# Chapter 9
# A Survey of Methods and Tools for Large-Scale DNA Mixture Profiling

**Emad Alamoudi, Rashid Mehmood, Aiiad Albeshri, and Takashi Gojobori**

## 9.1 Introduction

According to The American Heritage Medical Dictionary, DNA profiling is "the identification and documentation of the structure of certain regions of a given DNA molecule, used to determine the source of a DNA sample, to determine a child's paternity, to diagnose genetic disorders, or to incriminate or exonerate suspects of a crime [1]." DNA profiling (also named DNA typing, DNA fingerprinting, or DNA testing) which was first introduced in 1985 by Alec Jeffreys has changed the area of forensic science significantly [2]. Dr. Jeffreys has found that there are several regions in the human DNA that contain repeated DNA sequence. He found that these DNA sequence areas may differ from one person to another. Dr. Jeffreys was able to measure the variation in these DNA sequences by developing a unique identity test called Restriction Fragment Length Polymorphism (RFLP). The repeated DNA areas are called Variable Number of Tandem Repeats (VNTRs).

E. Alamoudi (✉) · A. Albeshri
Department of Computer Science, Faculty of Computing and Information Technology (FCIT), King Abdulaziz University, Jeddah, Saudi Arabia
e-mail: ealamoodi0004@stu.kau.edu.sa; aaalbeshri@kau.edu.sa

R. Mehmood
High Performance Computing Center, King Abdulaziz University, Jeddah, Saudi Arabia
e-mail: RMehmood@kau.edu.sa

T. Gojobori
Computational Bioscience Research Center (CBRC), King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia
e-mail: Takashi.Gojobori@kaust.edu.sa

R. Mehmood et al. (eds.), *Smart Infrastructure and Applications*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-13705-2_9

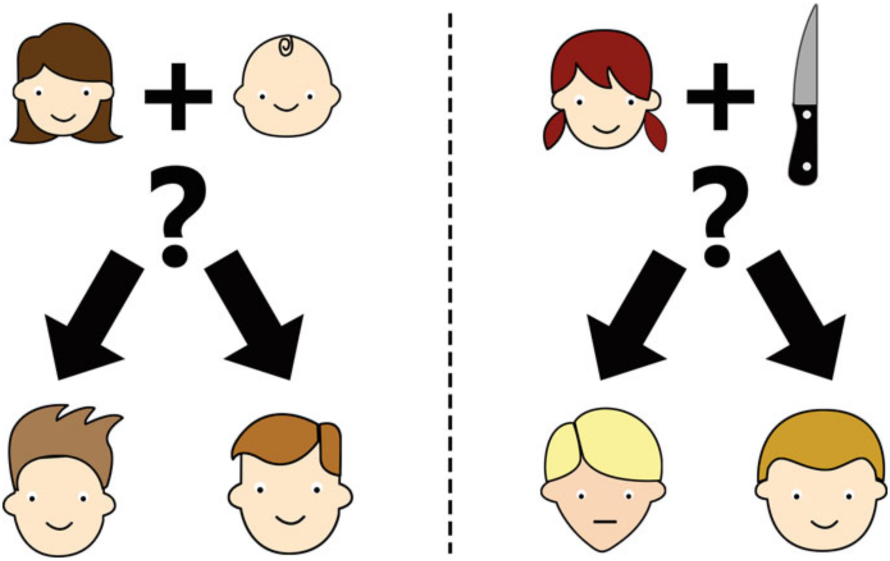


**Fig. 9.1** DNA profile interpretation can have multiple usages, such as determine child's father and find a criminal among suspects

Today, DNA profiling is helping in many cases to identify an innocent from guilty. Human Identity test can also be used in contexts such as missing people investigation, parentage test, ancestry test, and disaster victim identification (see Fig. 9.1).

The DNA typing is considered today to be the most useful tool in the hand of law enforcement. Moreover, computer databases which contain DNA information of criminals which was taken from crime scenes had helped to associate a crime to an offender. Due to having a specific set of Short Tandem Repeat (STR) loci in these massive databases, it is unlikely to see a new set of DNA markers to be introduced shortly [2].

In order for a DNA sample to be processed, several steps should be considered [2]. First, obtaining the DNA from a biological source. Second, assessing the amount of DNA recovered. Third, isolate the DNA from its cells by using Polymerase Chain Reaction (PCR), which is a technique for copying specific DNA areas. Finally, the STR alleles which have been generated from the previous step will be examined. Figure 9.2 shows the steps used in DNA sample processing.

However, many difficulties may occur during the procedure of producing a DNA profile that affects the analysis of the sample. One of these problems is the stochastic effects, which arise during DNA extraction. Other challenges are allele drop-out, PCR process, allele sharing, and PCR amplification artifacts. Such difficulties hardened the accurate interpretation of the DNA profile [3].

The result of the DNA sample processing will be compared to other sample or databases to check the similarity. If there is a match or "inclusion," this indicates

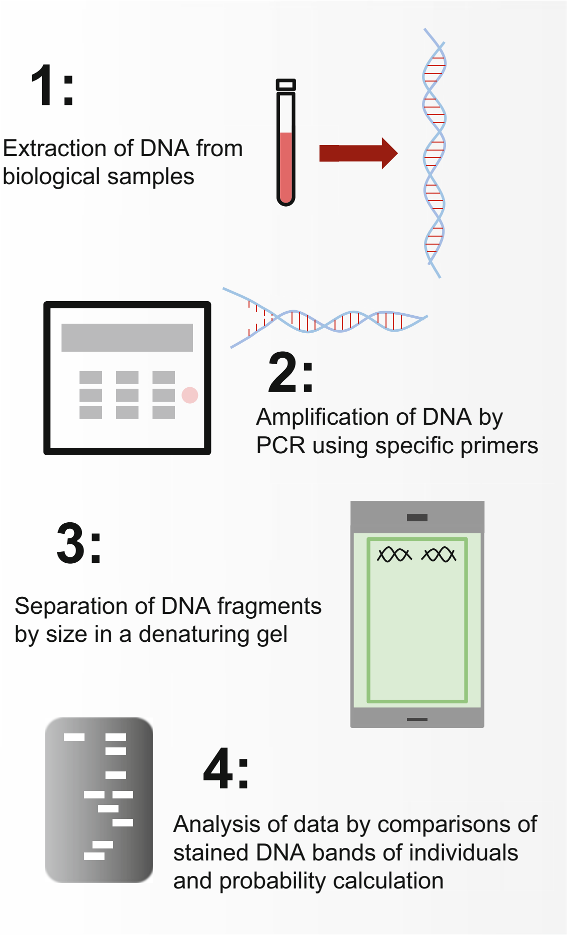

1:
Extraction of DNA from biological samples
2:
Amplification of DNA by PCR using specific primers
3:
Separation of DNA fragments by size in a denaturing gel
4:
Analysis of data by comparisons of stained DNA bands of individuals and probability calculation

**Fig. 9.2** The needed steps for DNA sample processing

that both samples were taken from the same source. On the other hand, if there is no match, the result would consider as "exclusion," which means there is no biological relation between the two samples [2]. A case report will be made by a forensic specialist explaining the result and containing random match probability answering the similarity question.

The Scientific Working Group on DNA Analysis Methods (SWGDAM) advise forensic report to contain a prediction of the number of contributors to the mixture that is under examination [3]. Usually, the number of contributors of a sample that taken from a crime scene is unknown. Therefore, an analyst should estimate it according to the electropherogram obtained. This assumption affects the final weight of DNA evidence [3].

In this chapter, we provide an extended review of DNA profiling methods and tools with a particular focus on their computational performance and accuracy. This is an extended version of our earlier work [4]. We have added further elaborations on the DNA profiling methods including DNA biology and genetics. Also, we discuss different HPC systems, namely, cloud, clusters, GPUs, and FPGAs. A background on parallel computing, MPI, OpenMP, and Java multithreading has been added. Additional DNA profiling tools have been reviewed and further explanation on the existing tools is provided. To the best of our knowledge, this is the first review work on DNA profiling tools.

Faster interpretations of DNA mixtures with a large number of unknowns and higher accuracies are expected to open up new frontiers for DNA profiling in the smart societies era. In the coming years, the complete genome sequencing technologies in a single or only a few cells will be easily available. These technologies may change the situation of DNA profiling completely. In this case, it is obvious to prepare appropriate statistical methods for that. It will be, therefore, important to prepare the mathematical and statistical algorithms for complete-genome-sequencing-based DNA profile. Emerging computational and big data developments [5], along with Internet of Things (IoT) [6] and smart society environments [7], will provide opportunities for new services related to DNA profiling.

The rest of the chapter is organized as follows. Section 9.2 describes background concepts related to this chapter including a background on DNA concepts, DNA profiling, parallel and High-Performance Computing (HPC). Section 9.3 discusses several methods for evaluating the DNA mixture statistically. Section 9.4 describes a number of approaches that rely upon the calculation of likelihood ratio to interpret DNA profile. We further discuss the importance of the Number of Contributors (NoC) in profiling a DNA mixture in Sect. 9.5. Some implementations that estimate the NoC was mentioned in the same section. Section 9.6 then illustrates notable DNA profiling tools. We conclude and give an outlook for the future of DNA profiling in Sect. 9.7.

## 9.2 Background Material

We now give a brief background of the various concepts and methods related to DNA profiling. The list of topics covered are DNA biology and genetics, forensic science, DNA mixture and its technologies, genetic markers, factors that increase the complexity of DNA profiling, likelihood estimator, the use of HPC in bioinformatics field, and HPC system and parallel frameworks.

### *9.2.1 DNA Biology and Genetics*

The basic unit of living species is the cell, which produces energy and raw materials. To keep a cell operating, thousands of proteins are required. An individual body usually contains 100 trillion cells [2]. All these cells come from a single cell called zygote, which is formed from the merging of the mother's egg and the father's sperm. All cells share the same genetic sequences. Inside the nucleus of the cell is a chemical substance called DNA, which encodes protein construction data and cell replication information.

DNA, or Deoxyribonucleic Acid, is acting like a blueprint for our bodies since it contains all the required information for passing down genetic attributes to next generations. The entire DNA of a cell is called a genome.

DNA serves two essential purposes: first, makes replication of itself; second, handles information about protein producing instructions. Its alphabet contains only four letters: Adenine (A), Thymine (T), Cytosine (C), and Guanine (G) [2]. These letters are known as nucleotides or bases. Different combination of these bases can make the difference between humans and other species. The human body contains around three billion nucleotides. Each nucleotide is linked to its complementary base through hydrogen bonds that link the bases. The complementary base for adenine is thymine, and it cannot pair up with either cytosine or guanine. On the other hand, cytosine can only pair up with guanine. Moreover, there are three hydrogen bonds that connect cytosine and guanine, and two bonds linking thymine and adenine. Therefore, the C-G base pair is a bit stronger than the A-T ones [2].

