Internal Report CS Bioinformatics Track 15–01 July 2015

Leiden University Computer Science Bioinformatics Track

Characterisation and Filtering of Systemic Noise in NGS Data with Applications in Forensics

Jerry Hoogenboom

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Characterisation and Filtering of Systemic Noise in Next Generation Sequencing of Short Tandem Repeats with Applications in Forensics

Jerry Hoogenboom

July, 2015

Abstract

In forensic analysis of DNA samples, short tandem repeat (STR) regions in the genome are selectively amplified with PCR to subsequently determine which alleles are present. The DNA polymerase used to amplify STRs with PCR is known to 'slip' occasionally, which results in an additional PCR product having one repeat unit more or less than the original allele. These so-called 'stutter products' may coincide with other alleles present in the sample and may therefore overshadow very small components of mixed DNA samples.

In this project, methods are developed to characterise the appearance of stutter products and other systemic noise in NGS data, to filter this noise in forensic DNA samples, and thereby to identify the true alleles in these samples, increasing the sensitivity of forensic DNA analysis for samples with very small contributions.

1 Introduction

Currently in forensic DNA analysis, a fixed set of STR markers in the genome is used to match DNA samples with a database of DNA profiles. An STR marker is a region in the genome with a highly repetitive DNA sequence (Figure 1). The repeat units are typically two to about 10 nucleotides in length and are repeated up to tens of times in a row. Because of the large number of possible combinations, the probability that any two people on Earth have the same number of repeats for each of the markers used is very close to zero [1]. Although the precise set of STR markers used in forensics varies worldwide, the core repeat typically consists of units of three to five nucleotides.

Figure 1. Anatomy of an STR region. Bottom: part of a DNA strand. Top: repeat elements of an STR on this DNA strand. This hypothetical STR consists of three repetitive regions, each having a different repeated sequence. Flanking regions are indicated by longer contiguous bars at the left and right ends.

These STR marker regions are selectively amplified using PCR. Capillary electrophoresis (CE) is then used to determine which alleles (identified by their lengths) of each of these markers are present in the sample. This results in a number of 'peaks' corresponding to the lengths of the different marker sequences. The height of each peak is related to the number of DNA molecules with that particular length that appear in the sample after amplification.

The DNA samples that appear in forensics are often mixtures of two or more individuals and the individual components often appear in different quantities. Detecting a minor contribution in a mixture, e.g., a contribution that comprises of only a few per cent of the entire mixture is challenging. The reverse is also true; in mixtures consisting of two components of 50% detecting either contribution becomes troublesome but the availability of all alleles can still be used to analyse whether somebody could be one of the contributors to the mixture [2].

Making matters worse, the PCR amplification step is not free of errors. The polymerase enzyme used in PCR is known to 'slip' when it encounters long stretches of repetitive sequences, omitting one or two repeats or even adding one (Figure 2) [3]. Because PCR is an iterative process, errors that appear in the first few cycles are also amplified and during this amplification more errors may be introduced. These imperfections of PCR result in smaller peaks that appear in addition to the 'real' peaks (Figure 3). These so-called 'stutter peaks' may overshadow the small contributions in a DNA sample.

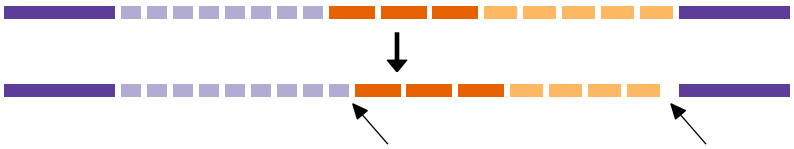

Figure 2. PCR artefacts caused by polymerase slipping. The PCR product will consist mainly of the original DNA sequence (top), but will also contain copies where an additional repeat is added (bottom, left arrow) or a repeat has been omitted (bottom, right arrow). Multiple such errors may have accumulated after multiple PCR iterations (bottom).

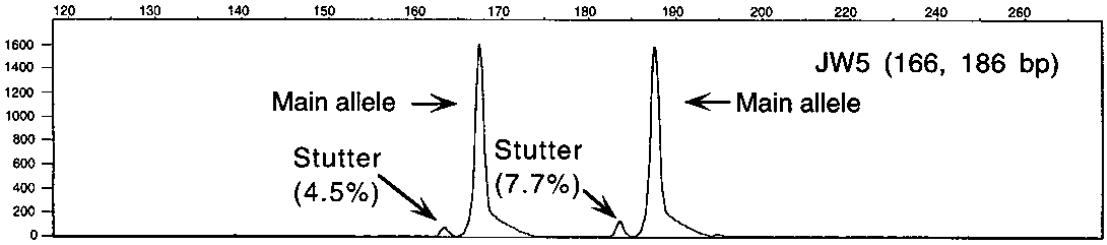

Figure 3. Example electropherogram from [4] showing stutter peaks. The x-axis corresponds to the number of base pairs. The y-axis corresponds to the amount of measured signal (i.e., DNA molecules).

1.1 Current state of the art

CE analysis has a number of major limitations. For instance, CE only analyses the length of a DNA molecule, ignoring any additional variation that may be present in the analysed region. Furthermore, because multiple markers may yield alleles of the same length, a number of different fluorescent dyes are used to simultaneously detect alleles of markers having overlapping size ranges. By assigning a different dye to each such marker, the colour of the signal can be used to determine from which marker the allele originated. Because there is some overlap in the emission and detection of the fluorescent colours, a very high signal of one colour may cause a so-called 'pull-up' peak to appear at a different colour [5, 6]. This pull-up peak may be incorrectly interpreted as an allele of the other marker, or it may mask the presence of a true allele of the other marker having the same length. This problem puts an upper limit on the detection range of CE analysis. Conversely, when a signal is too low, the corresponding peak may not exceed the background noise and consequently, an allele may be missed. This

detection range limits the ability to analyse the minor contribution in very uneven two-person mixtures, because after increased amplification to bring the minor contribution's allele peaks into the detection range, some of these peaks may have been overshadowed by pull-up artefacts from the major contribution.

Finally, depending on the fluorescent label used, CE usually needs several thousand DNA molecules before the signal becomes detectable. This results in skewed allele balances for peaks that are close to the lower detection limit. Allele balances are an important guide in the interpretation of mixtures [7].

Current research is targeted at replacing the CE step with a DNA sequencer. With a DNA sequencer, the actual sequence of the amplicon is obtained. Also, DNA sequencers have a much broader detection range than can be achieved with CE since the amount of sequences generated for a sample can easily be increased. With DNA sequencing it may be possible to detect even smaller components in mixed DNA samples than is currently possible. In addition, the precise DNA sequence can be taken into consideration, introducing extra specificity by the existence of SNPs (single nucleotide polymorphisms), alternative repeat units, and possible interruptions in stretches of repeats in the marker regions used.

The DNA sequencing reads are matched against a library of known marker alleles using TSSV [8]. In this library, each marker is described by its flanking sequences and a regular expression describing the repeat units it consists of, along with the expected number of times each repeat unit appears. Because the reads contain the DNA sequence, the number of repeats of each unit can be determined exactly. TSSV counts the number of reads that match the sequence of each allele found in the sample.

Because the PCR step is unaltered, the DNA sequencing method suffers from the same PCR stutter problem. However, the DNA sequencing method yields absolute numbers of reads and the precise number of repeats they contain for each repeat unit which opens opportunities for more detailed analysis.

1.2 The next step

This project, we target the characterisation of the appearance of stutter peaks and other systemic noise in STR sequencing data. The aim is to construct a statistical model of systemic noise with sufficient precision to be able to filter stutter peaks and other noise, potentially revealing small contributions in the DNA sample.

There are some obvious factors that influence the rate of stutters, for example DNA polymerase is much more likely to slip when confronted with a repeat unit that is repeated 20 times than when it is confronted with a repeat unit that is repeated only 10 times [4, 9, 10]. Interruptions in the repeat as small as a single base difference in one unit greatly limits the amount of stutter [11]. It has also been observed that '−1 stutter' (the omission of one repeat unit) is much more likely than '+1 stutter' (the addition of a repeat unit) [9]. Furthermore, shorter repeat units (e.g., dinucleotide) are more susceptible to stutter than longer ones (e.g., tetranucleotide) [3, 4, 9]. Because DNA sequencing yields clean and detailed data, it is expected that the influence of these factors (amongst others) can be estimated from a carefully defined set of training samples with high precision.

2 Materials and Methods

2.1 Sample preparation

Sequencing data was generated using the MiSeq Sequencer (Illumina). As template for sequencing libraries, a multiplex PCR was performed in which fragments of 17 STRs and Amelogenin were amplified. Adapters for sequencing (containing sample-specific barcodes) were ligated to the PCR products using the KAPA High Throughput Library Preparation Kit. A total of 271 reference samples were used in this project.

2.2 Paired-end read merging

Almost all STR amplicons are shorter than the maximum read length of 300bp offered by the MiSeq platform and are thus sequenced full length in both directions (paired-end sequencing). However, two samples in our data have FGA allele 45.2, the length of which slightly exceeds the maximum read length. This allele can therefore only be detected if the read pairs are merged together.

To merge a read pair, the reads are aligned such that the largest possible overlap is obtained while allowing a limited number of base mispairings in the overlapped region. When reads with a highly repetitive sequence are merged together this way, the resulting merged read may be one or more repeats shorter than the actual DNA molecule that was sequenced. It is therefore essential that the entire repeat structure was still contained in both reads and only the flanking sequences are extended by merging them.

Merging of paired-end reads was done using a modified version of FLASH 1.2.11, which writes the portion of the merged reads outside of the overlapped region in lower case [12].[1] Any reads in which either flanking sequence was completely in lower case was subsequently removed from the data because of the possibility that they were one or more repeats shorter than the actual DNA molecules they were generated from.

2.3 Linking reads to markers and alleles

The overlapped reads were then processed with TSSV 0.2.5, which assigns reads to specific markers and alleles [8]. TSSV recognises reads that match a specific pattern which is provided in a library file. This file contains, for each marker, sequences of the two flanking regions and a regular expression that describes the structure of the STR region in terms of its repeat units and their respective possible numbers of repeats.

TSSV generally works in two steps. First, it scans each read for the flanking sequences of each marker by computing alignments. The flanking sequences in the library used are 18 bases in length while allowing two substitutions per flank in the alignment. The reads are categorised as 'unrecognised' if no flanking sequence is found. Furthermore, both flanking sequences are required to have at least one upper case letter, which ensures that overlapped reads that are potentially truncated are categorised as 'unrecognised' as well.[2] Reads in which at least one flanking sequence is found are linked to the corresponding marker. They are categorised as 'no start' or 'no end' if the left or right flank is not found, respectively. The remaining reads proceed to the second step.

