Universidade Estadual de Campinas
Instituto de Computação

Tainá Turella Caetano dos Santos

An analysis of Probabilistic Genotyping Systems as high-integrity software

Uma análise de Sistemas de Probabilidade Genética como software de alta integridade

CAMPINAS
2023

Tainá Turella Caetano dos Santos

An analysis of Probabilistic Genotyping Systems as high-integrity software

Uma análise de Sistemas de Probabilidade Genética como software de alta integridade

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestra em Ciência da Computação.

Dissertation presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

Supervisor/Orientadora: Profa. Dra. Islene Calciolari Garcia

Este exemplar corresponde à versão final da Dissertação defendida por Tainá Turella Caetano dos Santos e orientada pela Profa. Dra. Islene Calciolari Garcia.

CAMPINAS
2023

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Silvania Renata de Jesus Ribeiro - CRB 8/6592

Sa59a

Santos, Tainá Turella Caetano dos, 1997-
An analysis of probabilistic genotyping systems as high-integrity software / Tainá Turella Caetano dos Santos. – Campinas, SP : [s.n.], 2023.

Orientador: Islene Calciolari Garcia.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Software - Verificação. 2. Programas de computador - Verificação. 3. Software - Validação. 4. Programas de computador - Validação. I. Garcia, Islene Calciolari, 1971-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações Complementares

Título em outro idioma: Uma análise de sistemas de probabilidade genética como software de alta integridade
Palavras-chave em inglês:
Computer software - Verification
Computer programs - Verification
Computer software - Validation
Computer programs - Validation
Área de concentração: Ciência da Computação
Titulação: Mestra em Ciência da Computação
Banca examinadora:
Islene Calciolari Garcia [Orientador]
Sandra Eliza Fontes de Avila
Renata Wassermann
Data de defesa: 21-12-2023
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0003-3900-0131
- Currículo Lattes do autor: http://lattes.cnpq.br/7006574774580851

Universidade Estadual de Campinas
Instituto de Computação

Tainá Turella Caetano dos Santos

An analysis of Probabilistic Genotyping Systems as high-integrity software

Uma análise de Sistemas de Probabilidade Genética como software de alta integridade

Banca Examinadora:

- Profa. Dra. Islene Calciolari Garcia
  Instituto de Computação, Universidade Estadual de Campinas - UNICAMP
- Profa. Dra. Sandra Eliza Fontes de Avila
  Instituto de Computação, Universidade Estadual de Campinas - UNICAMP
- Profa. Dra. Renata Wasserman
  Instituto de Matemática e Estatística, Universidade de São Paulo - USP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 21 de dezembro de 2023

Acknowledgements

This dissertation was feasible due to my spectacular advisor and her faith that this discussion matters even in more technical fields. I appreciate all the guidance you've provided and all the patience you have shown.

A special thanks to Professor Rediet Abebe, Ph.D. Nathan Adams and all the BEAAMO fellows. The summer we spent together was the starting point of this dissertation, and I'd never come this far without your support.

I never thought I'd be where I'm now, but I just got this far because of the people I had by my side during this journey. Pedro, Tamy, Thamiris, Edy, and so many others, thank you for putting up with me talking about this project all the time! I know it wasn't easy or pleasant, but it is over now.

This dissertation is a homage to Stella Turella and Cláudia Gomes, two women in my life that showed me that I could achieve anything I wanted, as long as I set my mind to it. You were, are, and will always be my role models, and I hope you are proud of everything I've accomplished so far.

Resumo

É de conhecimento geral que a tecnologia possui um grande papel na sociedade atual e que esta afeta diversas áreas, desde sistemas de recomendação disponíveis em redes sociais até software usado como evidência em casos criminais. Artefatos tecnológicos podem parecer inofensivos, mas quando cavamos mais a fundo nos mecanismos sociais que os alimentam e que são alimentados por eles, é fácil perceber que estes podem não ser tão justos ou até mesmo tão desconectados das disparidades sociais perpetradas por nós humanos.

Dependendo de quais esferas sociais tais algoritmos e sistemas atuam, os efeitos controversos de sua existência não supervisionada e inquestionada não são tão alarmantes, talvez porque o nível de integridade destas não seja alto o suficiente. É aceitável que, enquanto vemos nossos feeds em nossas redes sociais favoritas, não encontremos uma grande diversidade de posts. É menos aceitável quando um banco recusa um empréstimo a alguém porque um algoritmo de score de crédito, por motivos não bem definidos, sinalizou algum problema com o requerente. Mas seria inaceitável se um software utilizado como evidência criminal embasasse a prisão de uma pessoa inocente.

Ao usar software como evidência em tribunal, especificamente Software de Probabilidade Genética (PG), é preciso entender que uma saída deste software pode ser responsável pelo encarceramento de uma pessoa por longos períodos devido à gravidade dos crimes onde esta ferramenta se faz necessária. Portanto é imperativo compreender os mecanismos por trás deste software e garantir que ele passou por um processo minucioso de validação antes de sua aplicação em casos reais.

Existem muitas ferramentas de PG disponíveis para análise, mas nosso foco será limitado às ferramentas open source; LRmix/Forensim (utilizada no Brasil), EuroForMix, Kongoh, LikeLTD e LabRetriever. Nesta dissertação, nós comparamos o processo de validação seguido pelas pessoas desenvolvedoras das ferramentas citadas com *guidelines* propostas pela comunidade forense e verificamos se os padrões propostos pela IEEE são seguidos. Também são recuperadas métricas referentes à cobertura de código pelos testes e rápidas análises estáticas são realizadas no código.

Nosso objetivo é caracterizar quais testes deveriam ser tidos como obrigatórios em sistemas de alta integridade ao mesmo tempo em que analisamos se ambos os sistemas selecionados performam tais testes corretamente, mesmo em cenários complexos baseados em casos reais. Não temos a pretensão de desencorajar o uso deste tipo de software em circuntâncias onde as amostras são menos complexas, até porque estas análises se tornaram uma importante peça de evidencia em casos de abuso sexual, por exemplo. Aqui apenas trazemos percepções sobre possíveis brechas no processo de validação que poderiam ser danosas em casos complexos.

Abstract

As is widely known, technology plays a massive role in the current society, and it has effects in multiple areas, from recommendation systems available on social network feeds to evidentiary software. Technological artifacts might seem harmless, but when we dig deeper into the social mechanism that feeds and is fed by them, it is clear that they are neither fair or alienated from the societal disparities perpetrated by humans.

Depending on the social spheres in that algorithms, and systems act, the controversial effects of their unsupervised and unquestionable existence are not that alarming because their integrity level may not be high enough. It is acceptable if, while scrolling down on our favorite social network, we do not see diversified posts; it is less defensible when a bank declines someone's loan because a credit score algorithm flags them with no explainable reason; but it would be intolerable if an evidentiary software sent an innocent person to jail.

When using evidentiary software in court, specifically Probabilistic Genotyping Systems (PGS), it is necessary to understand that an outcome from this software can be responsible for incarcerating someone for long periods due to the gravity of the crimes where this tool is needed. Therefore it is imperative to understand the mechanisms behind this software and ensure that it has passed a thorough validation process before its application in casework.

There are many PGS available for evidence analysis, but our focus will be limited to the open source tools: LRmix/Forensim (used in Brazil), EuroForMix, Kongoh, LikeLTD e LabRetriever. In this dissertation, we compare the validation process performed by PGS developers with the guidelines proposed by the forensic community, and verify if they follow the standards proposed by IEEE. As the source code of all tools is available, we also retrieve a few metrics regarding the code coverage of tests and perform a brief static analysis of the software.

Our goal is to characterize which tests should be deemed as mandatory for high-integrity software at the same time that we analyze if the five selected tools perform them properly, even in complex casework based scenarios. We do not discourage the use of this type of software in simpler DNA mixtures, because they became a important piece of evidence in cases of sexual abuse, for example. We just provide insights about possible breaches in the validation process that could be damaging in complex cases.

List of Figures

List of Tables

Listings

Acronyms

ALPR Automated License Plate Reader. 19

AoS Atlas of Surveillance. 8, 18, 22

AUTORAD Autoradiogram. 27

BWC Body-worn camera. 19, 20

CPI Combined Probability of Inclusion. 30

CSS Cell-site Simulator. 21

DQ-alpha Polymarker tests. 28

EFF Electronic Frontier Foundation. 18–20

ENFSI European Network of Forensic Science Institutes. 25, 27, 31, 33, 36, 43, 44

EPG Electropherograms. 28, 29, 38, 39, 41

FBI Federal Bureau of Investigation. 27

FR Face Recognition. 20

GDPL General Data Protection Law. 15

IEEE Institute of Electrical and Electronics Engineers. 6, 7, 9, 26, 31–34, 36, 37, 46, 47, 49, 51, 61, 64, 71, 72

INMETRO National Institute of Metrology, Quality, and Technology. 26

LR Likelihood Ratios. 30, 31

MAC Maximum Allele Count. 29

NDIS National DNA Index System. 27

NIST The National Institute of Standards and Technology. 25, 26, 31, 33, 36, 38, 39, 43, 44, 61

NOC Number of Contributors. 15, 29, 40, 42, 43

OSAC Organization of Scientific Area Committees. 30

PCAST The U.S.A. President's Council of Advisors on Science and Technology. 25

PCR polymerase chain reaction. 26, 28, 29, 38, 43

PG Probabilistic Genotyping. 6, 15, 23, 33–36, 40, 43, 51, 61, 62, 65, 66, 72, 74, 75

PGS Probabilistic Genotyping Systems. 7, 8, 15, 24, 26, 29–36, 38, 39, 43, 44, 51, 63–65, 71–75

POI Person of Interest. 15, 20, 28, 30

QAS Quality Assurance Standards. 27

RFLP Restriction fragment length polymorphism. 26–28

RFP Request for Proposal. 46

RFU relative fluorescence units. 38

RMP Random Match Probability. 29

SE Software Engineering. 16, 31, 32, 37, 71–73

STR Short Tandem Repeat. 27, 28, 33, 38, 39, 41

SWGDAM Scientific Working Group on DNA Analysis Methods. 25–27, 30, 31, 33, 36, 39, 42–44

TWGDAM Technical Working Group on DNA Analysis Methods. 26

V&V Verification and Validation. 15–17, 19, 23, 24, 34, 46–51, 62–65

VVP Verification and Validation Plan. 46, 47, 62, 63

WoE Weight of Evidence. 23, 29, 30

Contents

Chapter 1

Introduction

A society that breathes technology forces its participants to understand how those technologies might impact their lives in multiple areas [51]. From evident fields like recommendations in social networks and door dash deliveries to less obvious ones like hiring [54], providing loans, and law enforcement, technologies control some of these actions and decisions. Due to specific algorithms, different people might have different outcomes on Google Search Engine [53] or in a different scenario could be incarcerated due to some result from a Probabilistic Genotyping (PG) system [1, 74].

The first records of using DNA evidence in court date back to the 80s in the United Kingdom [55] and United States[71]. What, at first, was a binary output stating the inclusion or the exclusion of a Person of Interest (POI), nowadays is the calculation of probabilities that support the prosecution or the defense hypothesis. The technology behind the collection and analysis of such evidence evolved a lot [6], and now it is possible to retrieve DNA from low-quality and low-quantity samples.

The use of Probabilistic Genotyping Systems (PGS) [18] is frequent when DNA evidence comes from mixtures. These mixtures can vary in complexity due to their Number of Contributors (NOC), degradation levels, inhibition levels, drop-out probabilities, drop-in probabilities, stutter, allele overlap, genetic diversity, and other factors. Therefore, validations of PG tools must ensure that these levels of casework complexity are covered, and this assurance comes from having representative data and an extensive validation process with complete documentation.

As PGS is an example of a technical and multidisciplinary software, it is necessary to understand the biology concepts that lead to the likelihood probabilities used as evidence in court and possible miscalculation fonts due to biological effects. We also need to remember that, as any software, this technology is also prone to error, but this can be mitigated by performing a thorough Verification and Validation (V&V) process.

The use of technology as a tool on public spheres is more and more debated, and recently some laws were approved to regulate and provide basic rights to citizens. Analysing the Brazilian scenario, the 5th art. chap. X of Federal Constitution [21] and the text of the General Data Protection Law (GDPL, in Portuguese LGPD) [10] ensure the citizen the right to privacy, and something similar can be found in most of the countries around the globe. This being said it is crucial to also certify if those rights are not being disregarded when new types of technology are inserted into the public sphere.

To fully comprehend this dissertation, it is necessary to define a few core concepts and specify some core entities that regulate and audit the DNA Analysis and Software Engineering (SE) fields. We begin with the SE concepts.

1.1 Software Verification and Validation

One of the focal points of this dissertation is the Verification and Validation (V&V) Process. In the Software Engineering context, verification processes ensure that the software complies with functional and non-functional requirements and perform tests to find mistakes in the code. Validation processes ensure that the system under development is in accordance with stakeholders' expectations.

V&V processes happen throughout the entire development and planning process [72]. When dealing with Agile methodology [1], it is positive to perform verification and validation activities from the requirements extraction until the final delivery of the system. In more classical approaches, in general, there is one quick verification of base requirements, and at the end, in-depth V&V tasks are performed.

The V&V process is only testing tasks (e.g., unity tests, component tests, integration tests), but this process evaluates much more than just that. Activities related to coding quality (e.g., clean code practices, code review, code coverage analysis), quality assurance (e.g., regression tests, feature tests), design review (e.g., accessibility, user experience), and management (e.g., documentation, requirements review) are all part of the V&V process and have equal weight when defining the quality of a given system.

Several studies and standards refer to V&V applicability and good practices [72, 42, 20, 28], and this happens because it is a common sense that products, including technological artifacts, must be developed and delivered following the needs and wills of its final consumers/users. It doesn't matter if the software is proprietary or open source; developers must always value good deliverance standards, regardless of the final purpose of the system. When the system under evaluation is deemed as a high-integrity, the verification and validation process becomes more extensive, because in these cases a software/hardware malfunctioning can cause serious social or even physical damage to the stakeholders.

One factor that influences Verification and Validation processes is the assessment of the integrity level of the software [42]. The integrity level can be summarized as a metric that informs how critical would it be if this software fails in some capacity. For example, aircraft's systems are categorized as high-integrity software [38] but the algorithm used by a bakery to account for their daily sales would be deemed low-integrity software. In this dissertation we debate a lot on how rigorous the V&V process should be when dealing with high-integrity systems.

[1] Agile methodology is an iterative and interactive project management framework that values constant feedback and development cycles.

1.2 Proprietary vs. open source code in public spheres

Before we dive into specific content on surveillance and evidentiary software, we'd like to provide context on proprietary and open source code. This is important because, given the context where surveillance and evidentiary software exist, it is reasonable to discuss where the intellectual properties and databases lay.

Proprietary code definition is code or software that is owned by the individual, company, or group who published this system, meaning that a given entity is responsible for its maintenance and publication and that its software can be treated as an opaque box to protect intellectual property rights. When we think about low-impact and low-integrity software, this might not threaten civilians' fundamental rights. When the software is present in the public sphere, this could mean that data from thousands of civilians is being collected, processed, stored, and sold , sometimes without their informed consent [39]. Alone, this is a privacy violation, but in cases where this data is evidence in court, we must analyze other consequences. Open source code, on the other hand, refers to software available for anyone to access, change, and distribute (under specific licenses) the code. In this dissertation context, it would increase transparency in court, allowing a more efficient and accurate defense.

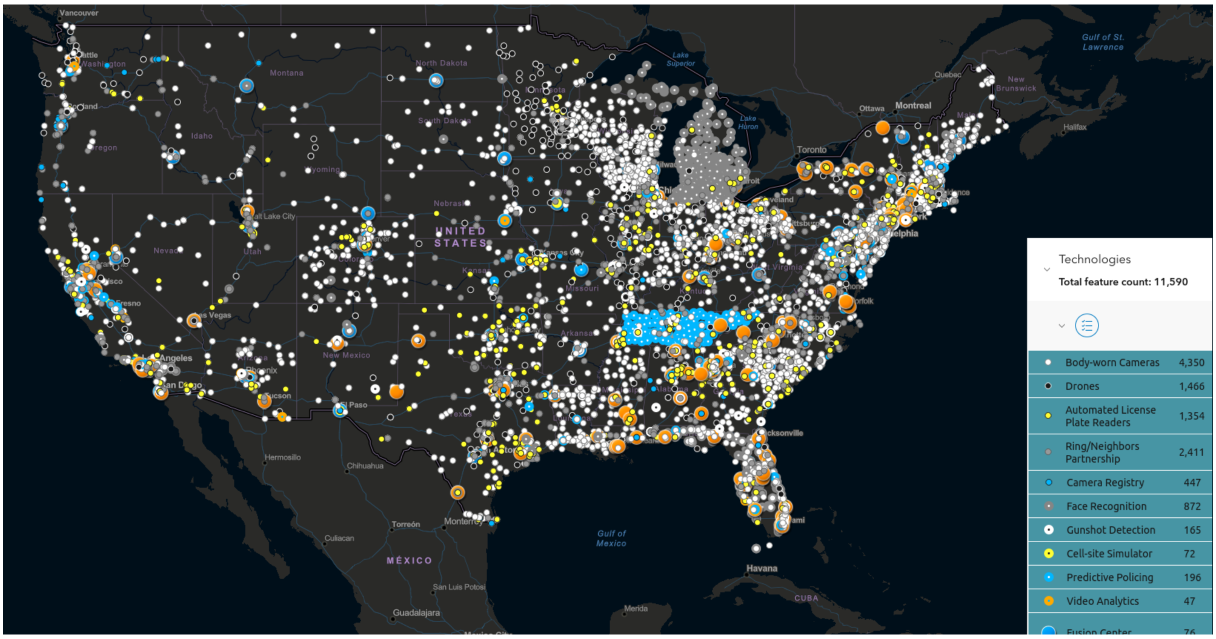
As advocated by [51, 3], the defendant and their legal advisors should be able to contest the evidence against them, but when proprietary software is involved in the mix, it becomes harder to assure this right. Companies behind evidentiary and surveillance software want to protect their product and the intelligence behind it, which is their right. Unfortunately, this opposes the rights of the accused party. Refuting [14], we say that open-source code would enhance the power of code review in a more active community and, most importantly, provide the minimum tools for defense attorneys who face the result from this type of software in court.

Another positive aspect of using open source code in public spheres is the opening to apply Linux's Law, which informally states: "With enough eyeballs, all bugs are shallow" [59]. But what does that mean in our context? In this dissertation, we will reinforce the need for a careful V&V process in high-integrity software, and a part of this process is also the existence of code review and the interaction with multiple stakeholders to make valid improvements in the software, as defined in Software Construction V&V activity [42], this aligns a lot with the Linux's Law because we reinforce the idea of a through verification and validation process. In the following sections, we describe surveillance and evidentiary software while highlighting the threats they can pose to civil society, even as open-source software.

Something that we also should value in open-source software is its transparency. There is no doubt about how this software works because all the computational job is in plain sight. Anyone with minimal knowledge, a computer, and time can unravel why the system prompts a specific result. Transparency is an excellent feature that should be mandatory in software applied to public spheres, especially if this software is used to generate evidence or monitor given groups, individuals, or regions.

1.3 Surveillance and evidentiary technologies

Surveillance technology is a category of technological artifacts that monitor a given environment and the activities of individuals or groups, claiming to provide security for the remaining members of society. Electronic Frontier Foundation (EFF) promotes a project called Atlas of Surveillance (AoS) that lists technologies as part of a surveillance network in the United States obtained with Open-Source Intelligence [2] and Crowdsourcing [3] methodologies. Figure [1.1] contains the updated map of surveillance technologies provided by EFF [4]. The map only shows information about the usage of surveillance technology in the US context, but as a technological hub, what the US law enforcement organizations currently use is generally exported to other countries.



