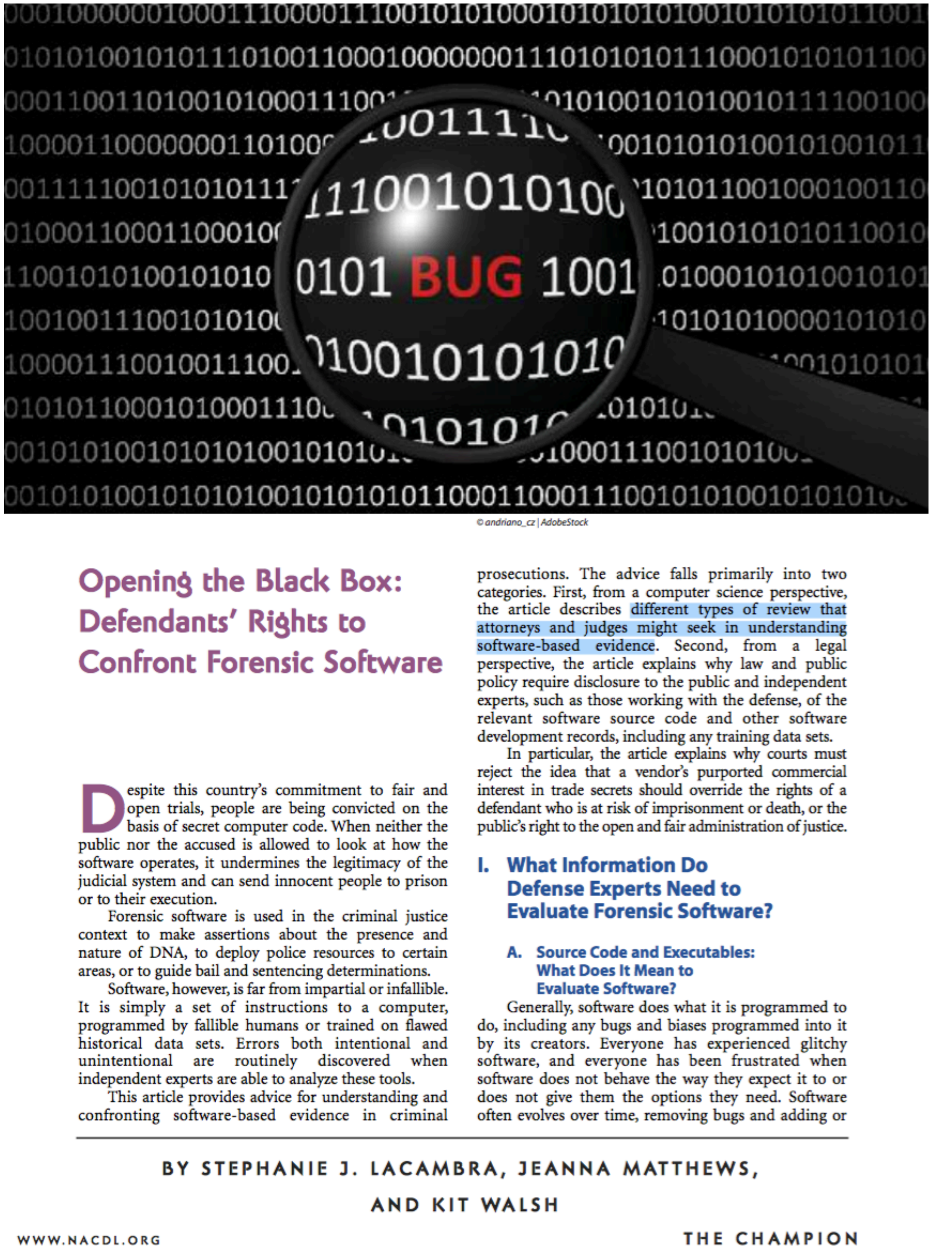
Following the outburst, the programmer becomes aware that hours have gone by unproductively and there is still no solution in sight. The programmer becomes listless. Posture often deteriorates.