DNA is composed of two twisted strands, or double helix, each of which comes from both parent. The DNA is divided into chromosomes; each chromosome acts like a container for the DNA molecule in a thread-like structure. A human genome is made up of 46 chromosomes or 23 pairs of chromosomes. Out of these 23 pairs, 22 pairs are autosomal chromosomes and one pair of the chromosome is for sex determination. Males will have X and Y chromosomes, whereas females will have two X chromosomes. Autosomal chromosomes are frequently used in human identity test [2], while the sex determination chromosome is usually used for sex determination tests.

A cell is called haploid if it contains only one set of chromosomes, like gamete cell (sperm and egg) However, if two sets of chromosomes do exist, a cell then is called diploid [2]. Triploid and tetraploid refer to having three or four sets of chromosomes, respectively.

A chromosome will have coding and noncoding areas: coding areas, or gene, are the regions that have the essential information for protein construction for cells. A gene size range between a few thousand and tens of thousands of base pairs [2]. A one-to-one comparison between biological and printed terms is presented in Fig. 9.3.

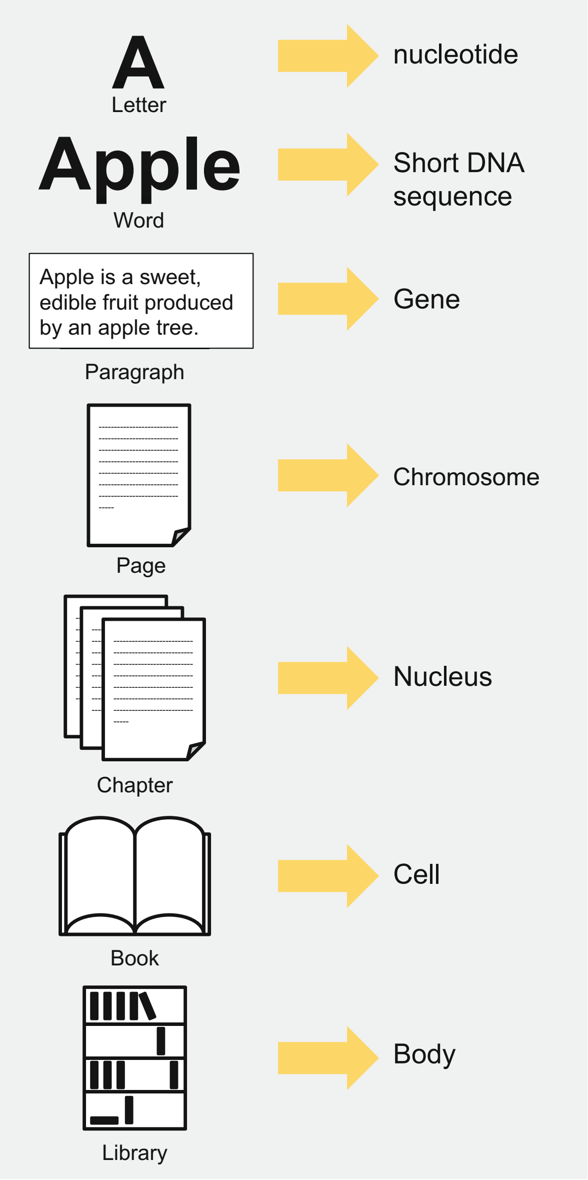


**Fig. 9.3** Comparison between printed and genetic terms

### 9.2.2 Forensic Science

Forensic DNA tests had a major influence on the evolution of the criminal justice system. Yet, the advancement of new technologies is enabling forensic labs to expand its capabilities and improved the sensitivity of the DNA interpretation.

Butler [8] thinks that this area would develop in the future in three main areas; DNA technologies will become faster, the sensitivity of extracting relative information will increase, and higher volume of data will be expected due to that sensitive nature. He argued that STR will remain the dominant genetic marker.

According to Butler [8], key challenges in the forensic science field are the subjectivity, inconsistency of the complex DNA mixture interpretations between different laboratories and analysts, and the need for training forensic analyst to enhance interpretation of DNA profiles.

### 9.2.3 DNA Mixture

A sample is called a DNA mixture when two or more individuals contribute to it. Under some circumstances, the interpretation of a mixture could be more challenging. Allele sharing is one of the factors that increase the difficulty of interpreting a profile [2]. If we have a two-person mixture, then we expected to observe only four alleles per locus. However, this rule may change if we have alleles overlapping or if we have heterozygous individuals. If we have more than four alleles per locus, then we might deal more than two people mixture [9].

DNA mixtures interpretation is a very demanding task [10]. Perez et al. define the DNA mixtures as when two or more people contribute to the same sample. They added that contributors include victims, perpetrators, or other people who interact with the crime scene. Yet, the mixture can be complex when it became a subject of allele drop-in or/and allele drop-out [11]. A detailed introduction to the DNA analysis on the forensic science domain was given by [2, 12]. Butler gives a historical overview explaining the evolution of the area. He also explains the structure of the DNA and its fundamental component.

### 9.2.4 Technologies for DNA Profiling

The topic of DNA profiling was improved by the new advances in the technology. Weedn and Foran [12] gave a general overview of the latest updates and challenges in the forensic science domain related to DNA profiling. STR followed by PCR amplification is one of the most used methods that regularly used in forensic labs [12]. Other markers such as Single Nucleotide Polymorphisms (SNP), Y chromosome STRs, and mitochondrial DNA are also considered. Weedn and Foran argued that the forensic DNA typing is the most dominant method in the forensic science laboratory. They mentioned that the forensic test usually performed with taking into consideration the court challenges. Therefore, the forensic science only uses a well-validated procedure, and all the laboratory processes should be documented. The protocols should be ready to be defended against legal attacks.

New technologies had not only increased the quality of profiling the DNA mixture, but also amplified artifacts such as stutter, variabilities, and baseline noise. Monich et al. [13] had introduced a quantitative signal model which forms the variability in a stutter, baseline noise, and allele peak height. They had also applied the chi-squared and Kolmogorov-Smirnov (KS) tests on the true peak heights and noise to test the fitness of various probability distribution classes. They argued that the interpretation of signal measured from a DNA sample used to be accomplished by using thresholding. Nonetheless, using thresholds during DNA analysis might lead to losing valuable information. For that reason, new methods that don't rely on threshold were developed.

### *9.2.5 Genetic Markers*

Many genetic markers are used for mixture analysis such as restriction fragment length polymorphism (RFLP), STR, SNP, Y chromosomes, and mitochondrial DNA (mtDNA). The number of contributors in a mixture can be identified by counting the number of Y-STR alleles [14]. mtDNA can be used to determine the number of contributors and also it can be used with degraded specimens.

**RFLP** The restriction fragment length polymorphism (RFLP) was a popular DNA analysis during the 1980s [12]. RFLP was introduced by Dr. Edwin Southern in 1975. It involves too much work, yet it reveals only a little. Therefore, it was replaced by other techniques which were more robust, sensitive, and affordable.

**STR** STR marker has been used for DNA mixture analysis for many years. Available commercial tools offer limited STR markers, which give limited statistical support for the inclusion of mixtures. Therefore, Y chromosome STR analysis has been introduced to give extra means for the analysis of mixtures in forensic cases.

**SNP** SNP is a genetic variation among individuals. It appears throughout a person's DNA. In a diploid human genome, which consists of around six billion base pairs, there are almost 15 million SNP sites [12]. However, this method has many problems. For example, it cannot be used when the suspect is unknown. Moreover, SNP is not compatible with STR databases, and establishing SNP database would require extra work [14].

**mtDNA** Mitochondrial DNA analysis is used in cases when tissues are lacking a nucleus. Since it is present at a high copy number in each cell, it has been used with highly degraded specimens. In forensic labs, mtDNA is wildly used to analyze shed hair that lack roots [12]. In addition, it can be used with fingernails and keratotic

**Table 9.1** A comparison between DNA typing methods in forensic labs

| DNA interpretation method | PCR-based | Date of introducing | Usefulness |
|---|---|---|---|
| RFLP | ✗ | 1980s | Regular caseworks |
| STR | ✓ | 1980s | Regular caseworks |
| SNP | ✓ | 2000s | Extremely degraded sample |
| Y-chromosome | ✓ | 2000s | Vaginal swabs in rape cases |
| mtDNA | ✓ | 1990s | Degraded sample and hairs |

It was inspired by [12]

skin. However, forensic labs do not highly adopt mtDNA because it depends on DNA sequencing, which is labor-intensive, slow, and expensive process [12].

The Spanish and Portuguese Working Group of the International Society for Forensic Genetics (GEP-ISFG) made a considerable effort toward standardizing and improving the accuracy of the mtDNA analysis.

Table 9.1 shows a comparison between some DAN profiling methods. The first column describes the genetic marker. Column 2 specifies whether or not the genetic maker is PCR-based. Column 3 states when the genetic marker starts to be active. The last column shows how the genetic marker can be used.

### *9.2.6 Factors Increasing the Complexity of DNA Profiles*

Different phenomena affect the complexity of interpreting a DNA profile. These factors include: the number of contributors, peak heights, stutter, a major peak masking, a stutter peak masking, population, drop-out probability, drop-in probability, and analytical threshold. No software had yet considered all these factors in its calculation [15]. Therefore, it is part of the challenges that face people who develop DNA mixture analysis tools to select which factor to model in their implementation.

### *9.2.7 Likelihood Estimator*

Likelihood ratio (LR) is the probability comparison between evidence under two propositions [2]. One is called the prosecution hypothesis, which assumes that the DNA collected from a crime scene goes to the suspect, whereas the other is the defendant hypothesis, which assumes that the matches between the suspect and the questioned sample happened coincidentally. The two considered propositions are mutually exclusive.

The likelihood ratio is calculated by putting the prosecution hypothesis as a numerator while putting the defendant hypothesis as a denominator [2]. The LR equation is:

**Table 9.2** The strength of evidence according to LR result [2]

| Likelihood ratio | Corresponding evidence |
|---|---|
| 1 to 10 | Limited support |
| 10 to 100 | Moderate support |
| 100 to 1000 | Moderate strong support |
| 1000 to 10,000 | Strong support |
| 10,000 or greater | Very strong support |

$$LR = Hp/Hd \tag{9.1}$$

If we assume that the suspect commits the crime (100% probability), which is the prosecution hypothesis, then $Hp = 1$. Additionally, if the STR typing result is heterozygous, the probability of the defendant hypothesis would be $Hd = 2pq$, where p and q are the occurrences of the allele one and two for a locus in a relevant population [2]. If we have a homozygous STR typing, then the probability of the defendant hypothesis would be $Hd = p^2$. Therefore, the equation would become:

$$LR = Hp/Hd = 1/2pq \tag{9.2}$$

Butler [2] said that if the final result was greater than one, then this result would support the prosecution side. While if it is less than one, then the defendant theory would be in favor.