Technologies

Total feature count: 11,590

○	Body-worn Cameras	4,350
●	Drones	1,466
●	Automated License Plate Readers	1,354
●	Ring/Neighbors Partnership	2,411
○	Camera Registry	447
●	Face Recognition	872
●	Gunshot Detection	165
●	Cell-site Simulator	72
●	Predictive Policing	196
●	Video Analytics	47
●	Fusion Center	76

Figure 1.1: United States map retrieved from AoS website plotting surveillance technology registered uses.

The EFF is a leading nonprofit founded in 1990 that defends digital privacy, free speech, and innovation. EFF's mission is to ensure that technology supports freedom, justice, and innovation for everyone, everywhere. The AoS project is a database of 10,600 data points in the US collected by hundreds of researchers, containing information about ten different categories of technologies and where they are in use; and it is available for download at the project's homepage. Of course, some of the surveillance technology developed have a positive impact — For example, body-worn cameras are known to reduce the levels of abusive use of force by law enforcement agents, but some are not ready to

[2] EFF defines it as: "a term used to describe gathering information that already exists online—from news stories, social media posts, press releases, or documents buried in government websites, often turned up through using advanced search engine techniques."

[3] EFF defines it as: "A research technique that involves recruiting a large number of people, such as students, members of your supporter base or readership, to collaborate on a single project."

[4] Map is updated constantly and made available at `https://atlasofsurveillance.org/atlas`. Last access: November 14th, 2023.

be taken that seriously as a measure of protection due to its known malfunctioning cases. In the following subsections, we present examples of surveillance software, an overview of its impacts on people's lives, and the issues that hide behind the false sense of security they provide [5].

We believe it is important to give a context on a few categories of surveillance technologies because most of them can become evidence in court. In this dissertation, we will not scrutinize the verification and validation process for all the technologies listed here, but this text could also be adapted to verify if it is possible to find the same V&V issues in other technologies.

Real-time crisis centers and fusion centers will not be described in this paper as they are not a technology that retrieves data, they are only responsible for the analysis and the decision-making on how to act given the data collected by surveillance technologies. The video analytics/computer vision category is also not described since it comprises any algorithm or model that runs over footage collected by some of the other technologies mentioned below.

Automated License Plate Reader (ALPR): Automated License Plate Readers are high-speed and computer-controlled camera systems that can capture car license plates, metadata regarding the photo (e.g., location, timestamp), and sometimes images of the passengers in the vehicle. The collected information might not seem much, but it allows law enforcement entities to determine any vehicle's whereabouts, possible routes, and travel patterns if the metadata is correctly combined and processed.

When the police or other agencies merge this information with vehicle ownership data, it is possible to have all the details about citizens' locations and routine habits, presenting a breach of privacy rights. The use of stationary or mobile ALPR could facilitate the targeting of specific groups and even the monitoring of people attending given facilities (e.g., religious buildings, abortion clinics, rehabilitation clinics).

Knowing that, anyone with an ALPR attached to their car can collect this type of data, not only law enforcement generates these databases, and thus private players can also assemble datasets that law enforcement or anyone who pays enough can purchase. Exaggerating this scenario, an ALPR is a weapon in cases where the local government and other entities violate individual rights, since the citizen has no idea about those records and who is recording them.

Naturally, there are also positive outcomes of ALPR usage. It could be part of relevant evidence to prove that a given person was at a crime scene, and with the use of hotlists [6] it can be helpful in missing person situations, kidnapping cases, sexual assault, and as proof of innocence as well.

Body-worn camera (BWC): Body-worn cameras are one of the most known categories of surveillance technologies. This technology has a dual purpose because, at the

[5] All the information available in the next subsections is based on studies and descriptions made by EFF and is available at `https://atlasofsurveillance.org/glossary`.

[6] A downloaded list of plates that is checked during police patrols by the ALPR system.

same time, it is a tool for law enforcement, and it also serves as a form of protection against police misconduct.

Starting with its public accountability role, when law enforcement officers correctly use body-worn cameras, it creates footage of all of the actions taken by that officer, which facilitates investigations in cases of abuse and violence against citizens. In areas where most of the population is marginalized, the use of BWCs can reduce the levels of police truculency, maybe helping to reduce the ratio of unfounded and unnecessary use of police deadly force. We highlight that only the proper use of BWCs can help decrease police violence because ill-intentioned officers can also manipulate what is recorded (but this manipulation could result in disciplinary actions against the officer).

Nonetheless, this technology is also a powerful form of surveillance because it can record a citizen's interactions and location, constituting a breach of privacy and fundamental rights. In Brazil, the US, and other countries, one of the fundamental rights of its citizens is the principle that no one is obliged to produce evidence against themselves. With the use of BWC, this becomes a blurred line because, without knowing, someone could generate this type of evidence.

Due to its popularity, multiple brands sell body-worn cameras, and some players also provide footage storage options. Initially, this is a better situation than a monopoly of the market, but it should be alarming that private companies can also store, therefore have access to, sensitive public data.

Face Recognition (FR): Face Recognition is a technology that allows the identification of an individual's facial features (e.g. eye distance, chin shape, nose shape) in photos and videos. Some applications can also run on mobile phones, giving the police a tool to identify wanted personas during patrols.

Unfortunately, this type of technology is prone to errors, and they are more common in the faces of non-white people. Studies, such as the one carried out by Joy Buolamwini and Deborah Raji [58], show that this type of technology is not prepared to deal with the diversity in our society. Therefore, it is scary that this technology is adopted by law enforcement blindly. But for a moment, we will pretend that this technology has good accuracy. In this scenario, this tool could help target people of interest to the government and help create an environment where anyone could be under surveillance all the time, especially if they are combined with different surveillance technologies and methods.

This type of technology is already common in airports and places where counter-terrorism methods are reasonably applied. Facial recognition systems are also integrated with security/surveillance cameras on public circuits, allowing the authorities to identify POI in concerts, football matches, and public protests. This technology threatens fundamental rights; as an example, protesters can easily be identified if their photos are available in any law enforcement database.

Surveillance Cameras and Camera Registry: Surveillance cameras are everywhere and could be considered the most popular method of surveillance listed by EFF. From stores and apartment condominiums to law enforcement agencies, this technology is used to create video footage of an environment and alone would be considered simply an evidentiary tool. However, this technology is generally associated with one or more types of surveillance methods becoming a threat to individual privacy. Something that also happens is a collaboration between businesses and law enforcement agencies in a form of a camera registry, meaning that a given store can provide its footage to police departments.

When associated with real-time crime centers (physical space where law enforcement agents analyse the information gathered by surveillance cameras) or just a few TVs assisted by a security guard in a store, cameras become a tool that can lead to early action on criminal activity and might give excuses for biased actions against marginalized groups. When associated with facial recognition systems, it becomes a tool to identify criminals or to persecute individuals. Therefore, as mundane as this technology might seem, the effects it perpetrates in society are massive.

Cell-site Simulator (CSS): Cell-site simulators are a type of equipment that poses as legitimate cellphone towers, tricking close-by devices into connecting to them instead of real ones. Passive CSSs do not transmit signals, they only extract and decode cellular transmission waves, they pose as listeners forcing devices to connect while also redirecting the signal to real towers. Active simulators are the ones that imitate cellphone towers; therefore, they are the ones that can intercept, redirect, decrypt, and record any communication made through them. This technology is, basically, a men-in-the-middle type of attack, as in Figure 1.2 [7].

CSS also poses a threat to fundamental rights, given that it is not possible to differentiate the connection from a real tower to one of these devices. Since they can intercept communication and its metadata (e.g. call duration, call receptor, SMS content), and also pinpoint a given device, they could be used to monitor persons of interest to law enforcement and in the worst scenarios, they could serve as a tool to surveil citizens. The positive aspect is that this technology is helpful in cases of kidnapping and terrorism threats, but it is a powerful tool to identify people attending protests and other public events.

Drone: Drones are aerial machines generally equipped with cameras in order to facilitate regional surveillance. They can record videos, take photos, intercept cellphone communications, and capture thermal and heat information while also giving an accurate location of the given footage. On top of that, some drones can also possess lethal and non-lethal weapons in order to perform operations with minimal risk to law enforcement officers but not necessarily minimal risk to civilians.

This type of technology is becoming more common as its price is dropping due to the increase in the number of market players. But given all they can do, their use as a surveillance method is alarming. As with other technologies mentioned in this section, drones become even more powerful when combined with other technologies, such as facial recognition or automatic license plate readers. They could easily surveil a person constantly.

[7] Figure retrieved from `https://www.eff.org/pages/cell-site-simulatorsimsi-catchers`. Last access: November 13th, 2023

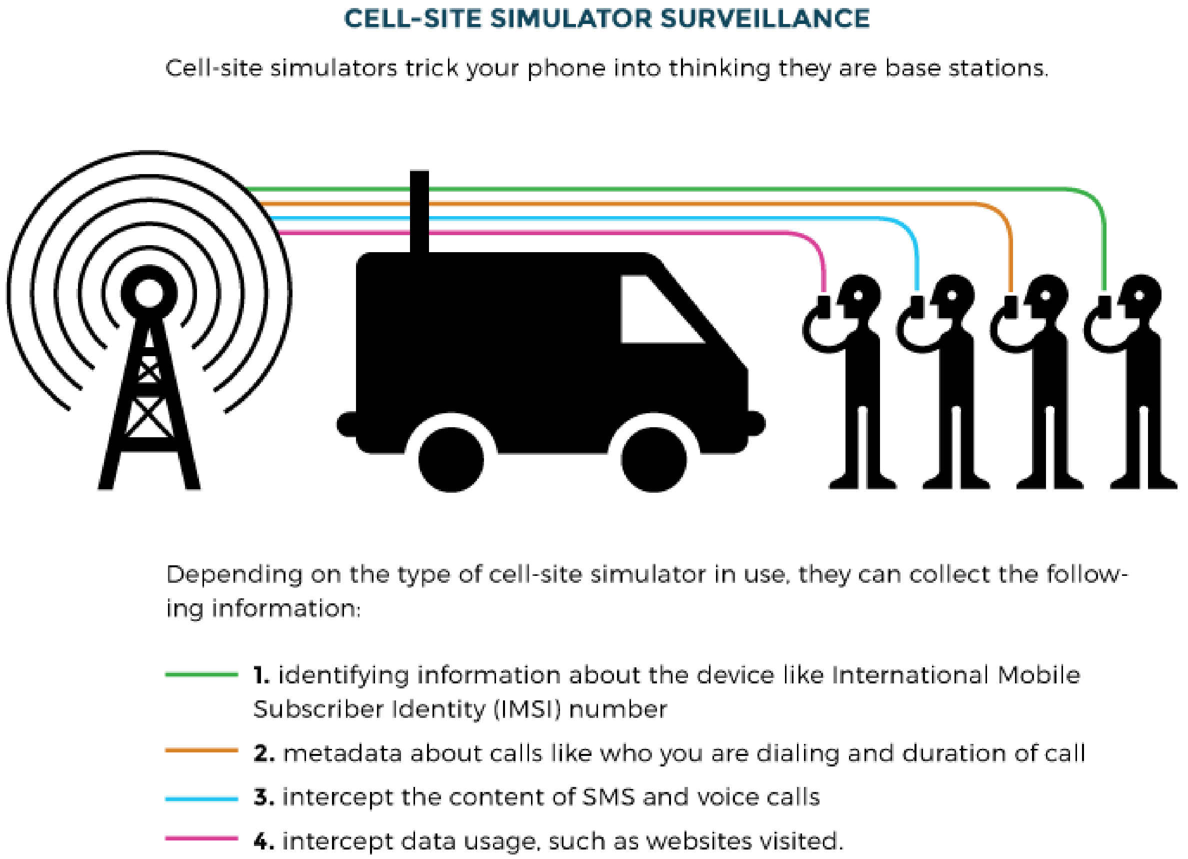

CELL-SITE SIMULATOR SURVEILLANCE
Cell-site simulators trick your phone into thinking they are base stations.
Depending on the type of cell-site simulator in use, they can collect the following information:
1. identifying information about the device like International Mobile Subscriber Identity (IMSI) number
2. metadata about calls like who you are dialing and duration of call
3. intercept the content of SMS and voice calls
4. intercept data usage, such as websites visited.

Figure 1.2: Example of Cell-site simulators communication retrieved from AoS website.

Therefore, drones threaten privacy rights, given the amount of information they can record and their carefree usage by police departments. As they can collect images of a broad range, they can film and photograph individuals not linked to illicit activity. These people could have their faces analyzed, and plates verified just because they were in the wrong place at the wrong time. Besides that, if armed, they can pose a threat to life itself.

Gunshot Detection: Gunshot detectors are equipment that detect gunfire-like sounds using microphones, sensors, and cameras, and then alert the local authorities, informing them of the exact location of the detector/sensor. In theory, they should reduce the response time of the police and other agencies when compared to an eyewitness who does the report. Besides the obvious functionality, gunshot detectors can also collect conversational audio that law enforcement uses as evidence in court.

Unfortunately, this technology is more flawed than their sellers advertise. One of the players in the market, ShotSpotter, is known for multiple cases where the sensors classified other sounds as gunshots [1]. As these detectors are placed in neighborhoods with a higher level of criminality, even when the detectors are wrong, the police goes to those places with the mentality that something is illegal happening. This can potentially lead to excessive policing in marginalized areas.

Predictive Policing: Predictive policing, in layman's terms, is the feasible version that current law enforcement could build of the movie "Minority Report" [62]. Law enforcement is not able to predict that a specific person will commit a murder but, by using an enormous dataset, it can try to pinpoint areas that are more prone to criminal activity.

An algorithm analyzes all the past information that police has over a region and locates areas that could possibly benefit from tougher policing. Despite the claims that it could enhance street safety, when we remember that this historical data is biased, we dread what could happen in neighborhoods where marginalized communities live [54].

Neighbors Partnership: Neighbors Partnership (e.g. Ring^{TM} from Amazon) is a device capable of sending footage, audio, and its location to law enforcement and aims to facilitate a citizen's report. This technology is ordinary in U.S. neighborhoods and helps people report unusual or suspect activity to agencies.

In theory, this is a collaborative way for communities to contact and help law enforcement, but unfortunately, given people's prejudice, multiple times, police is called unnecessarily. These devices can provide a sense of security for the people who live in a given neighborhood but increase the insecurity of pedestrians or drivers from marginalized and profiled communities, offending and disregarding some of their fundamental rights.

1.3.1 Evidentiary technologies

Most of the previously listed surveillance technologies can serve as evidence in court, so most could also fall under the category of evidentiary technologies. Breathalyzers, social media scrappers, and PG software are some of the widely used technologies in court, and given their importance, guidelines for validation and results reporting exist to ensure their fitness to casework [16, 68, 67, 69].

From a legal perspective in Brazil and the U.S., everyone under trial should have the right to defend themselves from the accusations. When it is someone's word against the defendant, the accusers must provide unequivocal proof that the accused committed a crime, and even then, the defendant can try to prove that the evidence is not strong enough. But how does that work when the evidence is the result of opaque-box algorithms [51, 3]?

V&V processes, in this context, become even more vital. A single mistake can change someone's life forever, especially in countries where being a previously convicted person reduces employment opportunities and carries a social stigma. We stress the importance of validation and on-point result reporting given the multidisciplinary audience that usually bases its opinions on evidentiary software. Judges, attorneys, and jury members may not be familiar with probabilistic terms or the mathematics behind the Weight of Evidence (WoE) calculation, but they must be taught and guided in a way that they can understand that the results in front of them are not black and white but possibly shades of grey.

1.4 Contributions and outiline

The main contribution of this dissertation is defining and explaining why Probabilistic Genotyping tools should be categorized as a high-integrity software. After that, we show the shortfalls in the V&V process for PG tools, by following the standard defined in [42], and we also verify if the introducing papers for all the open source Probabilistic Genotyping tools available in the market by the time we write this dissertation contemplate the guidelines [16, 68, 69, 67, 19] provided by multiple DNA Analysis expert organizations. We also provide insights about the code and GitHub communities for the target tools, and with that information, we derive assumptions about the developers of said systems. Our goal is to identify improvement areas in the Validation and Verification process for PGS, as well as to gather information about its current state, so we can better inform defense and prosecution of possible validity gaps.

To do so, in Chapter 2 we provide the reader with basic knowledge about organizations that provide guidelines and standards for DNA Analysis and Software Engineering fields, and explain the process behind DNA Analysis or the validation of Probabilistic Genotyping Software. At the end of Chapter 2, we list related publications that give insights on the current state of validation papers for PGS.

In Chapter 3, we define the method used to perform the analysis that can be found in Chapter 4. The latter aims to present the guidelines and standards in detail and shows tables that contain the assessment (True, Mostly True, Mostly False, and False) for each one of the factors/task/activities brought up by the standards. In this chapter, we also evaluate simple code and community metrics for LRmix, EuroForMix, Kongoh, LikeLTD, and LabRetriever.

Chapter 5 summarizes the discoveries made in this dissertation, presents solutions and open issues in this topic and also descriminates the limitations of this research.

Chapter 2

Background

Society and technology advances trigger and are triggered by each other. Allowing ourselves, for one instant, to do a quick digression, we remember the ancient nomad societies became sedentary groups because they adopted more effective agricultural techniques and new tools for their defense. We also recollect that the first Industrial Revolution came from a social necessity to enhance production, and when this evolution happened, our society explored new social contracts and found a new, but not necessarily better, way with capitalism.

With this simple example, we want to show that it is reasonable to say that society and technology cooperatively advance for good or evil purposes. We want to stress that they impact and influence each other in a way that it would be naive to believe that societal disparities and injustices would not affect the technology created.

Bringing the discussion to the 21st century, we can state that the Internet, Big Data, Artificial Intelligence, and other technological advances propelled the current society to a new phase of capitalism [76]. We can also identify that humanity developed new technological artifacts to fulfill other basal necessities. We have social networks to keep us connected, search platforms to answer our questions, and surveillance and evidentiary software to give the majority of the population a safety sensation. All of those systems should be properly tested before they are released to the public use, but the ones that can cause a social or even life threatening impact should abide by more strict V&V guidelines and standards.

2.1 Guidelines and standards to be considered

When we discuss the DNA analysis forensic community the names The National Institute of Standards and Technology (NIST), Scientific Working Group on DNA Analysis Methods (SWGDAM), and European Network of Forensic Science Institutes (ENFSI) are the main references. These organizations are responsible for some guidelines on validation and minimum requirements for developers and researchers during the developmental validation and reporting phases. The U.S.A. President's Council of Advisors on Science and Technology (PCAST), is another association that discusses the DNA Analysis subject, but we will not explore its publications in this dissertation given that most of the topics are based on SWGDAM's guidelines. During this research, no Brazilian entity is pointed out as relevant due to the lack of information on the use of this type of software. One entity that could probably regulate parts of DNA Analysis is National Institute of Metrology, Quality, and Technology (INMETRO), but no relevant guideline was found.

2.1.1 Institute of Electrical and Electronics Engineers

As we have mentioned, some of the evidence currently available in courts is based on software or hardware outputs, and for the Software Engineering field, an organization to remember is the Institute of Electrical and Electronics Engineers (IEEE).

IEEE, an organization created in 1884, is dedicated to advancing innovation and technological excellence for the benefit of humanity; it's the world's largest technical professional society. It serves professionals in the electrical, electronic, and computing fields and related areas of science and technology that underlie modern civilization.