Acceptance

The programmer finally accepts the situation, declares the bug a "feature", and goes to play some Quake.

- What types of review might attorneys and judges seek in understanding software-based/ computer-based evidence?

- Why law and public policy require disclosure of these materials to the public and independent experts?



© andriano_cz | AdobeStock

Opening the Black Box: Defendants' Rights to Confront Forensic Software

Despite this country's commitment to fair and open trials, people are being convicted on the basis of secret computer code. When neither the public nor the accused is allowed to look at how the software operates, it undermines the legitimacy of the judicial system and can send innocent people to prison or to their execution.

Forensic software is used in the criminal justice context to make assertions about the presence and nature of DNA, to deploy police resources to certain areas, or to guide bail and sentencing determinations.

Software, however, is far from impartial or infallible. It is simply a set of instructions to a computer, programmed by fallible humans or trained on flawed historical data sets. Errors both intentional and unintentional are routinely discovered when independent experts are able to analyze these tools.

This article provides advice for understanding and confronting software-based evidence in criminal prosecutions. The advice falls primarily into two categories. First, from a computer science perspective, the article describes different types of review that attorneys and judges might seek in understanding software-based evidence. Second, from a legal perspective, the article explains why law and public policy require disclosure to the public and independent experts, such as those working with the defense, of the relevant software source code and other software development records, including any training data sets.

In particular, the article explains why courts must reject the idea that a vendor's purported commercial interest in trade secrets should override the rights of a defendant who is at risk of imprisonment or death, or the public's right to the open and fair administration of justice.

I. What Information Do Defense Experts Need to Evaluate Forensic Software?

A. Source Code and Executables: What Does It Mean to Evaluate Software?

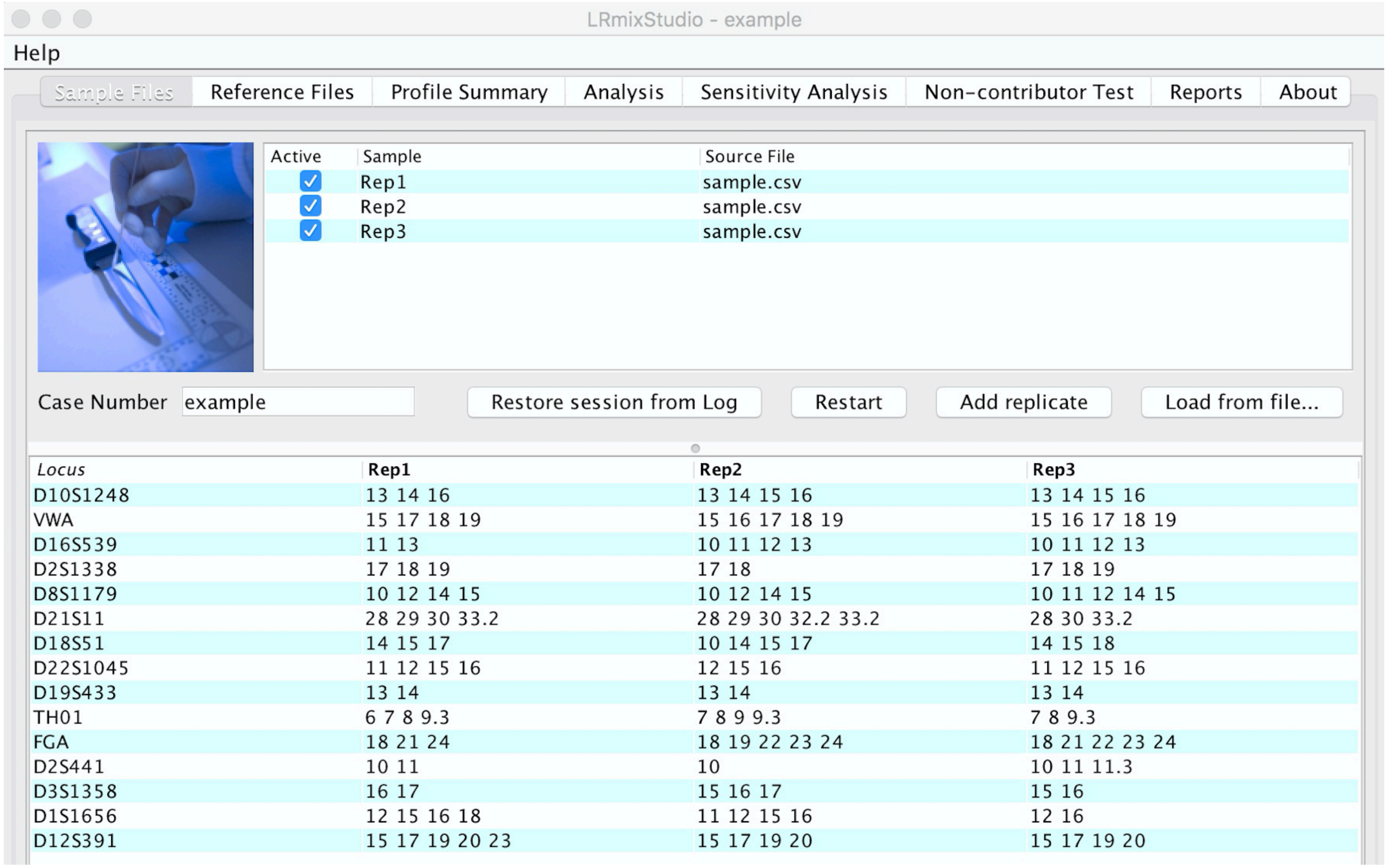
Generally, software does what it is programmed to do, including any bugs and biases programmed into it by its creators. Everyone has experienced glitchy software, and everyone has been frustrated when software does not behave the way they expect it to or does not give them the options they need. Software often evolves over time, removing bugs and adding or