[1] The required modifications have been shared with the creators of FLASH and will likely be included in a future release of the software. Meanwhile, it is available at https://github.com/Jerrythafast/FLASH-lowercase-overhang.

[2] The special handling of overlapped reads was introduced in TSSV 0.2.5 as part of this project and were developed together with the modifications made to FLASH.

In the second step, in reads that contain both flanking sequences of a marker, TSSV attempts to match the sequence between the flanks with the regular expression describing the STR region of that marker. If the expression matches, the read is categorised as 'known allele', otherwise it is categorised as 'new allele'. Both sets of sequences are written to two files each: a FASTA file containing each individual read and a CSV file that contains each unique sequence and the corresponding read counts.

The use of a regular expression to describe possible alleles limits the ability to detect novel alleles that do not match this pattern. Therefore, a specially prepared library is used which contains nonsense regular expressions that are guaranteed not to match. This essentially disables the second step of TSSV and causes all reads that contain both flanks of a marker to be categorised as 'new allele'. This way, any background noise sequences that do not match any currently known alleles are still included in the analysis of systemic noise as well.

The original library with full regular expressions is only used to rewrite the DNA sequences to short allele names. Sequences that do not match the regular expression can then be omitted from the visualisation for brevity, unless they appear in significant amounts, which could mark the presence of a novel allele. If the library is later changed to include a novel allele it is known immediately whether the new allele could also occur as systemic noise from other previously known alleles because it had been included in the analysis. Thus, it is not necessary to re-run the entire analysis pipeline because the regular expression in the library was only used for presentation purposes.

2.4 Estimating allele-specific systemic noise

For each allele of each marker, a profile of recurring systemic noise, including PCR stutter products as well as any other 'side products', was generated. Profiles were computed directly from homozygous samples where possible, but due to the large number of unique alleles homozygotes are rare for most markers. Therefore, these profiles were used merely as a guide for developing a method to estimate the noise profiles using all samples — homozygotes as well as heterozygotes — with high accuracy. To account for alleles that are not present among the training samples but may be encountered in the future, a method was developed to predict the occurrence of stutter in those alleles as well.

In all profiles the amounts of side products are expressed as a percentage of the number of reads of the originating allele. In the context of PCR stutter analysis, this quantity is often referred to as the 'stutter ratio' [10]. Here, the generalised term 'noise ratio' will be used to account for systemic noise other than stutter artefacts as well.

$$\text{Noise ratio} = \frac{\text{noise reads}}{\text{allele reads}} \times 100\%$$

In homozygous samples, 'noise reads' is simply the number of reads of a specific marker that do not match the allele. In heterozygous samples, determining whether non-allelic sequences are noise of one or both alleles and by which proportions is a major part of the problem of computing the noise ratio.

2.4.1 Estimating systemic noise in homozygous samples

Allele-specific noise profiles can be computed from homozygous samples directly by scaling the read counts of the sequences in that sample such that the allele is 100. The profiles thus computed from multiple samples with the same allele are averaged to obtain a single profile for the allele.

Profiles are computed separately for forward and reverse reads, because some strand bias may exist in the sequencing technology used. Sequences with a very low number of reads (e.g., less than 5) or a very low noise ratio (e.g., less than 0.5%) are ignored because such low amounts lead to unstable estimates.

2.4.2 Estimating systemic noise for all alleles from all samples

Because homozygotes are very rare for most alleles (Section 4.2), a different approach was needed to extract allele-specific profiles from heterozygous samples. We will make use of the observation that in heterozygous samples the profiles of the two alleles are simply added up after a scaling correction to account for differences in the amount of each allele present.

One of the challenges that has to be overcome is that the two alleles may have been present in different quantities (due to factors such as e.g., different PCR amplification efficiency) while at the same time the one allele may contribute some additional amount of the other allele (due to systemic noise such as PCR stutter effects) making it difficult to estimate both the original allele quantities as well as the amount of 'cross-contributions' between the two alleles. To solve this problem, a two-step iterative approach was taken, which is outlined in Algorithm 1.

In essence Algorithm 1 seeks a non-negative least squares solution to the equation $\mathbf{AP} = \mathbf{C}$, wherein $\mathbf{C}$ is an $N \times M$ matrix of constants derived from the observed read counts (see Figure 4), $\mathbf{A}$ is an $N \times N$ matrix summarising the estimated allele balance in the samples, and $\mathbf{P}$ is an $N \times M$ matrix containing the estimated profiles of systemic noise. N is the number of unique alleles among the observed samples and thus also the number of profiles produced and M is the number of unique sequences. It is possible to include additional sequences beyond the N alleles of the samples if this is deemed appropriate. For example, one may include all sequences that appear in at least 80% of the samples with a particular allele, since these sequences are probably the result of systemic noise as well. In any case, the first N columns in $\mathbf{P}$ and $\mathbf{C}$ should correspond to the N alleles of the samples and the order of the rows and columns should be the same (i.e., row n and column n in both $\mathbf{P}$ and $\mathbf{C}$ should correspond to the same sequence).

The input of Algorithm 1 consists of a $K \times M$ matrix $\mathbf{S}$ which contains the observed number of reads of each of the M sequences in each of the K samples. The genotype of each sample is provided as a set of indices $g_k, \forall g_k \leq N$.

Any element $\mathbf{P}_{n,m}$ of $\mathbf{P}$ can be interpreted as the amount of sequence m that is observed, on average, for every 100 reads of sequence n. Therefore, the algorithm initialises $\mathbf{P}$ to a diagonal $N \times M$ matrix with the elements on its major diagonal set to 100. The number 100 was chosen for practical reasons since it directly results in noise ratios expressed as percentages of the actual allele.

Similarly, the elements $\mathbf{C}_{n,m}$ of $\mathbf{C}$ can be interpreted as the total amount of sequence m that is observed in all samples with allele n. Line 6 in Algorithm 1 initialises $\mathbf{C}$. For homozygous samples, it scales the read counts of each sample $\mathbf{S}_k$ such that its allele $\mathbf{S}_{k,i}, i \in g_k$ is equal to 100 and then adds the scaled counts to row $\mathbf{C}_i$ of $\mathbf{C}$. Heterozygous samples are treated likewise twice — once for each of their alleles — except that the allele is scaled to 50 instead of 100 to compensate for the fact that the sample is added to two rows in $\mathbf{C}$, as compared to just one for homozygous samples.[3] Each row $\mathbf{C}_i$ of $\mathbf{C}$ thus contains the sum of the read counts of all samples that have allele i, with the read counts of each sample scaled such that allele $\mathbf{S}_{k,i}, i \in g_k$ becomes $100/|g_k|$.[4]

[3] One may also say that the samples are added once for each allele, adding them to the same row twice if their alleles are the same.

[4] Interestingly, because Algorithm 1 scales read counts to 100 divided by the number of alleles in the sample, it handles samples with more than two alleles without problems as well. Since Algorithm 1 makes no assumptions about

Algorithm 1 Systemic noise profile estimation

Require: $K \times M$ matrix $\mathbf{S}$ containing the K samples in the rows
Require: number of alleles $N \leq M$, corresponding to the first N columns of $\mathbf{S}$
Require: list g of sets g_k: the genotypes (indices of alleles $\leq N$) of samples $\mathbf{S}_k$
Return: $N \times M$ matrix $\mathbf{P}$ containing the profiles

function EstimateBackgroundProfiles($\mathbf{S}, g, N$)
  Initialisation:
  $\mathbf{P}_{n,m} \leftarrow 0, \quad 1 \leq n \leq N, \quad 1 \leq m \leq M$
  $\mathbf{P}_{n,n} \leftarrow 100, \quad 1 \leq n \leq N$ ▷ Set actual allele to 100
  $\mathbf{C}_{n,m} \leftarrow 0, \quad 1 \leq n \leq N, \quad 1 \leq m \leq M$
  for $k \leftarrow 1$ **to** K **do**
    $\mathbf{C}_{i,:} \leftarrow \mathbf{C}_{i,:} + \mathbf{S}_{k,:} \times \frac{100}{|g_k|} / \mathbf{S}_{k,i}, \quad \forall i \in g_k$ ▷ Compute separately for each i
  end for
  Optimisation:
  repeat
    Estimate allele balance:
    $\mathbf{A}_{i,j} \leftarrow 0, \quad \forall i, j \in [1 \dots N]$
    for $k \leftarrow 1$ **to** K **do**
      $\mathbf{Q}_{i,j} \leftarrow \mathbf{P}_{g_{k,i}, g_{k,j}}, \quad \forall i, j \in [1 \dots |g_k|]$
      $\mathbf{r}_i \leftarrow \mathbf{S}_{k, g_{k,i}}, \quad \forall i \in [1 \dots |g_k|]$
      $\mathbf{B} \leftarrow \left(\frac{100}{|g_k|} / \mathbf{r}^T \right) \times \textsc{nnls}(\mathbf{Q}^T, \mathbf{r}^T)^T$
      $\mathbf{A}_{g_{k,i}, g_{k,j}} \leftarrow \mathbf{A}_{g_{k,i}, g_{k,j}} + \mathbf{B}_{i,j}, \quad \forall i, j \in [1 \dots |g_k|]$
    end for
    Update profiles:
    $\mathbf{E} \leftarrow \mathbf{A}^T \mathbf{A}$
    $\mathbf{F} \leftarrow \mathbf{A}^T \mathbf{C}$
    repeat
      for $n \leftarrow 1$ **to** N **do**
        $\mathbf{P}_{n,:} \leftarrow (\mathbf{F}_{n,:} - \mathbf{E}_{n,\setminus n} \mathbf{P}_{\setminus n,:}) / \mathbf{E}_{n,n}$
        $\mathbf{P}_{n,:} \leftarrow \textsc{max}(\mathbf{P}_{n,:}, 0)$ ▷ Set negative elements to 0
        $\mathbf{P}_{n,n} \leftarrow 100$ ▷ Keep actual allele at 100
      end for
    until stop condition is met
  until stop condition is met
  return P
end function

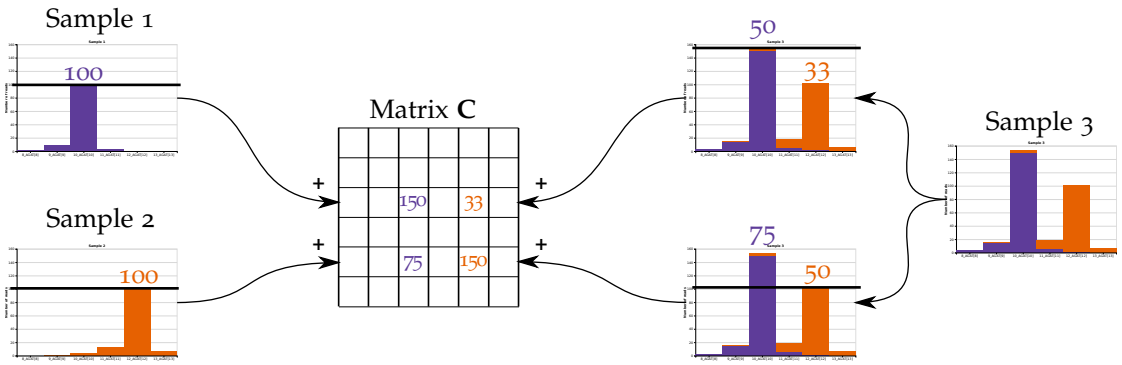

Figure 4. Construction of matrix **C** in Algorithm 1 by summing scaled read counts of the samples that share the same alleles. Left: Samples 1 and 2 are homozygous for the third and fifth allele, respectively. The read counts of both samples are scaled such that their true allele is equal to 100, after which they are added to the third and fifth row of matrix **C**, respectively. Right: Sample 3 is heterozygous, having both the third and fifth allele. Therefore, it is added to both the third and fifth row of matrix **C**, with its read counts scaled such that the third or fifth allele is equal to 50, respectively.