For software engineers, one of the most relevant standards associated with IEEE is the 1012-2016 - IEEE Standard for System, Software, and Hardware Verification and Validation [42]. For this dissertation, we use this standard to define the level of integrity of PGS and the validation process that this level of integrity software requires.

2.1.2 The National Institute of Standards and Technology

The NIST, which nowadays is part of the US Department of Commerce, was founded in 1901 and is one of the oldest physical science laboratories in the US. Its mission is to promote US innovation and industrial competitiveness by advancing measurement science, standards, and technology that enhances economic security and improves the population's quality of life.

NIST affairs are related to many areas, like intelligent electric power grids, computer chips, and products, like Probabilistic Genotyping Software, used by governmental organizations, including the US legal systems. Many services rely at some level on technology measurement and standards provided by the National Institute of Standards and Technology.

Regarding DNA Analysis, NIST has over 150 publications that discuss the development of new tools, the calibration of machines and models, and literature reviews of published papers on the subject. For this dissertation, the publication we will focus on is the NISTIR 8351 [16], which delimits some of the DNA Analysis principles while also providing some insight into the history of this forensic technique.

2.1.3 Scientific Working Group on DNA Analysis Methods

SWGDAM is the successor of the Technical Working Group on DNA Analysis Methods (TWGDAM) group, created in November 1988, as the US started to study and apply forensic DNA technology. TWGDAM provided a direction to the forensic DNA community by issuing guidelines for a DNA proficiency testing program, polymerase chain reaction (PCR), Restriction fragment length polymorphism (RFLP), DNA analysis conducted within laboratories, and many others.

The forensic DNA community followed these guidelines when implementing their DNA programs, and they became standards and minimum requirements in courts when prosecution or defense presents DNA evidence. SWGDAM is directly associated with the Federal Bureau of Investigation (FBI) and is responsible for recommending quality assurance standards and revisions. One of SWGDAM's most important responsibilities is the recommendation of revisions to the FBI's Quality Assurance Standards (QAS) for DNA Analysis. Laboratories that intend to participate in the National DNA Index System (NDIS) must comply with these QAS; according to Federal law.

For this dissertation, we will consider the following SWGDAM's publications: "Interpretation Guidelines for Autosomal STR Typing by Forensic DNA Testing Laboratories" [68], "Addendum to 'SWGDAM Interpretation Guidelines for Autosomal STR Typing by Forensic DNA Testing Laboratories' to Address Next Generation Sequencing" [69], "SWGDAM Guidelines for the Validation of Probabilistic Genotyping Systems, and Quality Assurance Standards for Forensic DNA Testing Laboratories and for Convicted Offender DNA Databasing Laboratories" [26].

2.1.4 European Network of Forensic Science Institutes

The purpose of the European Network of Forensic Science Institutes (ENFSI), operating since 1995, is to improve the mutual exchange of information, and the quality of forensic science delivery in Europe. ENFSI possesses 17 Expert Working Groups besides the general work in quality and competence management, research and development, and education and training. ENFSI, therefore, has been recognized as the monopoly organization in forensic science by the European Commission. For this research, the Working Group we are most interested in is the DNA Working Group, which supports the goals of ENFSI in the area of DNA casework analysis.

The ENFSI DNA Working Group nurtures discussions about the validation, introduction, and improvement of DNA analysis in casework, considering all aspects of DNA case analysis and case reporting [25]. This group fosters quality management systems, uniform best practices manuals/guidelines, exchanges between organizations regarding information and expertise, collaboration regarding reports and interpretation of DNA analysis, the collaboration between academia and industry, and other fronts related to training and standardization between laboratories and industry.

2.2 DNA analysis processes and results

DNA Analysis has changed since its first use in court [70]. Restriction fragment length polymorphism (RFLP) was the first methodology applied to DNA testing and nowadays mainly appears in court in post-conviction cases. The evidence generated and presented to judges was an Autoradiogram (AUTORAD), showing graphically to the forensic expert if a given allele was present. This method took a few months to generate its results and relied on great amounts of DNA templates (size of a coin) to produce reliable results. Despite issues with analysis duration and sample size, RFLP tests had positive delivery metrics for discriminative pallor. A coincidental match between a suspect and the true perpetrator would have a probability of 1 in 10 million.

The first polymerase chain reaction (PCR)[1] based tests were the Polymarker tests (DQ-alpha), the second generation of DNA tests. The critical bit of evidence came from a series of test strips that passed through analysis to detect the presence or the absence of blue marker dots. A particular pattern of these blue dots provided insights into which DNA molecules were present in a given sample. The amount of biological material needed to generate results for this test generation was 1% of the necessary material on RFLP tests. In a few hours, the results were available, but the discriminative pallor had a probability of one in a few thousand.

The latest generation of DNA tests is the Automated Short Tandem Repeat (STR)[2]. DNA profiles are generated as Electropherograms (EPG), from samples with around a hundred cells in just a few hours and with good discriminative pallor. The probability of a suspect coincidentally matching the perpetrator would be around one in a quintillion for major contributors, given the sample complexity.

Probabilistic Genotyping Tools are used in two possible situations, as shown in Figure 2.1, during a validation study or casework. In both scenarios (study and casework), the first step is the collection of DNA samples, and the last step is a report used in court to sustain the presented evidence. As in any process, the ones described here are prone to errors. Problems regarding the collection, such as degradation and inhibition, affect the analysis in a mixture and profile level. The remaining issues (i.e., stutters, spikes, blobs) are problems during the preparation of the sample and are known as STR Artifacts [46].

DNA collection is the process that retrieves biological material that becomes the Electropherograms (EPG) at the end of the DNA Analysis process. Nowadays, due to the increase of sensitivity in the analysis of DNA, any object/surface could be used as a source of this type of material, but experts try to focus on the collection of objects with a higher probability of having at least 500 and 1,000[3] pg[4] of template DNA. It is incredible that no more than one or two fingerprints can help prosecutors and defense attorneys to build a case, but at the same time that this sensibility is valuable, it also enhances the complexity of some analysis [46].

Collection on Casework: DNA collection retrieves biological material from crime scenes, POIs, and National Databases in casework. The reference profiles are collected from victims, POIs, or National Databases when the case has no lead. Experts run matches against National Databases in casework because it could return some possible suspects to the criminal case.

Collection on Validation Studies: DNA collection retrieves biological material from volunteers, simulations, and National Databases in validation studies. The Reference

[1] A technique for replicating DNA in the laboratory that amplifies specific fragments of DNA defined by primers. Primers are short nucleic acid s that provide a starting point for DNA synthesis.

[2] Short tandem repeats (STRs) are short repeated sequences of DNA (2–6 bp) that account for approximately 3% of the human genome [47]). The number of repeat units is highly variable among individuals, which offers a high power of discrimination when analyzed for identification purposes.

[3] DNA profiling kits generally recommend using between 500 and 1000 pg of template DNA.

[4] Picogram (pg) = 10^{-12} of a gram [70]. As an example, a fingerprint has roughly between 600 and 700 pg of DNA.

profiles are collected from volunteers, simulations based on allele frequency, or casework files. In PGS validation studies, the calculation of Random Match Probability (RMP) is made over profiles collected from National Databases.

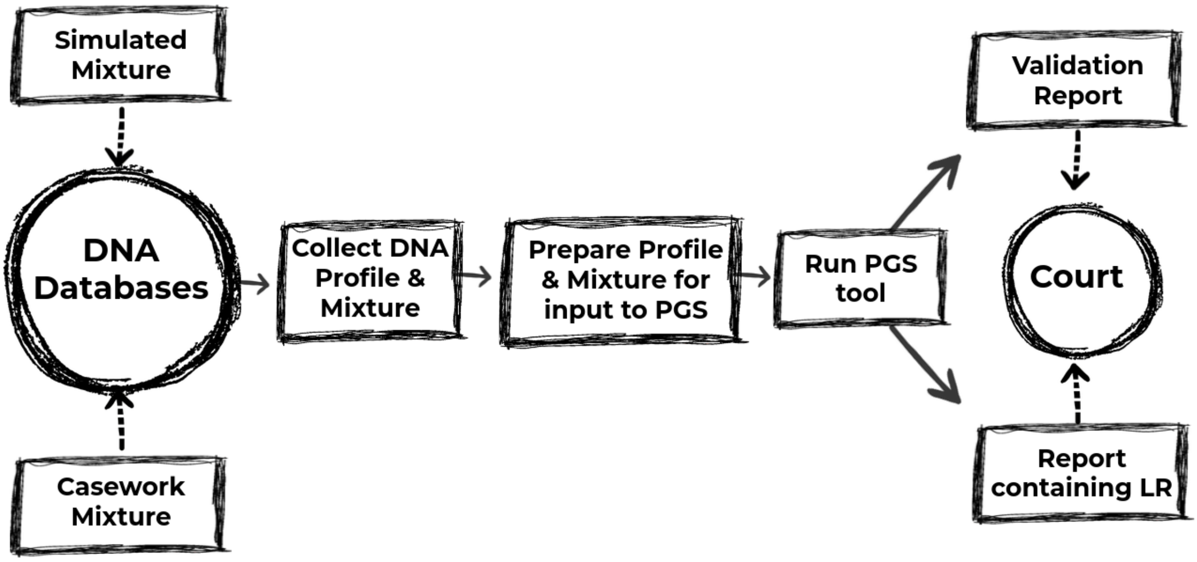


Figure 2.1: Flow of PGS - How DNA evidence is presented in court.

After the DNA collection step, the forensic analyst extracts (first step) and purifies(second step) the DNA. These processes are straightforward, being more complex when dealing with reproductive cells that require differential extractions to deal with their protein compositions.

The third step is the PCR amplification [70, 47] which is the process that amplifies DNA regions flanked by primers (marked regions). At each one of the 28 rounds of amplification, the DNA region suffers denaturation, annealing, and elongation resulting in a duplication of the areas at each round. In order to facilitate the further steps, the amplified DNA fragments receive a fluorescent label to be easier to identify each speck.

The fourth step is the size fractionation [70, 47] that is viable due to a technique that separates the molecules based on their size called capillary electrophoresis. The DNA goes towards the Genetic Analyzers machines, and a camera on the detector window captures the fluorescent signal. This process generates the raw data that is processed and turned into EPG. EPGs are the outputs analyzed as a DNA profile, and technical artifacts by the forensic expert and PGS add the WoE. Forensic experts calculate Random Match Probability (RMP) for weighting single source samples. The RMP represents the probability of a random individual matching a given genotype within a population; in the US, this information [60] about population allele frequency is available in the NDIS database.

Identifying a plausible Number of Contributors (NOC) to a sample is the first step when dealing with mixtures. Experts and algorithms calculate the Maximum Allele Count (MAC) to identify the minimum NOC that could generate a given allele distribution. PGS calculate the ratios of two probabilities that evaluate the evidence given at least two mutually exclusive propositions to assess the weight of evidence to mixtures; these ratios are called Likelihood Ratios (LR)[18, 73, 15]. In the past, experts attributed the WoE using Combined Probability of Inclusion (CPI) which is the probability of a random unrelated person being a possible contributor to the mixture.

In casework, the WoE is the value reported by the forensic analyst to prosecutors and public defenders to sustain their hypothesis during the trial. In validation, metrics are extracted from the WoE to provide a report on how accurate a given PGS or a given laboratory process is. Another point that seems interesting to highlight is the statistic consensus method, an approach that compares the likelihood ratio values provided by different software. For converging results the report presents the most conservative likelihood ratio value, and in divergent results, the DNA interpretation is considered inconclusive [30].

Public defenders must understand and require the complete set of data used to state the forensic analysis report as oriented in [29], since data misinterpretation can be harmful, as stated in [48]. The Organization of Scientific Area Committees (OSAC) for Forensic Science's Human Forensic Biology Subcommittee, with contributions from the Scientific Working Group on DNA Analysis Methods (SWGDAM), recently provided a document that describes a Human Forensic DNA Analysis Process Map [5], which illustrates some other procedures that might be performed during a sample analysis. However a lack of rigorous standardization inter-laboratory and intra-laboratory is also known [18, 66]. Figure 2.2 contains a summarized version of the DNA Analysis and DNA reporting steps.

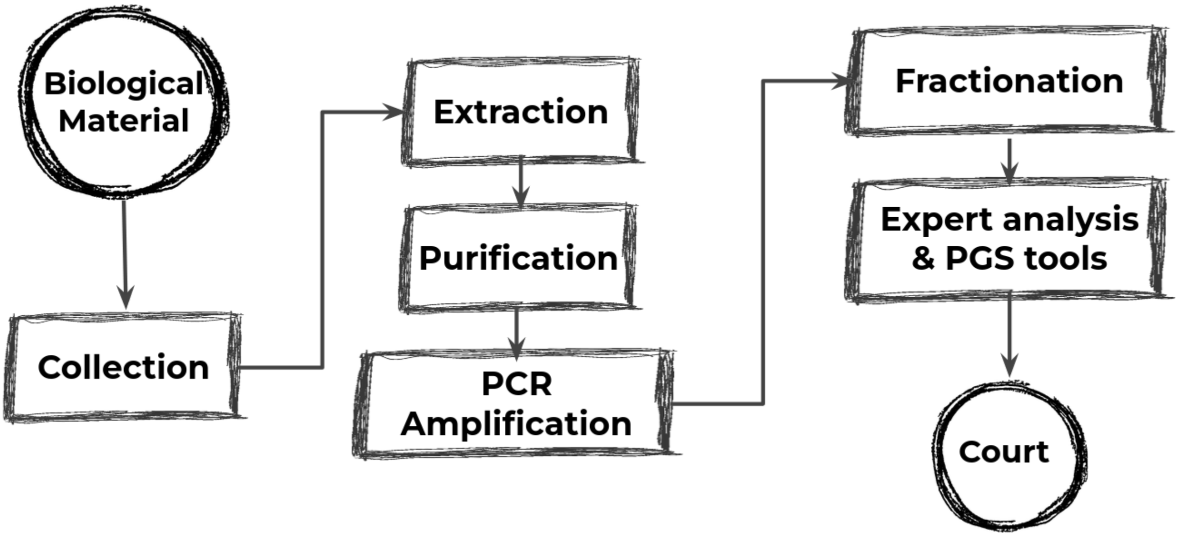


Figure 2.2: Milestones of DNA Analysis process.

2.2.1 Probabilistic Genotyping Software

PGS is a class of evidentiary statistical software used by law enforcement agencies to help link a genetic sample found at a crime scene to a person of interest (POI) [1, 18].

[5] The entire diagram scheme can be found at `https://www.nist.gov/system/files/documents/2022/05/05/OSAC%20Forensic%20Biology%20Process%20Map_5.5.22.pdf`. Last access: November 13th, 2023.

These tools receive as entries electropherograms containing the sample and the reference profiles of the contributors, and return LR that indicate the support of the evidence for the prosecution or defense hypothesis after performing the series of procedures mentioned before.

There are a few tools available when we discuss PGS. STRMix, TrueAllele, FST, LabRetriever, LRmix, EuroForMix, DNAStatistX, likeLTD, Kongoh, CEESit, MaSTR, LiRa, and DNA-VIEW are some of the probabilistic genotyping tools used to generate the reports presented in court. Generally, the developers introduce these tools to the forensic community in peer-reviewed publications that explain the theory behind the system itself while also providing a test set for validation purposes. These test sets try to generate evidence that the software is compliant with SWGDAM's, NIST's, or ENFSI's validation guidelines, but publications that should address developmental validation, do not abide by or mention IEEE standards.

Proper developmental validation of PGS is crucial, given the impact that it may have on a conviction. The existence of The Innocence Project [6], for example, is proof that sometimes experts might wrongfully interpret forensic evidence, therefore providing a testimony that would not be accurate in court, potentially leading to an invalid conviction. PGS developers should follow rigid standards while developing their tools, especially the ones defined for high-integrity systems.

2.3 Related Work

Figure 2.3 illustrates the relations between PGS publications, DNA databases, law enforcement, and the forensic community. This picture is a map for us to understand how issues in DNA databases and PGS validation can influence court decisions and how validation publications mold the level of trust bestowed on probabilistic genotyping software.

When discussing PGS validation, the forensic community focuses on their field standards. Laboratories and developers abide by SWGDAM's, other forensic, and State organizations guidelines to gain recognition for their processes, services, and products. Therefore, it is in their best interest that forensic laboratories and software developers publish their discoveries and provide results that can attest that they comply with market regulations.

It is possible to find a large batch of developmental validation studies demonstrating the casework fitness of PGS [49, 13, 5, 57, 23, 40, 34, 37, 35, 32], another set of publications discussing the accuracy of these same tools [63, 44, 11, 9, 52], many others that unfold the statistics and calculations behind probabilistic genotyping systems [56, 65, 45, 17, 36, 33, 4], and a few that focus on ethical aspects of introducing this tools in law enforcement [1, 51, 31, 7, 75, 24]. But something hard to find is publications that evaluate these systems from a Software Engineering (SE) perspective.

Buckleton, Bright, Cheng, and Taylor [14] try to present a report from SE's perspective, but they do not dig deep enough from a technical point of view. As a response to

[6] More information available at `https://innocenceproject.org/misapplication-of-forensic-science/`. Last access: November 13th, 2023.

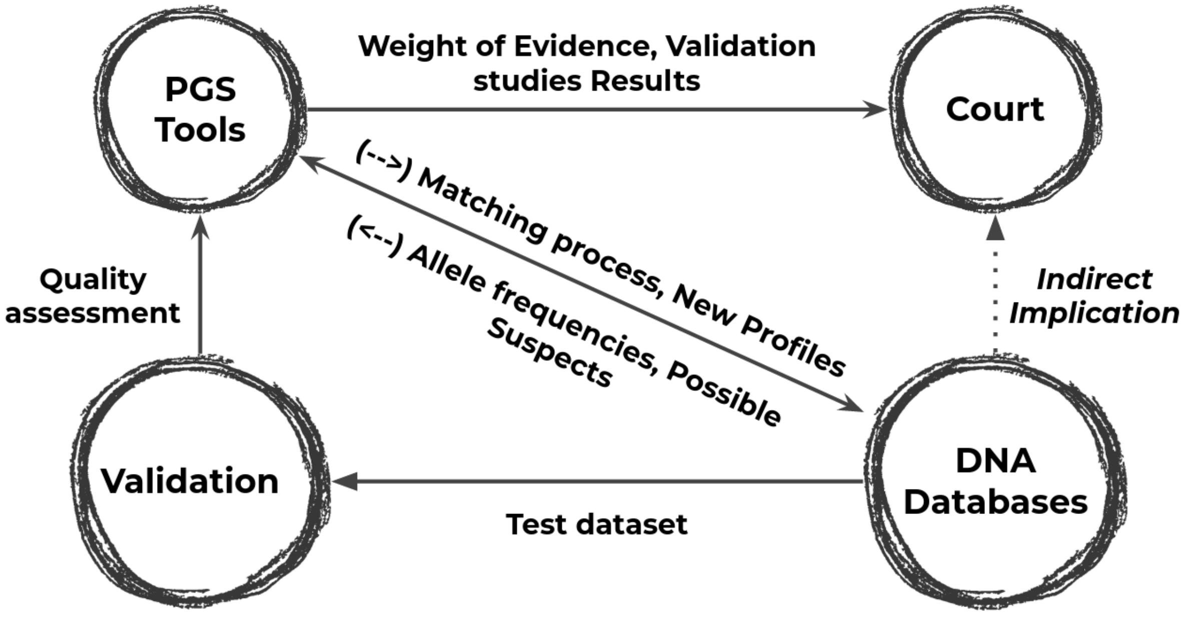

PGS Tools
Court
Validation
DNA Databases
Weight of Evidence, Validation studies Results
(-->) Matching process, New Profiles
(<--) Allele frequencies, Possible Suspects
Quality assessment
Indirect Implication
Test dataset

Figure 2.3: Implications of DNA databases and validation process in court.

evidence on anomalous results of the STRmix software, this paper describes the "code review" and testing processes performed by STRmix creators to verify the cause of the errors. In this publication, the authors borrow SE concepts like unity testing, bugs/defects, and code review, but they define them differently than the software engineering community. The paper revolves around the lack of well-defined software development and validation processes.