BY STEPHANIE J. LACAMBRA, JEANNA MATTHEWS, AND KIT WALSH

What does it mean to evaluate software?

What Information Do Defense Experts Need to Evaluate Forensic Software?

Executables

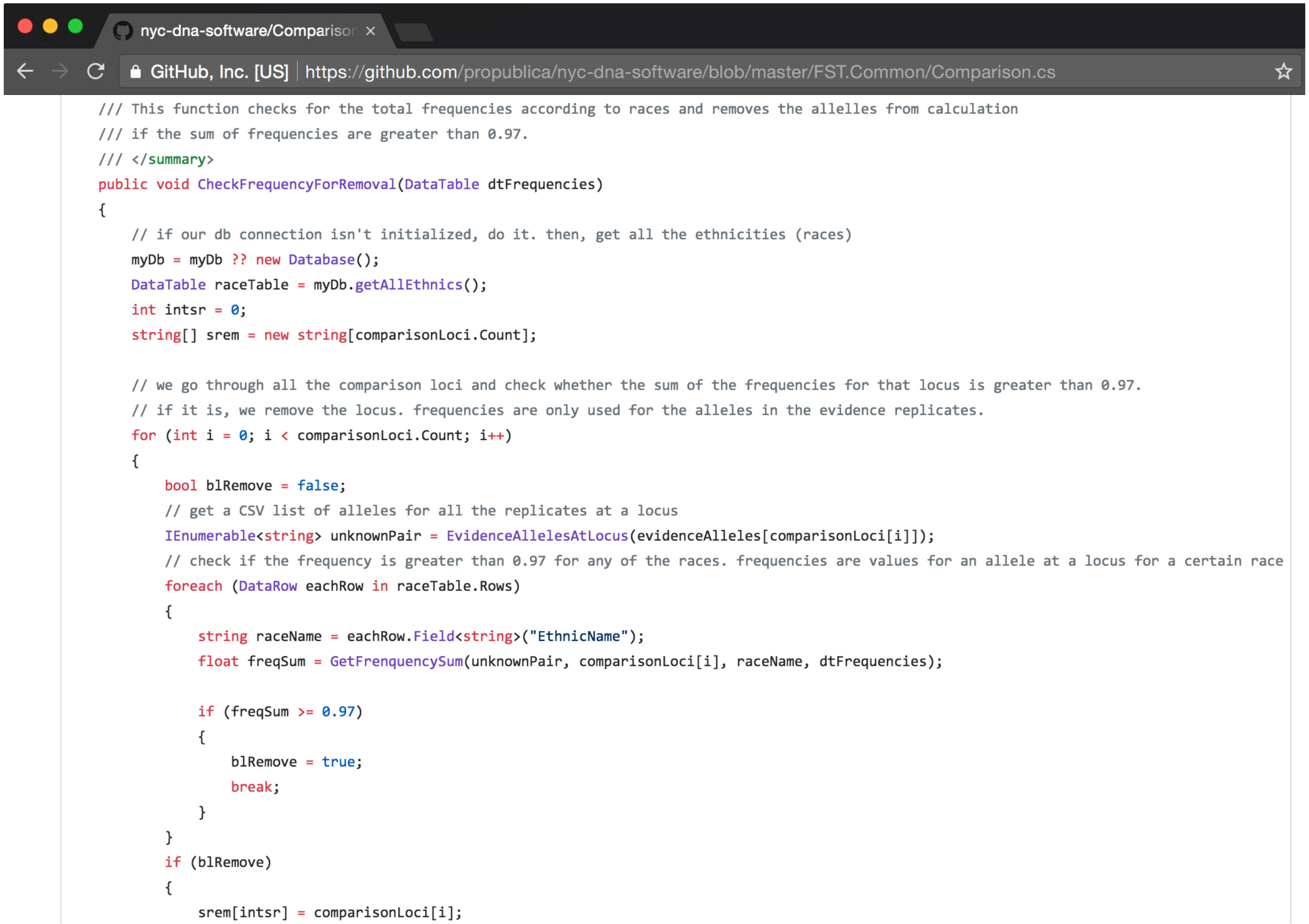


Active	Sample	Source File
☑	Rep1	sample.csv
☑	Rep2	sample.csv
☑	Rep3	sample.csv

Case Number example

Locus	**Rep1**	**Rep2**	**Rep3**
D10S1248	13 14 16	13 14 15 16	13 14 15 16
VWA	15 17 18 19	15 16 17 18 19	15 16 17 18 19
D16S539	11 13	10 11 12 13	10 11 12 13
D2S1338	17 18 19	17 18	17 18 19
D8S1179	10 12 14 15	10 12 14 15	10 11 12 14 15
D21S11	28 29 30 33.2	28 29 30 32.2 33.2	28 30 33.2
D18S51	14 15 17	10 14 15 17	14 15 18
D22S1045	11 12 15 16	12 15 16	11 12 15 16
D19S433	13 14	13 14	13 14
TH01	6 7 8 9.3	7 8 9 9.3	7 8 9.3
FGA	18 21 24	18 19 22 23 24	18 21 22 23 24
D2S441	10 11	10	10 11 11.3
D3S1358	16 17	15 16 17	15 16
D1S1656	12 15 16 18	11 12 15 16	12 16
D12S391	15 17 19 20 23	15 17 19 20	15 17 19 20

