Forensic Science International: Genetics 25 (2016) 191–197

Contents lists available at ScienceDirect

Forensic Science International: Genetics

journal homepage: www.elsevier.com/locate/fsig

Research paper

DNA Commission of the International Society for Forensic Genetics: Recommendations on the validation of software programs performing biostatistical calculations for forensic genetics applications

M.D. Coble[a,*], J. Buckleton[b,c], J.M. Butler[d], T. Egeland[e], R. Fimmers[f], P. Gill[g,h], L. Gusmão[i,j,k], B. Guttman[l], M. Krawczak[m], N. Morling[n], W. Parson[o,p], N. Pinto[j,k,q,r], P.M. Schneider[s], S.T. Sherry[t], S. Willuweit[u], M. Prinz[v]

[a] National Institute of Standards and Technology, Applied Genetics Group, Gaithersburg, MD, USA
[b] ESR, Private Bag 92021, Auckland 1142, New Zealand
[c] National Institute of Standards and Technology, Statistical Engineering Division (Guest Researcher), Gaithersburg, MD, USA
[d] National Institute of Standards and Technology, Special Programs Office, Gaithersburg, MD, USA
[e] Norwegian University of Life Sciences, Oslo, Norway
[f] Institute for Medical Statistics, Informatics, and Epidemiology, University Bonn, Germany
[g] Norwegian Institute of Public Health, Oslo, Norway
[h] University of Oslo, Oslo, Norway
[i] State University of Rio de Janeiro (UERJ), Rio de Janeiro, Brazil
[j] IPATIMUP, Institute of Molecular Pathology and Immunology of the University of Porto, Portugal
[k] Instituto de Investigação e Inovação em Saúde, University of Porto, Portugal
[l] National Institute of Standards and Technology, Software and Systems Division, Gaithersburg, MD, USA
[m] Institute of Medical Informatics and Statistics, Christian-Albrechts University of Kiel, Germany
[n] Section of Forensic Genetics, Department of Forensic Medicine, Faculty of Health and Medical Sciences, University of Copenhagen, Denmark
[o] Institute of Legal Medicine, Medical University of Innsbruck, Innsbruck, Austria
[p] Forensic Science Program, The Pennsylvania State University, PA, USA
[q] Institute for Research and Innovation in Health (I3S), University of Porto, Porto, Portugal
[r] Centre of Mathematics of the University of Porto, Porto, Portugal
[s] Institute of Legal Medicine, Faculty of Medicine, University of Cologne, Germany
[t] National Center for Biotechnology Information, National Library of Medicine, NIH, Bethesda, MD, USA
[u] Department of Forensic Genetics, Institute of Legal Medicine and Forensic Sciences, Charité—Universitätsmedizin, Berlin, Germany
[v] John Jay College of Criminal Justice, New York, USA

ARTICLE INFO

Article history:
Received 29 August 2016
Accepted 2 September 2016
Available online 4 September 2016

Keywords:
Biostatistical software
Software validation
Validation and verification
Forensic genetics
Software training and testing

ABSTRACT

The use of biostatistical software programs to assist in data interpretation and calculate likelihood ratios is essential to forensic geneticists and part of the daily case work flow for both kinship and DNA identification laboratories. Previous recommendations issued by the DNA Commission of the International Society for Forensic Genetics (ISFG) covered the application of bio-statistical evaluations for STR typing results in identification and kinship cases, and this is now being expanded to provide best practices regarding validation and verification of the software required for these calculations. With larger multiplexes, more complex mixtures, and increasing requests for extended family testing, laboratories are relying more than ever on specific software solutions and sufficient validation, training and extensive documentation are of upmost importance.

Here, we present recommendations for the minimum requirements to validate bio-statistical software to be used in forensic genetics. We distinguish between developmental validation and the responsibilities of the software developer or provider, and the internal validation studies to be performed by the end user. Recommendations for the software provider address, for example, the documentation of the underlying models used by the software, validation data expectations, version control, implementation and training support, as well as continuity and user notifications. For the internal validations the recommendations include: creating a validation plan, requirements for the range

* Corresponding author at: National Institute of Standards and Technology, Applied Genetics Group, 100 Bureau Drive MS 8314, Gaithersburg, MD 20899-8314, USA.
E-mail address: mcoble@nist.gov (M.D. Coble).

http://dx.doi.org/10.1016/j.fsigen.2016.09.002
1872-4973/Published by Elsevier Ireland Ltd.

of samples to be tested, Standard Operating Procedure development, and internal laboratory training and education. To ensure that all laboratories have access to a wide range of samples for validation and training purposes the ISFG DNA commission encourages collaborative studies and public repositories of STR typing results.