In Bright (2022) [12], a group of authors describes the regression testing of the same tool. The text explains more about the experiments and results, covers degradation aspects and mixtures, and proves that there are no significant differences between the two versions of the software but the report provided in the publication lacks a pattern when compared to IEEE standards.

The report on STRmix™Validation v2.3 is one of the most comprehensive reports from a SE and DNA analysis perspective but, in its conclusion, even if it mentions that there are some limitations to its use, it states "It has been shown that STRmix™v2.3 is suited for its intended use for the interpretation of profiles generated from crime scene samples" [7] which is just partially true, due to the border cases on the limitations.

The difference between this dissertation and the other publications listed here is that we want to explore this SE point of view by analyzing PGS as a high-integrity software. Therefore, we want to show that the current validation reports are shallow in some aspects, especially when dealing with complex casework. With this publication, we do not aim to entirely dismiss PGS, as it is a powerful and necessary technology in cases related to sexual harassment and simpler casework, but the developers of these tools should scrutinize them better before selling it as an all-knowing tool, that would never make a mistake because that's not the case when we add complexity levels to the analysis.

[7] Publication available at:`https://dfs.dc.gov/sites/default/files/dc/sites/dfs/page_content/attachments/STRmix%20Validation.pdf`. Last access: November 13th, 2023.

Chapter 3

Method

This chapter describes the method we followed to conduct the research behind this dissertation. Here we will explain; 1) Why we choose the DNA Analysis or Software Engineering standards and guidelines, 2) Why we choose LRmix, EuroForMix, Kongoh, LikeLTD, and LabRetriever as our case study software, 3) Explain which static code metrics we try to gather during the GitHub repository analysis of each software.

3.1 Guidelines and standards selection

We've selected the SWGDAM, NIST, ENFSI, and IEEE standards, given their relevance in their specific communities. As mentioned in Section 2.1, these organizations are responsible for regulating and inquiring about the use of forensic technology in law enforcement, or they control the software development and distribution chain. They became references all over the world when debating DNA Analysis, and the validation process performed by Kongoh developers[50] is an example of how global those guidelines are.

For this dissertation, the publication we focus on is the NISTIR 8351 [16], which delimits some of the DNA Analysis principles while also providing some insight into the history of this forensic technique. Section 2 of NISTIR 8351 presents a list of factors that affect measurement reliability and mixture complexity. As we are discussing validation, it seems fair to verify those factors, that might cause implications on the reliability of the software under analysis. They will be discussed and assessed during Chapter 4.

We also consider the following SWGDAM's publications: Interpretation Guidelines for Autosomal STR Typing by Forensic DNA Testing Laboratories [68], Addendum to "SWGDAM Interpretation Guidelines for Autosomal STR Typing by Forensic DNA Testing Laboratories" to Address Next Generation Sequencing [69] and SWGDAM Guidelines for the Validation of Probabilistic Genotyping Systems [67]. The highlights of each document are described and assessed for each of the PG tools in Chapter 4.

As we analyze the European PG tool EuroForMix, we also consider the ENFSI-BPM-DNA-01 [19] publication. This document highlights the best practices for internal validation of PGS and other quality metrics, which are relevant when validating DNA forensic software for casework use. All of the best practices and quality metrics are described and assessed for each of the PG tools in Chapter 4.

For software engineers, one of the most relevant standards associated with IEEE is the 1012-2016 - IEEE Standard for System, Software, and Hardware Verification and Validation[42]. For this dissertation, we use this standard to define the level of integrity of PGS and the validation process this level of integrity software requires. 1012-2016 - IEEE Standard defines 45 activities that developers and companies must follow in order perform proper V&V for high integrity software, these activities require more than 200 tasks. All the activities will be described and assessed for each of the PG tools in Chapter 4.

For each factor or activity, we evaluated if they were reported in the initial publications or the GitHub repositories, trying to be as faithful as possible to the descriptions made by the standards and guidelines. In cases where some factor was only mentioned we used the markers of mostly true or mostly false to indicate that at least there was an attempt to cover said factor.

3.2 Case study software choice

We chose LRmix due to its relevance in the Brazilian context [27] on PGS and the remaining because they complete the entire set of open source PGS. This aspect is important because this gives us access to the code behind the tools and like that we can perform a more informed assessment of the tool itself. In Chapter 5 we will point out other positive aspects of open source software in public spheres, but for now the most attractive feature is the code transparency.

To obtain a list of possible open-source PGS with at least one scientific paper we performed a search in Scopus with the query: `TITLE-ABS-KEY (("probabilistic" OR "probability" OR "likelihood ratio" OR "likelihood") AND ("DNA" OR "genotype" OR "DNA profile" OR "Forensic DNA") AND ("software tool" OR "software" OR "program") AND ("develop" OR "developmental" OR "development") AND "forensic").` This query returned a list of 35 papers that at least cited a PG tool, and then manually we checked the available information on the software websites to asses if they were truly open source. In each paper we also looked for references of other PG tools, so we could have a more comprehensive list of Probabilistic Genotyping Software. This process shows that the only available open source tools, by October of 2023, are: LRmix, EuroForMix, Kongoh, LikeLTD, and LabRetriever. MaSTR and CEESIt are mentioned as open-source PGS, but when we analyse the tools website and try to access the repository where the code should be available for public use, this is not accurate, therefore they will not be evaluated in this study.

In this dissertation, we do not provide a comparison of the probabilistic genotyping tools. The goal is to evaluate how the only five open-source software available in the market, by the time being, complied with international guidelines in their release moment. We also don't perform comparisons with the most recent publications related to those tools because we believe that, given the integrity level of the technology, what we point out in this dissertation should be part of the basic analysis for these technologies' release [22].

3.3 Code quality and GitHub community metrics

The communities around these tools are not that significantly different when we look into GitHub [1] [2] [3] [4] [5] metrics. Although the projects differ in technologies, most of them have only one main contributor, have less than four years, less than eleven open/closed issues, and less than ten forks. The only project that differs from the majority is LabRetriever that has three main contributors, ten years of development, and sixty seven open/closed issues. If we compare those projects with other ones with the same age, we could say that the community around PGS is not as active as others.

The metrics of static code analysis that we use are:

- Functional coverage: a metric that defines how much of the functional requirements of the software are covered. It helps to understand if the business requisites go under the proper analysis during testing. Since we aim to demonstrate that border cases might be disregarded, during PG tools testing, we believe that it is an important metric to be analyzed.

- Risk coverage: A metric that defines how the potential adverse risk of a system is scrutinized during the test phase. Again as we aim to demonstrate that some aspects of PG testing should be more strict, it is valuable to use a metric that assesses how the software could negatively impact the sphere where it acts.

Another metric that we intended to test is code coverage, but due to build issues on all projects this analysis was not feasible.

[1] GitHub for LRmix project: `https://github.com/smartrank/lrmixstudio`

[2] GitHub for EuroForMix project: `https://github.com/oyvble/euroformix`

[3] GitHub for Kongoh project: `https://github.com/manabe0322/Kongoh`

[4] GitHub for LikeLTD project: `https://github.com/cran/likeLTD`

[5] GitHub for LabRetriever project: `https://github.com/SCIEG/LabRetriever`

Chapter 4

Discussion, experiments and results

This chapter focuses on the discussion around Probabilistic Genotyping Software (EuroForMix, LRmix, Kongoh, LikeLTD, and LabRetriever) under the scope of their first appearance in peer-reviewed papers [2, 8, 50, 64, 43] and their current GitHub repository [1] [2] [3] [4] [5]. We put these sources in evidence because, in theory, when a system is outed to the community, the minimum verification and validation requirements must comply with the standards that structure its field, especially if this system impacts people's lives.

In this dissertation, we consider the following agencies/organizations as a source of standards for the field and the systems developed; NIST, SWGDAM, ENFSI, and IEEE. We appraise the initial trio due to its relevance for the DNA Analysis field, and the former is relevant because it states the minimum requirements valued by the Software Engineering field. We present two table that summarize if the introducing paper of EuroForMix, LRmix, Kongoh, LikeLTD, and LabRetriever, or the GitHub repository [6], abide by their stated indications.

Figures 4.1 and 4.2 represent the summarized results of this chapter, they contain the pie charts that illustrate how well each open-source PGS abides DNA Analysis (Figure 4.1) and Software Engineering standards (Figure 4.2). It is possible to see that most of the PG tools have a good coverage of DNA Analysis factors, but when we analyze Software Engineering standards the situation is completely different.

There is a big difference between the adequacy of the five Probabilistic Genotyping Software to the rules defined by organizations linked to the area of DNA Analysis and the rules defined by the IEEE. EuroForMix follows 59.38% of DNA Analysis guidelines (75.00% considering partially covered factors) but only abides by 3.05% of the tasks defined by IEEE standards (4.27% considering partially completed tasks). LRmix follows 65.63% of DNA Analysis guidelines (71.88% considering partially covered factors) but only abides by 3.05% of the tasks defined by IEEE standards (4.27% considering partially

[1] GitHub for LRmix project: `https://github.com/smartrank/lrmixstudio`

[2] GitHub for EuroForMix project: `https://github.com/oyvble/euroformix`

[3] GitHub for Kongoh project: `https://github.com/manabe0322/Kongoh`

[4] GitHub for LikeLTD project: `https://github.com/cran/likeLTD`

[5] GitHub for LabRetriever project: `https://github.com/SCIEG/LabRetriever`

[6] The last `pull` used for the analysis was made in October 2023. This disclaimer aims to inform that we used the most recent version available of the code during our analysis. We don't use the initial `commit` because it might not contain relevant code.

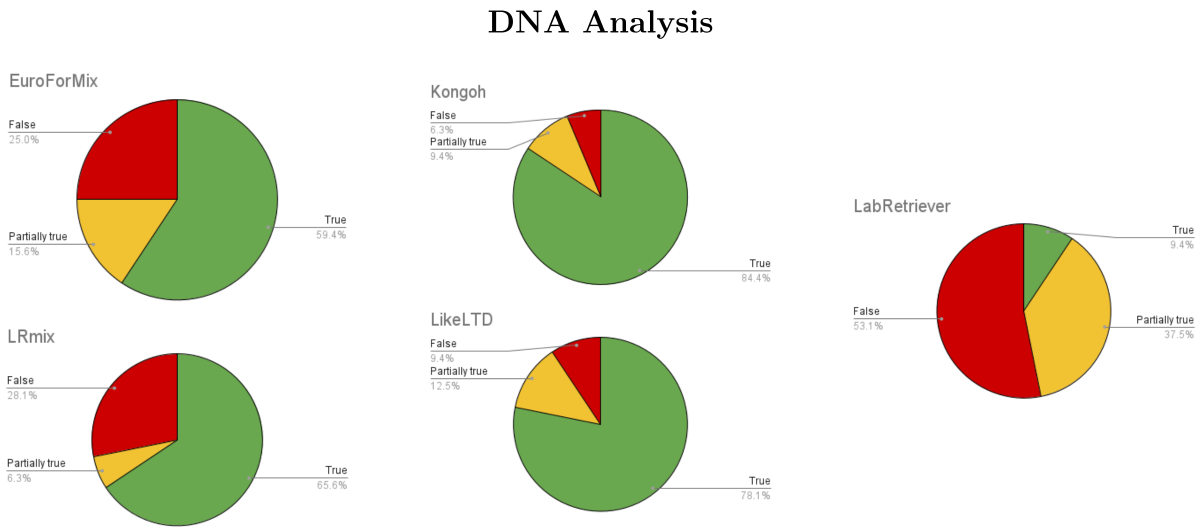


Figure 4.1: DNA Analysis result summarizing

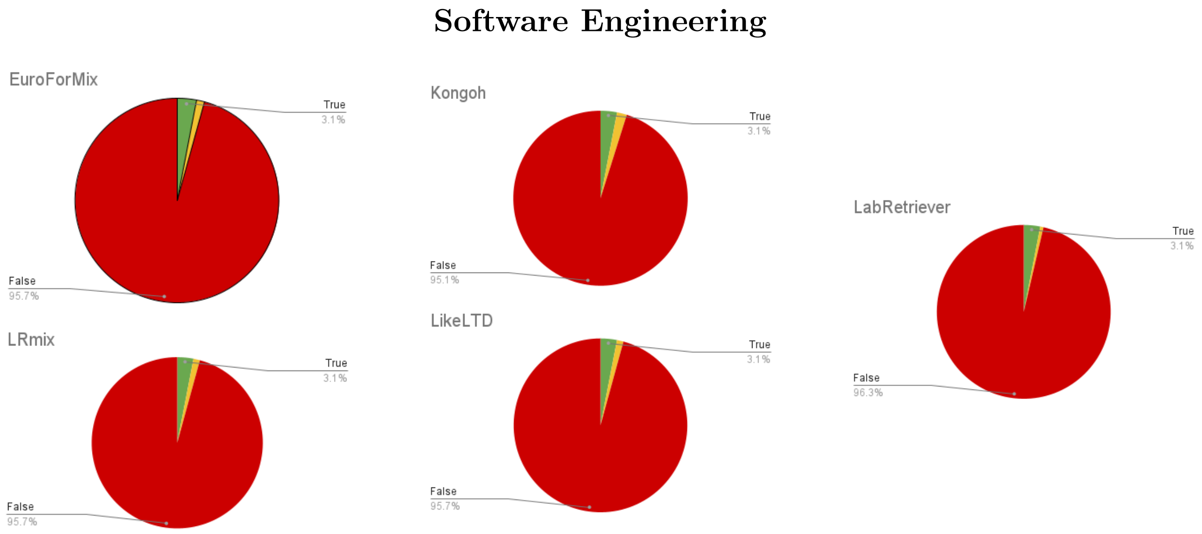


Figure 4.2: Software Engineering results summarizing

completed tasks). Kongoh follows 84.38% of DNA Analysis guidelines (93.75% considering partially covered factors) but only abides by 3.05% of the tasks defined by IEEE standards (4.88% considering partially completed tasks). LikeLTD follows 78.13% of DNA Analysis guidelines (90.63% considering partially covered factors) but only abides by 3.05% of the tasks defined by IEEE standards (4.27% considering partially completed tasks). And LabRetriever follows 9.38% of DNA Analysis guidelines (46.88% considering partially covered factors) but only abides by 3.05% of the tasks defined by IEEE standards (3.66% considering partially completed tasks).

The analysis of DNA Analysis factors and SE tasks in the Tables 4.1 and 4.2 is presented as; 1) True (●) if the factor or task is fully covered in the introducing paper or GitHub repository, 2) Partially true (●*) if most of the factor or task is covered in the introducing paper or GitHub repository, 3) Partially false (❍*) if most of the factor or task is not covered in the introducing paper or GitHub repository, and 4) False (❍) if the factor or task is not covered at all in the introducing paper nor GitHub repository. At the end of this Chapter, we deepen the discussion on our findings and deliberate what they mean from a validation perspective.

4.1 NIST

In this section we describe the factors that NIST evaluates as relevant for the validation of PGS, and verify if the introducing papers of EuroForMix, LRmix, Kongoh, LikeLTD, and LabRetriever, or their respective GitHub repositories, abide by the verification recommendations.

- **Peak position:** Peak positions are measured as migration time (raw data), nucleotides (against the size standard), and allele designations (against an allelic ladder). The accurate determination of peak locations is necessary for reliable STR allele designations.

- **Peak morphology:** This is when wide peaks result in poor resolution and the inability to fully separate STR alleles that differ by as little as a single nucleotide. Capillaries fail and resolution is lost after many CE sample injections. Failure to resolve similar length STR alleles may result in missing true contributor genotypes. Wide peaks may also size inaccurately designations.

- **Peak heights:** Measured in relative fluorescence units (RFU) and are generally proportional to the amount of PCR product detected. While an RFU value does not necessarily correspond to a specific number of picograms of DNA, variation in peak heights matters because this information is used to deconvolute mixture components into contributor genotype possibilities. Essential when calculating information impacted by peak heights, such as stutter percentages and peak height ratios.

- **Stutter products:** Produced during PCR amplification from slippage of the DNA strands while being copied, and are typically one repeat shorter or longer than their originating STR allele.They can be indistinguishable from true alleles of minor contributors and therefore impact DNA interpretation.

- **Spectral artifacts:** This is an anomaly of the detection process where fluorescent signal from one spectral channel bleeds through into an adjacent color channel (e.g., green into blue). Pull-up occurs from a saturating signal on the instrument detector. When low quantities of DNA are tested, it can be challenging to differentiate true alleles from amplification or detection artifacts. Spectral artifacts may also signal off-scale data in an EPG that should be avoided, as the stutter ratio will not be accurate.

- **Relative peak heights of allele pairs within a locus:** Heterozygous STR loci possess two alleles that differ in overall PCR product size. The peak heights of these two “sister” alleles can be compared in single-source samples to enable genotype assumptions in samples containing more than one contributor. Determines the limits of pairing alleles into genotypes with binary approaches and also helps to define parameters used for assigning potential genotypes and mixture ratios with PGS.

- **Assessing relative peak heights across loci in a DNA profile:** Provides an indication of the quality of a sample. With degraded DNA, peak heights decrease from left to right across an EPG (small-size to large-size STR alleles). Ratios between mixture components may dient becomes visible. Fails if the element is not visible after the timeout expires. This commaffer across tested loci.

- **Baseline noise:** Noise exists in all measuring systems. In a DNA profile EPG, noise is represented as jitter in the baseline signal. Enables an analytical threshold to be set and a lower limit of reliability to be established for peak heights.

- **Stochastic variation:** Measures approximation variations on allele quantity. Impacts recovered quantities of alleles from contributors and can lead to uncertainty in assigning alleles to genotypes and uncertainty in assigning genotypes to contributor profiles when examining small amounts of DNA.

- **Sharing of common alleles:** This happens when more than one contributor share alleles, and it can happen especially in samples where relatives are contributors. This factor influences the ability to estimate the number of contributors, particularly when combined with stochastic variation and the existence of stutter.

- **Number of contributors:** This factor corresponds to the number of known (or probable) contributors in a DNA sample. This factor influences the complexity of DNA analysis because it increases the probability of having alleles in the same loci, which can lead to mistakes in the contributors estimation and the probability itself.

4.2 SWGDAM

This section analyzes the SWGDAM's standards and guidelines and how EuroForMix and LRmix Software observes its considerations and regulations. We will scrutinize three documents that aim to guide laboratories while stating basic validation procedures for DNA analysis when discussing probabilistic genotyping. Most of the NIST requirements and attention points are replicated in this section. This happens because NIST uses SWGDAM as a source to define its own guidelines. With that said, a more high-level description will be available in this section.