Typically, the LR will have a higher ratio if the STR genotype is rear because of the reciprocal relationship. LR is the inverse of the locus estimated frequency [2]. Note that the likelihood ratio can be more complex depending on the mixture of the evidence.

The strength of the result of the likelihood ratio in terms of the prosecution's case can be interpreted numerically as presented in Table 9.2. Column 1 represents the LR value, while Column 2 is showing the corresponding strength of evidence.

### *9.2.8 HPC Systems*

In this section, we will explain four different types of HPC systems: FPGAs, clouds, GPUs, and clusters. Generally, FPGAs and GPU give better performance when algorithms are well designed, but they are extremely resource-constrained.

**Cloud** Usually, cloud NGS tools are built on the basis of the MapReduce framework [16]. Hadoop framework typically comes with MapReduce, and it distributes the work among compute cloud. MapReduce approach guarantees fault tolerance, load balancing, and redundancy. An example of a genome assembler that uses MapReduce framework is [17]. Nevertheless, privacy is still an issue when talking about cloud solutions.

**Clusters** Cluster HPC implementation usually combines Message Passing Interface (MPI) with another paradigm. MPI is used to distribute the task to other nodes (inter-node). On the other hand, the other paradigm usually takes care of the shared memory parallelism (intra-node). MPI + OpenMP is a common hybrid solution to perform fine- and coarse-grained optimization.

Optimize HPC implementation are much better than Hadoop solutions because fine-grained optimization is harder to achieve on Hadoop [16]. Consequently, Apache Spark was introduced to avoid Hadoop drawbacks. Still, well-tuned HPC implementation typically one order of magnitude faster than Apache Spark [16]. Apache Spark has the advantage of well-handling node failure and data replication.

A good future solution would combine HPC approaches and big data for processing NGS data. Such an approach has been successfully applied in domains such as machine learning [16].

**GPU** At its best performance, GPUs can give one order of magnitude better performance than CPUs [16]. CUDA is a programming language for general purpose applications runs at GPUs. Several NGS applications were successfully developed such as genome assembly [18], error correction [19], and k-mer counting [20].

However, developing an application to run on GPUs using CUDA requires a steep learning curve. It needs a deep understanding of GPUs architecture. As a result, very few tools have been targeting GPUs. Nevertheless, the new effort to develop highly optimized libraries such as NVBIO (https://developer.nvidia.com/nvbio) and the availability of languages like OpenACC might boost the GPUs effort in life science domain [16].

**FPGAs** FPGAs are chips that are able to be programmed that includes memory blocks and logic gates that can be configured manually. The configuration process usually is done through Verilog or VHDL programming languages [16]. FPGAs offer a highly scalable solution for NGS data. Example of FPGA-based tools includes FAssem assembler [21] and FADE tool for error correction [22]. Major drawbacks of using FPGAs-based are the long development cycle, and they are often not compatible to run on different FPGA generations. Yet, the new progress on higher level programming languages like OpenCL has smooth the way for the development of FPGAs-based solutions.

### *9.2.9 Parallel Frameworks*

Parallel technologies are interesting on how to get the maximum benefit of the multicore/many-core processors and networked computing resources.

Many architectures have been proposed to enhance the resource utilization, namely, symmetric multiprocessor architecture (SMP), non-uniform memory access architecture (NUMA), simultaneous multithreading architecture (SMT), single instruction multiple data architecture (SIMD), and graphics processing unit (GPU).

In addition, multiple parallel programming frameworks have been suggested such as OpenMP, MPI, and MapReduce.

Various memory architectures exist, namely, shared memory, distributed memory, and hybrid memory architecture [23]. Shared memory systems enable all processes within the system to share memory as global memory space. In distributed memory systems, each processor has its own memory that cannot be reached by others, and no global address is available. They communicate, and send and receive data, through the network. Finally, hybrid memory systems combine both shared and distributed memory architectures. In clusters of multi-core or many-core processors, all processors within the machine shared their memory within each other; however, different machines can communicate over the network.

**MPI** Message Passing Interface (MPI) is a library specification for message passing model for distributed memory systems. It has multiple implementations such as OpenMPI, MPICH, and GridMPI [23]. Each processor, when using MPI, will have its own memory; moreover, it still can access other processors' memory using network communication. MPI offers point-to-point, from one processor to another, and collective communication, from one or many processors to one or many processors. MPI can send and receive message between processes in different modes, such as block and non-block communication. The message size can be in gigabytes [23]. MPI can run on many platforms like Windows, OS X, Linux, and Solaris. Programs written with the help of MPI can run on a single machine or a cluster of machines.

**OpenMP** OpenMP is an interface (API) for shared memory parallelism. It facilitates the programming process since it provides a set of directives for synchronization, parallelization, and managing the shared memory among threads.

When compiling a software written using OpenMP, multithreaded programs will be generated. Then, threads will share the memory address which will smooth the communication among threads.

OpenMP helps software developers to build parallel programs without in-depth knowledge of multithreading mechanism. Fine-Grained parallelism can be maintained over the OpenMP directives. Multiple languages support OpenMP such as C, C++, Fortran, Java, and it can run on multiple platforms like UNIX, LINUX, and Windows.

**Java Multithreading** Java supports multithreading shared memory parallel program language, which enables developing parallel software [24]. Multithreading feature in Java allows the execution of more than one part of a program concurrently to achieve better utilization of the computer resources. This can be achieved in Java through two ways: (1) extend the thread class, (2) by using the runnable interface [24]. One process can have multiple threads that share the same address space. Thus, a synchronization mechanism is vital to ensure data protection. Java implicitly maintains synchronization by using a lock for each object [24].

Java also provides a parallelization through distributed memory system by using API called MPJ, MPI equivalent for Java. MPJ allows developing a parallel software to run on a cluster system [25].

### *9.2.10 High-Performance Computing in Bioinformatics*

Bioinformatics is a field that deals with massive data. Such data may require an extended time frame to be processed. Therefore, high-performance computing can help in shortening the time needed to finish the data processing. Perez et al. [10] discuss how HPC can help in solving bioinformatics problems. Authors had agreed that using advanced technologies had enabled remarkable discoveries in the medical field. They discussed different HPC systems which are used in bioinformatics area such as GPU computing. Graphics Processing Units (GPUs) are used to increase the computational capabilities of a group of PCs at a lower price. Moreover, they mentioned some HPC implementations in the bioinformatics field. These applications include Virtual Screening, Parallel Processing of Microarray Data, and Big Data Analytics and Network Models. In the end, authors had mentioned some drawbacks in the current HPC domain such as the energy consumption which can be overcome by using the virtualization concept, which enable sharing system hardware among different users. Other problems are the total cost of ownership and the high learning curve in upcoming programming models to influence their computational power.

Memeti et al. [26] had analyzed a DNA sequence on a heterogeneous platform that works with the Intel Xeon Phi coprocessor. These heterogeneous platforms usually come with one or more Xeon Phi devices and one or two general purpose CPUs. Researchers had introduced a parallel algorithm which can assign the workload of DNA sequence analysis to the different Xeon devices and host general purpose CPUs. This parallel implementation was aiming to reduce the overall analysis time. They also introduced a machine learning method that can predict the performance of the proposed algorithm on both the host and device. Finally, they evaluated the performance of their proposed method using human and animals' DNA on a platform that consists of an Intel Xeon Phi 7120p device with 61 core and two 12-core Intel Xeon E5 CPUs.

Bell and Gray [27] had given an overview of the history of supercomputer since the 1960s. Moreover, they tried to predict the future and how the next trend would be. They illustrated 50 years of evaluation in the high-performance computing domain. Authors argued that in 2001, there existed two major types of architectures: clusters of scalar multiprocessors and clusters of Cray-style vector supercomputers. They said that in the 1960s, Seymour Cray had proposed a parallel instruction implementation using parallel and pipelined function units. In 1982, Cray's research had reached to the multiprocessor (XMP) structure which helped to introduce the current supercomputer architecture. This architecture was sharing 10% of the market in 2001. However, a single node had reached its limit. So, to go beyond that, a cluster architecture was proposed. In the 1980s, a cluster by CMOS-based killer micros had overcome the single node by better performance, scalability, and lower price. In 1993, NASA was looking for a supercomputer that satisfies its need which was 1 Gflops workstation. To achieve that, a Beowulf project was established which cost $40,000. In 2001, 28 Beowulfs were among the Top500 fastest supercomputers. In the end, authors had expected that there would be two possible paths for supercomputers to evolve in the future. One is an application-centric vector supercomputer. While the other concentrate on peta-scale datasets where users can get access to data.

Diegoli et al. [28] had estimated the recombination rate among 15 X STR markers by using data of genotype from 158 families and following earlier suggested a likelihood-based method which allows for single-step mutation. The computational challenges from the previous study were overcome by introducing a multi-core parallelization on the HPC system. Authors had argued that X STR is useful in forensic science due to a number of features such as their ease of haplotype inference because of the male hemizygosity and their particular mode of inheritance. They also added that few studies had systematically estimated the recombination rate among X STRs. Nonetheless, none of these studies had been comprehensive as their study.

To write an algorithm that can utilize an HPC system, a person should be able to deal with parallel programming languages. However, when writing an algorithm, different bugs may occur. Laguna et al. [29] had described the latest updates in designing a saleable debugging tool. They argue that debugging a parallel program is more difficult than debugging a serial one. Authors had focused on three dynamic debugging methods in both parallel programs and MPI instructions. The first dynamic approach is discovering scaling bugs, which helps to find bugs that are latent at a small scale while manifesting themselves at a larger scale. Vrisha is an example of this technique. Second, behavior-based debugging, this technique is based on observing the behavior of the processor. This helps to reduce the huge number of parallel processors into a small number of behavioral groups. AutomaDeD framework is a simple model of task behavior that saves information related to patterns and timing in each task's control flow. The information allows the developer to detect performance problems. Finally, software defects in MPI, MPI library implementations have suffered from software bugs, especially when ported to new machines. Many of these bugs are hard to find by average programmers. FlowChecker is an example of software that can detect MPI bugs. In the end, authors had focused their attention on three main problems that are still open in the domain which are programmability challenges, performance bugs, and detecting silent data corruptions.

In DNA profiling, the use of HPC has been limited. MPI has not been used. Most of the parallel tools have been developed using Java threads (e.g., LRmix Studio, CeesIt, and NOCIt), OpenMP (e.g., LikeLTD), and Snow parallel package in R (e.g., Kongoh and Euroformix). A distributed memory implementations of DNA profiling methods have not been reported to date.