Published by Elsevier Ireland Ltd.

1. Introduction

Forensic genetics is experiencing an increase in data volume and complexity, and the interpretation of these data is becoming more and more dependent upon the use of appropriate bio-statistical computer programs. Software for calculating likelihood ratios to evaluate trace evidence or competing kinship scenarios has been in use for many years now, and several groups have described validation exercises of either in-house, open source, or commercial software packages [1–15].

These publications vary notably in terms of the validation approach taken, and standardized reporting of which quality measures were invoked, which tests have been successfully completed, and which software documentation was available. This information is not only of interest to the forensic scientist but also to the legal community. For quality measures, a distinction must be drawn between the responsibility of the software developer or provider, e.g. for code review, version control, documentation of the underlying theory and validation against known data sets, and the responsibility of the end user, e.g. internal validation under local laboratory conditions, formulation of standard operating procedures (SOPs), and training and competency testing.

International industry standards apply to software validation, verification [16] and test documentation [17]. These standards can be simplified and extrapolated [18] to forensic genetics. For internal validation, the goal is similar to other analysis methods: to test the proper function and assess accuracy and limitations of the methods. Previous recommendations on forensic method validation and application of genetic analyses are useful to be read in conjunction with these guidelines [19–25].

The International Society for Forensic Genetics (ISFG) has convened a DNA Commission to establish validation guidelines for bio-statistical software to be used in forensic genetics. Examples include software to calculate statistics for: single-source samples, autosomal DNA mixtures of two or more individuals with no drop-out, or where drop-out and drop-in are possible, paternity and kinship testing, and haploid marker interpretation. The goal of the DNA Commission was to carve out a consensus view on the minimum requirements for the validation (is it doing the right thing?) and verification (is it doing the thing right?) of a software program (V&V) [16] and to describe the software test documentation (STD) [17] to be generated by the software provider. The DNA Commission differentiated developmental from internal (laboratory) validation and emphasizes that the software used is an integral part of the evidential process and should not be treated as a separate and isolated component.

2. Provider responsibilities and developmental validation

The software developer has the burden to specify and document the assumptions and genetic/statistical models underlying the software program and refer to mathematical/statistical proofs or provide these with the software. Prior to promoting their software for practical use, the provider or developer must conduct a developmental validation demonstrating that the intended calculations are being performed correctly and that they provide the expected results. The data sets used for validation should be made publicly available alongside the validation results, as is outlined below.

2.1. Underlying models and developer's validation

Recommendation 1

Bio-statistical software for forensic genetic applications should be accompanied by scientific papers or information or guidance materials, such as a user manual, describing the underlying method. The population genetic and data model(s) used should be explicitly described and disclosed to allow the reproducibility of all the computations by other means (algebraic formulae, other software programs or statistical approaches) as publication in peer-reviewed journals

The DNA Commission encourages software providers or developers to report the theoretical assumptions underlying their product or refer to already published models. We also encourage the publication of the design and outcome of their developmental validation in peer-reviewed journals. We discourage insufficiently documented or described software where the end user cannot adequately explain to the trier of fact (e.g. judge or jury) the theoretical basis of the software used.

Recommendation 2

Bio-statistical software for forensic genetic applications should be validated according to particular requirements and specific intended use. The software developer's validation should use publicly-available data sets or disclose the used data set otherwise. The result of the software developer's validation and its environment (hardware and software dependencies) should be documented and disclosed

One of the principles of scientific research is that any new finding should be amenable to independent replication. The DNA Commission therefore encourages software providers or developers to verify and validate their software (e.g. by generating or using validation data sets with known outcomes) along with the parameters necessary for the software to work (e.g. population allele counts for frequency calculation). Verification may be assessed using code review. This information could then be publicized so as to support interested laboratories with their own internal training and explorative testing of the software.

The test cases of the validation data should be designed so as to cover all of the software functionality, to be complex enough to detect installation errors, and to be generic enough to also serve as a basis for testing the consistency of future versions of the validated software. Although the goal of internal validation is not to repeat developmental validation, making the data and parameters used for the latter publicly available may add extra benefit in that it would allow laboratories to investigate the local performance of the software under the conditions of the developmental validation, if they so wish. Validation test results should be documented (and disclosed) following a test plan [17] as well as system requirements and platform (hardware and software) specification.