4.2.1 Addendum & SWGDAM Interpretation Guidelines for Autosomal STR Typing by Forensic DNA Testing Laboratories

The SWGDAM Interpretation Guidelines for Autosomal STR Typing by Forensic DNA Testing Laboratories publication and its addendum is one of the most cited standards in peer-reviewed papers, even though it was designed for binary interpretation approaches instead of, for instance, probabilistic genotyping. Its objective is to state guidelines for best practices and not a formal standard for the DNA Analysis field. These documents declare that they mainly address not-so-complex mixtures (two contributors mixtures, not low template samples, etc.), even though the concepts should still be valid for more complex cases.

In this dissertation, we don't focus on the sections related to laboratory procedure validation, and we use the following points to assess the quality of the papers and documents that inquire about the validation process of EuroForMix, LRmix, Kongoh, LikeLTD, and LabRetriever:

- **Possible genotype combinations:** The determination of possible genotype combinations of the contributors present on the mixture is a factor that contributes during validation because it allows different analysis perspectives.

- **Data support:** Besides the fact that validation reports must support the affirmations provided about the PG tool under analysis, this factor is also aligned with genotype combinations, casework distribution, and many other factors that force the tool to deal with different samples. Being thoughtful of this factor during validations enhances the strength of affirmations such as "This system is fit for use".

- **Casework distribution with known contributors:** It is necessary to run the Probabilistic Genotyping Software against sample distributions it would face in court. Otherwise, it is not safe to say that it would work in real scenarios.

- **Source population database disclosure:** The source of the population database(s) used in any statistical analysis must be disclosed, so it is possible to reproduce the findings and analyze if the allele frequency or its distributions are bias-free.

- **Homozygous and Heterozygous typing results:** Homozygous typing results happen when at least one mixture contributor possesses identical alleles for a particular locus (e.g., sex chromosome in females). Heterozygous typing results happen when at least one mixture contributor has non-identical alleles for a given locus (e.g., sex chromosome in males). This factor is related to the concepts of peak heights and similar ones available in Section 4.1.

- **Multiple locus profiles:** Is the combination of the alleles found at two or more loci in a single individual, adding complexity to the sample because it can affect peak heights.

- **Mixtures and single sources:** Mixtures are samples with two or more contributors, and single sources are samples that contain the DNA of a single individual. This factor is related to the NOC in a mixture defined in Section 4.1.

- **Biological relationships and allele sharing hypothesis:** The allele sharing hypothesis is defined as when two or more contributors to a mixture have identical alleles in a locus, an ordinary trace when there is a biological relationship between contributors (e.g., close relatives). This factor is related to the sharing of common alleles concept available at Section 4.1.

- **Analytical threshold:** This identifies the lowest value at which DNA can be distinguished from baseline noise. This factor relates to the baseline noise concept available in Section 4.1.

- **Non-allelic peaks:** Sttuter products, blobs, and spikes are the most common causes of non-allelic peaks, which literally means peaks in the EPG that do not derive from alleles. This factor relates to the stutter products concept available at Section 4.1.

- **Instrumental artifacts:** Peaks caused by issues in the equipment. This factor relates to the concept of spectral artifacts available in Section 4.1.

- **Stochastic effects of low-template amplification:** Issues during the amplification process, when some part of the DNA is not properly processed. This factor relates to the stochastic variation concept, available in Section 4.1.

- **Intra-locus peak height ratios (PHR):** Comparison between two "sister" alleles that help the generation of genotype assumptions in mixtures. This factor relates to the concepts of peak heights and similar ones available in Section 4.1.

- **Number Of Contributors:** Number of possible individuals collaborating in a mixture. This factor is available in Section 4.1.

- **Mixture ratios:** This factor investigates how ratios of contributing individuals affect the final analysis.

- **Minor and major contributors:** This factor is related to mixture ratios. Minor contributors are individuals that have the smaller part of DNA affecting the sample, and major contributors are the antithesis of minor contributors. It is important to validate different ratios of contribution to simulate casework.

- **Degradation:** Any biological material can suffer degradation, which is the process of losing some part of material. This factor tends to add complexity to the analysis of peak heights.

- **Allelic drop-in:** This factor is related to sample contamination, spectral artifacts, and stutter products. They cause a deviation in the allele peak heights and generation of probable genotypes.

- **Allelic dropout:** This factor is related to situations when there is not enough amplification of an allele, also affecting peak heights analysis and generation of probable genotypes.

4.2.2 SWGDAM Guidelines for the Validation of PG Systems

Guidance is provided herein for the validation of probabilistic genotyping software used for the analysis of autosomal STR typing results. In this dissertation we use the following features to assess the quality of the papers and documents that inquire about the validation process of EuroForMix, LRmix, Kongoh, LikeLTD, and LabRetriever:

- **Audit trails:** Audit trails to track changes to system data and/or verification of system settings in place each time a calculation is run.

- **Publication of Scientific Principles:** The underlying scientific principle(s) of the probabilistic genotyping methods and characteristics of the software should be published in a peer-reviewed scientific journal. The underlying scientific principles of probabilistic genotyping include, but are not limited to, modeling of stutter, allelic drop-in and drop-out, Bayesian prior assumptions such as allele probabilities, and statistical formulae used in the calculation and algorithms.

- **Sensitivity tests:** This factor assesses the ability of the system to reliably determine the presence of contributor's DNA over a broad variety of mixtures. This should be evaluated using various sample types (e.g., different numbers of contributors, mixture proportions, and template quantities).

- **Specificity tests:** This factor assures that the evaluation of the ability of the tool to provide reliable results for non-contributors. This should be evaluated using various sample types (e.g., different numbers of contributors, mixture proportions, and template quantities).

- **Precision tests:** Studies should evaluate the variation in likelihood ratios calculated from repeated software analyses of the same input data. This should be evaluated using various sample types (e.g., different numbers of contributors, mixture proportions, and template quantities).

- **Case-type and control samples:** The use of manufactured samples that resemble casework material. This factor relates to the concept of casework distribution with known contributors available at Section 4.2

- **Accuracy tests:** This factor relates to the assessment of the accuracy of the calculations performed, as well as allele designation functions.

- **Artifacts:** Artifacts are the simplified name of the spectral artifacts available in Section 4.1.

- **Peak heights (Intra-locus peak height variation and Inter-locus):** This factor is related to the concepts of peak heights and similar ones available at Section 4.1, particularly those that make comparisons of "sister" peaks in a locus or between loci.

- **Single and multi-sourced samples:** Single-sourced samples are those where there is only one individual contributing to the genetic material available. Multi-sourced samples, also known as mixtures, are samples where there are two or more contributors. This factor relates to the NOC in a mixture defined in Section 4.1.

- **Allelic Drop-out:** As mentioned in the previous SWGDAM guideline analysis, this factor relates to situations when there is not enough amplification of an allele, also affecting peak height analysis and generation of probable genotypes.

- **Degradation:** As mentioned in the previous SWGDAM guideline analysis, this factor relates to situations in which part of the DNA material is lost. This factor tends to add complexity to the analysis of peak heights.

- **Inhibition:** PCR inhibitors are any factor which prevents the amplification of nucleic acids through the polymerase chain reaction (PCR). PCR inhibition is the most common cause of amplification failure when sufficient copies of DNA are present.

- **Allelic Drop-in:** As mentioned in the previous SWGDAM guideline analysis, this factor relates to sample contamination, spectral artifacts, and stutter products. They cause a deviation in the allele peak heights and generation of probable genotypes.

4.3 ENFSI

This section analyzes ENFSI's guidelines on best practices and how EuroForMix and LRmix Software follow these directives. Most of the items were present in NIST and SWGDAM items previously stated, showing that DNA Analysis as a field has an established basic set of tests when discussing PGS.

4.3.1 ENFSI-DNA-BPM-01

This document lists best practices for evaluating PG software within a laboratory before its application in casework. But in our case, we focus on the metrics they believe are important during the validation and not necessarily the process within the laboratory. We focus on those metrics because they point out crucial types of samples and characteristics that PG software generally assesses.

- **2-person or 3-person mixtures:** Specific types of mixtures containing two and three contributors respectively. This factor relates to the NOC in a mixture defined in Section 4.1.

- **Mixtures with and without drop-out:** This factor defines the necessity to test samples with and without situations when there is not enough amplification of an allele. This relates to the concept of allele drop-out available at Section 4.2.

- **Mixtures with relatives:** Mixtures with relatives relates to the sharing of common alleles concept available at Section 4.1.

- **Mixtures with unrelated contributors:** This factor relates to the fact that all variations of the mixture should be part of a validation set in PGS. The samples that contemplate this requirement are the ones that don't contain the DNA material of biologically related subjects.

- **Known non-contributors to a mock crime-sample:** Creating mixtures with known contributors, but in casework proportions relate to the concept of casework distribution with known contributors available at Section 4.2.

- **Drop-in probability:** This factor defines the necessity of testing samples with contamination, spectral artifacts, and stutter products. This relates to the concept of allelic drop-in available in Section 4.2.
- **Allele peak height:** As mentioned in Section 4.1, this factor measures the heights of allelic and non-allelic peaks in a DNA sample.
- **Degradation:** This factor relates to the DNA material loss that occurs over time in a sample and is available at Section 4.2.
- **Stutters:** This factor relates to the concept of stutter products available at Section 4.1.
- **Analytical threshold:** This factor relates to the concept of analytical threshold available in Section 4.1.
- **Background noise:** This factor relates to the concept of background noise available in Section 4.1.

Table 4.1: Assessment of DNA Analysis standards compliance for the five open-source PGS introducing papers following NIST, SWGDAM and ENFSI guidelines.

NIST					
Factor	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Peak position	●	●	●	●	●*
Peak morphology	●	●	●	●*	❍
Peak heights	●	●	●	●	❍*
Stutter products	●	●	●	●	❍*
Spectral artifacts	❍	❍	❍	❍	❍
Relative peak heights of allele pairs within locus	●	●	●	●	❍*
Assessing relative peak heights across loci in a DNA profile	●	●	●	●	❍
Baseline noise	●	❍	●	●	●*
Stochastic variation	●	●	●*	●*	●*
Sharing of common alleles	●	●	●	●	●*
Number of contributors	❍*	●	●	●	●*
SWGDAM					

Factor	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Possible genotype combinations	●	●	●	●	❍*
Data suport	●*	●*	●*	●*	●*
Casework distributions with known contributors	●	●	●	●	❍
Source population database disclosure	●	●	●	●	❍
Homozygous and heterozygous typing results	❍	❍	●	●	❍
Mixtures and single sources	●*	●*	●*	●*	❍
Multiple locus profiles	●	●	●	●	●
Mixture ratios	❍	❍	●	●	●*
Minor contributors	❍	❍	●	●	●*
Major contributors	❍	❍	●	●	●*
Degradation	●	❍	●	●	●*
Allelic drop-in	●	●	●	●	●*
Allelic drop-out	●	●	●	●	●*
Audit trails	❍	❍	❍*	❍*	❍
Publication of scientific principles	●	●	●	●	●
Sensitivity tests	●*	●	●	●	❍*
Specificity tests	●*	●	●	●	❍*
Precision tests	●*	●	●	●	❍*
Accuracy tests	●	●	●	●	❍*
Inhibition	❍	❍	●	❍	❍
			ENFSI		
Factor	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Mixtures with unrelated contributors	●	●	●	●	●*

4.4 IEEE

The IEEE standard, besides defining software integrity level, divides the verification and validation process into activities and those activities have tasks. IEEE defines a system as level 4 (high integrity) if the behavior of the system, in combination with its environment, causes (1) catastrophic consequences for which the likelihood of the behavior occurring is at most occasional or if this behavior causes (2) critical consequences for which the likelihood of the behavior occurring is at most probable. The definition of catastrophic and critical consequences is at least when a major and permanent injury, partial loss of mission, major system damage, or major financial or social loss can happen when using the system.

We define below the activities and the tasks that IEEE standard defines as mandatory for a level 4 system, and later we assess if the introducing papers perform these activities as expected. The assessment was performed by reading the introducing papers and looking for references that could resemble the task's deliverables, for example if the task defined that a given document should be part of its complete delivery we assesed if at least the content of this document was present in the paper.

4.4.1 Common Management Process

The activities and tasks for the Common Management Process are explained in this subsection.The processes follow the IEEE 15288:2015 standard [61].

- **V&V Management (Common):** The purpose of V&V Management activity is to build, sustain, determine, and direct the V&V plan and efforts to satisfy the project's technical goals. This activity is responsible for the development and revision of the Verification and Validation Plan (VVP) and allowing the stakeholders to identify possible process improvements. It is important to reinforce that the management activities of this process must focus on V&V.

- **Acquisition Support V&V process:** This activity ensures that the system follows the Acquisition process defined by IEEE 15288:2015 [61]. The acquisition process determines the steps the system must abide by to obtain a product or service following the acquirer's requirements. The Acquisition Support V&V Process oversees the implementation of interfaces planned with the acquirer, the system requirements stated in the Request for Proposal (RFP) inclusion, and supplies reports/results to support the system acceptance by the acquirer.

- **Supply Planning V&V process:** This activity ensures that the system follows the Supply process defined by IEEE 15288:2015 [61]. The acquisition process determines the steps the system must abide by to provide a product or service following the agreed requirements. The Supply Planning V&V Process oversees the implementation of interfaces planned with the acquirer, and the inclusion of system requirements stated in the Request for Proposal (RFP).

- **Project Planning V&V process:** The Project Planning V&V Process assures that the project scope is complete, as well as the definition of its activities. The Supply Planning V&V Process delivers a VVP synchronized with the Project Plan.

- **Configuration Management V&V process:** The Configuration Management V&V Process secures that the system's configuration management supports verification and validation activities as established by IEEE 15288:2015 [61]

4.4.2 Verification and Validation Process

The activities and tasks for the Verification process are explained in this subsection. The processes follow the IEEE 15288:2015 [41].

- **Business or Mission Analysis V&V process:** The Business or Mission Analysis V&V process assures that the outcomes of the Business or Mission Analysis process are achieved. In a successful implementation of this process, objective evidence is developed to assess whether the organizational strategy and the opportunity space align, the consistency of the solution space with the problem space, the initial life cycle compatible with the preferred solution space, any necessary systems or services are available for business or mission analysis, and the root of traceability is established from the organizational level to the solution itself.

- **Stakeholder Needs and Requirements Definition V&V process:** The purpose of this process is to assure that the outcomes of the Stakeholder Needs and Requirements Definition process are accomplished. As a result of its successful implementation, the generation of evidence that assesses if the stakeholder requirements specify the characteristics and context of the use of services and operational concepts as required. This evidence must also define all constraints on a system solution, be traceable to the original stakeholders, be complete, unambiguous, correct, and accurate, and be valid by measurable analyses or tests.

- **System Requirements Definition V&V process:** The purpose of the System Requirements Definition V&V process is to assure that the outcomes of the System Requirements Definition process have been achieved. This process should provide evidence that specifies all required characteristics, functional and performance requirements, interface requirements, and requirements for qualification, safety and security, human factors engineering, and user documentation for the system. This evidence must determine all constraints that will affect the architecture of the system and the means to realize it. The system requirements must be unique, complete, unambiguous, consistent with all other prerequisites, implementable, verifiable, and traceable to the stakeholder requirements. This process provides a basis for verifying that each system requirement can be satisfied.

- **Architecture Definition V&V process:** The purpose of this process is to assure that the outcomes of the Architecture Definition process are provided. As a result of the successful implementation of the process assess if the system architecture (i.e., hardware, software, interfaces, and communication) satisfies the system requirements and its feasibility. It also verifies if the architecture is based on specified selection criteria, the basic definition of verification of system elements is defined, and the establishment of the basis for the integration of system elements.

- **Design Definition V&V process:** The purpose of the Design Definition V&V process is to guarantee that the outcomes of the Design Definition process are achieved. As a result of the implementation of this process, we can assess the definition of the design characteristics of each system element and its enablers, the consolidation of interfaces between system elements, and the establishment of the system design. In this phase, we also identify and make available any systems required for the design definition activities, and we establish the traceability of the design characteristics to the architectural elements of the system architecture.

- **System Analysis V&V process:** The purpose of this process is to assure that the outcomes of the System Analysis process are accomplished. As a result, we assess whether the strategy for the system analysis is complete and appropriate for it's importance, and if this analysis support's the conclusions and recommendations of the system.

- **Implementation V&V process:** The purpose of the Implementation V&V process is to assure that the outcomes of the Implementation process have been achieved. As a result of the success of this process, we can assess whether the implementation follows the requirements and design definitions for the product, given the complete recorded evidence of the implementation process.

- **Integration V&V process:** The purpose of the Integration V&V process is to assure the successful implementation of the Integration process. As a result of this process, we assess the conformance of system elements to the defined architectural and design definitions and verify if the implementation and integration are as expected. This process provides evidence of whether the integrated system meets the pre-established requirements, and if this integration strategy is consistent with the system architecture. Human integration is also evaluated by this process and non-conforming integration actions are registered and reported at the end of this process. It is mandatory that the test plan is updated to verify the completeness and quality of the integration process.

- **Transition V&V process:** The purpose of the Transition V&V process is to achieve the outcomes of the Transition process. After implementing this process, we should have a comprehensive and explicitly documented transition strategy for the system. The system must be in its designed operational location, deliver all specified services per its governing requirements, and be sustainable by enabling systems.

- **Operation V&V process:** The purpose of the Operation V&V process is to assure the outcomes of the Operation process. As a result of the successful implementation of the Operation V&V process, we obtain evidence to assess whether the operation strategy satisfies stakeholder requirements and needs.

- **Maintenance V&V process:** The purpose of this process is to assure that the outcomes of the Maintenance process have been achieved. As a result, we obtain objective evidence to assess whether the maintenance strategy is comprehensive and explicitly documented, whether corrective actions resolve the issue or negative impact, and the evaluation of the effect of corrective, adaptive, perfective, and preventive changes on stakeholder and system requirements, architecture, design, and implementation. This evaluation occurs on problem reports, fixing actions, and trends.

- **Disposal V&V process:** The purpose of the Disposal V&V process is to guarantee that the outcomes of the Disposal process are achieved. As a result of the implementation of the Disposal V&V process, we can assess whether the Disposal plan defines system boundaries and identifies system elements, is commensurate with the complexity and risk of the disposal, and accounts for all system elements. And last but not least, we verify if this disposal plan addresses environmental considerations, applicable laws, regulations, and organizational policies and procedures.

4.4.3 Software Verification and Validation Process

The activities and tasks for the Software Verification process are explained in this subsection. The processes follow the IEEE 15288:2015 [41].

- **Software Concept V&V process:** The purpose of the Software Concept V&V process is to assure that the Software Requirements Analysis process is completed. As a result, it is possible to assess whether the allocation of system requirements, the solution satisfies the software requirements, and whether there is no incorporation of false assumptions into the solution.

- **Software Requirements Analysis V&V process:** The purpose of this process is to assure that outcomes of the Software Requirements Analysis process, the Software Qualification Testing process, and the Software Acceptance Support process are accomplished. The successful implementation of this process results in evidence that assesses the correctness, completeness, accuracy, testability, and consistency of software requirements.

- **Software Design V&V process:** The purpose of this process is to guarantee that outcomes of the Software Architectural Design process, the Software Detailed Design process, the Software Integration process, the Software Qualification Testing process, and the Software Acceptance Support process are accomplished. As a result, it is possible to assess whether the software design is correct, accurate, compliant, and a complete transformation of the software requirements.

- **Software Construction V&V process:** The purpose of the Software Construction V&V process is to assure that outcomes of the Software Construction process, the Software Integration process, the Software Qualification Testing process, and the Software Acceptance Support process are achieved. As a result, it is possible to assess whether the transformations from the software design into code, database structures, and related machine-executable representation are correct, accurate, and complete.