After matrices **P** and **C** are initialised, the algorithm enters its main loop wherein it alternately estimates the allele balance in the samples (matrix **A**) and refines the least squares fit of the profiles **P**. The main loop is exited and the profiles are returned when the sum of the squared errors,

$$\sum_{n=1}^{N}\sum_{m=1}^{M}(\mathbf{D}_{n,m})^2, \quad \mathbf{D}=\mathbf{C}-\mathbf{AP}$$

is reduced by less than 0.01% in one iteration.

Estimation of the allele balance is done for every sample in isolation. At line 11, the elements in **P** that correspond to cross-contributions between the alleles $i, j \in [1 \ldots |g_k|]$ of sample k are extracted. Similarly, the corresponding elements from $\mathbf{S}_k$ are extracted at line 12. For heterozygous samples, this expands to (shortening $g_{k,i}$ to i for brevity):

$$\mathbf{Q} \leftarrow \begin{bmatrix}\mathbf{P}_{i,i} & \mathbf{P}_{i,j}\\ \mathbf{P}_{j,i} & \mathbf{P}_{j,j}\end{bmatrix} = \begin{bmatrix}100 & \mathbf{P}_{i,j}\\ \mathbf{P}_{j,i} & 100\end{bmatrix}$$

$$\mathbf{r} \leftarrow \begin{bmatrix}\mathbf{S}_{k,i} & \mathbf{S}_{k,j}\end{bmatrix}$$

At line 13, a non-negative least squares algorithm is employed to estimate the allele balance within the sample. The NNLS function can be any algorithm that solves $\mathbf{JK}=\mathbf{L}$ for $\mathbf{K}$ subject to $\mathbf{K} \geq 0$ in the least squares sense, e.g., [13]. Line 13 of Algorithm 1 uses this function to solve $\mathbf{bQ}=\mathbf{r}$ for $\mathbf{b}$ (by solving $\mathbf{Q}^T\mathbf{b}^T=\mathbf{r}^T$), which gives an estimation of the proportions in which the alleles are present in the sample. The resulting row vector $\mathbf{b}$ is left-multiplied by a column vector with the same scaling factors as previously calculated at line 6. The result is, for heterozygotes, a 2×2 matrix $\mathbf{B}$.[5] Finally, at line 14, the elements of $\mathbf{B}$ are added to their corresponding elements of $\mathbf{A}$.

With the allele balance estimates all added to **A**, the second step in the main loop of Algorithm 1 is to update the profiles **P** such that they form a non-negative least squares solution to the equation $\mathbf{AP}=\mathbf{C}$ subject to the additional requirement that the elements on the

the number of alleles each sample can have, it is possible to use mixed samples to compute systemic noise profiles as well. This has not been tested, however.

[5]Note that for homozygotes, $\mathbf{B}$ has a single element with a value of 1.

diagonal of **P** must be 100. Lines 16–24 implement NNLS(**A**, **C**) with this additional requirement enforced on line 22. Indeed, with the omission of line 22, lines 16–24 are a general-purpose implementation of the NNLS function. This implementation is based on [13].

Separate profiles for the numbers of forward and reverse strand reads can be constructed by doubling the number of columns in **P** and **C**, where the left half corresponds to the forward stand and the right half to the reverse strand. This ensures that the same allele balance matrix **A** is used for both strands.

2.5 Predicting stutter in novel alleles

Although Algorithm 1 can estimate a profile of systemic noise for each allele in the database, it is not capable of generating profiles for other alleles. Still, if a DNA sample is encountered that contains an allele that was not in the database, it is desirable to be able to filter any systemic noise it may produce. To this end, a method was developed to predict the sequence and corresponding amount of PCR stutter artefacts that would be produced for any allele of a given marker.

Previous studies have shown that the amount of stutter is strongly correlated with the length of the repeated sequence [4, 9, 14] and even more so with the length of uninterrupted repeats [10, 11]. Therefore, it is useful to relate the noise ratio of stutter products with the length of the repeat sequence in bases. This is done in homozygous samples on a per-marker and per-repeat-unit basis.

To this end, each of the alleles is scanned for all positions where a particular repeat unit (e.g., the sequence 'AGAT') is repeated and the length of this repeated sequence — in base pairs, including incomplete repeats — is recorded. For each sample with this allele, the number of reads that lack exactly one repeat is counted. Reads that combine the loss of one repeat with one or more other differences (e.g., substitutions, or stutter in another stretch of repeats in the same allele) are included in this count. These counts are then used to compute the noise ratios of individual stutter sites and a polynomial function is fitted to quantify the relationship between the length of the repeat and the stutter ratio.

This analysis is repeated for each unique repeat unit of between one and six bases (inclusive), treating cyclically equivalent units (e.g., 'ATAG' for 'AGAT') and their respective reverse complements (e.g., 'CTAT' or 'ATCT') synonymously. The amount of +1 stutter and −2 stutter is analysed in the same way.

Because some differences in stutter rate exist between the different markers, a separate function is fitted for each marker. When there is insufficient data per marker, it is also possible to fit a polynomial function to all data at once. When doing so, the least significant coefficient (the intercept) can still optionally be estimated separately for each marker, which yields a set of polynomial functions with the same shape shifted vertically by a marker-specific amount.

The quality of fit is assessed by computing the coefficient of determination,

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \overline{y})^2}$$

where

$$\hat{y}_i = \begin{cases} f_i & f_i \geq 0 \\ 0 & f_i < 0 \end{cases}$$

with y_i the noise ratios of the reference samples, $\overline{y}$ the mean, and f_i the polynomial function's estimate of the noise ratio of sample i. The R^2 will be close to unity when the function is a good fit and lower otherwise.

A multi-step process has been developed to determine the optimal polynomial fit for each repeat unit and marker. Previous studies suggested this correlation is linear [9–11], but quadratic polynomials are considered here as well if a linear relationship seems inappropriate. The optimal functions are determined by repeating the following process for each unique repeat unit and each type of stutter (i.e., –2, –1, and +1 stutter).

In each step, the (least squares) fits with R^2 values below 0.7 are rejected. In some steps, the negative samples (i.e., those with zero stutter) are excluded when estimating the parameters of the fit. R^2 values are always computed including the negative samples, even if those were excluded when estimating the parameters. There are often plenty of negative samples because even repeats of less than two full units, which generally do not produce any detectable amount of stutter, are included (e.g., 'AGATAG' is considered a 6bp repeat of 'AGAT'). By using them in the computation of the R^2 score only, these negatives guard against 'over-fitting' on the non-zero samples, since they will greatly reduce the R^2 score if the resulting $\hat{y}_i$ values for these data points are not zero.

1. Fit one line for each marker, all having the same slope. Negative samples are ignored. This helps to get a good estimate of the slope for markers with little non-zero data.

2. Fit one line for each marker, ignoring negative samples. Markers with sufficient non-zero data will now obtain a better fit by allowing the slope to vary per marker.

3. Fit a quadratic polynomial for each of the markers for which no linear fit was obtained, all having the same shape and again ignoring the negative samples.

4. Fit another quadratic polynomial for those markers, ignoring the negative samples but allowing different shapes for each marker. Replace the previous quadratic fit for those markers for which an improved R^2 score is achieved.

5. Finally, fit one last quadratic polynomial for the markers for which no linear fit was obtained, this time including the negative samples. Replace the previous quadratic fit for those markers for which an improved R^2 score is achieved.

This way, a linear fit is obtained if possible, falling back to a quadratic fit as needed. Higher-order polynomials are not used because the increased flexibility comes with an increased risk of over-fitting.

The polynomial functions thus produced can be used to compute a predicted amount of stutter for a repetitive sequence of a given length. By combining the output of the functions for all possible sites of stutter in an allele, a complete stutter profile can be predicted for that allele.

2.6 Filtering systemic noise in samples of unknown genotype

To be able to filter systemic noise in samples of unknown genotype, one first needs to determine which alleles are likely present in the sample. To this end, the method of Algorithm 1 is essentially reversed, i.e., we will now solve for $\mathbf{a}$ in $\mathbf{aP} = \mathbf{c}$, where $\mathbf{c}$ is a row vector with the sample's read counts for the M sequences in the profiles and $\mathbf{a}$ is a row vector with the estimated amount of each of the N profiles in the sample. $\mathbf{P}$ is the $N \times M$ matrix of profiles produced by Algorithm 1. Solving for $\mathbf{a}$ is done in the non-negative least squares sense as before. Again, the profiles and read counts of both strands can be included in this calculation at the same time by doubling the number of columns in $\mathbf{P}$ and $\mathbf{c}$. The resulting vector $\mathbf{a}$ will then give estimated allele contributions that best fit both the forward and reverse strand reads in the sample at the same time.

Background-corrected read counts can then be computed by first subtracting the scaled profiles from the sample's read counts

$$\mathbf{d} \leftarrow \mathbf{c} - \mathbf{a}\mathbf{P}$$

and then adding the total size of each profile to the corresponding allele, i.e.,

$$\mathbf{d}_n \leftarrow \mathbf{d}_n + \mathbf{a}_n \sum_{m=1}^{M} \mathbf{P}_{n,m}, \quad \forall n \in [1 \ldots N]$$

Note that $\mathbf{d}$ may have negative elements if the sample contains a lower amount of a certain sequence than was predicted by the profiles of its dominant alleles.