The validity of the results obtained from a given validation data set should also be assessed by way of comparison to the results obtained through hand calculations of algebraic formulae (if possible), using alternative statistical approaches where applicable

e.g. paternity index or the Random Match Probability (RMP), qualitative conclusions drawn by trained analysts [7], or through the use of similar software [8]. Validation exercises should include simulated or real samples with a known underlying scenario. Simulations should cover all relevant aspects of the behavior of genotypes, e.g. mutations, silent alleles, marker linkage, linkage disequilibrium, or population substructure. All input and output data file formats should be documented and/or validated as well. Where applicable external references defining the file format should be included. The DNA Commission encourages the use of open and license free file formats.

Mixture analysis software should be validated on test data involving both known donors (contributors in the mixture which explain the hypothesis of the prosecution (Hp true)) and known non-donors (contributors not in the mixture which explains the hypothesis of the defense (Hd true)), with scenarios underlying the data that cover the range likely to be encountered in casework. The representativeness of the data should cover, as a minimum, the number of contributors, mixture ratios of contributors and DNA template amounts. False donors may be created by simulation or may be real. For the Hp true samples the LR should be largely above 1. The proportions of samples producing a LR less than 1 for Hp true and greater than 1 for Hd true should be noted. The results of these experiments should be disclosed. Circumstances where the LR is above 1 for Hd true or less than 1 for Hp true should be discussed.

For kinship testing software, computations should be performed comparing the likelihoods of the (available) individuals related through the pedigree A (Hp) or through the pedigree B (Hd), under the established assumptions of the program. Samples for Hp and Hd true can be obtained from casework or (preferably) from simulated data. Tests for different levels (and types) of kinship defining Hp and Hd should be computed. The results of these experiments should be disclosed, namely through the plotting of true and false positive rates (in the sense of adopting Hp) for various thresholds of the LR.

Examples of using ground-truth data to test the performance of software can be found in [8,14,26].

2.2. Version control

Recommendation 3

Each version and build of a software should be distinguishable by a version and build number. Each version and build of a software should be validated independently. Exceptions or exclusion of specific tests should be documented.

Software development is often incremental. Amendments to a program may involve alteration of the core algorithms or may be merely cosmetic (such as improving the user interface). If software has been developed in separate parts, any change to one part may bear a risk of consequential changes in the other parts. This has to be taken into account when validating revised software components separately, even though such partial testing may greatly lower the efforts for developmental validation and for internal revalidation by the laboratories.

Providers or developers must label their software by version numbers and a build number to completely identify the software. Every significant change to the code in a released version should be given a unique version number. Whereas additions to the code that, for example, only affect the display of results may not require a change in version number, systems should be in place ensuring that substantial changes cannot be made to the software without changing the version number. All material made available with regard to the developmental validation must be linked to the applicable version number. All software documentation also needs to be clearly tied to a specific version of the software.

Retired versions and documentation should be archived by the providers or developers so as to ensure the possibility of reusing these versions if required, e.g. for review of old cases. Many laboratories are moving towards the use of probabilistic software for mixture interpretation, and consequently often face requests from both prosecution and defense to re-interpret historical cases, especially where “inconclusive” results were obtained by other means of interpretation. We anticipate that future probabilistic bio-statistical software programs will necessitate the review of today’s interpretational methods. It is important to retain retired software versions and the associated documentation of these programs.

2.3. Education and training to the end user from the provider

Recommendation 4

The software provider or developer should create instructions on how to validate and configure the software prior to use in a laboratory. These instructions should form the basis of any internal validation plan to be designed by users.

Recommendation 5

Any bio-statistical software should be accompanied by a user manual enabling a trained user to understand and explain the principles of the software functions and to use the software correctly.

Recommendation 6

Any potential user should have access to sufficient knowledge to use the software in a reasonable way. It is the responsibility of the laboratory to make sure that it has sufficient training resources and provides sufficient support to users to demonstrate that a proposed implementation is ‘fit-for-purpose’.

Laboratories validating software should also create their own examples to test the limits of the software of interest. Guidance from the developer or provider could be valuable to allow the laboratory to develop the most sensible and efficient strategy for validation.

Implementation instructions of stand-alone software should include hardware specifications and troubleshooting information. It is anticipated that the known data sets (either generated by the provider or the testing laboratory) with previously established outcomes will be used to verify proper on-site performance as discussed in Recommendation 2.