- **Software Integration V&V process:** This process guarantees that the Software Integration process is successfully implemented. As a result, it is possible to validate software and system requirements during the implementation of each software component.

- **Software Qualification Testing V&V process:** The purpose of the Software Qualification Testing V&V process is to assure that the outcomes of the Software Qualification Testing process are achieved. As a result, it is possible to assess whether the integrated software product satisfies its requirements.

- **Software Acceptance Testing V&V process:** This process assures the complete implementation of the Software Acceptance Support process and the Software Operation process As a result, it is possible to assess whether the software satisfies its acceptance criteria, and the customer can determine whether or not to accept the integrated software product.

- **Software Installation and Checkout V&V process:** The purpose of the Software Installation and Checkout V&V process is to assure that outcomes of the Software Installation process and the Software Acceptance Support process are achieved. As a result of the successful implementation of the Software Installation and Checkout V&V process, it is possible to assess whether the software installation in the target environment is correct.

- **Software Operation V&V process:** This process assures that outcomes of the Software Operation process are achieved. As a result, it is possible to verify the evaluation of new constraints in the system, proposed system changes and their impacts on the software, and correctness and usability aspects for operating procedures.

- **Software Maintenance V&V process:** This process guarantees that outcomes of the Software Maintenance process are achieved. As a result, it is possible to evaluate software changes and their impact on the system. It also allows the assessment of operational anomalies, migration requirements, and retirement requirements. V&V tasks are re-performed.

- **Software Disposal V&V process:** This process aims to assure that the Software Disposal process is completed. As a result, it is possible to assess the inclusion in software requirements of constraints in the software disposal strategy and whether disposal leaves the system in an agreed-on state.

4.4.4 Hardware validation

As PG tools have interfaces with the machines that provide the electropherograms and even the DNA collection kits, it would be reasonable to perform the V&V tasks related to hardware, but as this machinery can vary a lot from laboratory to laboratory, we believe that this step of validation should be performed in a laboratory implementation level.

Table 4.2: Assessment of IEEE standards compliance for the five open-source PGS introducing papers.

Activity: V&V Management (Common)					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
VVP Generation	❍*	❍*	❍*	❍*	❍
Interface With Other Processes	●*	●*	●*	●*	❍
Proposed/Baseline Change Assessment	❍*	❍*	❍*	❍*	❍
Management Review of the V&V Effort	❍*	❍*	❍*	❍*	❍
Management and Technical Review Support	❍	❍	❍	❍	❍
Identify Process Improvement Opportunities in the Conduct of V&V	❍	❍	❍	❍	❍
V&V Final Report Generation	●*	●*	●*	●*	●*
Activity: Acquisition Support V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Scoping the V&V Effort	❍	❍	❍	❍	❍
Planning the Interface between the V&V Effort and Supplier	❍	❍	❍	❍	❍
System Requirements Review	❍	❍	❍	❍	❍
Acceptance Support	❍	❍	❍	❍	❍

Activity: Supply Planning V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Planning the Interface between the V&V Effort and Supplier	❍	❍	❍	❍	❍
Contract Verification	❍	❍	❍	❍	❍
Activity: Project Planning V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Project Planning Strategy Assessment	●*	●*	●*	●*	●*
Activity: Configuration Management V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Configuration Management Assessment	●*	●*	●*	●*	❍
Activity: Business or Mission Analysis V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Business or Mission Analysis Results Evaluation	●	●	●	●	●
Traceability Analysis	●	●	●	●	●
Criticality Analysis	❍	❍	❍	❍	❍
Hazard Analysis	❍*	❍*	❍*	❍	❍
Security Analysis	❍	❍	❍*	❍	❍
Risk Analysis	❍*	❍*	❍*	❍*	❍*
Activity: Stakeholder Needs and Requirements Definition V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Stakeholder Needs and Requirements Evaluation	❍	❍	❍	❍	❍
Traceability Analysis	❍	❍	❍	❍	❍
Criticality Analysis	❍	❍	❍	❍	❍

Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Activity: System Requirements Definition V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Requirements Evaluation	❍	❍	❍	❍	❍
Interface Analysis	❍	❍	❍	❍	❍
Traceability Analysis	❍	❍	❍	❍	❍
Criticality Analysis	❍	❍	❍	❍	❍
System Integration Test Plan V&V	❍	❍	❍	❍	❍
System Qualification Test Plan V&V	❍	❍	❍	❍	❍
System Acceptance Test Plan V&V	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Activity: Architecture definition V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Architecture Evaluation	●*	●*	●*	●*	●*
Interface Analysis	●	●	●	●	●
Requirements Allocation Analysis	●	●	●	●	●
Traceability Analysis	●*	●*	●*	●*	●*
Criticality Analysis	❍	❍	❍	❍	❍
System Integration Test Design V&V	❍	❍	❍	❍	❍
System Qualification Test Design V&V	❍	❍	❍	❍	❍

System Acceptance Test Design V&V	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Activity: Design Definition V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Design Evaluation	❍	❍	❍	❍	❍*
Interface Analysis	❍	❍	❍	❍	❍
Traceability Analysis	❍	❍	❍	❍	❍*
Criticality Analysis	❍	❍	❍	❍	❍
System Integration Test Case V&V	❍	❍	❍	❍	❍
System Qualification Test Case V&V	❍	❍	❍	❍	❍
System Acceptance Test Case V&V	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Activity: System analysis V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
System Analysis Strategy Evaluation	❍	❍	❍	❍	❍
System Analysis Results Evaluation	❍	❍	❍	❍	❍
Activity: Implementation V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Implementation Strategy Assessment	●*	●*	●*	●*	●*

System Element Implementation Analysis	❍	❍	❍	❍	❍
System Element Interaction Analysis	❍	❍	❍	❍	❍
Criticality Analysis	❍	❍	❍	❍	❍
System Integration Test Procedure V&V	❍	❍	❍	❍	❍
System Qualification Test Procedure V&V	❍	❍	❍	❍	❍
System Acceptance Test Procedure V&V	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Activity: Integration V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
System Integration Strategy Assessment	●	●	●	●	●
System Integration Test Execution V&V	❍	❍	❍	❍	❍
System Element Interaction Analysis	❍	❍	❍	❍	❍
System Qualification Test Execution V&V	❍	❍	❍	❍	❍
Activity: Transition V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Transition Strategy Evaluation	❍	❍	❍	❍	❍
Transition Demonstration Assessment	❍	❍	❍	❍	❍

System Acceptance Test Execution V&V	❍	❍	❍	❍	❍
Activity: Operation V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Operating Procedures Evaluation	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Activity: Maintenance V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
System Maintenance Strategy Assessment	❍	❍	❍	❍	❍
System Maintenance Execution Assessment	❍	❍	❍	❍	❍
Activity: Disposal V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Disposal Plan Evaluation	❍	❍	❍	❍	❍
Activity: Software Concept V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Concept Documentation Evaluation	❍	❍	❍	❍	❍
Requirements Allocation Analysis	❍	❍	❍	❍	❍
Traceability Analysis	❍	❍	❍	❍	❍
Criticality Analysis	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Activity: Software Requirements Analysis V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**

Requirements Evaluation	❍	❍	❍	❍	❍
Interface Analysis	❍	❍	❍	❍	❍
Traceability Analysis	❍	❍	❍	❍	❍
Criticality Analysis	❍	❍	❍	❍	❍
Software Qualification Test Plan V&V	❍	❍	❍	❍	❍
Software Acceptance Test Plan V&V	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Activity: Software Design V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Design Evaluation	❍	❍	❍	❍	❍
Interface Analysis	❍	❍	❍	❍	❍
Traceability Analysis	❍	❍	❍	❍	❍
Criticality Analysis	❍	❍	❍	❍	❍
Software Component Test Plan V&V	❍	❍	❍	❍	❍
Software Integration Test Plan V&V	❍	❍	❍	❍	❍
Software Component Test Design V&V	❍	❍	❍	❍	❍
Software Integration Test Design V&V	❍	❍	❍	❍	❍
Software Qualification Test Design V&V	❍	❍	❍	❍	❍
Software Acceptance Test Design V&V	❍	❍	❍	❍	❍

Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Activity: Software Construction V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Source Code and Source Code Documentation Evaluation	❍	❍	❍	❍	❍
Interface Analysis	❍	❍	❍	❍	❍
Traceability Analysis	❍	❍	❍	❍	❍
Criticality Analysis	❍	❍	❍	❍	❍
Software Component Test Case V&V	●*	●*	●*	●*	❍
Software Integration Test Case V&V	●*	●*	●*	●*	❍
Software Qualification Test Case V&V	●*	●*	●*	●*	❍
Software Acceptance Test Case V&V	●*	●*	●*	●*	❍
Software Component Test Procedure V&V	❍	❍	❍	❍	❍
Software Integration Test Procedure V&V	❍	❍	❍	❍	❍
Software Qualification Test Procedure V&V	❍	❍	❍	❍	❍
Software Component Test Execution V&V	●*	●*	●*	●*	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍

Activity: Software Integration V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Software Integration Test Execution V&V	●*	●*	●*	●*	❍
Traceability Analysis	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍

Activity: Software Qualification Testing V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Software Qualification Test Execution V&V	●*	●*	●*	●*	❍
Traceability Analysis	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍

Activity: Software Acceptance Testing V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Software Acceptance Test Procedure V&V	❍	❍	❍	❍	❍
Software Acceptance Test Execution V&V	●*	●*	●*	●*	❍
Traceability Analysis	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍

Activity: Software Installation and Checkout V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Installation Configuration Audit	❍	❍	●*	❍	●*
Installation Checkout	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍

Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Activity: Software Operation V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Evaluation of New Constraints	❍	❍	❍	❍	❍
Operating Procedures Evaluation	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Activity: Software Maintenance V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
VVP Revision	❍	❍	❍	❍	❍
Anomaly Evaluation	❍	❍	❍	❍	❍
Criticality Analysis	❍	❍	❍	❍	❍
Migration Assessment	❍	❍	❍	❍	❍
Retirement Assessment	❍	❍	❍	❍	❍
Hazard Analysis	❍	❍	❍	❍	❍
Security Analysis	❍	❍	❍	❍	❍
Risk Analysis	❍	❍	❍	❍	❍
Task Iteration	❍	❍	❍	❍	❍
Activity: Software Disposal V&V process					
Task	**EuroForMix**	**LRmix**	**Kongoh**	**LikeLTD**	**LabRetriever**
Software Disposal Evaluation	❍	❍	❍	❍	❍

4.5 Further explanation of the tabled results

The goal of this section is to explain the reasons behind the true (●), partially true (●*), mostly false (❍*) and false (❍) assessments, when comparing to the standards and guidelines definitions, in the previous sections of this chapter. Here we outline the logical thought followed during the papers evaluation in a way that it can be openly debated.

The first disclaimer we'd like to provide is regarding the amount of "❍*" and "●*" evaluations for LabRetriever. This happens because the paper does not report tests performed using the software described, the publication only cites tests that were performed using the model in which the before mentioned software is based on. Therefore, most of the factors we can't assess for this tool, leading to a bigger percentage of uncertainty and non-compliance.

4.5.1 DNA Analysis evaluations

Reviewing findings on DNA Analysis factors, given the nature of these systems, most of them take into consideration peak position when calculating the likelihood ratios. LabRetriever is the only one that receives a partially true evaluation because there is no direct citation of the use of peak position for the metrics. The peak morphology factor is used to also consider the presence of stutters, therefore the system LikeLTD receives a partial true, given that it openly says that stutters are accounted for, but LabRetriver receives a false because it openly says that it doesn't perform tests regarding stutter, even though they admit that it is a factor that can interfere in DNA Analysis results.

Most of the tools incorporate peak heights in their models, but LabRetriever is not part of that group, even though it is possible to manually insert this information in the software there is no evidence that this factor is tested. The only factor that is not tested by any of the tools is related to spectral artifacts.

EuroForMix fails to thoroughly test the NOC, because the mixtures tested in the paper contain only two contributors, and there is no attempt to run the model using a case with more contributors. All the systems, excluding LRmix and LabRetriever mention and exhibit evidence of tests for baseline noise. Nonetheless, most systems abide by NIST guidelines, but there are still validity gaps (e.g. samples with relative, simulations of spectral artifacts, and bigger exploration of peak morphology).

Data support is true for all the PG tools but with constraints because the data presented can only support the systems under the same circumstances of the tests. Sensitivity, specificity, and precision tests for EuroForMix are only partially valid because the tests were restrained to two person-mixtures. In these cases it is impossible to confirm the validity for other scenarios, but this does not mean that the systems do not work, this means that there is not enough evidence to support that it would work in other scenarios.

EuroForMix and LRmix papers lack information about mixture ratios, minor contributors, major contributors, inhibition, and audit trails. The only papers that attempt to provide an evidence similar to an audit trail are Kongoh and LikeLTD. For LRmix, degradation is marked as false because there is no clear statement that the degradation applied to the samples matches degradation levels in the casework. It is possible to see in all papers the use of mixtures with unrelated contributors, and this validation is important because a considerable part of the court cases involve at least one victim and a perpetrator who, in general, have no biological relation.

4.5.2 Software Engineering evaluations

Differently from what we observed for DNA Analysis guidelines, all systems fail to comply with most of the requirements of the IEEE Standard for System, Software, and Hardware Validation and Verification.

Of the *Common V&V Management* activities, only the Final Report Generation is a deliverable for the five tools. In this scenario, the Final Report would be the introduction paper, given that the details of the tests performed are available there. But if we analyze what should be a part of this document, none of the systems would have delivered a complete final report. Another concerning fact is that the VVP generation, one of the most vital tasks in the Validation and Verification process, is not even created. This gives margin to the non-compliance of basic DNA Analysis factor tests because, as there is no clear statement of basic tests from the beginning, they could be forgotten during the validation.

The Interface with Other Processes, we consider the DNA collection process as a part of this process on EuroForMix, LRmix, Kongoh, and LikeLTD. The tools before mentioned perform a series of tests to prove the validity and fitness of their solutions to the DNA Analysis problem, and to do so they collect samples in various ways. LabRetriever on the other hand, does not perform tests in this introduction paper, it only makes a reference to trials realized over the model that it is based on.

Acquisition Support and Supply Planning activities are a bit abstract when focusing on Probabilistic Genotyping Software, but given the integrity level of the system, they are mandatory. Given the definition of the Supply Planning activity, it would be correct to assume that all validation factors are part of the product's requirement, meaning the non-deliverance or uncertainty of accurate deliverance of one of them would be a deviation from the standard.

As we are analyzing the introducing paper, in theory there are no visible adjustments or proposed changes in the papers, nevertheless EuroForMix, LRmix, Kongoh, and LikeLTD prepare a basic unit testing suite that could help them assess some parts of the deliverables related to the Proposed Baseline Change Assessment task. With this unit test suite they could, to a certain extent, verify if changes have a negative impact in the software and due to that we add a mostly false assessment to them. Regarding LabRetriever, the developers go on the opposite direction in this sense. Given that the code presented in this papers is already a modification of a different model, they should propose change assessment as structured as possible.

As for the Management Review of the V&V Effort task, the existence of a VVP is important and as none of the software really delivers one it is harder to assess the compliance to this task. However if we take into account that the simple description of the tests performed are a base for a VVP, we could say that EuroForMix, LRmix, Kongoh, and LikelTD have a document to use as a source to this management analysis. However, besides the reiteration that some factors can induce more complexity to a sample analysis, there is no risk evaluation on how negatively those factors can impact the software. They summarize the V&V effort and have results that can guarantee that the software works under specific circumstances, but the remaining parts of the task are not accomplished. LabRetriever, on the other hand fails completely because in this case there is not even a base for a VVP given that the tests are performed in different papers using a different model.

From the point of view of the Acquisition Support, we must define the final user of the system. If we consider only the level of forensic technicians, the interface of the PG tools provides enough information for them to make a final report to the tribunal. But if we consider that attorneys should also be able to comprehend the results to contest them in favor of their client, the means to do so should also be available. From this optics and considering a bigger context, PGS fails to provide means to exercise the right to confront your accuser [51], in this case, the Probabilistic Genotyping Systems.

All systems receive a "●*" for the Project Planning V&V process because there is no formal declaration of the VVP, as mentioned before. It is evident that a form of project planning exists because no one develops a complex system without it. Nevertheless, in these cases, the planning lacked vital steps of the process, especially those regarding verification and validation. It might seem counterintuitive that, we assigned "●*" for the Configuration Management V&V for EuroForMix, LRmix, Kongoh and LikeLTD. This happens because, even though there is no transparent and previously planned VVP, the system provides the platform for the necessary tests. We reinforce that the systems did not perform all the DNA Analysis basic validation tests, but the structure was available to do so. LabRetriever is the only system left out because there is no evidence that this platform is available for basic testing.

Probabilistic Genotyping Software development happens due to an opportunity space created by the necessity of supplying evidence in a court of law. Given that the entire organizational structure behind this software category spins around that vacuum, it is reasonable to say that the "Business or Mission Analysis Results and Evaluation" and the "Traceability Analysis" are complete. But when the remaining tasks of the Business or Mission Analysis V&V activity are under analysis, it is noticeable that elemental requirements are left aside. In this activity, companies or groups developing PGS must state integrity levels, potential hazards, and security analysis, but none of those exist in the introduction validation papers evaluated or in their GitHub repositories.

As we are dealing with highly sensitive data, there is nothing more personal than your DNA, the security assessment should be more extensive. Kongoh is the only platform that openly defines how the data is being kept and how is the process of acquiring it. But Given the level of criticality of this system they all should perform a better analysis on this matter. In this scenario we are talking only about the data used for validation studies, as there is no guarantee nor description of security procedures for data treatment during the use of the software.

All five software fail to complete the Stakeholder Needs and Requirements Definition V&V process. Keeping in mind that this dissertation sees all DNA Analysis validation factors as system and stakeholders requirements, all marked as "false" configure a lack of definition completeness, correctness, consistency, and testability. The readability component is suitable for the forensic technician stakeholder category; however, both tools disregard attorneys, judges, and others related to the law field as stakeholders. The System Requirements Definition V&V and the Design Definition V&V processes undergo the same fail criteria, because any activity that demands a review or analysis of functional system requirements is doomed to the same result due to the missing DNA Analysis factors.

Something that is also a common issue for all tools is the lack of proper documentation. While they have user manuals that are also important when we think about the broader distribution of said systems, there is no documentation on requirements, test plans, or the development process. For high-integrity systems, this documentation process should be detailed; otherwise, this opens a breach for validity issues. Imagine that aircraft constructors fail to provide documentation of a manufacturing process, and a critical failure occurs in that aircraft. The implications of this lack of documentation could be catastrophic for the company; and the same thing should happen in cases where software inflicts harm to people.

Regarding the Architecture Definition V&V process, the systems comply with everything related to architectural needs but fail again to provide the documentation that states the test planning. All test design tasks aren't completed as expected at this integrity level. Another process in which issues in implementation exist is the System Analysis V&V because this activity is a step where the entire planning and all the assumed conditions and requirements are evaluated.

One process where LabRetriever performs a bit better than the remaining systems is the Design Evaluation V&V. This happens because this is the only publication that has a basic explanation of the interface of the software that is delivered to the final user.Considering the System Implementation V&V activity, the Implementation Strategy Assessment and the System Element Interactions tasks are performed on both tools; however, the documentation lacks the minimal structure defined by IEEE standards. Evaluating both software on the remaining processes, as they are also a reiteration of previous failed ones, they are not delivered as they should.