Source Code


nyc-dna-software/Comparison ×
GitHub, Inc. [US]
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];
```

Information from the Development Process

- Design documents
- Testing plans and results

Experience with Deployed Software

- Bug reports
- Change logs
- Revision Control

Machine Learning Systems

- Training sets

- Learn from, but don’t reproduce, the past

What are the arguments that law and public policy require disclosure of these materials to the public and independent experts?

Due Process

- Due process entitles the defense to review the prosecution's evidence
- Due process prohibits burden shifting to the defense.
- Requiring the defense to make a showing of materiality before granting access to the forensic software materials runs contrary to basic constitutional guarantees
- Commitment to open and public trials

Disclosure of Trade Secret Information

- Disclosure of trade secrets under protective orders is common in civil cases
  - Even when the parties are direct competitors with an interest in profiting from the proprietary information of the other
- Should be easier for a defendant trying to defend their life and liberty to access and assess forensic software, as compared to a party with a mere economic interest
- Prosecution Typically Cannot Establish That Disclosure Subject to a Protective Order Would Cause Harm

Stanford Law Review

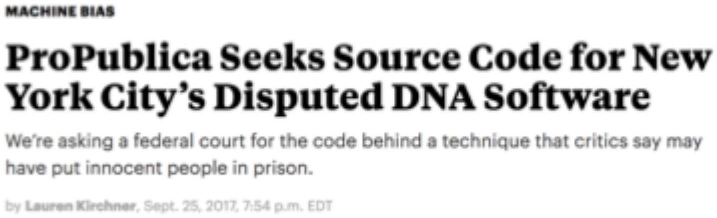