2.7 Data visualisation

Visualisation of data is done using Vega [15]. With Vega, one specifies graphing rules in a JavaScript Object Notation (JSON) formatted file. The Vega runtime, a JavaScript library, uses these rules to produce a graph that visualises a data set that can be supplied separately in various file formats. Vega can run embedded on a web page where it offers interactivity by exposing a JavaScript programming interface, or it can run on Node.js which makes it possible to include Vega in automated analysis pipelines. Graphs can be rendered in Scalable Vector Graphics (SVG) or rasterised Portable Network Graphics (PNG) image formats.

All graphs in this work are created with Vega and use the 'four-class PuOr' (purple–orange) colour scheme provided by ColorBrewer [16]. ColorBrewer is an online tool originally designed for cartographers that guides its users to selecting a colour scheme that exhibits desirable properties such as being interpretable to people with red–green colour blindness and remaining interpretable after black-and-white photocopying or printing. The 'four-class PuOr' colour scheme was selected because it exhibits all these properties. Using photocopy-safe colours is especially important in forensics, where documents are often printed or copied to be kept in paper files.

3 Results

3.1 Systemic noise in homozygous samples

Although the number of homozygous samples in the training data set is limited, the use of homozygous samples has one clear advantage over estimating the noise profiles from heterozygotes: there can be no ambiguity about the originating allele of any observed sequence. Therefore, a clear overview of what kind of systemic noise is to be expected and in what amounts can be obtained with little effort.

Figure 5 gives an overview of the systemic noise in eight homozygous samples having a D7S820 allele with 11 'TCTA' repeats and a stretch of nine 'A' bases, separated by a non-repetitive section of 11 bases. The largest amount of noise in these samples is due to stutter in the 'TCTA' repeat, with the −1 stutter product producing a noise ratio of about 6.2% and the +1 stutter product about 0.7%. The stretch of 'A' bases exhibits stutter as well, on average about 4.8% for the −1 stutter product and 1.9% for the +1 stutter product. This is an interesting result: the 'A' 9-repeat exhibits much higher +1 stutter than the 'TCTA' 11-repeat, whereas the −1 stutter of the 'A' repeat is lower. The former is in agreement with previous reports that shorter repeat units generally show higher stutter rates [3, 9], whereas the latter can be

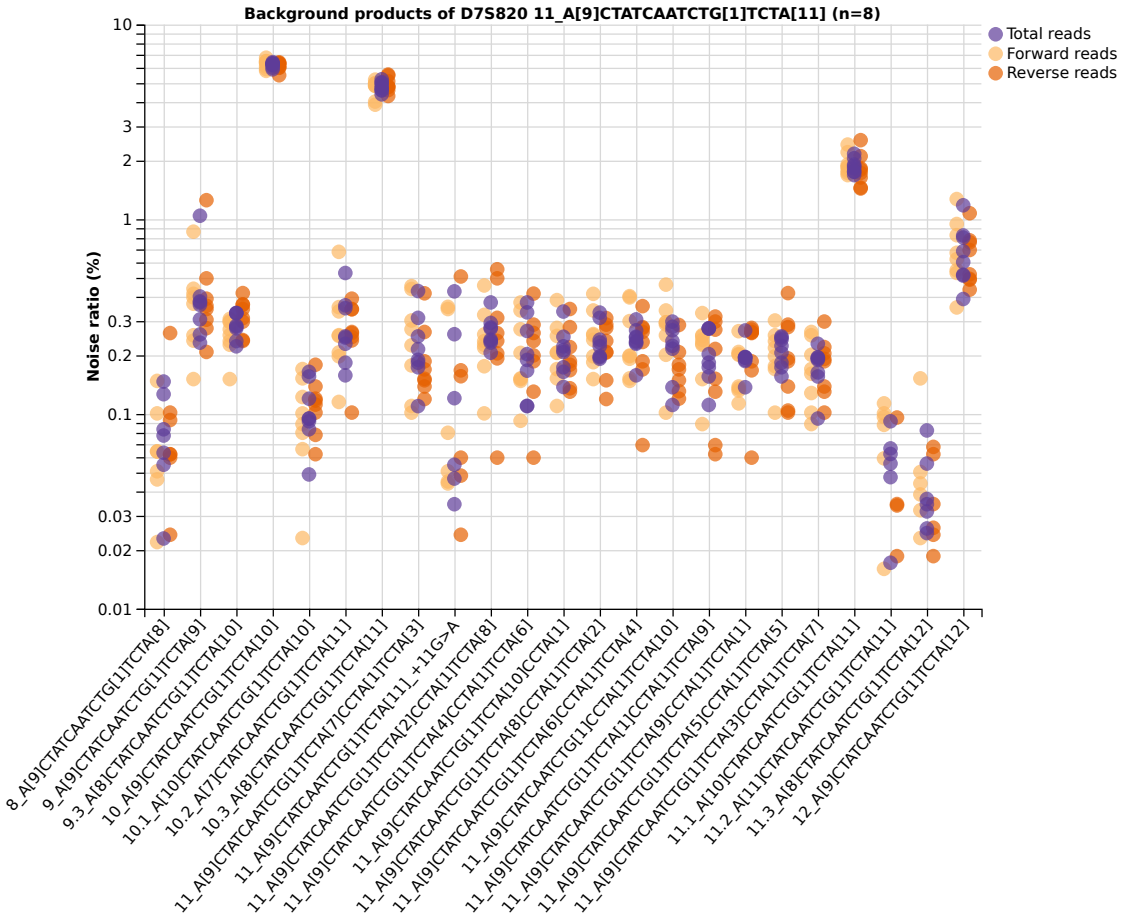

Figure 5. Example noise profile of a D7S820 allele with 11 'TCTA' repeats and a stretch of nine 'A' bases, generated from eight homozygous samples. Three data points are plotted for each observed sequence per sample, corresponding to the noise ratio of forward strand reads, reverse strand reads, and total reads respectively. Both repetitive regions in the allele exhibit stutter and various substitutions appear to occur regularly as well, albeit in very small quantities.

explained by the fact that a higher number of repeats results in a higher stutter rate as well [4, 9, 10].

Furthermore, it is interesting to note that the noise ratio of the sequence that exhibits a –1 stutter in both repeats is about 0.3%: this is equal to the 6.2% for the 'TCTA' repeat multiplied by the 4.8% for the 'A' repeat (Figure 5, third column).

Besides stutter products, various other sequences with single base substitutions are observed. While they generally have a noise ratio below 0.5% for the allele in Figure 5, there are other cases where specific substitutions show up with a much higher noise ratio. One such example are the alleles of the marker D19S433, all of which start with a 'TCCT' repeat followed by 'ACCT', i.e., 'TCCT[n]ACCT[1]'. Figure 6 shows data from 41 samples that are homozygous for D19S433 and exhibit the sequence 'TCCT[n–1]ACCT[2]', i.e., a T-to-A substitution in the last repeat, with a noise ratio of about 0.7–1.3%. This substitution, among other similar examples in other markers, appears in such high amounts that it is desirable to be able to filter it as well; stutter products are thus no longer the only type of systemic noise in forensic STR analysis.

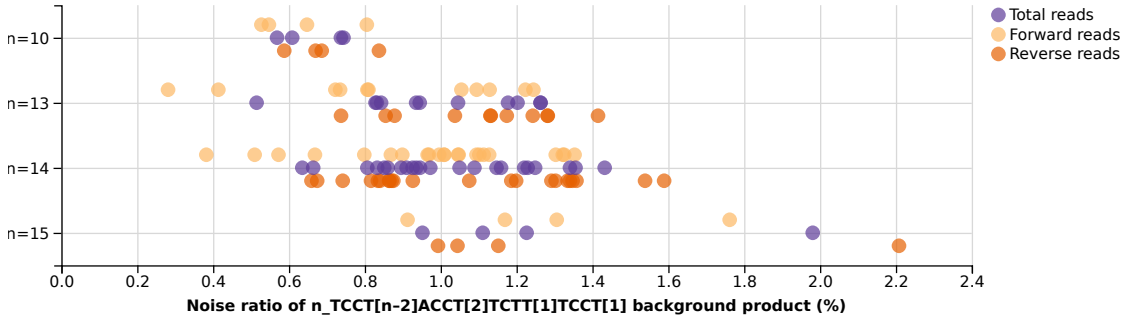

Figure 6. The noise ratio of T-to-A substitutions in the last repeat of D19S433 alleles in 41 homozygous samples. The actual alleles of these samples are summarised as 'n_TCCT[n–1]ACCT[1]TCTT[1]TCCT[1]'.

3.2 Systemic noise from all samples

To overcome the limitations of using only homozygous samples, Algorithm 1 was introduced. Using this algorithm it is possible to estimate average noise ratios for each observed sequence with each allele in the training data set. The strength of the algorithm lies in the fact that it continuously re-estimates the allele balance in the heterozygous samples and then assigns the observed noise sequences in a least squares optimal way to the alleles of the samples.

For n unique alleles, m unique sequences, and g alleles in the genotypes of the samples (i.e., one for every homozygous sample plus two for every heterozygous sample), Algorithm 1 has $\mathcal{O}(nm)$ space complexity. The initialisation phase has $\mathcal{O}(g)$ time complexity for the construction of $\mathbf{C}$. In the iterative part of the algorithm, the estimation of allele balance has $\mathcal{O}(g)$ time complexity as well. The profile update step has $\mathcal{O}(nm)$ time complexity. It must be noted however, that the number of iterations does not depend on any of these figures. Depending on the exact input, in exceptional cases the algorithm may converge slowly and thus require more iterations. With our set of 271 reference samples, the algorithm typically completed in about 6 seconds per marker on an Intel Xeon X5670 CPU running at 2.93 GHz. The marker for which the algorithm runs the longest is Amel, which is not an STR marker but instead a sex test which has only $n = 3$ unique alleles ('X', 'Y', and 'X_+41C>T') and $m = 12$ unique sequences. The algorithm converges slowly for this marker, requiring over 70 iterations compared to about 20 for the others. Overall, the algorithm takes an average 326 milliseconds per iteration on this machine.

The profiles produced by Algorithm 1 can be interpreted as the mean noise ratio among the samples with a given allele. Because our data are ratios, the data likely follows a log-normal distribution. Figure 5 also seems to suggest this; while the data is plotted on a log scale, it does not show signs of bias. If the same data were plotted on a linear scale, a bias towards zero would be evident (not shown).

In Figure 7 the noise ratio of –1 stutter and +1 stutter products of two alleles in homozygous samples are compared against a standard normal distribution and the log of the same data points is compared against the same distribution. When log-normally distributed data is plotted against a normal distribution, a 'banana-shaped' curve is obtained, the slope of which increases as one goes from left to right on the x-axis. Although especially the graphs of CSF1Po suffer from the low number of data points, for +1 stutter this banana shape is apparent and is largely eliminated by taking the log of the data. For the higher noise ratios of the –1 stutter products (means of 7.5 for CSF1Po and 2.2 for TPOX versus 0.9 and 0.3 respectively for +1 stutter), the data was already closer to the normal distribution.