User manuals should also have version control for the former to match the software actually in use. Every released version of the software should be accompanied by a comprehensive user manual, or a comprehensive description of the introduced modifications (in case of minor changes). The user manual should be linked to the software version (e.g. use of the software version number on every page). The manual should include a description of the theoretical basis of the software or references to publications or other work describing the basis of the implemented methods. Changes from previous versions should be detailed within the documentation. A separate version history listing the changes introduced for each version release should also be available.

The manual should be standalone or provide detailed references to the available literature. If training is a prerequisite for obtaining the software in the first place, then the manual should provide all instructions in conjunction with that training. In any case, trained users should understand the principles and limitations of the software sufficiently well to represent and explain the results in court. If training is not a prerequisite for obtaining the software, then the user manual must be sufficient that an untrained user can also competently use the software.

As far as training is concerned, the DNA Commission endorses practical in-house training sessions, or remote training (either live or recorded); or at a minimum adequate written material required to meet this recommendation.

Not only the laboratory and the prosecution, but also the defense must have access to suitable information, and the defense may need to investigate significant aspects of the performance of the software for a specific case. Scientists working for the defense should be allowed to attend training and should be permitted to obtain or purchase the software after meeting any training requirements.

2.4. Software updates and continuity

Recommendation 7

To ensure continued availability of software in the future, it is recommended that software source code is placed in a secure repository (e.g. GitHub or an escrow account) and that the algorithms are described in sufficient detail to allow for reimplementation. It is the responsibility of the customer of software to ensure that they have a legal basis to access the code in the event of a supplier ceasing to trade or withdrawing support

The DNA Commission does not consider examination of the source code to be a useful fact-finding measure in a legal setting. A rigorous validation study (both developmental and internal) should be sufficient to reveal shortcomings or errors in coding. There should be sufficient public information available to allow for independent reimplementation as described in recommendation 1. However, if requested by the legal system, the code should be made available subject to the software provider's legitimate copyright or commercial interests being safeguarded. Supervised access to the code under a "no copy" policy is acceptable.

If the software follows the open source principle, the DNA Commission encourages open-source developers to publish their source code using systems such as SoftwareX (http://www.journals.elsevier.com/softwarex/) as Supplementary data. Language specific repository systems such as CRAN (https://cran.r-project.org/) or general ones like GitHub (https://github.com) should be utilized where publishing is unsuitable or impossible.

Sharing of the source code can be useful for collaborative efforts or further development, improvements, or modifications. The sharing of source code does not release the developer from their obligation to rigorously document, verify and validate their software.

Recommendation 8

Custodians of software used for forensic genetics purposes should establish a system allowing them to notify users about quality assurance issues and updates. Software bugs (and their fixes) together with a list of changes should be disclosed.

During the time a given piece of software is in use, new limitations or programming faults almost inevitably will be discovered. The impact of such faults should be investigated by the providers or developers and disclosed together with the fix. However, it is important that knowledge of any newly arisen problems is shared transparently with end users and other stakeholders in the judicial process. Corrective actions must be triggered as needed and end users prevented from continued use of outdated or flawed versions. This requires, as a minimum, a link between the providers and developers on the one hand, and end users and interested third parties on the other that may even be unknown to the providers or developers themselves. This link could be drawn, for example, by a website where critical information is made available, or a registration system whereby the provider or developer can contact users directly.

2.5. Randomness

Recommendation 9

Software using algorithms with components of randomness, such as Monte Carlo methods or random permutations, should have a feature to set this function to a stable state/mode that allows for repeated testing or recalculation (e.g. the user should be able to set the seed for initiating a Monte Carlo process to allow for repeated analyses of the same data set).

Some software programs utilize randomness (e.g. model the drop-out probability as part of a Markov Chain Monte Carlo, determine a p-value by random permutation or random selection as part of a bootstrap process). These use a random number generator which starts from an initial number, known as the seed, and apply an algorithm that produces a sequence of numbers that have little relationship to each other. The series will eventually repeat itself, although usually only after a very long time.

It may be necessary to reproduce results after the fact and reanalyze one specific run in exactly the same fashion, for example as part of verifying the software after a change, or due to a retrospective investigation. Since this can only be achieved by using the same seed in the second run that was used in the first run, it is desirable that the seed is reported as part of the output of each run, and that the end user can set a particular seed for a run themselves, if they so wish.