## 9.3 DNA Profiling: General Methods

Several methods had been proposed to evaluate a DNA mixture statistically. Likelihood ratio, the combined probability of inclusion/exclusion (CPI/CPE), and a modified random match probability (mRMP) are some examples of these methods [30]. In February 2000, the FBI's DNA Advisory Board had strongly recommended the first two methods to be used [2]. Moreover, in 2006, the International Society of Forensic Genetics (ISFG) had emphasis on the value of likelihood ratio [30]. There are six steps to interpreting a DNA mixture which was first described by Tim Clayton in 1998 [2]. First, we need to identify the existence of a mixture. Second, the allele peaks should be selected. Third, we need to determine the possible number of contributors. Fourth, compute an approximation of the ratio of the people who contribute to the sample. Fifth, we need to calculate all potential genotype combinations. Finally, a reference sample comparison should be made.

In the CPI approach, an equal weight is given to all possible genotype combinations. Therefore, a lot of information is being wasted when using this approach which makes it inefficient when working with distinct genotypes [30]. This approach does not require prior knowledge of the number of contributors because it is evaluating all genotypes' combination based on the evidence profile [30].

The Random Match Probability (RMP), on the other hand, is usually used with single-source samples; therefore, a modified random match probability (mRMP) was proposed to deal with more single-source samples [30]. Unlike CPI, this approach requires prior knowledge of the number of contributors in the mixture and will not work well with low-level profiles. An example of two- and three-person mixtures calculations using mRMP was described in [31].

According to Bille et al., LR is the most dominant method of evaluating a DNA mixture. However, both mRMP and LR make use of the available information in the sample where CPI does not tend to do so.

More detailed analysis of the three methods and their advantages and weaknesses can be seen in Butler's book "Advanced Topics in Forensic DNA Typing: Interpretation" [30].

## 9.4 DNA Profiling Using Likelihood Ratio

LR is considered as the most appropriate and powerful approach for calculating the weight of DNA evidence. There are three methods using LR that are widely described in the literature. The first method is the binary model, which is the simplest yet it cannot handle complex mixture [32]. Second, the semi-continuous, which is the most used by scientists since it is easy to understand and explain, but it still neglects relevant information [33]. Finally, the continuous which overcomes most of the previous models' shortcomings. It utilizes most of the available information provided by the sample, yet it is harder to be accepted and explained in a courtroom

[32]. These models may involve a human or computerized process depending on the complexity of the approach. Kelly et al. [33] had made a comparison between these three approaches which are suggested by the DNA Commission of the ISFG.

Many frameworks that interpret complex DNA profiles rely on the likelihood ratios approach such as [11]. Gill et al. had mentioned a set of guidelines which can help to evaluate any complex mixture. In addition, they provide some features for any model that might deal with complex interpretation such as the ability to incorporate several contributors. They emphasize the fact that the calculation must be provided in a fast manner.

Most of the likelihood ratio-based analysis require the number of contributors to be given before the analysis start. For instance, [34–39] rely on the number of contributors on their analysis.

However, others had tried to avoid using it in their interpretation, such as [40, 41]. Russell et al. had developed a semi-continuous method that can calculate the likelihood ratios without previous knowledge about the contributor's number. Their simple model has the abilities to calculate the statistical weight to inclusions. They had also provided a limit test which will guarantee the absence of any false inclusion by chance. To test the proposed unconstructed likelihood ratio (UCLR) model, researchers had collected a set of DNA mixtures with known contributors in different ratios. The result shows good performance on three people mixture. However, the performance becomes worse as the number of contributors increased.

## 9.5 Estimating Number of Contributors for DNA Profiling

Today, most applications that interpreted the DNA profile do require the number of contributors to be available as input [40]. Different methods have been developed to conclude the number of contributors in a DNA mixture. One of these methods is called Maximum Allele Count (MAC). This approach calculates the minimum number of contributors who might contribute to a sample by counting the observed alleles at each locus. Nevertheless, this method may not be valid to work in a complex mixture because of the complexity of allele sharing [42]. New methods that were proposed do not only rely on the number of observed alleles, but also on the frequencies of observing the allele in the population. Biedermann et al. [43] had developed a probabilistic method that performs a Bayesian network to conclude the number of contributors in DNA mixture. The new approach performs better than MAC with a degraded DNA sample and a higher number of contributors. Maximum Likelihood Estimator (MLE) is another method used to estimate the number of contributors. It tries to maximize the likelihood value of the DNA profile [44].

Haned et al. [45] had compared MAC and MLE. The efficiency of both methods had been analyzed and compared for identifying two to five-person mixtures. Three different situations were used to test both methods. First, when all contributors belong to the same population and when allele occurrences are known. Second, when allele occurrences are not known, which may occur in population subdivision.

Finally, a condition of partial profiles and how it could affect the estimation accuracy. MAC method is used to set the lower bound that can clarify the number of alleles in a mixture. Haned et al. believe that MAC is unreliable since there is a chance for allele sharing between people which called the masking effect. The result of the comparison supports the use of MLE when a mixture contains more than three contributors. However, when three or two people contribute to a mixture, MAC would perform better.

However, as the number of contributors increased the risk would increase. Haned et al. [46] had analyzed the risk of dealing with three-, four-, and five-person mixture. They have done that by comparing the gold standard LR to the casework LR. The gold standard LR is when the number of contributors and genotypes are known which means the availability of all required information to compute LR per contributor. Authors showed the result and the implied thoughts of analyzing high order mixture in the forensic domain. Haned et al. argued that the low template DNA mixture of three-, four-, and five-person are common in forensic casework, yet it is hard to interpret.

Many methods are used today to evaluate the number of contributors in a sample such as [3, 9, 10, 47]. Perez et al. had created a strategy that could find out the number of contributors from two to four-person mixtures for both low template and high template DNA amounts. The proposed strategy helped to provide a useful tool to differentiate between high and low template two-, three-, and four-person mixtures. The four-person mixtures show some difficulties due to the allele sharing phenomena.

Egeland et al. focus on calculating the number of contributors in a mixture by maximizing the likelihood. The proposed approach is based on single SNP. The method tried to answer two questions: Is it a mixture? And if yes, then how many markers are required and how they should be selected. One of the recommendations that was driven from the result was regarding the number of markers needed to calculate the number of contributors which is 100 markers.

A typical algorithm for finding the best allele pair in a locus to interpret a mixture is presented in Algorithm 9.1. Such a process is essential when calculating the number of contributors in a DNA profile. Moreover, it is considered as a performance bottleneck.

On the other hand, Marciano and Adelman [48] proposed a machine learning approach that can estimate the number of contributors in a mixture. Their approach can handle mixtures with up to four contributors. The testing phase of this method shows a good result. The model first will be trained on a set of data, then it will be able to guess the number of contributors in DNA sample correctly. According to Marciano and Adelman, such a problem perfectly fits the domain of machine learning. The abundance of human mixture data can help to train the model well.

Yet, the machine learning approach suffers from several drawbacks. First, the quality of the result depends on the trained data. Uncorrected data may harm the system and lead to faulty results. Second, a training phase is always required before using the system. Such a phase is time-consuming and it might need to be redone many times. Third, the accuracy of the system starts to shape up after working

**Algorithm 9.1.** calculate locus's best allele pair that give best interpretation of the sample

```
procedure $GeneProbCalc(stepSize, noc, lname, revLoci, forLoci, DNAmass, AlleleAtLoci, LDO)$
//noc=number of contributors, lname=locus name, LDO= Locus Drop Out
    $locAlleles = AlleleAtLoci[locusname]$
    $MeanAndStd = Meanstd(locusname)$ //find mean and stddev
    for $i$=0 to $stepSize$ do
        $g$=random array between 0 and 1 with size noc
        for $j$=0 to $noc$ do
            for $k$=0 to 2 do
                $r$=Generate random number that does not exceed the interval of the locus
                $allele = AlleleRange[r]$ //get the allele in the selected interval for a specific locus
                Add allele to Peakscumulative
                $contMass$=g[i-1]*DNAmass
                if Rand() < ExpVal($locusName, LDO, contMass$) then
                    $ValidAlleles$.add($allele$)
                    $weight[allele] = weight[allele] + contMass$
                end if
            end for
        end for
        for $aName$=ValidAlleles.start to ValidAlleles.end do //aName=Allele Name
            if locAlleles $contains$ allele then
                $(mean, variance)$ =Meanstd($weight[allele]$) //find the mean and stddev
                if $revLoci[lName]$ && Rand()<ExpVal($lname, RevStutDropO, weight$) then
                    $rMu$=ExpVal2($lName, mean, weight[allele]$) $* allele.height$
                    $rSigma$=ExpVal2($lName, Stddev, weight[allele]$) $* allele.height$
                    $revAlleleStut = aName$ - 10 // get the reverse
                    $fowStutPeak = Peakscumulative[allele]$
                end if
                $means[revAlleleStut] = means[revAlleleStut] + rMu$
                $variances[revAlleleStut] = variances[revAlleleStut] + rSigma * r$
                if $forLoci[lName]$ && Rand()>ExpVal($lName, forStutDropO, weight$) then
                    $fMu$=ExpVal2($lName, Mean, allele.weight$) $* allele.height$
                    $fSigma$=ExpVal2($lName, Stddev, allele.weight$) $* allele.height$
                    $fowAlleleStut = aName$ + 10 // get forward
                    $fowStutPeak = Peakscumulative[allele]$
                end if
                $means[fowAlleleStut] = means[fowAlleleStut] + rMu$
                $variances[fowAlleleStut] = variances[fowAlleleStut] + rSigma * rSigma$
            end if
        end for
        for $temp$=Peakscumulative.start to Peakscumulative.end do
            $mean.add(temp.allele, MeanAndStd[0])$
            $variances.add(temp.allele, MeanAndStd[1] * MeanAndStd[1])$
        end for
        $locusProb$=calcLocusPeakHeightsProb($Peakscumulative, means, variances$)
        $Summation+ = locusProb$
        if $locusProb > currMax$ then
            $currMax = locusProb$
            for $alleleName$=selectedValidAlleles.start to selectedValidAlleles.end do
                $currMaxAlls.add(alleleName)$
            end for
        end if
    end for
    $result.add(Summation, currMax, currMaxAlls)$
    Return $result$
end procedure
```

**Algorithm 9.1** A typical algorithm for calculating locus's best allele pair that gives the best interpretation which helps in finding the number of unknowns in a DNA mixture (algorithm inspired by NOCIT tool [3])

large data of human DNA mixtures. Such data may not be easily available. Finally, as the maximum number of contributors increase, the accuracy of the prediction will be declined. Authors said that they didn't go up to five contributors because misclassification of five contributors may occur on four contributors mixture [48].