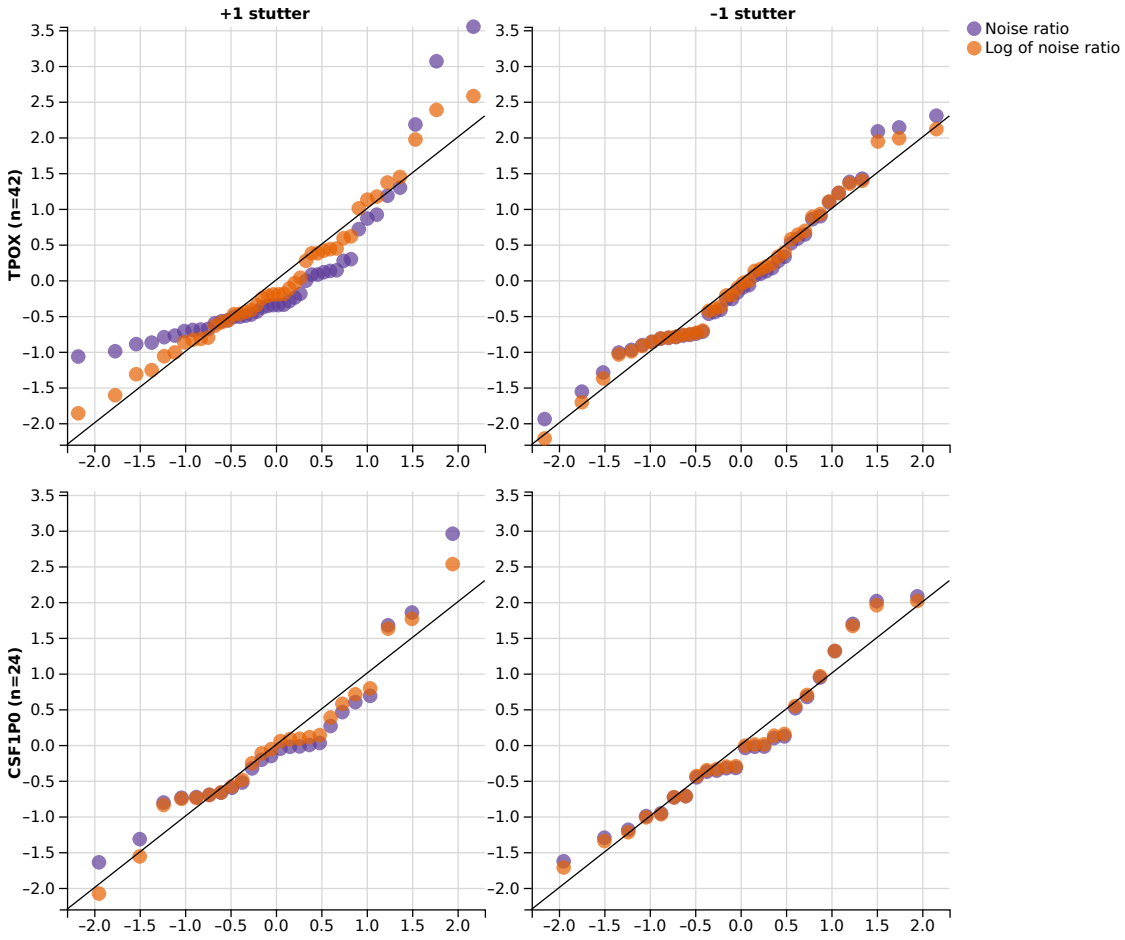

Figure 7. A quantile–quantile (Q–Q) plot of the noise ratio of the +1 stutter (left) and −1 stutter (right) in 42 TPOX 8_TGAA[8] (top) and 24 CSF1Po 12_CTAT[12] (bottom) homozygotes. Quantiles of the noise ratio and those of the natural logarithm of the noise ratio are both compared against the standard normal distribution.

If the data are indeed log-normally distributed, i.e.,

$$\log(X) \sim \mathcal{N}(\mu, \sigma^2), \quad \sigma > 0$$

then the mode of the data is always lower than the mean:

$$\begin{aligned} \operatorname{mode}[X] &< \operatorname{E}[X] \\ \exp(\mu - \sigma^2) &< \exp(\mu + \sigma^2/2) \end{aligned}$$

Thus, it is to be expected that the majority of the samples will have a lower noise ratio than the mean that Algorithm 1 produces. The difference between the mode and the mean grows larger as the shape parameter σ^2 of the log-normal distribution increases. This value can be computed from homozygous samples by

$$\sigma^2 = \ln\left(1 + \frac{\operatorname{Var}[X]}{(\operatorname{E}[X])^2}\right)$$

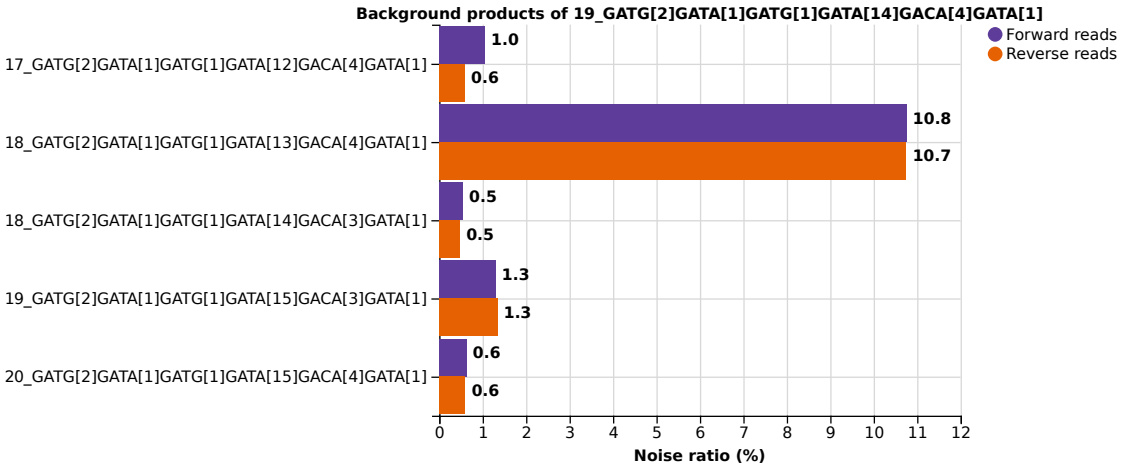

Figure 8. Systemic noise profile of vWA allele 19_GATG[2]GATA[1]GATG[1]GATA[14]GACA[4]GATA[1], estimated from three homozygotes and 36 heterozygotes using Algorithm 1. Only background products with an average noise ratio above 0.50% are shown. Various stutter products are visible, as well as one common substitution.

It is not trivial to compute the variance per allele from heterozygous samples, however, making this computation problematic when heterozygous samples are included. Therefore, the profiles produced by Algorithm 1 currently only contain mean noise ratios.

An example noise profile is visualised in Figure 8. Being estimated almost exclusively from heterozygous samples, this example clearly demonstrates the accuracy of the method in determining which background products have likely originated from this allele. This profile contains 21 more background products with noise ratios between 0.2% and 0.5%, all of which differ from the allele by a single nucleotide substitution. Some background products below 0.2% are more difficult to interpret or are not specific to this allele (not shown). A similar profile was constructed for all alleles (of all markers) for which two or more samples were available.

3.3 Predicted stutter in novel alleles

In order to predict systemic noise profiles for alleles that are not present in the set of reference samples, polynomial functions were fitted that relate the length of a repeat to the noise ratio of the occurrence of stutter in this repeat. As outlined in Section 2.5, a separate polynomial is fitted for each unique repeat unit and each marker.

Table 1 summarises all results obtained for repeats of 'A' and 'AGAT' and for some markers the data is visualised in Figure 9 as well. Data in the five columns labelled 'Line 1'–'Quad 3' of Table 1 correspond to the R^2 scores of the fits computed in each of the five steps outlined on page 11, respectively.

3.4 Filtering systemic noise

With the profiles of stutter products and other systemic noise created with the methods outlined above, the true allelic components of samples with unknown genotypes can be estimated by computing a best fit of all profiles to the sample simultaneously. Various approaches can then be taken to call alleles in the sample and to quantify their proportions.

The simplest approach is to interpret the coefficients of the fit of the profiles as the amounts of the alleles those profiles represent, e.g., interpret 'x times profile A and y times profile B' as

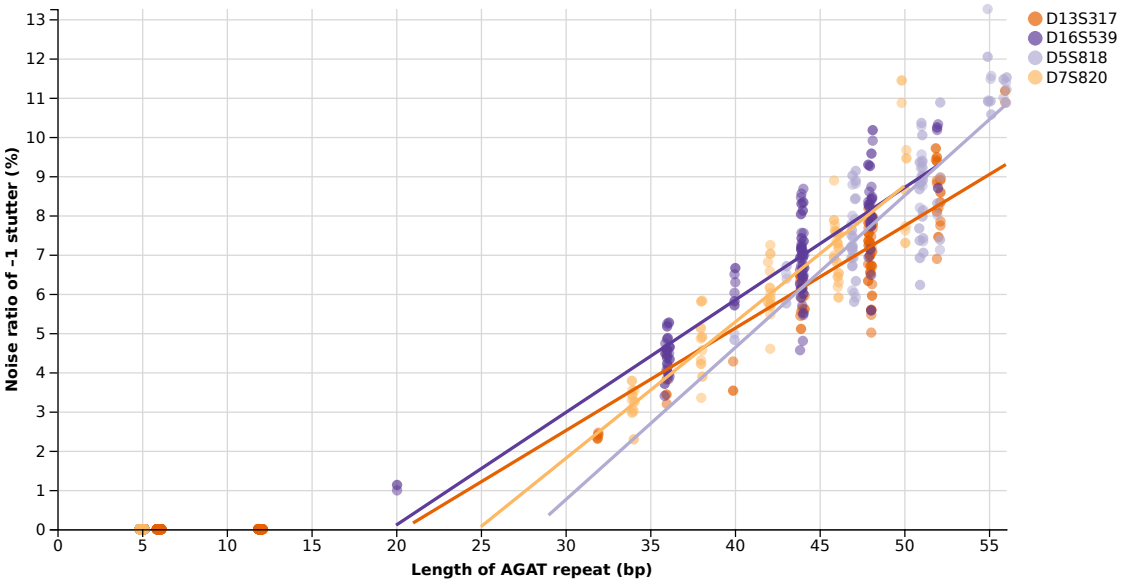

Figure 9. Comparison of the noise ratio of –1 stutter with the repeat length in the 'AGAT' repeat sequence of four markers. Data points of all homozygous samples are plotted, as well as the corresponding fitted lines listed in Table 1.