3. Internal validation

Internal validation refers to empirical studies performed either within a laboratory or outsourced to a third party entity to ensure that the software runs properly within the relevant laboratory. It should cover a wide range of the functionality of the software and all relevant parameter settings of the software. Unless the software will only be used on pristine samples with complete genotypes, the validation needs to address variations in multiplexes, cycle numbers, clean-up chemistries, injection strategies, or equipment that may be used in casework. Internal validation should be planned carefully. The plan should include (at a minimum) the objectives outlined in recommendations 10 through 13. Developmental validation information should be gathered from the provider or developer and laboratories should be familiar with the content of this material before starting their internal validation.

The goal of an internal validation study is to explore the limitations of the software and test the reliability, robustness, and reproducibility of the system. Samples that mimic the types of cases encountered should be tested. These will primarily include "mock" samples. Real casework samples can also be used. The challenge with using real casework samples is that the "ground truth" composition of the mixture components may be difficult to determine, especially with very low level minor contributors.

Some laboratories may be restricted with their use of casework data for validation activities. Where previous interpretation methods resulted in an inclusion of a person of interest, broadly one should expect an inclusionary likelihood ratio for the interpretation of the same profile using probabilistic genotyping software.

3.1. Developing a plan and sample testing

Recommendation 10

Before initiating the validation of a software program, the laboratory should develop a documented validation plan. The software should have a completed and up to date developmental validation along with other supporting materials such as

publications describing the models, propositions and parameters used by the software and a user's manual.

Recommendation 11

The laboratory should test the software on representative data generated in-house with the reagents, detection instrumentation, and analysis software, used for casework. If a laboratory employs variable DNA typing conditions (e.g. within variation in the amplification and/or electrophoresis conditions to increase or decrease the sensitivity of detection of alleles and/or artifacts), then these types of profiles should also be tested as part of the internal validation plan.

Recommendation 12

The laboratory should consider the range of samples expected to be analyzed in casework to define the scope of application of the software. Internal validation should address (1) true donors and non-donors and/or (2) related and unrelated individuals across a range of situations that span or exceed the complexity of the cases likely to be encountered in casework.

Planning is crucial for any validation exercise to be successful. In addition to identifying suitable staff to conduct the necessary experiments, the information technology resources required for running the software should be scrutinized as well. Moreover, some of the experiments called for in Recommendation 11 may be redundant under certain circumstances. For example, if a laboratory is validating software for kinship analysis, then varying the amplification or electrophoresis conditions is usually unnecessary because only the specific alleles (and not the variation in peak heights) are required for software validation.

The consideration of both known contributors and known non-contributors is important to determine the limits of any software for mixture interpretation [27]. Mixtures should be gauged against profiles of true donors (i.e., ground truth known trials) to test the sensitivity of the software whereas a comparison to non-contributors is necessary to test its specificity. Where previous interpretation resulted in an inclusion of a person of interest, one should expect an inclusionary likelihood ratio for the same profile using the software under validation; deviations should be discussed in the validation report.

Determination of the limits of the software is important to establish the types of profiles that are suitable for handling by the laboratory. It is acceptable to manipulate the input data so as to create challenging profiles with the desired properties to test.

Probabilistic software, especially for low-level DNA mixtures, may allow a laboratory to widen the scope of their casework in terms of the type of evidence handled. However, there may also be a temptation to submit all complex mixtures to particularly versatile software. Therefore, the community is reminded of a previous recommendation of the DNA Commission [20] that is still valid:

> (Gill et al., 2006, Recommendation 8): If the alleles of certain loci in the DNA profile are at a level that is dominated by background noise, then a biostatical interpretation for these alleles should not be attempted.

Recommendation 13

The laboratory should determine whether the results produced by the software are consistent with the laboratory's previously validated interpretation procedure if the data and/or method exist.

In general, known samples are used as part of the internal validation and the results from previous validation exercises (for example, a simple spreadsheet to calculate kinship statistics for parent-child trios) should be compared to the output of new software. One would expect the results of the different procedures to be sufficiently similar.

3.2. Standard operating procedure development

Recommendation 14

In addition to the user manual, the laboratory should develop standard operating procedures based upon the internal validation data outlining the types of cases and data to which the software can be applied, the source of population allele frequencies, the testing of one or more propositions, reporting, and how software updates are performed regularly.

The SOP for any laboratory should take into account both the developmental and internal validations. They should guide end users on when and how to use the software and when it should not be used. The latter can be achieved by providing explicit guidance on the limitations of the software. The SOP should be detailed enough to ensure consistent use of the software across the laboratory. It is important to note with both kinship analysis [21] and forensic evidence evaluation [20], the construction of clearly defined hypotheses (propositions) is critical, and the key assumptions underlying the computational process will affect the final interpretation of the output [28–30].