## 9.6 Software Tools for DNA Profiling

A number of tools are available that implement various DNA profiling methods. These include DNA MIX [49], Euroformix [34], LRmix [36], LRmix Studio [32, 50], TrueAllele [35], LikeLTD [38], Lab Retriever [15], CeesIt [37], NOCIt [3], DNAMixture [51], Forensim [52], MixtureCalc, Mixture Analysis [53], FamLink kinship [54], DNA Mixture Separator [55], and STRmix [56]. We will review the most notable tools in this section. At the end of this section, we will provide a comparison between the selected tools.

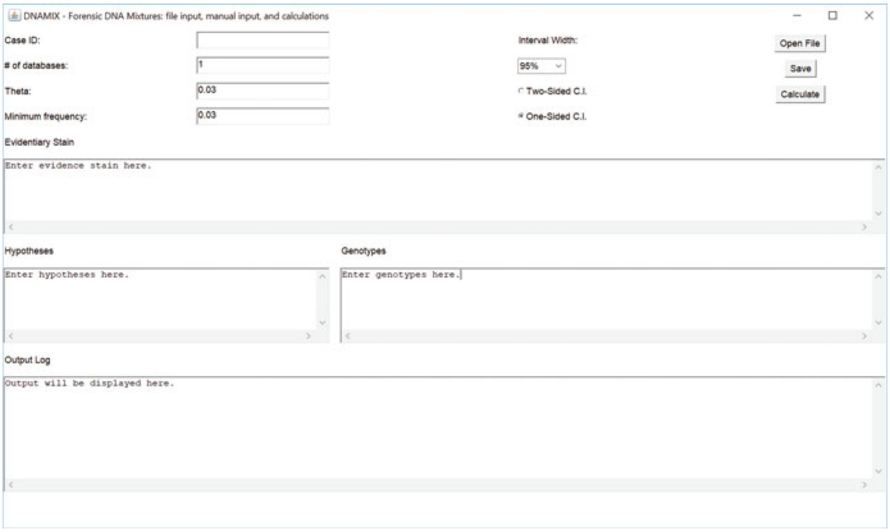
### *9.6.1 DNA Mix*

There are three versions of this software, and all of them are open sources. The third version is the most notable and powerful one among the three, and is based on [49]. This version is written in Java and is appropriate for complex mixtures as well as single-contributor stains. The software will ask for the database, stains, genotype, and hypothesis to be inputted.

On the latest version, dependency of all alleles was carried by contributors to the DNA mixture. All contributors will be assumed to belong to the same population, which will increase the effect that is being considered. Authors of DNA MIX did ignore the probability of null alleles. Thus, only homozygous contributors contribute a single allele to a profile. A simple GUI has been developed in this version (Fig. 9.4).

### *9.6.2 LRmix Studio*

LRmix Studio is a software designed to interpret complex DNA profiles. It was built on its previous version, which called LRmix; however, LRmix Studio is much faster and more flexible. It can measure the probative value of any (autosomal STR-based) DNA profile [50]. It can handle uncertainty in the DNA mixture from the allelic drop-out and drop-in. Moreover, it is written in Java, and it is open source under the GPLv3 license (Fig. 9.5).



DNAMIX - Forensic DNA Mixtures: file input, manual input, and calculations

Case ID:

# of databases: 1

Theta: 0.03

Minimum frequency: 0.03

Interval Width: 95%

○ Two-Sided C.I.

● One-Sided C.I.

Evidentiary Stain

Enter evidence stain here.

Hypotheses

Enter hypotheses here.

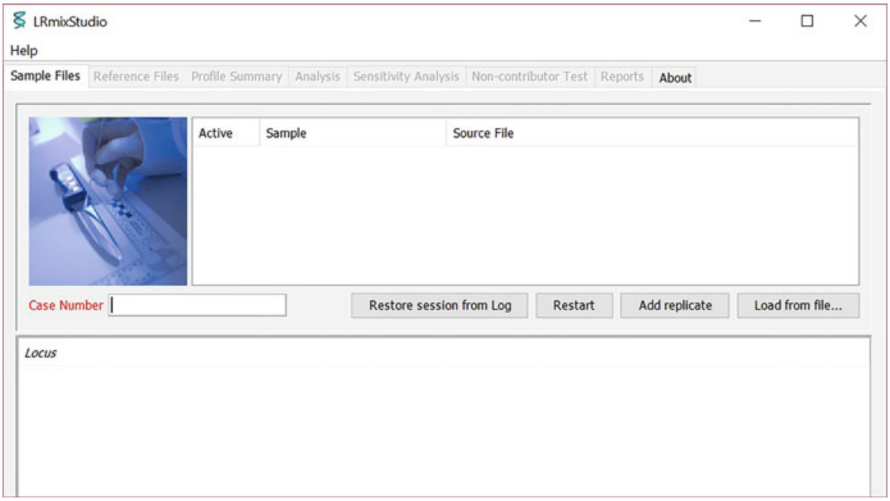
Genotypes

Enter genotypes here.

Output Log

Output will be displayed here.

**Fig. 9.4** The user interface for DNAMIX v3.2



LRmixStudio

Case Number

**Fig. 9.5** The user interface for LRmix Studio v2.1.3

This software is following the semi-continuous model of interpreting DNA profiles. Both the prosecution and the defense hypotheses assume that contributors are unrelated. Yet, under the defense hypothesis, contributors can be related to an unknown contributor.

If there are missing data in the reference profiles, LRmix Studio tool will be unable to work properly. Moreover, it cannot deconvolute DNA profile because it does not explicitly include the information of the peak height.

### 9.6.3 TrueAllele

TrueAllele is a software that computes DNA interpretation automatically. It can infer genetic profiles from all sorts of DNA samples. The software applies the continuous model; however, no open source version of the code is available. It was written in Matlab. Analysis followed by a comparison of TrueAllele is presented on [35] using real information that has been taken from actual cases.

TrueAllele can separate complex DNA profiles into its component genotypes. For each locus for a given contributor, the genotype and the uncertainty of that genotype are labeled using the probability distribution over the potentials of the allele pair.

TrueAllele applies the MCMC (Markov Chain Monte Carlo) statistical search to sample from the joint posterior probability distribution. For each locus in every contributor, the posterior probability for the genotype is going to be calculated. Thus, to remove the examination bias, the genotype will be inferred exclusively from the evidence data [57].

### 9.6.4 Lab Retriever

Lab Retriever [15] is a free software developed to estimate the likelihood ratios that combine a probability of drop-out. It was built on the top of another software called LikeLTD which was written in R language. The front end of the software was developed using CSS, JavaScript, Python, and HTML. On the back end, authors rewrote the code using C++ to acquire more speed. The software uses the semi-continuous model. It computes likelihood ratios for up to four unknown contributors to a DNA sample.

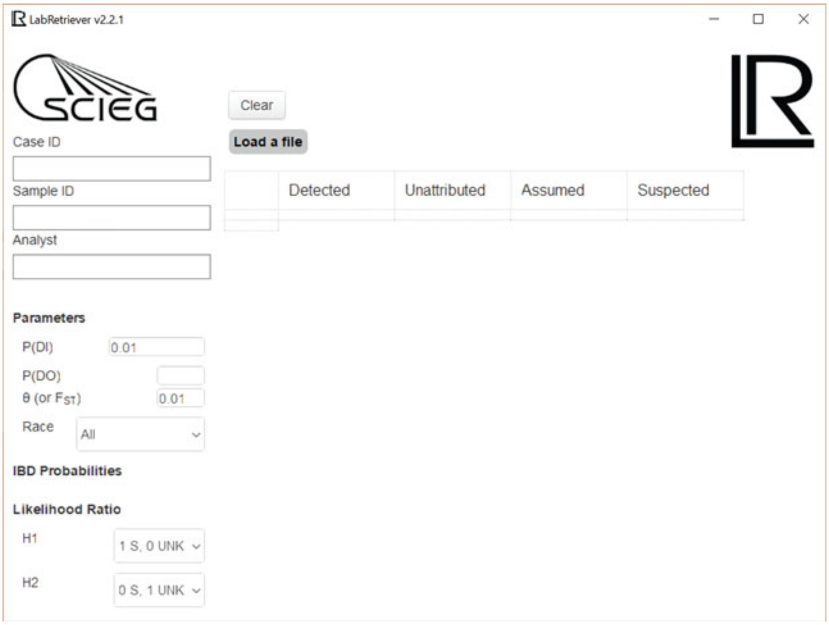
Lab Retriever uses dynamic programming to speed up the computation, which will avoid iterating over all genotypes. This tool estimates the likelihood ratio and compares the evidence under various hypotheses, while still allow for drop-out of alleles.

In order for the system to work, the user must specify as an input the following: The detected alleles in the evidence profile, the suspect genotype, the genotype of other contributors, the considered hypotheses, and the database of allele frequency.

Moreover, several parameters should be specified such as the probability of drop-in and drop-out and the co-ancestry adjustment value (Figs. 9.6 and 9.7).

### 9.6.5 CeesIt

CeesIt (CEES: computational evaluation of evidentiary signal) [37] is a method that integrates two features of the continuous approach to calculate the LR and its distribution which are conditioned on the defense hypothesis and the linked



Case ID

Sample ID

Analyst

Parameters

P(DI) 0.01

P(DO)

θ (or $F_{ST}$) 0.01

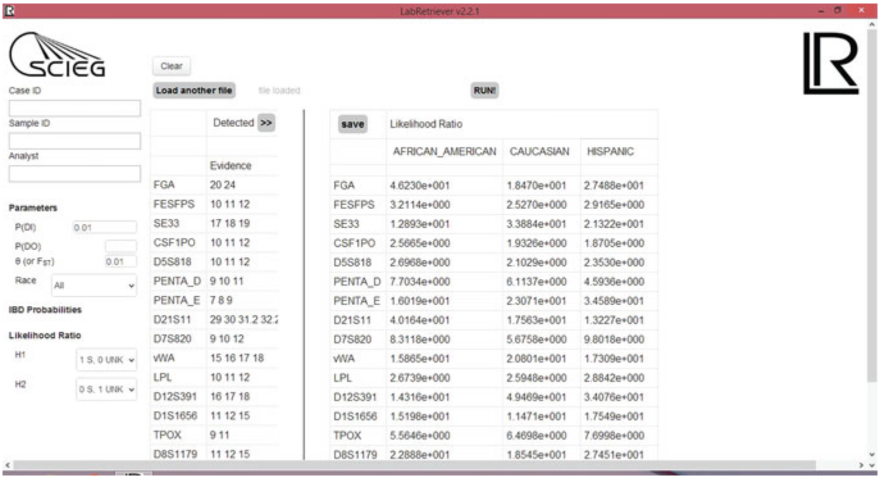
Race All

Likelihood Ratio

H1 1 S, 0 UNK

H2 0 S, 1 UNK

**Fig. 9.6** The user interface for Lab Retriever v2.2.1



Case ID

Sample ID

Analyst

Parameters

P(DI) 0.01

P(DO)