Table 1. R^2 scores for various polynomials fitted to data from homozygous samples, relating repeat length in bases with the stutter ratio of a repeat of that length. The R^2 score of the selected function is printed in bold face. See the steps at page 11 for an overview of how the different lines and quadratic functions are fitted.

St.	Marker	Line 1	Line 2	Quad 1	Quad 2	Quad 3	Function
Repeat unit 'A'							
–2	D7S820	—	**0.7435**	0.7071	—	0.7848	$0.0004192x - 0.001164$
–1	D7S820	—	**0.9589**	0.9367	0.9892	0.9886	$0.008726x - 0.02512$
–1	PentaD	—	—	0.7840	**0.8431**	0.8075	$0.001330x^2 - 0.01051x + 0.02032$
+1	D7S820	—	**0.9736**	0.9824	0.9841	0.9830	$0.003994x - 0.01373$
+1	PentaD	—	—	0.7047	**0.7687**	—	$0.0002721x^2 - 0.002584x + 0.00578$
Repeat unit 'AGAT'							
–2	D3S1358	0.8378	**0.8392**	0.8271	0.8489	0.8368	$0.0003436x - 0.01096$
–2	D8S1179	0.8381	**0.8546**	0.8569	0.8584	0.8544	$0.0004279x - 0.01638$
–2	D13S317	0.7555	**0.7732**	0.7854	0.7965	0.7903	$0.0003714x - 0.01359$
–2	D21S11	**0.8355**	0.8235	0.8603	0.8636	0.8505	$0.0002909x - 0.009071$
–1	CSF1Po	**0.9744**	0.9740	0.9776	0.9780	0.9776	$0.002363x - 0.03370$
–1	D3S1358	0.9582	**0.9669**	0.9604	0.9723	0.9723	$0.002997x - 0.04721$
–1	D7S820	0.9564	**0.9712**	0.9710	0.9716	0.9715	$0.003475x - 0.08626$
–1	D5S818	0.9717	**0.9777**	0.9783	0.9789	0.9784	$0.003874x - 0.1088$
–1	D8S1179	0.9734	**0.9748**	0.9731	0.9765	0.9758	$0.002150x - 0.02601$
–1	D13S317	0.9794	**0.9824**	0.9874	0.9875	0.9876	$0.002611x - 0.05324$
–1	D16S539	0.9762	**0.9786**	0.9791	0.9796	0.9795	$0.002866x - 0.05630$
–1	D21S11	**0.9513**	0.9509	0.9707	0.9743	0.9739	$0.002363x - 0.03373$
–1	vWA	0.9746	**0.9752**	0.9757	0.9800	0.9801	$0.002393x - 0.01747$
+1	D3S1358	0.7933	**0.7939**	0.7946	—	0.7953	$0.0002483x - 0.006287$
+1	D7S820	0.7838	**0.7883**	0.7846	0.8025	0.7842	$0.0002719x - 0.005846$
+1	D13S317	0.8202	**0.8253**	0.8225	0.8277	0.8222	$0.0002936x - 0.007356$
+1	D16S539	**0.7527**	0.7350	0.7499	—	0.7526	$0.0002095x - 0.002698$
+1	D21S11	0.7223	**0.8154**	—	0.8277	0.8227	$0.0003003x - 0.004534$

'x times allele A and y times allele B'. However, with this approach it is not possible to detect novel alleles or alleles for which the estimated amounts of noise differ too much from what is found in this particular sample.

Alternatively, the fitted profiles can be used to reduce the read counts of background noise products. This correction step helps to find the true alleles in the sample, whether those be novel alleles or not, by removing the systemic noise produced by known alleles. As an optional extra step, the filtered noise may be added to the originating allele, which may greatly help the actual alleles to stand out even more.

As an example, Figure 10 shows the effects of filtering background noise in two two-person mixtures. The same two persons contributed to both samples, but their DNA was mixed in two different ratios; 1:9 and 1:99. The major contributor is homozygous for D5S818 allele 12_CTCT[1]-ATCT[12]_+13A>G and heterozygous for D8S1179 with alleles 10_TCTA[10] and 13_TCTA[13]. The minor contributor is heterozygous for D5S818 with alleles 9_ATCT[10]_+13A>G and 13_CTCT[1]ATCT[13] and heterozygous for D8S1179 with alleles 14_TCTA[1]TCTG[1]TCTA[12] and 15_TCTA[1]TCTG[1]TCTA[13]. The alleles of the major contributor are clearly visible. Before filtering, the alleles of the minor contributor appear in roughly the same quantities as the –1 stutter products of the major contributor in the 1:9 ratio sample and are indistinguishable in the more extremely skewed 1:99 ratio sample. After filtering the minor contribution becomes clearly visible in the 1:9 ratio scenario, regardless of whether the filtered noise is added to the originating alleles or not. The 1:99 ratio scenario remains challenging, with the –1 stutter products of the major contribution still higher than the true alleles of the minor contributor.

3.5 Forensic DNA Sequencing Tools

The tools that were developed during this project are collected in a Python package called 'FDSTools' (Forensic DNA Sequencing Tools) which has been submitted to the Python Package Index (PyPI).[6] This way, the tools can easily be installed on any computer with an internet connection using the Python package installer 'pip'.

3.5.1 Main characterisation and filtering tools

The two most important tools in the toolbox are BGEstimate, which is used to compute background noise profiles from a set of reference samples and BGCorrect, which uses the profiles created with BGEstimate to filter systemic noise in case samples.

BGEstimate This tool is to be used on a set of reference samples to compute profiles of systemic noise for each allele found in the samples, which is done using Algorithm 1. It is therefore necessary that the genotypes of the samples are provided and that the samples contain no contaminations.

It may be beneficial to use a TSSV library with nonsense regular expressions to include all observed sequences in the analysis, as described in Section 2.3. In this setting, BGEstimate operates on the 'newalleles.csv' output files of TSSV. It is also possible to use a normal TSSV library, in which case the 'knownalleles.csv' files should be used instead.

BGEstimate analyses only those noise sequences that appear in at least 80% of the samples with a particular allele by default, but this percentage can be overridden by the user. Furthermore, a minimum number of reads, minimum number of samples per allele, and minimum noise ratio to report can be specified to filter the input and the results.

The data in Figure 8 were generated using BGEstimate.

[6] Available at https://pypi.python.org/pypi/fdstools/.

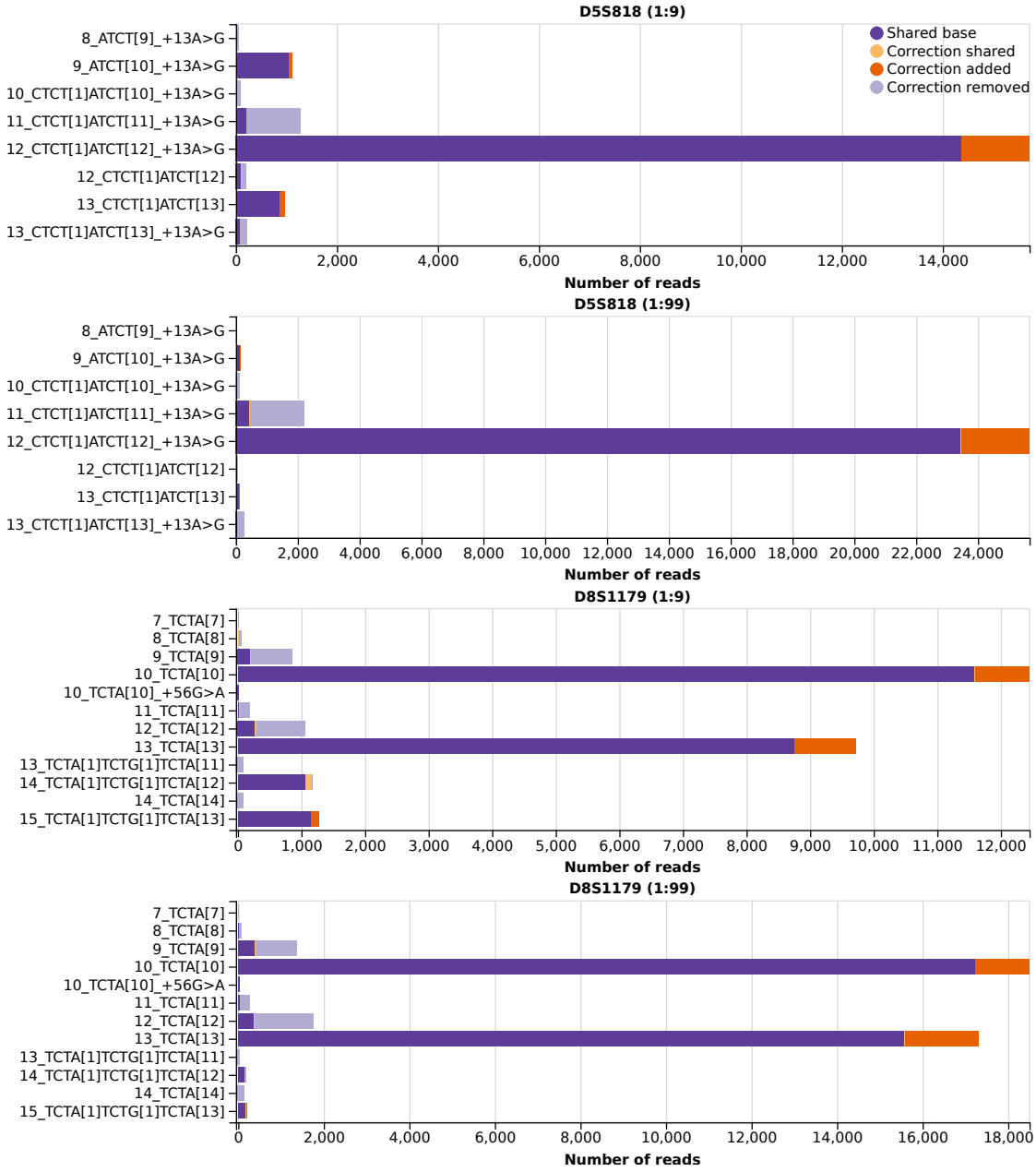

Figure 10. Examples of two two-person DNA samples with mixture ratios of 1:99 and 1:9 respectively. Detected sequences of D5S818 and D8S1179 above about 0.1% of the highest allele are shown (lower amounts would be invisible). Read counts before filtering correspond to 'shared base' plus 'correction shared' plus 'correction removed'. After filtering background noise, reads counts correspond to 'shared base'. When background noise is added to the originating alleles, read counts correspond to 'shared base' plus 'correction shared' plus 'correction added'.