Prior to training laboratory staff on new SOPs, the instructions should be tested on a controlled data set to verify that workflow laid out by the SOPs performs as expected.

Software bug-fix releases should be installed with priority according to a plan as part of the SOP. The laboratory should define a general policy on software updates and upgrades in terms of validation and personnel responsibilities.

3.3. Training and education

Recommendation 15

The laboratory should develop and follow a policy or procedure for the training of software end users in the laboratory.

Training laboratory personnel on the use of bio-statistical software is mandatory and must include a range of cases and require a competency test as a qualifying exam. In addition, the DNA Commission recommends that basic training on likelihood ratios and proposition building should be an integral part of the professional qualification of forensic geneticists.

The training policy should outline the prerequisite competencies for an examiner using the software. For example, if the software requires manual elements such as removal or recognition of artifacts, or for mixture software the assignment of a number of contributors, then these are prerequisite competencies. For each competency mandated for the examiner using the software, the exact learning outcome, the examination strategy, and the expectations required to pass the exam should be defined.

Additional proficiency testing and continuous competency monitoring of the software users is also recommended. The ISFG encourages the participation of external collaborative exercises such as proficiency testing workshops and interlaboratory studies [31,32] to develop a "community of users".

Recommendation 16

The DNA Commission encourages the forensic community to establish a public repository of typing results from adjudicated casework covering a wide range of kinship cases and mixture samples including different challenging scenarios like low-level mixtures and related contributors. The data need to be in a universal, useful file format. The repository should be governed by a neutral organization providing equal access to all interested international parties.

Mock or case-like samples may be a useful alternative for the repository. Meta-data associated with the submitted profiles should include relevant information such as the kit used, PCR

cycle conditions, the separation polymer used, the CE system electrophoretic injection parameters, and any other relevant information about the sample.

The DNA Commission envisions the repository to become a rich resource for both, the initial testing of new software and continuing training programs. For example, a set of candidate family reference data from NIST [33] available at http://www.cstl.nist.gov/biotech/strbase/kinship.htm was used by one laboratory to confirm the concordance between a kinship software program and algebraic calculations verified by a spreadsheet program [34]. Likewise, the Biomedical Forensic Sciences program at Boston University (USA) has developed a training website (http://www.bu.edu/dnamixtures/) with a variety of single-source and mixture profiles for testing and training.

3.4. Additional guidance on software usage and application

Cosmetic modifications such as a change in the graphical interface of the program, or changes in the reporting format, may not require developmental validation but should be subjected to additional tests to ensure that the changes do not affect the interpretation of the software output. This may be achieved by running a range of identical cases before and after the changes, followed by comparative reviewing of the output. Core changes to the implemented algorithms should be subjected to additional developmental validation prior to their release.

In addition to supporting internal laboratory validation, it is recommended that software providers or developers, together with laboratories and other stakeholders, create Supporting information targeted towards the legal community. This information shall be made up such that it allows end users to successfully debate the scientific merits of the software in admissibility hearings and court cases. In jurisdictions employing an adversarial system, this should include a defense access policy.

If the cost to purchase the software is prohibitive, access, at reasonable or no cost, to an executable version of the software for use in a particular case, along with sufficient support that the defense could realistically run the software with some understanding should be provided. If alternative validated software using similar, scientifically sound and widely accepted algorithms is available, then the defense scientist may use this different software to analyze the case in question. There may be examples where the analysis of one and the same evidence with different software produces statistical output that may lead to differing conclusions. This could possibly cause confusion in the legal system although it should not be interpreted as one software being "better" than the other. It is important instead that the end users understand the underlying assumptions, models, and limitations of the software used.

Acknowledgements

The authors would like to thank Dr. Bruce Weir (University of Washington) and June Guiness (UK Forensic Science Regulation Unit) for their input and helpful discussions. Points of view in this document do not necessarily represent the official position or policies of the U.S. Department of Commerce. Certain commercial equipment, instruments, or materials are identified in this paper in order to specify experimental procedures as completely as possible. In no case does such identification imply a recommendation or endorsement by the U.S. Department of Commerce nor does it imply that any of the materials, instruments or equipment identified are necessarily the best available for the purpose.