Volume 70 May 2018

ARTICLE

Life, Liberty, and Trade Secrets: Intellectual Property in the Criminal Justice System

Rebecca Wexler*

Brown Institute Magic Grant: Decoding Differences in Forensic DNA Software

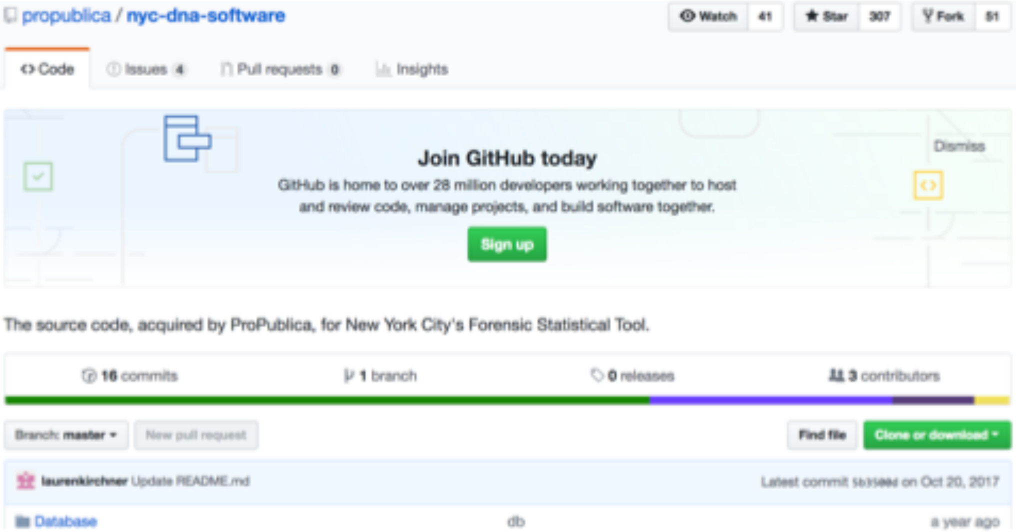


MACHINE BIAS

ProPublica Seeks Source Code for New York City's Disputed DNA Software

We're asking a federal court for the code behind a technique that critics say may have put innocent people in prison.

by Lauren Kirchner, Sept. 25, 2017, 7:54 p.m. EDT



Join GitHub today

GitHub is home to over 28 million developers working together to host and review code, manage projects, and build software together.

The source code, acquired by ProPublica, for New York City's Forensic Statistical Tool.

laurenkirchner Update README.md — Latest commit [illegible] on Oct 20, 2017

Database — db — a year ago

Methods

- Independent, third-party, adversarial testing and review
  - Automated testing harnesses
  - Common file formats and settings
  - Source code analysis
- Recommendations
  - Clear advice for judges, defense attorneys, journalists
  - Sample requirements for software systems, targeting the procurement phase

Procurement Phase Wishlist

- When public money used for criminal justice software, require! or at least give credit for:
    - Software artifacts: bug reports, internal testing plans and results, software requirements and specifications, risk assessments, design documents, etc.