θ (or $F_{ST}$) 0.01

Race All

Likelihood Ratio

H1 1 S, 0 UNK

H2 0 S, 1 UNK

| | Detected >> |
|---|---|
| | |
| | Evidence |
| FGA | 20 24 |
| FESFPS | 10 11 12 |
| SE33 | 17 18 19 |
| CSF1PO | 10 11 12 |
| D5S818 | 10 11 12 |
| PENTA_D | 9 10 11 |
| PENTA_E | 7 8 9 |
| D21S11 | 29 30 31.2 32.2 |
| D7S820 | 9 10 12 |
| vWA | 15 16 17 18 |
| LPL | 10 11 12 |
| D12S391 | 16 17 18 |
| D1S1656 | 11 12 15 |
| TPOX | 9 11 |
| D8S1179 | 11 12 15 |

| save | Likelihood Ratio | | |
|---|---|---|---|
| | AFRICAN_AMERICAN | CAUCASIAN | HISPANIC |
| | | | |
| FGA | 4.6230e+001 | 1.8470e+001 | 2.7488e+001 |
| FESFPS | 3.2114e+000 | 2.5270e+000 | 2.9165e+000 |
| SE33 | 1.2893e+001 | 3.3884e+001 | 2.1322e+001 |
| CSF1PO | 2.5665e+000 | 1.9326e+000 | 1.8705e+000 |
| D5S818 | 2.6968e+000 | 2.1029e+000 | 2.3530e+000 |
| PENTA_D | 7.7034e+000 | 6.1137e+000 | 4.5936e+000 |
| PENTA_E | 1.6019e+001 | 2.3071e+001 | 3.4589e+001 |
| D21S11 | 4.0164e+001 | 1.7563e+001 | 1.3227e+001 |
| D7S820 | 8.3118e+000 | 5.6758e+000 | 9.8018e+000 |
| vWA | 1.5865e+001 | 2.0801e+001 | 1.7309e+001 |
| LPL | 2.6739e+000 | 2.5948e+000 | 2.8842e+000 |
| D12S391 | 1.4316e+001 | 4.9469e+001 | 3.4076e+001 |
| D1S1656 | 1.5198e+001 | 1.1471e+001 | 1.7549e+001 |
| TPOX | 5.5646e+000 | 6.4698e+000 | 7.6998e+000 |
| D8S1179 | 2.2888e+001 | 1.8545e+001 | 2.7451e+001 |

**Fig. 9.7** Lab Retrievers v2.2.1 interface in action

**Table 9.3** The running time of CeesIt under different number of contributors

| Number of contributors | Average time (minutes) |
|---|---|
| 1 | 7 |
| 2 | 50 |
| 3 | 140 |

p-value. It combines stutter, drop-out, and noise in its calculation. For calibration information, it uses a single-source sample with known genotypes. It calculates the LR for a selected Person of Interest (POI) on a questioned sample, together with the p-value and LR distribution.

To assess the performance of CeesIt, it was tested using 303 sample files ranging between one and three contributors, and the mass of the sample was ranging between 0.016 and 1 ng. The analysis results show a dependency on the number of contributors. Therefore, a good estimation for the number is critical for an accurate result.

The running time of the tool depends on the number of contributors. As the number increased, the time complexity will increase too. See Table 9.3 for more details on CeesIt running time.

Multithreaded is already implemented on CeesIt to increase resource utilization to acquire more speedup.

The software was written in Java and is available as a (.jar) file. An in-depth analysis of the software was presented on [37].

### *9.6.6 LikeLTD*

LikeLTD is a software that is used for computing the likelihood of DNA profile evidence, including complex mixtures. It has been written in R. However, since the fifth version, the computation-intensive areas in code have been rewritten in C to be executed in parallel. This software applies the continuous model of calculating the Likelihood ratio. These areas include the computation of genotype combinations for unknown contributors, computing allele doses for each genotype combination, dose adjustments for relatedness, heterozygosity, drop-out, and power.

The runtime of the peak height model is much slower than the runtime of the discrete model, yet it yields a higher evidence weight (see Table 9.4). The time complexity of the peak height model scales up with the number of unknown contributors, the number of observed peaks, and the number of replicates in the

**Table 9.4** The runtime of calculating the Weight of Evidence (WoE) using the two different models for the laboratory case [38]

| Hypothesis | Model | WOE | Runtime (Minutes) |
|---|---|---|---|
| Q/X + K1 + U1 | Discrete | 2.3 | 14 |
| | Peak height | 8.2 | 23 |
| Q/X + U1 + U2 | Discrete | 0.5 | 38 |
| | Peak height | 7.8 | 200 |

profile. Other parameters that increase the runtime are the modeling double-stutter or over-stutter. Parallelism was achieved on the C++ code by using a shared memory parallelism (OpenMP).

The runtime of the algorithms was recorded using a node with eight Intel Core I7 processors (3.1 Hgz per core) and with 15 Gb of RAM. The result is presented in Table 9.4. The first column describes the hypothesis that was applied. Two hypotheses were used. Q is a contributor to the crime scene profile under the Hp while X is the unknown individual under Hd that assumes to contribute to the profile instead of Q. The hypotheses may specify the number of K which represent the known contributors whereas U is the unknown contributors. The second column indicates the used model whether it uses discrete or peak height. The last two columns are showing the weight of evidence and the corresponding running time.

### *9.6.7 DNAMixture*

DNAMixture is a statistical model that calculates and analyzes DNA sample for one or more contributors [51]. It uses Bayesian network representation to speed up the computation and allow analysis of mixtures which contain several unknown contributors. Alleles observing process is objective, and it does not depend on a subjective preprocessing of the DNA profile [58]. Such a preprocessing can lead to more errors. The model has been tested on some real case and the results were sensible and robust [58].

This software has been written in R and follows the “fully continuous” statistical model. Its authors claim to develop all methodology within a framework for consistent analysis and transparency. The application does not have a graphical user interface, which requires a basic experience in R. DNAMixture relies on an R package called “Hugin.” Hugin is used to compute the Bayesian network. DNAMixture is not parallelized, yet the Hugin package is.

The computational complexity of the model depends on several factors. The running time of DNAMixture when there are five unknown contributors took 3 h on a regular desktop machine [58]. Authors claim that they perform analysis on several cases which takes 35 min; when they analyze the same cases using another tool called TrueAllele [57], the runtime goes to 36 h [58].

### *9.6.8 Kongoh*

Kongoh [59] is an open-source application based on the continuous model for interpreting DNA sample. This model deals with artifacts and allelic drop-out ratio on its calculation, but it doesn't consider allele drop-in probability. It performs a Monte Carlo simulation based on the probability distributions of the given parameters. Next, gamma distributions will be used to approximate the peak heights that were generated by the simulation.

The number of contributors is not required to be given as an input. Kongoh can determine the number of contributors when it ranges from one to four. However, the accuracy will be affected when the number of contributors increases to reach 33% when the number of contributors becomes four. Kongoh can handle sample with a small amount of DNA, and also with degraded DNA samples. The software has a graphical user interface. R language was used to write Kongoh and its source code is available online.

On a standard desktop computer, one mixture might take around 10 h when hypothesizing 1–4 contributors. However, when hypothesizing 1–3 contributors, the runtime will decrease remarkably to a few minutes [59]. Its performance was compared to EuroForMix (version 1.7) and LRmix Studio (version 2.1.3) in [59]. In the future, authors of Kongoh are looking to use newer STR typing kits with higher sensitivity.

### *9.6.9 EuroForMix*

EuroForMix is a software based on the fully continuous approach to estimate STR DNA profiles from a complex DNA sample of contributors with artifacts. It is available as an open source. EuroForMix was written in R language. Nonetheless, the likelihood function was written in C++ to speed up the computation. The software introduces a parallel implementation, since the v0.5.0, using snow R package. The parallel implementation will only be considered when a number of unknowns are at least 3 (not performed yet for database searching or non-contributor simulation). A number of processes will be similar to the number of random start points required in the optimization.

Euroformix requires a significant amount of computational time when the number of unknown contributors is four or more. Table 9.5 gives an approximation time complexity for each number of unknown contributors. From the table, it is clear that the time consumed when we have four unknown contributors was too much. Column 1 describes the number of contributors while Column 2 gives the corresponding time taken.

**Table 9.5** An approximate overview of the time taken to calculate the LR depend on the number of unknown contributors [60]

| Number of unknown contributors | Runtime |
|---|---|
| 1 | 1 s |
| 2 | 1 min |
| 3 | 30 min |
| 4 | 24 h |

**Table 9.6** The runtime using a different maximum number of contributors [3]

| Number of contributors | Time range (Mode) |
|---|---|
| 1 | <1 min (0.2 min) |
| 2 | 15–30 min (17 min) |
| 3 | 30 min–1.5 h (1 h) |
| 4 | 1–5 h (4 h) |
| 5 | 5–20 h (14 h) |

### *9.6.10 NOCIt*

NOCIt [3] analyzes the DNA sample to calculate the number of contributors in a mixture. Java programming language was used to write the software. It determines the number of contributors (from 1 to 5). NOCIt can only interpret an autosomal STRs data which are independent of each other. Moreover, the software is not developed to deal with a stutter.

The execution time of [3] depends on the maximum number of contributors, the number of loci/alleles considered and the processing speed of the computer. It is also dependent on whether multiple runs of NOCIt are occurring at the same time, i.e., two NOCIt interfaces are open at once and running two separate samples. Table 9.6 provides the runtime of NOCIt. The first column gives the number of contributors, whereas the second column describes the range of time taken to analyze that number. The result was collected from a dual-core laptop with Intel® CoreTM i5-3380 CPU @ 2.9 GHz (Fig. 9.8).

### *9.6.11 STRmix*

STRmix is a probabilistic genotyping application which performs the continuous model of interpreting the DNA profile. The DNA profile interpretation is based on a Markov Chain Monte Carlo (MCMC) sampling model [39]. It calculates the likelihood ratio which is the probability of the DNA evidence under two hypotheses, defense and prosecution hypotheses.

It was built to interpret single and mixed DNA profiles. Moreover, it follows the SWGDAM recommendations. It utilizes information that extracts from a DNA sample, such as peak height, to calculate the probability of a DNA profile for all possible genotype combinations. The software considers aspects such as allele drop-in, allele drop-out, and stutter. The software has been written in Java, and it's only available for purchase.