References

[1] C.H. Brenner, Symbolic kinship program, Genetics 145 (1997) 535–542.
[2] J.A. Riancho, M.T. Zarrabeitia, A Windows-based software for common paternity and sibling analyses, Forensic Sci. Int. 135 (2003) 232–234.
[3] J. Drábek, Validation of software for calculating the likelihood ratio for parentage and kinship, Forensic Sci. Int. Genet. 3 (2009) 112–118.
[4] M.W. Perlin, M.M. Legler, C.E. Spencer, J.L. Smith, W.P. Allan, J.L. Belrose, B.W. Duceman, Validating TrueAllele® DNA mixture interpretation, J. Forensic Sci. 56 (2011) 1430–1447.
[5] A. Röck, J. Irwin, A. Dür, T. Parsons, W. Parson, SAM: String-based sequence search algorithm for mitochondrial DNA database queries, Forensic Sci. Int. Genet. 5 (2011) 126–132.
[6] K. Slooten, Match probabilities for multiple siblings, Forensic Sci. Int. Genet. 6 (2012) 466–468.
[7] A.A. Mitchell, J. Tamariz, K. O'Connell, N. Ducasse, Z. Budimlija, M. Prinz, T. Caragine, Validation of a DNA mixture statistics tool incorporating allelic drop-out and drop-in, Forensic Sci. Int. Genet. 6 (2012) 749–761.
[8] P. Gill, H. Haned, A new methodological framework to interpret complex DNA profiles using likelihood ratios, Forensic Sci. Int. Genet. 7 (2013) 251–263.
[9] D. Kling, A.O. Tillmar, T. Egeland, Familias 3—extensions and new functionality, Forensic Sci. Int. Genet. 13 (2014) 121–127.
[10] R. Puch-Solis, T. Clayton, Evidential evaluation of DNA profiles using a discrete statistical model implemented in the DNA LiRa software, Forensic Sci. Int. Genet. 11 (2014) 220–228.
[11] C.D. Steele, M. Greenhalgh, D.J. Balding, Verifying likelihoods for low template DNA profiles using multiple replicates, Forensic Sci. Int. Genet. 13 (2014) 82–89.
[12] R.G. Cowell, T. Graversen, S.L. Lauritzen, J. Mortera, Analysis of forensic DNA mixtures with artefacts, J. R. Stat. Soc. Ser. C: Appl. Stat. 64 (2015) 1–48.
[13] G. Dørum, D. Kling, C. Baeza-Richer, M. García-Magariños, S. Sæbø, S. Desmyter, T. Egeland, Models and implementation for relationship problems with dropout, Int. J. Legal Med. 129 (2015) 411–423.
[14] D. Taylor, J. Buckleton, I. Evett, Testing likelihood ratios produced from complex DNA profiles, Forensic Sci. Int. Genet. 16 (2015) 165–171.
[15] Ø. Bleka, G. Storvik, P. Gill, EuroForMix: an open source software based on a continuous model to evaluate STR DNA profiles from a mixture of contributors with artefacts, Forensic Sci. Int. Genet. 21 (2016) 35–44.
[16] IEEE Standard for System and Software Verification and Validation, in IEEE Std 1012-2012 (Revision of IEEE Std 1012-2004), pp. 1–223, May 25 2012; 10.1109/IEEESTD.2012.6204026.
[17] IEEE Standard for Software and System Test Documentation, in IEEE Std 829-2008, pp. 1–150, July 18 2008; 10.1109/IEEESTD.2008.4578383.
[18] General Principles of Software Validation; Final Guidance for Industry and FDA Staff, U.S. Department Of Health and Human Services – Food and Drug Administration—Center for Devices and Radiological Health & Center for Biologics Evaluation and Research. (2002) available at: http://tinyurl.com/m39b2g.
[19] N. Morling, R.W. Allen, A. Carracedo, H. Geada, F. Guidet, C. Hallenberg, W. Martin, W.R. Mayr, B. Olaisen, V.L. Pascali, P.M. Schneider, Paternity Testing Commission of the International Society of Forensic Genetics: recommendations on genetic investigations in paternity cases, Forensic Sci. Int. 129 (2002) 148–157.
[20] P. Gill, C.H. Brenner, J.S. Buckleton, A. Carracedo, M. Krawczak, W.R. Mayr, N. Morling, M. Prinz, P.M. Schneider, B.S. Weir, DNA Commission of the International Society of Forensic Genetics: recommendations on the interpretation of mixtures, Forensic Sci. Int. 160 (2006) 90–101.
[21] D.W. Gjertson, C.H. Brenner, M.P. Baur, A. Carracedo, F. Guidet, J.A. Luque, R. Lessig, W.R. Mayr, V.L. Pascali, M. Prinz, P.M. Schneider, N. Morling, ISFG: recommendations on biostatistics in paternity testing, Forensic Sci. Int. Genet. 1 (2007) 223–231.
[22] P. Gill, L. Gusmão, H. Haned, W.R. Mayr, N. Morling, W. Parson, L. Prieto, M. Prinz, H. Schneider, P.M. Schneider, B.S. Weir, DNA Commission of the International Society of Forensic Genetics: recommendations on the evaluation of STR typing results that may include drop-out and/or drop-in using probabilistic methods, Forensic Sci. Int. Genet. 6 (2012) 679–688.
[23] Scientific Working Group on DNA Analysis Methods (SWGDAM), Validation Guidelines for DNA Analysis Methods (2012), pp. 1–13. http://media.wix.com/ugd/4344b0_cbc27d16dcb64fd88cb36ab2a2a25e4c.pdf.
[24] J.A. Bright, I.W. Evett, D. Taylor, J.M. Curran, J. Buckleton, A series of recommended tests when validating probabilistic DNA profile interpretation software, Forensic Sci. Int. Genet. 14 (2015) 125–131.
[25] Scientific Working Group on DNA Analysis Methods (SWGDAM), Guidelines for the Validation of Probabilistic Genotyping Systems (2015), pp. 1–12. http://media.wix.com/ugd/4344b0_22776006b67c4a32a5ffc04fe3b56515.pdf.
[26] D. Taylor, Using continuous DNA interpretation methods to revisit likelihood ratio behaviour, Forensic Sci. Int. Genet. 11 (2014) 144–153.
[27] P. Gill, J. Curran, C. Neumann, A. Kirkham, T. Clayton, J. Whitaker, J. Lambert, Interpretation of complex DNA profiles using empirical models and a method to measure their robustness, Forensic Sci. Int. Genet. 2 (2008) 91–103.
[28] J. Buckleton, J.A. Bright, D. Taylor, I. Evett, T. Hicks, G. Jackson, J.M. Curran, Helping formulate propositions in forensic DNA analysis, Sci. Justice 54 (2014) 258–261.