        - Lack of software standards in traditionally non-computing fields (e.g. DNA)
    - Source code
    - No clauses preventing third party review or publishing of defects found
    - Access to executables for third party testing
    - Scriptable interfaces to facilitate automated testing
    - Bug bounties

- Reward/Fund/Incentivize non-profit third party entities to do independent testing and find problems!

Case of Forensic Statistical Tool (FST)

- Probabilistic genotyping software
- Developed by Office of the Chief Medical Examiner (OCME)
- NY State Commission on Forensic Science approves FST for use in casework

Problems/Changes

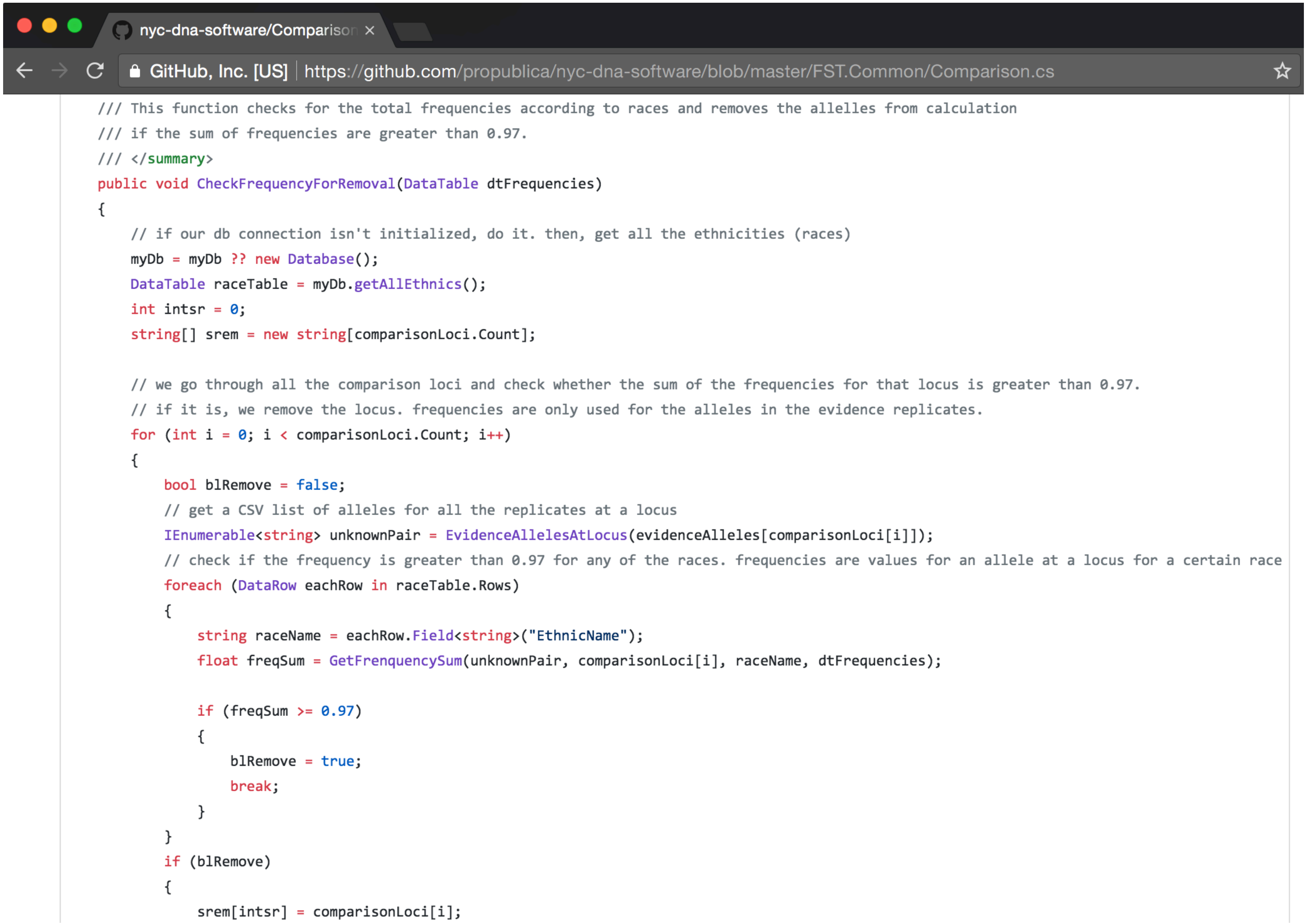
- April 2011 online, within weeks offline
- June 2011 back online after repairs/modifications

- For some samples reanalyzed post-modification, likelihood ratio “values were slightly modified as expected.”
  - -Quality Control Test of Forensic Statistical Tool (FST) Version 2.0, June 30, 2011

- “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.”
  - -Affidavit of Eugene Lien, OCME Assistant Director, July 17, 2017

What were the changes?

- In retrospect we know
- ~70 line function, checkFrequencyForRemoval

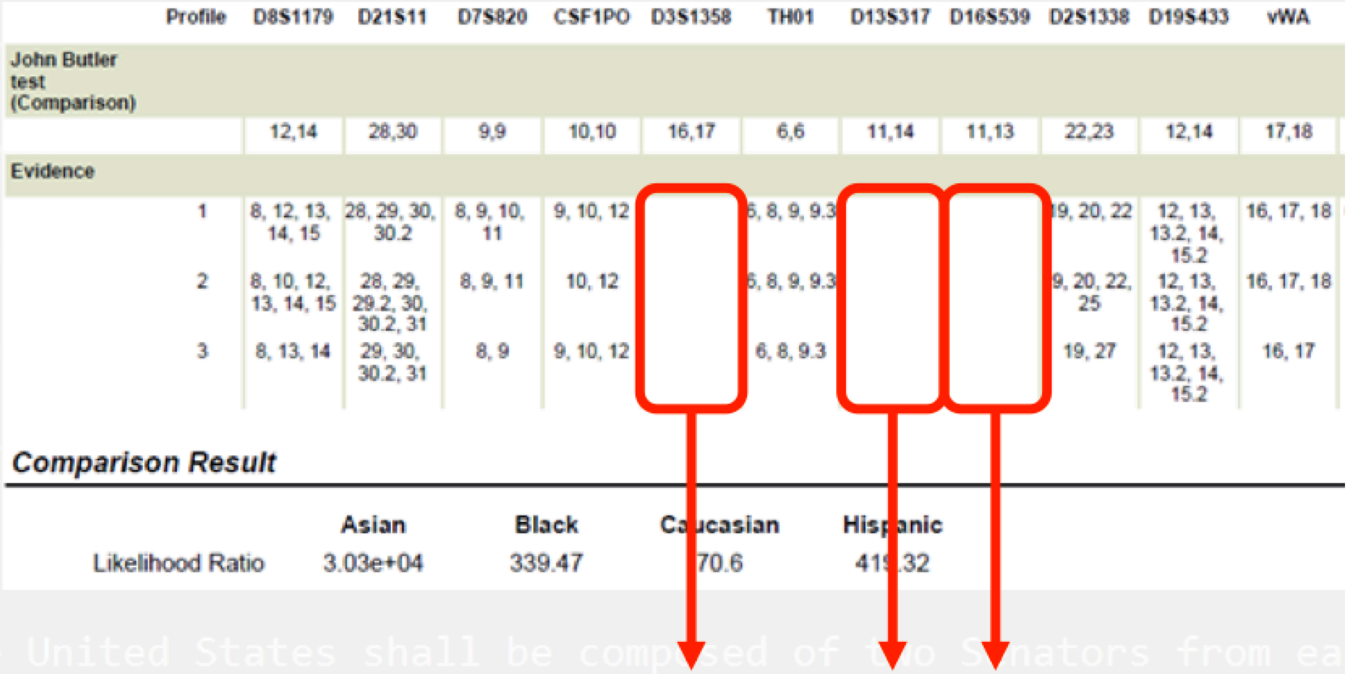

nyc-dna-software/Compariso
GitHub, Inc. [US]
https://github.com/propublica/nyc-dna-software/blob/master/FST.Common/Comparison.cs