The Integration V&V process is not as successful due to the previously mentioned issues about the acceptance criteria, especially if they involve functional system requirements. Activities such as Transition V&V, Operation V&V, Maintenance V&V, and Disposal V&V are evaluated as unsuccessful in all cases, mainly because the developers failed to provide an environment for stakeholders that want to collaborate with the tools. All systems fail during the build phase, and it is not clear why the tools do not work. The snippets and executable codes available in the repositories are functional; nonetheless, given that the systems are open source, it was expected that anyone could be able to run their code if following the correct instructions.

Now, we dive into less abstract activities and verify if software components abide by the definitions made in the planning/ideation phase. Given the documentation gaps and the obvious conceptual issues, especially regarding system requirements, we'll summarize other attention points and remove our focus on the elementary DNA Analysis validation factors that should, but aren't, complied. Non-functional requirements, aside from compatible OS, are not defined.

In this hands-on moment of a software lifecycle, more than ever, documentation is crucial, especially for open-source systems. All systems fail to deliver the activities defined in Chapter 9 of IEEE 1012-2016 standard [42], and even though they perform unit and component testing, to a given extent, the remaining tasks are forgotten. We say that the tests are executed to a given extent because there is no rigor in the test case definition.

The lack of documentation here might be one of the causes of the reduced community interest in these tools. Given that all cited PGS do not work if not in a previously built version, how could we expect that a thriving community of contributors would rise? One of the related papers [14] states that EuroForMix production bugs prove that there is no positive side of keeping open-source code, but this claim is too generalistic and bases its knowledge on a community of only three contributors.

As mentioned in Chapter 2 there are positive outcomes from using open-source software, especially in a public sphere, where the results from this software can be used as evidence in court. The use of open-source code in these scenarios reinforces the fundamental rights of citizens, by simply allowing them to question and scrutinize tools outputs.

Based on the previous analysis, we state that the five PG tools under evaluation fail to provide complete validation and verification documents. Most of them should be provided due to the basic requirements of the Software Engineering knowledge area and to strengthen their use in court. However, the DNA Analysis factors they do not comply with impose a threat to these tools general use. They provide breaches in validity that could be used to exploit the criminal system.

As mentioned before, the goal of this dissertation is not to completely invalidate PGS but to show that there is ground for improvement. We also don't want to discourage people from other fields from developing their tools; simply we point out that, as there are standards, guidelines, and laws guiding their respective fields, Software Engineering also has minimal regulations that should be followed.

4.6 Code and community metrics

In terms of community metrics, the PG tools under analysis are similar, and our study will be strictly qualitative. Most of them have only one main contributor, have less than four years, less than eleven open/closed issues, and less than ten forks. The only project that differs from the majority is LabRetriever that has three main contributors, ten years of development, and sixty seven open/closed issues. These factors indicate that the software's community is not extensive or active. Compared to other open-source programs, it is noticeable that PGS has a small group of interested and expert developers.

When we analyze the code itself, the first big difference is the programming language in which the software was developed. LRmix is a Java program divided into the ordinary structure of a *main* and a *test* folder. EuroForMix, is an R program that respects the code organization of a source (`src`) and test folder. Kongoh and LikeLTD are also written in R language, but the folders are not named in a conventional way. LabRetriever is the only system that has a mix of programming languages, having C++ for the *backend* and optimization of the code and Python, JavaScript, and CSS for the *frontend* development.

The intersection of the V&V process with code metrics is represented by functional and risk coverage metrics. Using the factors defined by DNA Analysis expert organizations as a sort of System Requirements definitions we could say that the functional coverage is, on average, 59.38% of the scope of requirements for most tools, and that happens because some of the factors are only partially complied. Regarding risk coverage, as there is no formal description of a Risk Analysis, it is not possible to measure it in the same way. To be fair, all papers perform calculation for both prosecution and defense hypotheses, which could be seen as a way to validate, in known contributors samples, if the system is performing accordingly.

Unit Testing is a process in which the smallest testable parts of an application, called units, are individually evaluated for proper use. This process is ordinary in the industry but generally overlooked during research, but in our case, both PG tools propose a set of unit tests and some end-to-end tests to prove their fitness to casework.

In LRmix, it is possible to find in the folder '`src/test`' the examples of tests performed to identify if the code runs as expected. As LRmix is written in Java, *JUnit* is the framework used for tests. By analyzing the test suite it is possible to confirm that most of the defined requirements for the system are evaluated. Unfortunately due to environment and build issues we couldn't run the code, therefore we can't measure code coverage for this test suite. In Listing 4.1 we can see an example of a unit test developed for this tool.

```java
/**
 *
 * @author dejong
 */
public class AlleleTest {

    public AlleleTest() {
    }

    @BeforeClass
    public static void setUpClass() {
    }

    @AfterClass
    public static void tearDownClass() {
    }

    @Before
    public void setUp() {
    }

    @After
    public void tearDown() {
    }

    /**
     * Test of getAllele method, of class Allele.
     */
    @Test
    public void testGetAllele() {
        System.out.println("getAllele");
        final Allele instance = new Allele("1");
        final String expResult = "1";
        final String result = instance.getAllele();
        assertEquals(expResult, result);
    }
```

Listing 4.1: Example of unit testing with JUnit for LRmix

In EuroForMix, Kongoh and LikeLTD, we also find examples of a Unit Testing using the *testthat* framework and *svUnit* library . Again we can see evidence that the system requirements are covered by test examples, but it is not possible to extract more conclusive metrics regarding test coverage due to build issues. In Listings 4.2, 4.3, and 4.4 we can see examples of a unit test developed for those tools. LabRetriever is the only tool that does not provide evidence of unit testing.

```r
#Testing that the numerical inference of gamma distribution is working
#NEW VERSION FOR v3.4.0
#rm(list=ls());library(euroformix);library(testthat)

#PART 1: beta < 1
n = 30 #number of samples
x = seq(10,300,l=n)
dec = 3 #decimals numbers

set.seed(1)
th = c(1000,0.8,0.6) #true parameters
y=rgamma(n,shape=(2/th[2]^2)*th[3]^((x-125)/100),scale=th[1]*th[2]^2)
#y = c(2434.10045389206,5255.45799208187,4930.54859484562,
   3346.79665881895,4516.40953253601,3482.67267894867,
   3084.75989033535,1849.68456609539,1253.68266225878,
   1319.74306160061,1528.96067970406,1684.7590275416,
   845.986838779711,2453.67407627856,2066.84798535382,
   2342.70302678474,2072.08528478055,1256.09166426717,
   67.6821918881922,1495.59122820608,2022.12297997056,
   459.826219293013,533.707145276646,632.617057156043,
   667.27247237923,321.461122468519,1241.59025314987,
   352.539656179777,1381.81288737032,1022.15471305975)

set.seed(1)
test_that("fitted gamma distr with degrad:", {
  thhat1 = fitgammamodel(y,x)#,offset = 125, scale = 100,plott=T)
  expect_equal(round(thhat1,dec),c(968.935 ,  0.682  , 0.514))
})

set.seed(1)
test_that("fitted gamma distr without degrad :", {
  thhat2 = fitgammamodel(y)#,offset = 125, scale = 100,plott=T)
  expect_equal(round(thhat2,dec),c(932.056, 1.081))
})


#PART 2: beta > 1
th = c(1000,0.8,1.2) #true parameters
set.seed(1)
y=rgamma(n,shape=(2/th[2]^2)*th[3]^((x-125)/100),scale=th[1]*th[2]^2)
#plot(x,y)
set.seed(1)
test_that("fitted gamma distr with degrad (noDEG) :", {
  thhat1 = fitgammamodel(y,x,restrictDeg = TRUE)#,offset = 125, scale =
   100,plott=T)
  expect_equal(round(thhat1,dec),c(1007.097  ,  0.707  , 0.999))
})
```

```r

set.seed(1)
test_that("fitted gamma distr with degrad (noDEG):", {
  thhat2 = fitgammamodel(y,x,restrictDeg = FALSE)#,offset = 125, scale = 100,plott=T)
  expect_equal(round(thhat2,dec),c(1007.097,    0.707, 1.059))
})
```

Listing 4.2: Example of unit testing with Testthat for EuroForMix

```r
test_that("estGamma", {
  peakOneL <- 15
  sizeOneL <- 121.35
  numMc <- 1000
  tempMean <- 7000
  mrOneC <- 0.3
  degOneC <- -0.0025
  sizeMean <- 215.76
  bsrPeakOneL <- 11.1547
  fsrPeakOneL <- 11.1547
  dsrPeakOneL <- 11.1547
  aeParamOneL <- c(0.111, 37.5)
  hbParamOneL <- c(0.0113, 198)
  bsrParamOneL <- c(0.00943, -0.0463, 27.1)
  fsrParamOneL <- c(0.000402, 0.00101, 1996)
  dsrParamOneL <- c(0.000463, -0.00150, 1036)
  m2srParamOneL <- c(0, 0)
  minAE <- 0.058
  minHb <- 0.046
  maxBSR <- 0.7
  maxFSR <- 0.35
  maxDSR <- 0.13
  maxM2SR <- 0.12

  set.seed(1)
  ephData <- estEPH(peakOneL, sizeOneL, numMc, tempMean, mrOneC, degOneC, sizeMean, bsrPeakOneL, fsrPeakOneL, dsrPeakOneL, aeParamOneL, hbParamOneL, bsrParamOneL, fsrParamOneL, dsrParamOneL, m2srParamOneL, minAE, minHb, maxBSR, maxFSR, maxDSR, maxM2SR)
  gammaAl <- estGamma(ephData[[1]])
  gammaBs <- estGamma(ephData[[2]])
  gammaFs <- estGamma(ephData[[3]])
  gammaDs <- estGamma(ephData[[4]])
  gammaM2s <- estGamma(ephData[[5]])

  expect_equal(round(gammaAl[1, 1], 5), 25.86290)
  expect_equal(round(gammaAl[2, 1], 5), 9.67065)
  expect_equal(round(gammaBs[1, 1], 6), 8.388144)
  expect_equal(round(gammaBs[2, 1], 6), 1.829865)
  expect_equal(round(gammaFs[1, 1], 7), 0.2816043)
  expect_equal(round(gammaFs[2, 1], 7), 21.9175423)
  expect_equal(round(gammaDs[1, 1], 7), 0.3040519)
  expect_equal(round(gammaDs[2, 1], 7), 8.6015010)
  expect_equal(gammaM2s[1, 1], 0)
  expect_equal(gammaM2s[2, 1], 0)
})
```

Listing 4.3: Example of unit testing with Testthat for Kongoh

```r
## Test unit ’maximize’
library(svUnit)

###############################################################
# The new two functions are to set up the unit test environment
###############################################################

.setUp <-
function () {
  ## Specific actions for svUnit: prepare context
  if ("package:svUnit" %in% search()) {
    .Log <- Log() ## Make sure .Log is created
    .Log$..Unit <- "inst/unitTests/runit_maximize.R"
    .Log$..File <- ""
    .Log$..Obj <- ""
    .Log$..Tag <- ""
    .Log$..Msg <- ""
    rm(..Test, envir = .Log)
  }
  # Sets threads to 2 or less. This is a CRAN requirement.
  if(.Call(likeLTD::.cpp.nbthreads) > 2) {
    nb_threads_in_test = .Call(likeLTD::.cpp.nbthreads)
    .Call(likeLTD::.cpp.set_nbthreads, as.integer(2))
  }
}

.tearDown <-
function () {
  ## Specific actions for svUnit: clean up context
  if ("package:svUnit" %in% search()) {
    .Log$..Unit <- ""
    .Log$..File <- ""
    .Log$..Obj <- ""
    .Log$..Tag <- ""
    .Log$..Msg <- ""
    rm(..Test, envir = .Log)
  }
  # Reset number of threads to what it was.
  if(’nb_threads_in_test’ %in% ls()) {
    .Call(likeLTD::.cpp.set_nbthreads, as.integer(nb_threads_in_test))
    rm(nb_threads_in_test)
  }
}

###############################################################
# Then data functions
###############################################################

ref.data.path <- function() {
  path = Reduce(file.path, c("extdata", "hammer", "hammer-reference.csv"))
  system.file(path, package="likeLTD")
}
csp.data.path <- function() {
  path = Reduce(file.path, c("extdata", "hammer", "hammer-CSP.csv"))
  system.file(path, package="likeLTD")
}

###################################
# Finally, the unit-test themselves
###################################
test_estimates <- svTest(function() {

  cspProfile = read.csp.profile(csp.data.path())
  knownProfiles = read.known.profiles(ref.data.path())

  if(! "estimates" %in% ls(.GlobalEnv))
    estimates <- getFromNamespace("estimates", "likeLTD")

  checkEquals(array(c(0.575, 0.500)),
              estimates(knownProfiles[1, ], cspProfile))
  checkEquals(array(c(0.625, 0.500)),
              estimates(knownProfiles[2, ], cspProfile))
  checkEquals(array(c(0.750, 0.675)),
              estimates(knownProfiles[3, ], cspProfile))
})
```

Listing 4.4: Example of unit testing with svUnit for LikeLTD

Chapter 5

Conclusion

DNA Analysis is relevant evidence in court, and with this dissertation, we do not aim to diminish its value. We want to point out that from a SE point of view, the tools EuroForMix, LRmix, Kongoh, LikeLTD, and LabRetriever do not abide by international standards, and from a forensic analysis perspective, there is no in-depth investigation of some peculiarities on casework samples, and this opens precedence for issues in court decisions that involve complex mixtures.

During this dissertation, we often do a parallel between Brazil and the U.S., and this happens because it is reasonable to consider the U.S. as a hub of technology development, the U.S. has an influence globally when we think about software/hardware development and due to the research relationship that the author of this dissertation have with UC Berkeley. Given that this publication is an MSc. dissertation at a Brazilian university, we believe it is crucial to adapt the text to reflect the Brazilian reality.

We'd like to reinforce that almost every type of surveillance technology can be used in court, in a way that they would also become evidentiary technologies, but not all the artifacts deemed as evidentiary are used in surveillance, and the example of that, in this dissertation, is PGS.

In previous chapters, we stated a big difference between the adequacy of the five Probabilistic Genotyping Software to the rules defined by organizations linked to the area of DNA Analysis and the rules defined by the IEEE. EuroForMix follows 59.38% of DNA Analysis guidelines (75.00% considering partially covered factors) but only abides by 3.05% of the tasks defined by IEEE standards (4.27% considering partially completed tasks). LRmix follows 65.63% of DNA Analysis guidelines (71.88% considering partially covered factors) but only abides by 3.05% of the tasks defined by IEEE standards (4.27% considering partially completed tasks). Kongoh follows 84.38% of DNA Analysis guidelines (93.75% considering partially covered factors) but only abides by 3.05% of the tasks defined by IEEE standards (4.88% considering partially completed tasks). LikeLTD follows 78.13% of DNA Analysis guidelines (90.63% considering partially covered factors) but only abides by 3.05% of the tasks defined by IEEE standards (4.27% considering partially completed tasks). And LabRetriever follows 9.38% of DNA Analysis guidelines (46.88% considering partially covered factors) but only abides by 3.05% of the tasks defined by IEEE standards (3.66% considering partially completed tasks).

Focusing on DNA Analysis, it is noticeable that there is no scrutiny over complex samples in a way that the analysis doesn't necessarily match the level of complexity of casework mixtures. An example is degraded samples. Most papers mention tests with degradation, but there is no disclosure of its extent, so it is impossible to establish that the systems are suited to this type of sample. When we shift our attention to Software Engineering standards, the scenario is not better since the developers follow only a small amount of the required activities or tasks for a system with this level of criticality or integrity. There is no clear documentation about the requirements, and the verification and validation plan does not exist.

One of the non-abided activities is the Design Evaluation V&V process defined by IEEE standards. One of the many aspects that this activity evaluates is the proper use of truncation and rounding in computational systems [42]. For PGS, as we deal with baseline thresholds and likelihoods, it is expected for rounding and truncating methods to go under analysis. A rounding error could favor a person's conviction when discussing evidentiary and surveillance software, so there should be a clear statement of how this threshold is stipulated, what is a conservative error margin, and this should be easily available and understandable to a non-technical stakeholder (e.g. judges, attorneys, police enforcement). There were many other points where the available documentation on the tools failed SE standards, but the more critical ones are related to a lack of transparency and requirement evaluation alongside diverse stakeholders.

Moving on to the results of code and community metrics, our analysis reinforces the idea that a small cluster of developers is responsible for nurturing these systems, but also lights up some curiosity about the background of their contributors. We do not mean to disregard their contributions, but maybe point out that, from an SE perspective, all PG tools lack information for proper validation and fail to deliver initial documentation of the development process. We hypothesize that if the GitHub communities for those tools were more active, this documentation issue and the lack of exploration of basic stakeholder requirements would not be present. Thinking about the development community, maybe if the knowledge of the existence of these tools was broader and if more people knew about the real impact that they have in our society, it would be possible for ethical hacking communities and even other interested developers to collaborate in the repositories, to bulletproof, as much as possible, the applications. In addition, if more diverse stakeholders were present in the requirement definition process, maybe the vacuums regarding the casework sample variety would be attended to, and even the lack of proper result communication when considering attorneys and judges who might not be familiar with the formal forensic output of the system.

It is known that open-source code has advantages for software development, but when we discuss software that will be used in the public sphere the transparency obtained with open-source code should be mandatory. Linux's Law helps us defend that open-source code is more prone to be comprehensively tested because there are more eyes on it. Especially when we think about technologies that have a reasonable impact on public affairs and individuals' lives, open-source software gives an opening for society itself to state its opinion and try to enhance these systems.

In the Computer Science field, it is desirable to be the first person/group to develop a technology or to make breakthrough advances in some specific model, and sometimes developers don't evaluate the damage that said implementation could cause to our society due to this rush to be an innovator. Auditing and revising the V&V process of computational systems posterior to the software launch is a palliative approach, but this risk evaluation should be part of the process as a whole. Therefore, we claim involving people from multiple backgrounds during the ideation and design phase is necessary. In the same way that many companies understand that it is crucial to have an inclusive and diverse environment to develop new products, in academia and public policies, it is vital to understand the society affected by a given system. It is necessary to assess if this system is adequate in a new environment and if it is fit to solve the specific society problems where they will be applied.

We also stress the assessment of the real necessity for this kind of surveillance technology. By this we mean; is this a real need in our society, or do we as humans resort to these technologies to give us a false sense of security? A hypothesis is that we might develop them to surround ourselves with imaginary fences or protections, and they do protect most of us, but definitely not all. One thing that is important to mention is how public policies and social groups can impose the creation of new technologies. Taking PGS as an example, there is a real need to provide more accurate and strong evidence for the jury, making the existence of this system a fair demand. But when we think about Predictive Policing, the need for this type of technology comes from a subconscious need to feel safe, but does it really protect all of us? But let's quit this more philosophical discussion as it is not the focus here.

An assumption can be made regarding the lack of perception that technological artifacts have real impact, therefore the more "social" side of Software Engineering is generally dismissed. There are no evaluations on potential hazards because people don't see how lines of code could impact the society. We must stop imagining software as a separated sphere of our lives, because it is more integrated than ever and will be more integrated each day.

5.1 Research limitations

This dissertation, as any other research, has limitations. One of our limitations was the access to the proper code to build the applications, this forced us to decrease the scope of our analysis to something more qualitative than quantitative. Even though we study all the available open source PGS, by the time of this research, we are restricted to the evidence the developers provide, therefore they might have performed other SE tasks, but as they don't appear in the available reports we must evaluate them as non-existent.