[29] T. Hicks, A. Biedermann, J.A. de Koeijer, F. Taroni, C. Champod, I.W. Evett, The importance of distinguishing information from evidence/observations when formulating propositions, Sci. Justice 55 (2015) 520–525.
[30] S. Gittelson, T. Kalafut, S. Myers, D. Taylor, T. Hicks, F. Taroni, I.W. Evett, J.-A. Bright, J. Buckleton, A practical guide for the formulation of propositions in the Bayesian approach to DNA evidence interpretation in an adversarial environment, J. Forensic Sci. 61 (2016) 186–195.
[31] A.R. Thomsen, C. Hallenberg, B.T. Simonsen, R.B. Langkjær, N. Morling, A report of the 2002–2008 paternity testing workshops of the English speaking working group of the International Society for Forensic Genetics, Forensic Sci. Int. Genet. 3 (2009) 214–221.
[32] L. Prieto, H. Haned, A. Mosquera, M. Crespillo, M. Alemañ, M. Aler, F. lvarez, C. Baeza-Richer, A. Dominguez, C. Doutremepuich, M.J. Farfán, M. Fenger-Grøn, J. M. García-Ganivet, E. González-Moya, L. Hombreiro, M.V. Lareu, B. Martínez-Jarreta, S. Merigioli, P. Milans Del Bosch, N. Morling, M. Muñoz-Nieto, E. Ortega-González, S. Pedrosa, R. Pérez, C. Solís, I. Yurrebaso, P. Gill, Euroforgen-NoE collaborative exercise on LRmix to demonstrate standardization of the interpretation of complex DNA profiles, Forensic Sci. Int. Genet. 9 (2014) 47–54.
[33] K.L. O'Connor, S.P. Myers, E.L.R. Butts, C.R. Hill, J.M. Butler, P.M. Vallone, Candidate reference family data: a tool for validating kinship analysis software, 21st International Symposium On Human Identification, San Antonio, TX, October 11–14, 2010 http://www.cstl.nist.gov/biotech/strbase/pub_pres/Promega2010_OConnor.pdf. http://www.cstl.nist.gov/biotech/strbase/kinship.htm.
[34] D.A. Azevedo, G.R.B. Souza, I.H.E.F. Silva, L.A.F. Silva, Genetic kinship analysis: a concordance study between calculations performed with the software Familias and algebraic formulas of the American Association of Blood Banks, Progress in Forensic Genetics 14: Proceedings of the 24th International ISFG Congress (2011) e186–e187.