```
/// 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];
```

Dropping some data that supports the defense hypothesis without notice!

LR > 1 supports inclusion as a contributor

LR < 1 supports exclusion as a contributor



Profile	D8S1179	D21S11	D7S820	CSF1PO	D3S1358	TH01	D13S317	D16S539	D2S1338	D19S433	vWA
John Butler test (Comparison)											
	12,14	28,30	9,9	10,10	16,17	6,6	11,14	11,13	22,23	12,14	17,18
Evidence											
1	8, 12, 13, 14, 15	28, 29, 30, 30.2	8, 9, 10, 11	9, 10, 12		6, 8, 9, 9.3			19, 20, 22	12, 13, 13.2, 14, 15.2	16, 17, 18
2	8, 10, 12, 13, 14, 15	28, 29, 29.2, 30, 30.2, 31	8, 9, 11	10, 12		6, 8, 9, 9.3			9, 20, 22, 25	12, 13, 13.2, 14, 15.2	16, 17, 18
3	8, 13, 14	29, 30, 30.2, 31	8, 9	9, 10, 12		6, 8, 9.3			19, 27	12, 13, 13.2, 14, 15.2	16, 17

Comparison Result

	Asian	Black	Caucasian	Hispanic
Likelihood Ratio	3.03e+04	339.47	70.6	41[illegible]32

Removing data at 3 loci that is...

Exclusionary 0.53

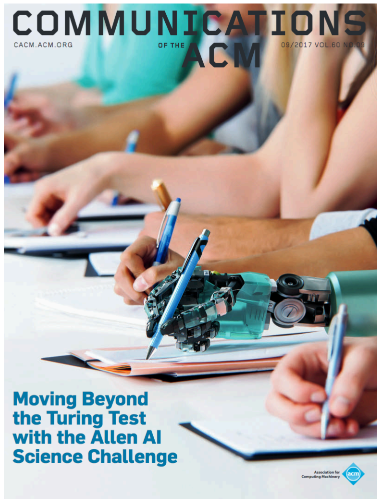
3.1 1.3 Inclusionary

Where is debugging?

- What hope would an individual have of seeing whether a change like this impacted them?
- How likely to find at all without source code access?
- Work of debugging falling to individual defense teams
- You’re just complaining because you’re guilty?

Algorithmic Transparency and Accountability Efforts

- US-ACM/EUACM Statement on Algorithmic Transparency and Accountability
  - 7 principles
  - Awareness, access and redress, accountability, explanation, data provenance, auditability, validation and testing
- CACM article “Toward Algorithmic Transparency and Accountability”


COMMUNICATIONS OF THE ACM
CACM.ACM.ORG
09/2017 VOL.60 NO.09
Moving Beyond the Turing Test with the Allen AI Science Challenge

https://www.acm.org/binaries/content/assets/public-policy/2017_usacm_statement_algorithms.pdf

AI Ethics and Anti-Patterns

- Upcoming article in AI Magazine and talk at 2019 AI For Good
- 10 Common Anti-Patterns
  - 1. Learn from the Past Without Remembering the Context
  - 2. Learning from Humans Without Remembering Human Bias and the Possibility of Malicious Training
  - 3. Using Data You Have Rather than the Data You Need
  - 4. Transparency without Accountability
  - …..

Human vs. computer-based decision making?

- Human decision making flawed as well *but* at least we recognize it

- Myth of logical, unbiased, nearly infallible computer decision making

- Possibility of more reproducible, accurate, fair decisions *but* to achieve that we will need investments in debugging, skepticism and explanation
  - Bail vs. automated risk assessments

- Important to preserve our ability to question in general and in a specific case

Decision-Making Landscape

- Big decisions about the lives of individuals are being made in a partnership between human decision-makers and computer systems.
- Fundamentally changing the landscape of our societal decision-making processes
- In the process of automation, moving to new platforms, are we undermining principles on which societies have been built?
- What are we doing to secure and debug these complex socio-technical systems? To build in incentives for debugging and improvement?