BGCorrect BGCorrect uses the profiles computed by BGEstimate to annotate DNA samples, operating on the 'knownalleles.csv' or 'newalleles.csv' output files of TSSV. The sample is analysed as outlined in Section 2.6. A number of columns are added to the file that give the number of reads that noise filtering would remove (i.e., the expected number of reads that were due to systemic noise) and the number of reads that noise correction would add (i.e., the expected number of 'missed' reads of a true allele). These figures are given separately for each strand.

After a database of systemic noise profiles is obtained using BGEstimate, a forensic laboratory may use BGCorrect to filter systemic noise in case samples.

The data in Figure 10 were generated using BGCorrect.

3.5.2 Other analysis tools

Besides the two main tools outlined above, a number of additional analysis tools have been developed. These are included in FDSTools as well.

BGPredict BGPredict can be used to compute stutter profiles for novel alleles by relating the length of the repeat sequence with the noise ratio of stutter products as visualised in Figure 9. BGPredict returns the coefficients of the fitted functions along with their R^2 scores and the domain of the supporting data. Like BGEstimate, it may operate on the 'newalleles.csv' or 'knownalleles.csv' output files of TSSV depending on the library used.

Users may select the degree of the fitted polynomials, the maximum repeat unit length, the type of stutter to analyse, and whether the polynomials of all markers should have the same shape. Furthermore, various filters can be applied, such as ignoring samples with less than a given amount of stutter or requiring a minimum number of reads per sample.

In essence, BGPredict is the foundation for the procedure outlined in Section 2.5. Currently, there is no tool that automatically executes the steps at page 11. Instead, users are required to call BGPredict with various different arguments manually. Plans exist to extend BGPredict with the ability to automatically determine the most suitable fitting method for each repeat unit and marker. Also, the output of BGPredict is currently not compatible with the output of BGEstimate and thus can not be used as the input for BGCorrect directly.

Stuttermark Stuttermark operates on the 'knownalleles.csv' output file of TSSV. Stuttermark allows users to specify the maximum expected noise ratio of stutter products (defaulting to 15% [17, 18] for −1 stutter and 4% for +1 stutter if not specified). All sequences are compared in pairs and whenever two sequences differ by the omission or addition of exactly one repeat unit, the provided noise ratios are used to compute the maximum possible number of reads of the one sequence that could be attributed to stutter originating from the other sequence. By default, Stuttermark assumes no stutter occurs when a unit is repeated less than three times.

A total maximum possible amount of stutter is obtained for each sequence by adding up the maximum amounts of stutter for each possible originating allele. Any sequence that was observed in no more than the thus computed number of reads is marked as a stutter product.

Stuttermark relies entirely on the maximum expected stutter ratios provided by the user to remove the burden of interpreting the stutter products in forensic DNA samples. It has already proven its use in the absence of a database of systemic noise profiles, while the methods for estimating those were still being developed [19].

CommonBackground This tool computes noise ratios of all systemic noise found in homozygous samples, using either the 'knownalleles.csv' or 'newalleles.csv' output files of TSSV. When visualised, this data looks like Figure 5. It has the same filtering options as BGEstimate. Being a simpler, homozygotes-only predecessor of BGEstimate, its primary use is to check the accuracy of BGEstimate and to allow for the calculation of the variance of the computed noise ratios.

Blame Blame can be used to find particularly 'dirty' (i.e., contaminated) samples among the reference samples. To do this, it requires background noise profiles estimated by BGEstimate and operates on either the 'knownalleles.csv' or 'newalleles.csv' output files of TSSV, depending on which files were used as the input for BGEstimate.

Blame has three operating modes; 'common' (the default) prints the overall most occurring contaminant alleles per marker,[7] 'highest' prints the samples with the highest single contaminant per marker, and 'dirtiest' prints the samples with the highest total amount of contaminants per marker. Running blame in the 'highest' or 'dirtiest' mode thus helps the user identify reference samples that are not suitable for generating background noise profiles using BGEstimate, whereas running in 'common' mode helps identifying a common background signal that may be present among all samples.

3.5.3 Visualisation tools

Vega graph specifications and accompanying web pages have been created to visualise the output of each of these tools. When viewed in a web browser, the web page provides additional controls that allow the user to filter the data, switch between linear and logarithmic scales, or select different subsets of the data (e.g., when visualising background noise profiles, the user may select an allele to view the corresponding profile). The web pages also offer the option to save the displayed image as a Scalable Vector Graphics (SVG) or rasterised Portable Network Graphics (PNG) file, so that they can be imported into documents.

Vega graph specifications allow various data transformations to be applied and data can be loaded dynamically from various types of files. This allows output data from the tools to be visualised with Vega immediately. These capabilities are nicely illustrated by 'Samplevis' (short for 'sample visualisation'). Like all other visualisations, Samplevis consists of a web page and a Vega graph specification. The web page contains a file selection input element such as those used by websites to implement file upload functionality. When the user selects a sample data file (such as an output file of BGCorrect, or the 'knownalleles.csv' output file of TSSV) in Samplevis, this file is automatically parsed by Vega and a graph similar to Figure 10 is produced.

The web pages provide a graphical user interface to interact with the visualisations. It is not necessary to use a web server, since Vega runs entirely in the web browser and files can be loaded from local storage. Furthermore, the web pages can be bypassed by running Vega in 'headless mode' using Node.js, which also allows for generating various graphs as part of an automated analysis pipeline.

[7] The tool is called 'Blame' because the default mode might produce a DNA profile identifying the person who handled the samples.

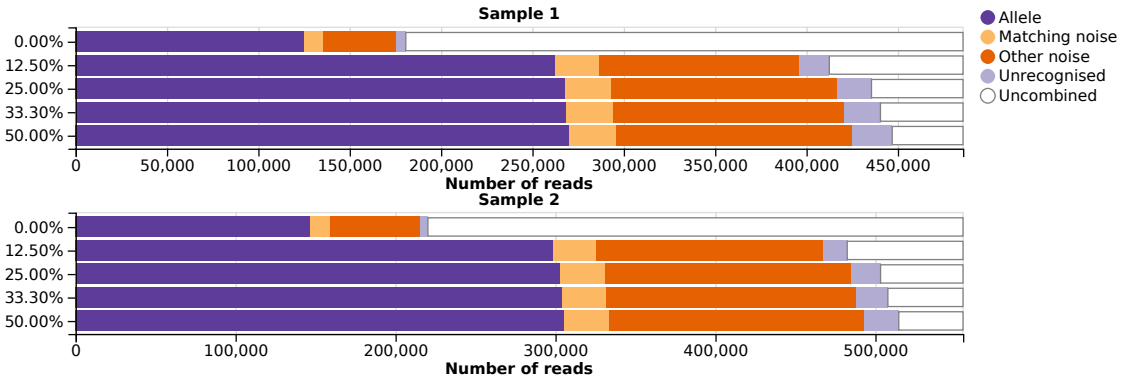

Figure 11. Analysis of overlapped reads while allowing various different amounts of mismatches in the overlapped region in two samples. Sample 1 has a long FGA allele with allele number 45.2. Increasing the maximum number of allowed mismatches yields more reads with the sample's true allelic sequence, suggesting that in those cases the higher quality base at mismatch positions is indeed the correct base. Almost all mismatches that, when resolved by FLASH, did not result in a true allelic sequence, resulted instead in sequences that do not match the regular expression in the TSSV library. As a result, such mismatches will not affect sample analysis.
'Allele': The number of reads with the sample's true allelic sequences. 'Matching noise': The number of reads with other sequences that match any marker's regular expression (mainly stutter products). 'Other noise': The number of reads with sequences that do not match the regular expressions, corresponding to 'new alleles' in TSSV output. 'Unrecognised': The number of reads that did not contain both flanking sequences of any marker. 'Uncombined': The number of read pairs that were not combined.

4 Discussion

4.1 Paired-end read merging

We have tested overlapping the reads while allowing one mismatch in every eight, four, three, and two nucleotides in the overlapped region and investigated the effects on data corresponding to samples with and without the long FGA allele 45.2 (Figure 11). With FLASH, the output base is the one with the higher quality score, defaulting to the 'Read 1' base if the scores are identical. Interestingly, increasing the maximum number of allowed mismatches resulted in a significant increase in the number of FGA allele 45.2 reads, a mild increase the number of reads of various other alleles, and no apparent increase in the number of non-allelic sequences (Figure 12). This means that it is beneficial to use paired-end read merging even for alleles that fit within the read length.

One possible explanation is that the mismatches occur because of erroneous base calls near the end of each read, where the quality scores are generally lower. When overlapping the two reads, the higher-quality beginning of the complementary read allows the errors to be recognised and corrected. Furthermore, merging the reads might 'repair' reads where the trimming of adapter sequences has cut into the flanking sequences which are used to link the reads to specific markers.

4.1.1 Read direction imbalance

Before the paired-end read overlapping step was introduced, only the data from the 'Read 1' file was used and the 'Read 2' file remained unused. The markers PentaD, FGA, and to a lesser extent D18S51 and PentaE exhibited strong strand bias at that time, where the recognised allelic reads of these markers were almost exclusively in the same orientation. The same strand dominated in 'Read 1' and 'Read 2', even across different sequencer runs. After overlapping

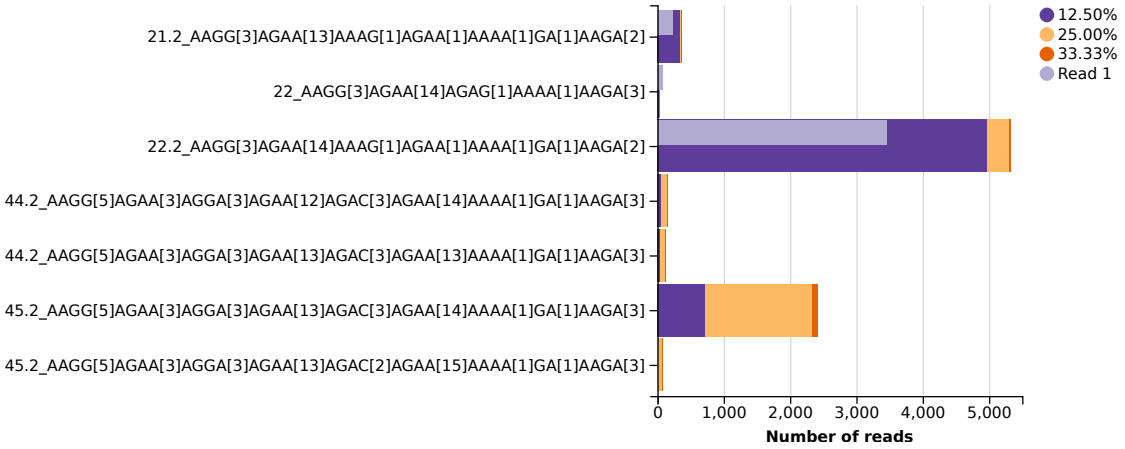

Figure 12. Recovery of a long FGA allele by paired-end read merging. As the number of allowed mismatches in the overlapped region of the reads is increased from 12.5% to 25%, many more reads of allele 45.2 are recovered, while the number of reads corresponding to stutter products of this allele does not increase as much. The number of recognised reads in the 'Read 1' file (i.e., without paired-end read merging) is displayed as well. Clearly, the sorter allele 22.2 benefits from read pair overlapping as well.

the complementary reads the number of reads in the dominating orientation increased slightly, but the number of reads in the other orientation increased by such an extent that no apparent strand bias remained.