One other point we'd like to mention is the scope of the project. We focus our analysis on the U.S. and Brazil due to its familiarity to the authors. We reinforce that the U.S. is a hub of technology and it makes this situation more comfortable, but we believe that it is necessary to point out that our scope focuses in countries where we had a stronger opinion on how they deal with this technological matter.

The evaluations performed in this research, even though they are based in the standards and guidelines before mentioned, they are grounded on the interpretation of said guidelines. The authors add the rationale behind the evaluations, so that it would be possible to arrive in similar conclusions in a future analysis of this study, but something that could be performed is the reassessment with multiple evaluators in a way that we could arrive in conclusions with a agreement method. Regarding the proposal for future work, we want to emphasize that, as it wasn't the focus of this research, the authors did not try to exhaust the literature looking for evidence that those questions were/are not studied by a different research group.

Finally, as we have been reiterating throughout the text, the missing documents and non compliance to the standards analyzed do not invalidate the use of PGS as a whole. Those verification and validation breaches do not necessarily induce an error in the system. They just don't support claims of universal casework well-functioning. To asses such thing we'd need to create a PG tool from ground scratch following all that is stated in the standards and guidelines before mentioned.

5.2 Future Work

The goal of this dissertation is to continue and expand the current discussion around evidentiary software and social issues, but many other opportunities to perform research in this intersection exist. In the upcoming paragraphs, we mention some of them and draw a draft of future research questions that would be valuable to tackle. Therefore, the next topics are more of a provocation than established truths.

Starting with a more practical approach to future research, to expand and enrich the EFF Atlas of Surveillance is one of the paths forward for this research. Considering the Brazilian context, we could officialize a partnership with EFF or create our version of this database.

As governments won't stop demanding new types of surveillance technology, informing the population about its existence and monitoring possible abuse and misuse cases of those technologies is necessary. Developing a platform where citizens could search and report the use of these technologies could be a way to disseminate the threats behind their existence, and that could also help attorneys prepare better defenses against those technologies.

Besides those benefits, entities like The Innocence Project could benefit from more information about possible validation issues in PGS or binary DNA Analysis. As there is no complete regulation of these tools, we must monitor them as a society and not allow them to become an excuse for biased decisions or arrests.

In this dissertation, we focused a lot on how the software validation process should proceed, but another aspect that deserves attention is the quality of the DNA databases used in those validations and the possible ethical issues around them.

Some of the publications that we cite in Section 2.3 are validation studies of Probabilistic Genotyping tools, and in those studies, the systems were evaluated against samples collected within the laboratories or from volunteer donation or by not-so-ethical methods (e.g., blood bank purchases). In the same way that we now have red flags against face-recognition training datasets, we should also raise them against DNA datasets. We need to scrutinize the their assembling process not just to understand possible bias sources (e.g.

population used, quality of the sample), but also to understand if the data is protected as demanded by law. Some questions that ca emerge from this analysis could lead to validity problems in the PGS validation publications, not to mention the ethical issues behind some of them.

Deviating a bit from the validation process, we should also worry about companies assembling their own DNA databases - for instance, ancestrality test companies. These companies, especially in countries with a known enslavement history, thrive on the curiosity and need for ancestral reconnecting. They collect more DNA each day, and at some point, we must question what they are doing with our DNA samples. Even if they keep them in a way to force anonymity, do they share this data with other companies or government agencies?

Replication studies are also welcome in this field. As in multiple science realms, the Reproducibility and Replication crisis is real when discussing PGS validation studies. Dealing with DNA Analysis is an expensive and tricky business. In general, publications do not provide the datasets used, some of the analysis is made over proprietary software, so licenses become an issue, and even when we deal with open source software, some of them are not distributed in a way that it is possible to easily build the application from scratch.

This field lacks standards that force companies and interested researchers to provide all that is necessary to do a proper evaluation of the code and its results in a posterior moment.

With all that said, once again, we say that the goal of this dissertation is not to dismiss or invalidate PG tools. Our purpose here is to show that the standards mentioned are proof that there are breaches in PGS that need to be solved in order to sustain their unquestioned use in court.

Bibliography

[1] Rediet Abebe, Moritz Hardt, Angela Jin, John Miller, Ludwig Schmidt, and Rebecca Wexler. Adversarial scrutiny of evidentiary statistical software. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 1733–1746, New York, NY, USA, 2022. Association for Computing Machinery.

[2] David J. Balding and John Buckleton. Interpreting low template DNA profiles. *Forensic Science International: Genetics*, 4(1):1 – 10, 2009.

[3] Steven M Bellovin, Matt Blaze, Susan Landau, and Brian Owsley. Seeking the source: Criminal defendants' constitutional right to source code. *Ohio St. Tech. LJ*, 17:1, 2021.

[4] CCG Benschop, J Jong, L Merwe, and H Haned. Adapting a likelihood ratio model to enable searching DNA databases with complex STR DNA profiles. *Proceedings of the 27th ISHI, Promega. com*, 2016.

[5] Corina CG Benschop, Jerry Hoogenboom, Pauline Hovers, Martin Slagter, Dennis Kruise, Raymond Parag, Kristy Steensma, Klaas Slooten, Jord HA Nagel, Patrick Dieltjes, et al. DNAxs/DNAstatistx: Development and validation of a software suite for the data management and probabilistic interpretation of DNA profiles. *Forensic Science International: Genetics*, 42:81–89, 2019.

[6] Frederick R Bieber, John S Buckleton, Bruce Budowle, John M Butler, and Michael D Coble. Evaluation of forensic DNA mixture evidence: protocol for evaluation, interpretation, and statistical calculations using the combined probability of inclusion. *BMC genetics*, 17(1):1–15, 2016.

[7] Alex Biedermann, Silvia Bozza, Franco Taroni, and Colin Aitken. Reframing the debate: a question of probability, not of likelihood ratio. *Science & Justice*, 56(5):392–396, 2016.

[8] Øyvind Bleka, Geir Storvik, and Peter Gill. Euroformix: An open source software based on a continuous model to evaluate STR DNA profiles from a mixture of contributors with artefacts. *Forensic Science International: Genetics*, 21:35 – 44, 2016. Cited by: 170.

[9] DNA Advisory Board. Statistical and population genetics issues affecting the evaluation of the frequency of occurrence of DNA profiles calculated from pertinent population database (s). *Forensic Science Communications*, 2(3):1–8, 2000.

[10] Brasil. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). *Diário Oficial [da] República Federativa do Brasil*, 2018.

[11] Jo-Anne Bright, Shan-I Lee, John Buckleton, and Duncan Taylor. Revisiting the STRmix™ likelihood ratio probability interval coverage considering multiple factors. *bioRxiv*, pages 2021–06, 2021.

[12] Jo-Anne Bright, Judi Morawitz, Duncan Taylor, and John Buckleton. Regression test of various versions of STRmix, 2022.

[13] Jo-Anne Bright, Duncan Taylor, Catherine McGovern, Stuart Cooper, Laura Russell, Damien Abarno, and John Buckleton. Developmental validation of STRmix™, expert software for the interpretation of forensic DNA profiles. *Forensic Science International: Genetics*, 23:226–239, 2016.

[14] John Buckleton, Jo-Anne Bright, Kevin Cheng, and Duncan Taylor. Probabilistic genotyping code review and testing. *arXiv preprint arXiv:2205.09788*, 2022.

[15] John Buckleton and James Curran. A discussion of the merits of random man not excluded and likelihood ratios. *Forensic Science International: Genetics*, 2(4):343–348, 2008.

[16] J Butler, H Iyer, R Press, MK Taylor, PM Vallone, S Willis, and DNA Mixtur e Interpretation. A nist scientific foundation review (nistir 8351-draft). *National Institute of Standards and Technology, Gaithersburg*, 2021.

[17] John M Butler, Michael D Coble, and Peter M Vallone. STRs vs. snps: thoughts on the future of forensic DNA testing. *Forensic science, medicine, and pathology*, 3:200–205, 2007.

[18] Michael D Coble and Jo-Anne Bright. Probabilistic genotyping software: an overview. *Forensic Science International: Genetics*, 38:219–224, 2019.

[19] R CODE. Best practice manual for the internal validation of probabilistic software to undertake DNA mixture interpretation, 2017.

[20] James S Collofello. *Introduction to software verification and validation.* Carnegie Mellon University, Software Engineering Institute, 1988.

[21] BRASIL. Constituição(1988). *Constituição da República Federativa do Brasil promulgada em 5 de outubro de 1988: atualizada até a Emenda Constitucional n. 48, de 10-8-2005 :.* Coleção Saraiva de Legislação. Saraiva,, São Paulo :, 38 ed edition, 2006.

[22] Martin Croxford and Roderick Chapman. Correctness by construction: A manifesto for high-integrity software. *CrossTalk*, 18(12):5 – 8, 2005.

[23] Sacramento County District. Internal validation of STRmix™ v2.4, 2017.

[24] Michael D Edge and Jeanna Neefe Matthews. Open practices in our science and our courtrooms. *Trends in Genetics*, 38(2):113–115, 2022.

[25] ENFSI. Best Practice Manual for Human Forensic Biology and DNA Profiling – ENFSI-DNA-BPM-03. Standard, ENFSI, Wiesbaden, Germany, July 2018.

[26] FBI. Quality assurance standards for forensic DNA testing laboratories. Standard, FBI, Washington DC, USA, September 2011.

[27] SERVIÇO PÚBLICO FEDERAL. XVI Relatório da rede integrada de bancos de perfis genéticos (RIBPG), 2023.

[28] Marcus S. Fisher. *Software verification and validation: An engineering and scientific approach*. Springer, 2007.

[29] Robert Fjellstrom, Jeffrey Steiner, and Paul Beuselinck. Tetrasomic linkage mapping of rflp, pcr, and isozyme loci in l. *Crop Science*, 43:1580, 07 2003.

[30] Paolo Garofano, Denise Caneparo, Giuseppina D'Amico, Marco Vincenti, and Eugenio Alladio. An alternative application of the consensus method to DNA typing interpretation for low template-DNA mixtures. *Forensic Science International: Genetics Supplement Series*, 5:e422–e424, 2015.

[31] N Georgiou, RM Morgan, and JC French. The shifting narrative of uncertainty: a case for the coherent and consistent consideration of uncertainty in forensic science. *Australian Journal of Forensic Sciences*, pages 1–17, 2022.

[32] P Gill and H Haned. A new methodological framework to interpret complex DNA profiles using likelihood ratios. *Forensic Science International: Genetics*, 7(2):251–263, 2013.

[33] Peter Gill, Colin P Kimpton, Andrew Urquhart, Nicola Oldroyd, Emma S Millican, Stephanie K Watson, and Terry J Downes. Automated short tandem repeat (STR) analysis in forensic casework—a strategy for the future. *Electrophoresis*, 16(1):1543–1552, 1995.

[34] Frank M Götz, Holger Schönborn, Viktoria Borsdorf, Anne-Marie Pflugbeil, and Dirk Labudde. Genoproof mixture 3—new software and process to resolve complex DNA mixtures. *Forensic Science International: Genetics Supplement Series*, 6:e549–e551, 2017.

[35] Susan A Greenspoon, Lisa Schiermeier-Wood, and Brad C Jenkins. Establishing the limits of TrueAllele® casework: A validation study. *Journal of forensic sciences*, 60(5):1263–1276, 2015.

[36] H Haned, T Egeland, D Pontier, L Pene, and P Gill. Estimating drop-out probabilities in forensic DNA samples: a simulation approach to evaluate different models. *Forensic Science International: Genetics*, 5(5):525–531, 2011.

[37] Hinda Haned. Forensim: An open-source initiative for the evaluation of statistical methods in forensic genetics. *Forensic Science International: Genetics*, 5(4):265 – 268, 2011. Cited by: 91.

[38] Les Hatton. *Safer C: Developing Software for in High-Integrity and Safety-Critical Systems*. McGraw-Hill, Inc., USA, 1995.

[39] Joanne Hinds, Emma J. Williams, and Adam N. Joinson. "it wouldn't happen to me": Privacy concerns and perspectives following the cambridge analytica scandal. *International Journal of Human-Computer Studies*, 143:102498, 2020.

[40] Mitchell M Holland, Teresa M Tiedge, Abigail J Bender, Sidney A Gaston-Sanchez, and Jennifer A McElhoe. MaSTR™: an effective probabilistic genotyping tool for interpretation of STR mixtures associated with differentially degraded DNA. *International Journal of Legal Medicine*, 136(2):433–446, 2022.

[41] IEEE. ISO/IEC/IEEE International Standard - Systems and software engineering – Software life cycle processes. Standard, IEEE Computer Society, Los Alamitos, CA, November 2008.

[42] IEEE. IEEE Standard for System, Software, and Hardware Verification and Validation. Standard, IEEE Computer Society, Los Alamitos, CA, May 2017.

[43] Keith Inman, Norah Rudin, Ken Cheng, Chris Robinson, Adam Kirschner, Luke Inman-Semerau, and Kirk E. Lohmueller. Lab retriever: A software tool for calculating likelihood ratios incorporating a probability of drop-out for forensic DNA profiles. *BMC Bioinformatics*, 16(1), 2015. Cited by: 38; All Open Access, Gold Open Access, Green Open Access.

[44] Tim Kalafut, Simone Pugh, Peter Gill, Sarah Abbas, Marie Semaan, Issam Mansour, James Curran, Jo-Anne Bright, Tacha Hicks, Richard Wivell, et al. A mixed DNA profile controversy revisited. *Journal of Forensic Sciences*, 67(1):128–135, 2022.

[45] Tim Kalafut, Curt Schuerman, Joel Sutton, Tom Faris, Luigi Armogida, Jo-Anne Bright, John Buckleton, and Duncan Taylor. Implementation and validation of an improved allele specific stutter filtering method for electropherogram interpretation. *Forensic Science International: Genetics*, 35:50–56, 2018.

[46] Dan E Krane. Evaluating forensic DNA evidence: essential elements of a competent defense review, 2004.

[47] ES Lander, LM Linton, and B Birren. Initial sequencing and analysis of the human genome [published correction appears in nature. 2001; 411 (6838): 720]. *Nature*, 409(6822):860–921, 2001.

[48] Steven P Lund and Hari Iyer. Likelihood ratio as weight of forensic evidence: a closer look. *Journal of Research of the National Institute of Standards and Technology*, 122:1, 2017.

[49] Sho Manabe, Chie Morimoto, Yuya Hamano, Shuntaro Fujimoto, and Keiji Tamaki. Development and validation of open-source software for DNA mixture interpretation based on a quantitative continuous model. *PLoS One*, 12(11):e0188183, 2017.

[50] Sho Manabe, Chie Morimoto, Yuya Hamano, Shuntaro Fujimoto, and Keiji Tamaki. Development and validation of open-source software for DNA mixture interpretation based on a quantitative continuous model. *PLoS ONE*, 12(11), 2017. Cited by: 37; All Open Access, Gold Open Access, Green Open Access.

[51] Jeanna Matthews, Marzieh Babaeianjelodar, Stephen Lorenz, Abigail Matthews, Mariama Njie, Nathaniel Adams, Dan Krane, Jessica Goldthwaite, and Clinton Hughes. The right to confront your accusers: Opening the black box of forensic DNA software. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 321–327, 2019.

[52] Geoffrey Stewart Morrison. In the context of forensic casework, are there meaningful metrics of the degree of calibration? *Forensic Science International: Synergy*, 3:100157, 2021.

[53] Safiya Umoja Noble. *Algorithms of Oppression: How Search Engines Reinforce Racism.* NYU Press, 2018.

[54] Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy.* Crown Publishing Group, USA, 2016.

[55] Sundararajulu Panneerchelvam and Mohd Nor Norazmi. Forensic DNA profiling and database. *The Malaysian journal of medical sciences : MJMS*, 10 2:20–6, 2003.

[56] Vincenzo L Pascali. A novel computational strategy to predict the value of the evidence in the snp-based forensic mixtures. *Plos one*, 16(10):e0247344, 2021.

[57] Mark W Perlin, Jamie L Belrose, and Barry W Duceman. New york state true allele® casework validation study. *Journal of Forensic Sciences*, 58(6):1458–1466, 2013.

[58] Inioluwa Deborah Raji, Timnit Gebru, Margaret Mitchell, Joy Buolamwini, Joonseok Lee, and Emily Denton. Saving face: Investigating the ethical concerns of facial recognition auditing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 145–151, 2020.

[59] Eric Raymond. The cathedral and the bazaar. *Knowledge, Technology & Policy*, 12(3):23–49, 1999.

[60] Janette W Redman and Margaret C Kline. Allele frequencies for 15 autosomal STR loci on us caucasian, african american, and hispanic populations. *J Forensic Sci*, 48(4), 2003.

[61] IEEE Computer Society. ISO/IEC/IEEE International Standard - Systems and software engineering – System life cycle processes. Standard, IEEE Computer Society, Los Alamitos, CA, May 2015.

[62] S. Spielberg, Gerald R. Molen, Bonnie Curtis, Walter F. Parkes, and Jan de Bont. Minority report, 2002.

[63] Christopher D Steele and David J Balding. Statistical evaluation of forensic DNA profile evidence. *Annual Review of Statistics and Its Application*, 1:361–384, 2014.

[64] Christopher D. Steele, Matthew Greenhalgh, and David J. Balding. Evaluation of low-template DNA profiles using peak heights. *Statistical Applications in Genetics and Molecular Biology*, 15(5):431–445, 2016.

[65] Harish Swaminathan, Muhammad O Qureshi, Catherine M Grgicak, Ken Duffy, and Desmond S Lun. Four model variants within a continuous forensic DNA mixture interpretation framework: Effects on evidential inference and reporting. *PloS one*, 13(11):e0207599, 2018.

[66] SWGDAM. Guidelines for the validation of probabilistic genotyping systems. Standard, Scientific Working Group on DNA Analysis Methods, US, June 2015.

[67] SWGDAM. SWGDAM Guidelines for the Validation of Probabilistic Genotyping Systems. Standard, SWGDAM, Washington DC, USA, May 2015.

[68] SWGDAM. Swgdam interpretation guidelines for autosomal STR typing by forensic DNA testing laboratories. Standard, Scientific Working Group on DNA Analysis Methods, US, January 2017.

[69] SWGDAM. Addendum to swgdam interpretation guidelines for autosomal STR typing by forensic DNA testing laboratories to address next generation sequencing. Standard, Scientific Working Group on DNA Analysis Methods, US, April 2019.

[70] William C Thompson, Simon Ford, Travis Doom, Michael Raymer, and Dan Krane. Evaluating forensic DNA evidence: essential elements of a competent defense review. *The Champion*, 27(16):16–25, 2003.

[71] US. Andrews vs. state. In *533 So. 2d 841 (Fla. Dist. Ct. App. 1988)*, 1988.

[72] D.R. Wallace and R.U. Fujii. Software verification and validation: an overview. *IEEE Software*, 6(3):10–17, 1989.

[73] Bruce S Weir, CM Triggs, L Starling, LI Stowell, KAJ Walsh, and J Buckleton. Interpreting DNA mixtures. *Journal of Forensic Science*, 42(2):213–222, 1997.

[74] Emily West and Vanessa Meterko. Innocence project: DNA exonerations, 1989-2014: review of data and findings from the first 25 years. *Alb. L. Rev.*, 79:717, 2015.

[75] Matthias Wienroth, Rafaela Granja, Veronika Lipphardt, Emmanuel Nsiah Amoako, and Carole McCartney. Ethics as lived practice. anticipatory capacity and ethical decision-making in forensic genetics. *Genes*, 12(12):1868, 2021.

[76] Shoshana Zuboff. *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. PublicAffairs, 1st edition, 2018.