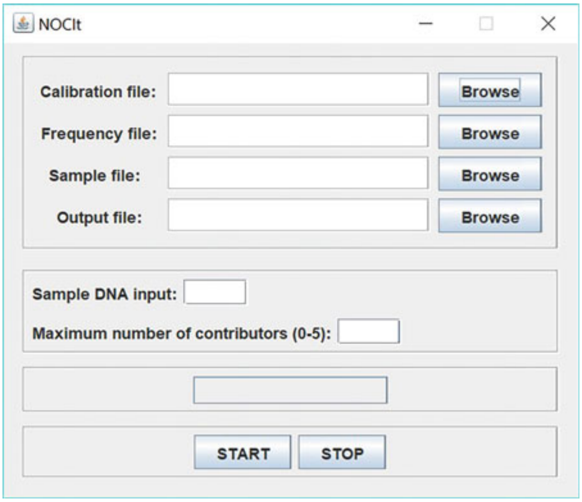


**Fig. 9.8** The user interface for NOCIt v15

Moretti et al. [39] had tested STRmix and they argued that it can be used to interpret single-source profiles and mixtures of two, three, four, and five persons.

### *9.6.12 A Comparison of the DNA Profiling Tools*

A general comparison between the selected tools is presented in Table 9.7. The first column gives the names of the software. Columns 2–8 provide information about various features of the software. Column 2 gives information on whether the software has a GUI or not. Column 3 and 4 are illustrating if the selected software considers the phenomena of drop-in and stutter on its interpretation. Column 5 describes the model that used to calculate LR. The sixth column describes the programming language that used to build the selected software. Column 7 indicates the availability of source code. The last column describes the used parallel framework. Note that the table is missing some information due to either the lack of resource for some software or because of the inability to access the software's source code.

A timeline that shows the history of introduction of the compared tools is presented in Fig. 9.9.

## 9.7 Conclusion

Interpreting DNA mixture is a common practice in forensic science domain. It is a complicated process that requires an extended period of time. We gave an overview of the DNA profiling field. A historical background, along with its application

**Table 9.7** A general comparison between the review softwares

| | GUI | Drop-in | Stutter | Calculation model | Language | Source Code | Parallelism |
|---|---|---|---|---|---|---|---|
| LRmix studio [15, 32, 50] | Yes | Yes | – | Semi-continuous | Java | Yes | Java multithreading |
| TrueAllele [34, 35] | Yes | Yes | Yes | Continuous | Matlab | No | – |
| DNAMIX V.3 [49] | Yes | – | – | – | Java | Yes | No |
| Euroformix [34] | Yes | Yes | Yes | Continuous | R, C++ | Yes | Snow package |
| CeesIt [37] | Yes | Yes | Yes | Continuous | Java | No | Java multithreading |
| NOCIt [3] | Yes | Yes | Yes | Continuous | Java | No | Java multithreading |
| DNAMixtures [51] | No | Yes | Yes | Continuous | R | Yes | No |
| Kongoh [59, 61] | Yes | No | Yes | Continuous | R | Yes | Snow package |
| LikeLTD [38] | No | Yes | Yes | Continuous | R, C | Yes | OpenMP |
| Lab Retriever [15] | Yes | Yes | – | Semi-continuous | C++ | Yes | No |
| STRmix [39, 56] | Yes | Yes | Yes | Continuous | Java | No | – |

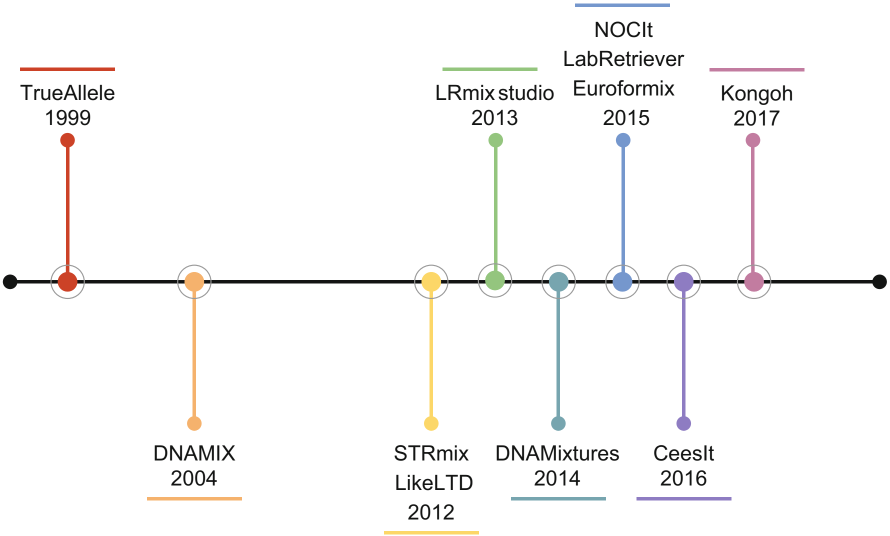


**Fig. 9.9** DNA mixture analysis tools introduced over the time. This timeline describes the year of introduction of each tool

was mentioned. We, then, discuss the needed steps to sample a DNA mixture and what are the required technologies. After that, we reviewed the literature based on their classification into describing DNA profiling in general. We focus later on approaches that follow the Likelihood Ratio model. We also reviewed the various tools and compared their performance and accuracy. This is an extended version of our earlier work [4].

In the end, we would suggest the use of Euroformix and LikeLTD for DNA profiling since they are already performing parallelism. They both utilize most of the available information in the DNA sample because they follow the continuous model for calculating the LR value. The source code for the two software is available for assessment and modification. However, Euroformix provides a GUI which gives it a slight advantage over LikeLTD for users who have no technological expertise.

A frequent necessity to apply these tests might raise the need to speed up the runtime of such analysis. The computational complexity has been the major deterring factor holding the area advancements and applications. An improvement would give a chance to interpret mixtures with a larger number of unknowns and within a shorter time frame. The investigation of the relevant literature reveals that the current approaches for parallelization of DNA profiling rely on shared memory parallelization. A distributed implementation is needed to speed up the computations allowing for the use of a large number of cores and processors. This is our ongoing research, which will be reported in the near future. Faster interpretations of DNA mixtures with a large number of unknowns and higher accuracies are expected to open up new frontiers for DNA profiling in the smart societies era.

In the coming years, the complete genome sequencing technologies in a single or only a few cells will be easily available. These technologies may change the situation of DNA profiling completely. In this case, it is obvious to prepare appropriate statistical methods for that. It will be, therefore, important to prepare the mathematical and statistical algorithms for complete-genome-sequencing-based DNA profile. High-performance computing will play a key role in speeding up DNA profiling methods, particularly those HPC techniques which exploit domain-specific data and algorithmic patterns [62], system heterogeneity (e.g., disks for space, and accelerators for speed) for its advantage [63], and virtual organization models (similar to grids [64]) for information sharing across organizational boundaries. Hierarchical system structures will be needed to localize and optimize data and computations [65]. Internet of Things (IoT) would be integrated in smart city systems to create innovative services [7] and deal with big data-related challenges [6]. Mobile, fog, and cloud computing [5, 66–68] will enable dynamic system environments, seamlessly connecting users and systems.

**Acknowledgments** The work carried out in this chapter is supported by the HPC Center at the King Abdulaziz University.

## References

1. The American Heritage medical dictionary. Houghton Mifflin Co., Boston (2007)
2. Butler, J.M.: Fundamentals of Forensic DNA Typing. Academic Press/Elsevier (2010)
3. Swaminathan, H., Grgicak, C.M., Medard, M., Lun, D.S.: NOCIt: a computational method to infer the number of contributors to DNA samples analyzed by STR genotyping. Forensic Sci. Int. Genet. **16**, 172–180 (2015)

4. Alamoudi, E., Mehmood, R., Albeshri, A., Gojobori, T.: DNA profiling methods and tools: a review. In: Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST. pp. 216–231. Springer, Cham (2018)
5. Arfat, Y., Aqib, M., Mehmood, R., Albeshri, A., Katib, I., Albogami, N., Alzahrani, A.: Enabling smarter societies through Mobile big data fogs and clouds. Procedia Comput. Sci. **109**, 1128–1133 (2017)
6. Alam, F., Mehmood, R., Katib, I., Albogami, N.N., Albeshri, A.: Data fusion and IoT for smart ubiquitous environments: a survey. IEEE Access. **5**, 9533–9554 (2017)
7. Mehmood, R., Alam, F., Albogami, N.N., Katib, I., Albeshri, A., Altowaijri, S.M.: UTiLearn: a personalised ubiquitous teaching and learning system for smart societies. IEEE Access. **5**, 2615–2635 (2017)
8. Butler, J.M.: The future of forensic DNA analysis. Philos. Trans. R. Soc. Lond. Ser. B Biol. Sci. **370**, 577–579 (2015)
9. Paoletti, D.R., Krane, D.E., Raymer, M.L., Doom, T.E.: Inferring the number of contributors to mixed DNA profiles. IEEE/ACM Trans. Comput. Biol. Bioinforma. **9**, 113–122 (2012)
10. Perez, J., Mitchell, A.A., Ducasse, N., Tamariz, J., Caragine, T.: Estimating the number of contributors to two-, three-, and four-person mixtures containing DNA in high template and low template amounts. Croat. Med. J. **52**, 314–326 (2011)
11. Gill, P., Haned, H.: A new methodological framework to interpret complex DNA profiles using likelihood ratios. Forensic Sci. Int. Genet. **7**, 251–263 (2013)
12. Weedn, V.W., Foran, D.R.: Forensic DNA typing. In: Molecular pathology in clinical practice. pp. 793–810. Springer International Publishing, Champions (2016)
13. Monich, U.J., Grgicak, C., Cadambe, V., Wu, J.Y., Wellner, G., Duffy, K., Medard, M.: A signal model for forensic DNA mixtures. In: 2014 48th Asilomar Conference on Signals, Systems and Computers. pp. 429–433. IEEE (2014)
14. Tao, R., Wang, S., Zhang, J., Zhang, J., Yang, Z., Sheng, X., Hou, Y., Zhang, S., Li, C.: Separation/extraction, detection, and interpretation of DNA mixtures in forensic science (review)
15. Inman, K., Rudin, N., Cheng, K., Robinson, C., Kirschner, A., Inman-Semerau, L., Lohmueller, K.E.: Lab retriever: a software tool for calculating likelihood ratios incorporating a probability of drop-out for forensic DNA profiles. BMC Bioinformatics. **16**, 298 (2015)
16. Schmidt, B., Hildebrandt, A.: Next-generation sequencing: big data meets high performance computing. Drug Discov. Today. **22**, 712–717 (2017)
17. Chang, Y.-J., Chen, C.-C., Chen, C.-L., Ho, J.-M.: A de novo next generation genomic sequence assembler based on string graph and MapReduce cloud computing framework. BMC Genomics. 13 Suppl 7, S28 (2012)
18. Li, D., Liu, C.-M., Luo, R., Sadakane, K., Lam, T.-W.: MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Bioinformatics. **31**, 1674–1676 (2015)
19. Liu, Y., Schmidt, B., Maskell, D.L.: DecGPU: distributed error correction on massively parallel graphics processing units using CUDA and MPI. BMC Bioinformatics. **12**, 85 (2011)
20. Erbert, M., Rechner, S., Müller-Hannemann, M.: Gerbil: a fast and memory-efficient k-mer counter with GPU-support. Algorithms Mol. Biol. **12**, 9 (2017)
21. Varma, B.S.C., Paul, K., Balakrishnan, M., Lavenier, D.: FAssem: FPGA Based Acceleration of De Novo Genome Assembly. In: 2013 IEEE 21st Annual International Symposium on Field-Programmable Custom Computing Machines. pp. 173–176. IEEE (2013)
22. Ramachandran, A., Heo, Y., Hwu, W.M., Ma, J., Chen, D.: FPGA accelerated DNA error correction, https://iwe.pure.elsevier.com/en/publications/fpga-accelerated-dna-error-correction, (2015)
23. Kang, S.J., Lee, S.Y., Lee, K.M.: Performance comparison of OpenMP, MPI, and MapReduce in practical problems. Adv. Multimed. **2015**, 1–9 (2015)
24. Hamidi, B., Hamidi, L.: Synchronization Possibilities and Features in Java, vol. 1, p. 75 (2015)
25. Carpenter, B., Getov, V., Judd, G., Skjellum, A., Fox, G.: MPJ: MPI-like message passing for Java. Concurr. Pract. Exp. **12**, 1019–1038 (2000)

26. Memeti, S., Pllana, S.: A machine learning approach for accelerating DNA sequence analysis. Int. J. High Perform. Comput. Appl. 1–17
27. Bell, G., Gray, J.: What' S Next in Computing ? 45, 91–95 (2002)
28. Diegoli, T.M., Rohde, H., Borowski, S., Krawczak, M., Coble, M.D., Nothnagel, M.: Genetic mapping of 15 human X chromosomal forensic short tandem repeat (STR) loci by means of multi-core parallelization. Forensic Sci. Int. Genet. **25**, 39 (2016)
29. Laguna, I., Ahn, D.H., De Supinski, B.R., Gamblin, T., Lee, G.L., Schulz, M., Bagchi, S., Kulkarni, M., Zhou, B., Chen, Z., Qin, F.: Debugging high-performance computing applications at massive scales. Commun. ACM. **58**, 72–81 (2015)
30. Butler, J.M.: Advanced topics in forensic DNA typing: interpretation
31. Bille, T., Bright, J.-A., Buckleton, J.: Application of random match probability calculations to mixed STR profiles. J. Forensic Sci. **58**, 474–485 (2013)
32. Garofano, P., Caneparo, D., D'Amico, G., Vincenti, M., Alladio, E.: An alternative application of the consensus method to DNA typing interpretation for low template-DNA mixtures. Forensic Sci. Int. Genet. Suppl. Ser. **5**, e422–e424 (2015)
33. Kelly, H., Bright, J.-A., Buckleton, J.S., Curran, J.M.: A comparison of statistical models for the analysis of complex forensic DNA profiles. Sci. Justice. **54**, 66–70 (2014)
34. Bleka, Ø., Storvik, G., Gill, P.: 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**, 35 (2016)
35. Perlin, M.W., Dormer, K., Hornyak, J., Schiermeier-Wood, L., Greenspoon, S.: TrueAllele casework on Virginia DNA mixture evidence: computer and manual interpretation in 72 reported criminal cases. PLoS One. **9**, e92837 (2014)
36. Gill, P., Haned, H., Eduardoff, M., Santos, C., Phillips, C., Parson, W.: The Open-source software LRmix can be used to analyse SNP mixtures. Forensic Sci. Int. Genet. Suppl. Ser. **5**, e50 (2015)
37. Swaminathan, H., Garg, A., Grgicak, C.M., Medard, M., Lun, D.S.: CEESIt: a computational tool for the interpretation of STR mixtures. Forensic Sci. Int. Genet. **22**, 149–160 (2016)
38. Balding, D.J., Steele, C.: The likeLTD software: an illustrative analysis, explanation of the model, results of performance tests and version history. UCL Genet. Inst. **1**, 1–49 (2014)
39. Moretti, T.R., Just, R.S., Kehl, S.C., Willis, L.E., Buckleton, J.S., Bright, J.-A., Taylor, D.A., Onorato, A.J.: Internal validation of STRmix™ for the interpretation of single source and mixed DNA profiles. Forensic Sci. Int. Genet. **29**, 126–144 (2017)
40. Taylor, D., Bright, J.-A., Buckleton, J.: Interpreting forensic DNA profiling evidence without specifying the number of contributors. Forensic Sci. Int. Genet. **13**, 269–280 (2014)
41. Russell, D., Christensen, W., Lindsey, T.: A simple unconstrained semi-continuous model for calculating likelihood ratios for complex DNA mixtures. Forensic Sci. Int. Genet. Suppl. Ser. **5**, e37–e38 (2015)
42. Paoletti, D.R., Doom, T.E., Krane, C.M., Raymer, M.L., Krane, D.E.: Empirical analysis of the STR profiles resulting from conceptual mixtures. J. Forensic Sci. **50**, JFS2004475–JFS2004476 (2005)
43. Biedermann, A., Bozza, S., Konis, K., Taroni, F.: Inference about the number of contributors to a DNA mixture: comparative analyses of a Bayesian network approach and the maximum allele count method. Forensic Sci. Int. Genet. **6**, 689–696 (2012)
44. Haned, H., Pène, L., Sauvage, F., Pontier, D.: The predictive value of the maximum likelihood estimator of the number of contributors to a DNA mixture. Forensic Sci. Int. Genet. **5**, 281–284 (2011)
45. Haned, H., Pène, L., Lobry, J.R., Dufour, A.B., Pontier, D.: Estimating the number of contributors to forensic DNA mixtures: does maximum likelihood perform better than maximum allele count? J. Forensic Sci. **56**, 23–28 (2011)
46. Haned, H., Benschop, C.C.G., Gill, P.D., Sijen, T.: Complex DNA mixture analysis in a forensic context: evaluating the probative value using a likelihood ratio model. Forensic Sci. Int. Genet. **16**, 17–25 (2015)

47. Egeland, T., Dalen, I., Mostad, P.F.: Estimating the number of contributors to a DNA profile. Int. J. Legal Med. **117**, 271–275 (2003)
48. Marciano, M.A., Adelman, J.D.: PACE: probabilistic assessment for contributor estimation—a machine learning-based assessment of the number of contributors in DNA mixtures. Forensic Sci. Int. Genet. **27**, 82–91 (2017)
49. Curran, J.M., Triggs, C.M., Buckleton, J., Weir, B.S.: Interpreting DNA mixtures in structured populations. J. Forensic Sci. **44**, 987–995 (1999)
50. Haned, H., De Jong, J.: LRmix Studio 2.1 user manual. (2016)
51. Graversen, T.: Statistical and Computational Methodology for the Analysis of Forensic DNA Mixtures with Artefacts, https://ora.ox.ac.uk/objects/uuid:4c3bfc88-25e7-4c5b-968f-10a35f5b82b0, (2014)
52. Forensim: An open-source initiative for the evaluation of statistical methods in forensic genetics. Forensic Sci. Int. Genet. 5, 265–268 (2011)
53. Gill, P., Sparkes, R., Pinchin, R., Clayton, T., Whitaker, J., Buckleton, J.: Interpreting simple STR mixtures using allele peak areas. Forensic Sci. Int. **91**, 41–53 (1998)
54. Kling, D., Egeland, T., Tillmar, A.O.: FamLink – a user friendly software for linkage calculations in family genetics. Forensic Sci. Int. Genet. **6**, 616–620 (2012)
55. Tvedebrink, T., Eriksen, P.S., Mogensen, H.S., Morling, N.: Evaluating the weight of evidence by using quantitative short tandem repeat data in DNA mixtures. J. R. Stat. Soc. Ser. C Applied Stat. **59**, 855–874 (2010)
56. Developmental validation of STRmix™, expert software for the interpretation of forensic DNA profiles. Forensic Sci. Int. Genet. 23, 226–239 (2016)
57. Perlin, M.W., Hornyak, J.M., Sugimoto, G., Miller, K.W.: TrueAllele genotype identification on DNA mixtures containing up to five unknown contributors*, vol. 60, p. 857 (2015)
58. Cowell, R.G., Graversen, T., Lauritzen, S.L., Mortera, J.: Analysis of forensic DNA mixtures with artefacts. J. R. Stat. Soc. Ser. C Applied Stat., 64. 1–48 (2015)
59. Manabe, S., Morimoto, C., Hamano, Y., Fujimoto, S., Tamaki, K.: Development and validation of open-source software for DNA mixture interpretation based on a quantitative continuous model. PLoS One. **12**, e0188183 (2017)
60. Bleka, Ø.: An introduction to EuroForMix (v1.8). 2016, 1–59 (2016)
61. Manabe, S.: Kongoh version 1.0.1 User Manual. 1–12 (2017)
62. Mehmood, R., Crowcroft, J.: Parallel iterative solution method for large sparse linear equation systems. Comput. Lab. Univ. 22 (2005)
63. Mehmood, R.: Serial disk-based analysis of large stochastic models. In: Validation of Stochastic Systems. pp. 230–255. Springer, Berlin, (2004)
64. Altowaijri, S., Mehmood, R., Williams, J.: A quantitative model of grid systems performance in healthcare organisations. In: 2010 International Conference on Intelligent Systems, Modelling and Simulation. pp. 431–436. IEEE (2010)
65. Mehmood, R., Crowcroft, J., Hand, S., Smith, S.: Grid-level computing needs pervasive debugging. In: The 6th IEEE/ACM International Workshop on Grid Computing, 2005. p. 8 pp. IEEE (2005)
66. Tawalbeh, L.A., Mehmood, R., Benkhlifa, E., Song, H.: Mobile cloud computing model and big data analysis for healthcare applications. IEEE Access. **4**, 6171–6180 (2016)
67. Tawalbeh, L.A., Bakhader, W., Mehmood, R., Song, H.: Cloudlet-Based Mobile Cloud Computing for Healthcare Applications. In: 2016 IEEE Global Communications Conference (GLOBECOM). pp. 1–6. IEEE (2016)
68. Muhammed, T., Mehmood, R., Albeshri, A., Katib, I.: UbeHealth: A personalized ubiquitous cloud and edge-enabled networked healthcare system for smart cities, https://ieeexplore.ieee.org/document/8382164/, (2018)