Closer inspection of the sequences of these markers revealed that the dominating strand for all of these markers had a core repeat unit of 'AAAG' or 'AAAAG'. The recovered reads thus consisted of repeats of 'CTTT' and 'CTTTT'. The recovered PentaD reads of one sample were analysed to see why they were not recognised prior to overlapping with the complementary read. Of the 7164 recovered reads, two reads were previously unrecognised because neither flanking sequence was recognised by TSSV, three reads because only the flanking sequence before the STR was unrecognised, 1451 reads because only the flanking sequence after the STR was unrecognised, and 5708 reads had both flanking sequences recognised but the STR sequence did not match that of the expected allele. This data suggests the Illumina MiSeq introduces base call errors specifically in sequences that are rich in 'T' and devoid of 'A' and 'G' bases. While it is usually able to call the bases after this region correctly, in about 20% of the reads this did not happen and the sequencer continued to make excessive incorrect base calls. This effect is also evident in Figure 12, where the vast majority of recovered allele 22.2 reads are on the reverse strand (not shown).

The fact that these reads can be recovered by FLASH, which selects the base with the highest quality score in case the reads do not agree, suggests the base quality scores obtained from the MiSeq are of sufficient accuracy to correct for this error.

4.2 The rare alleles problem

The markers used in forensics have been chosen for their high variability, such that any individual on Earth is expected to have a unique combination of alleles. As a result, homozygosity is relatively uncommon. For the vast majority of the unique alleles in our data set of 271 samples no homozygous samples are available and hardly any alleles have more than a very small number of homozygous samples (Table 2). Thus, it is a necessity to estimate the systemic noise from heterozygous samples.

Table 2. Number of alleles per marker when a minimum number of samples per allele is required. If there are insufficient samples for a given allele, the heterozygous samples with that particular allele can not be used to estimate the profiles of their respective other alleles as well, resulting in a rapid loss of unique alleles. Due to the large amount of unique alleles, alleles having homozygous samples are a minority for most markers.

	Samples					Heterozygotes				
	1+	2+	3+	4+	5+	1+	2+	3+	4+	5+
Marker	Alleles									
Amel	3	3	2	2	2	1	1	1	1	1
CSF1Po	9	9	9	8	6	4	3	3	3	3
D2S1338	55	34	26	23	21	12	7	4	3	1
D3S1358	21	17	15	13	12	6	6	5	5	2
D5S818	23	20	14	14	14	8	7	5	3	2
D7S820	32	25	21	20	18	10	6	5	4	2
D8S1179	27	22	17	17	17	8	6	4	4	4
D13S317	31	23	18	16	15	10	7	5	3	3
D16S539	17	15	13	13	12	9	6	6	5	3
D18S51	19	16	14	11	10	8	7	7	6	3
D19S433	20	15	13	12	12	8	6	4	4	2
D21S11	59	35	25	21	18	10	3	3	3	2
FGA	34	24	16	12	12	8	6	4	4	3
PentaD	24	16	15	13	13	8	7	4	4	3
PentaE	19	17	17	16	16	14	8	5	5	3
TH01	14	12	8	8	8	7	4	4	4	4
TPOX	11	11	11	10	10	8	5	4	3	3
vWA	24	20	17	15	14	7	6	6	3	3

Algorithm 1 solves this problem by assigning the noise seen in heterozygous samples to either allele in a least squares optimal way. In case there is only one heterozygous sample with a particular allele though, the optimal solution is to assign any excess noise not explained by the profile of the other allele (because the sample may contain a higher than average amount of some sequences) to the profile of this allele. As a result, the estimated noise profile of this allele contains many traces of the other allele of the sample.

There are two possible solutions to this problem. The obvious solution is to simply ignore the profile of this allele because it likely contains contributions of the systemic noise of another allele which in general can not be reliably quantified. However, there is a subtle issue with this solution: the sample still influences the profile of the more common allele. And this influence is biased, since any amount of background noise above the average of the other samples with this allele was assigned to the profile of the other allele.

Therefore, the better solution is to exclude the sample from the analysis for this marker altogether. But this has one important side effect: there is also one less sample available to estimate the profile of the more common allele. For markers with many unique alleles, especially D2S1338 and D21S11, this sets off a chain reaction in which additional alleles are repeatedly removed because there was only one sample with that allele left. The problem worsens if the required number of samples per allele is increased to three or more in an attempt to get more reliable estimates (Table 2). Ultimately, a trade-off needs to be made between generating profiles for more alleles and generating fewer, more reliable profiles based on a larger minimum number of samples per allele.

4.3 Location of repeat may influence stutter

While fitting polynomials to the stutter ratio of homozygous samples, an interesting observation was made. If the shape of the polynomials is held constant, the noise ratio of –1 stutter in the 'AGAT' repeat in vWA and D3S1358 alleles was notably higher than that of the same repeat (with the same length) in other markers. For example, when fitting lines with equal slope, the slope parameter of the lines is 0.002363. The intercepts of vWA and D3S1358 are –0.01668 and –0.01583, respectively. The average intercept of the nine markers with this repeat is –0.03145. The lowest intercepts are –0.04188 and –0.03911 for D13S317 and D7S820, respectively.

D3S1358 and vWA share an interesting property: the first base in the 'AGAT' core repeat of these markers appears only 16 (for vWA) or 21 (for D3S1358) bases after the primer, whereas the repeat is located at a distance of 72 or 76 bases from the primer in D13S317 and 69 bases in D7S820. This puts the repeat in the middle of the D13S317 and D7S820 amplicons, whereas it is very close to one end in vWA and D3S1358. This suggests that the noise ratio may be systematically lower as the repeat is closer to the middle of the amplicon. The accuracy of predictions by BGPredict may therefore be improved by including the location of the repeat in the analysis. Furthermore, if variation between markers can be eliminated this way, it may be possible to predict the noise ratio of stutter in a marker-independent way.

5 Conclusion

Application of Next Generation Sequencing (NGS) analysis to forensic DNA samples has the potential to greatly increase sensitivity of forensic DNA analysis, allowing the detection of much smaller contributions than is currently possible with capillary electrophoresis (CE). However, this increased sensitivity also reveals many artefactual sequences, including the same PCR stutter artefacts as those that occur in CE analysis as well as artefactual sequences that differ from the originating allele by various single base substitutions.

The precise output of NGS analysis, which consists of exact sequences and raw read counts, allows for quantification of the occurrence and subsequent filtering of systemic noise. With the tools developed in this project, which have been published as a Python package named 'FDSTools', systemic noise can be quantified accurately on a per-allele basis in a setting dominated by heterozygous samples. Furthermore, this information can be used to filter systemic noise in forensic DNA samples, revealing small contributions that would have been overshadowed by systemic noise if no filtering was applied.

Finally, various visualisation tools were developed to ease interpretation and publication of the results.

6 Acknowledgements

I would like to thank my supervisors Kristiaan van der Gaag and Jeroen Laros for their invaluable input and the fruitful discussions during our weekly meetings. And many thanks to Kris and Rick for doing all the lab work. Without them I would not have had any data to work with in the first place.

References

[1] M. A. Jobling and P. Gill. Encoded evidence: DNA in forensic analysis. *Nat. Rev. Genet.* 5(10):739–51, Oct. 2004.

[2] P. Gill et al. Interpreting simple STR mixtures using allele peak areas. *Forensic Sci. Int.* 91(1):41–53, Jan. 1998.

[3] X. Y. Hauge and M. Litt. A study of the origin of 'shadow bands' seen when typing dinucleotide repeat polymorphisms by the PCR. *Hum. Mol. Genet.* 2(4):411–5, Apr. 1993.

[4] P. S. Walsh, N. J. Fildes, and R. Reynolds. Sequence analysis and characterization of stutter products at the tetranucleotide repeat locus vWA. *Nucleic Acids Res.* 24(14):2807–12, July 1996.

[5] J. E. Lygo et al. The validation of short tandem repeat (STR) loci for use in forensic casework. *Int. J. Legal Med.* 107(2):77–89, Apr. 1994.

[6] R. L. Sparkes et al. The validation of a 7-locus multiplex STR test for use in forensic casework. (II) Artefacts, casework studies and success rates. *Int. J. Legal Med.* 109(4):195–204, July 1996.

[7] M. R. Bill et al. PENDULUM—a guideline-based approach to the interpretation of STR mixtures. *Forensic Sci. Int.* 148(2–3):181–9, Mar. 2005.

[8] S. Y. Anvar et al. TSSV: a tool for characterization of complex allelic variants in pure and mixed genomes. *Bioinformatics*, 30(12):1651–9, June 2014.

[9] D. Shinde, Y. Lai, F. Sun, and N. Arnheim. Taq DNA polymerase slippage mutation rates measured by PCR and quasi-likelihood analysis: $(CA/GT)_n$ and $(A/T)_n$ microsatellites. *Nucleic Acids Res.* 31(3):974–80, Feb. 2003.

[10] C. Brookes, J.-A. Bright, S. Harbison, and J. S. Buckleton. Characterising stutter in forensic STR multiplexes. *Forensic Sci. Int.* 6(1):58–63, Jan. 2012.

[11] M. Klintschar and P. Wiegand. Polymerase slippage in relation to the uniformity of tetrameric repeat stretches. *Forensic Sci. Int.* 135(2):163–6, Aug. 2003.

[12] T. Magoč and S. L. Salzberg. FLASH: fast length adjustment of short reads to improve genome assemblies. *Bioinformatics*, 27(21):2957–63, Nov. 2011.

[13] M. N. Schmidt, O. Winther, and L. K. Hansen. Bayesian non-negative matrix factorization. In: *Independent Component Analysis and Signal Separation*. Springer, 2009, 540–7.

[14] A. A. Westen et al. Assessment of the stochastic threshold, back- and forward stutter filters and low template techniques for NGM. *Forensic Sci. Int.* 6(6):708–15, Dec. 2012.

[15] *Vega: A Visualization Grammar*. URL: http://trifacta.github.io/vega/.

[16] M. Harrower and C. A. Brewer. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. *Cartogr. J.* 40(1):27–37, June 2003. URL: http://colorbrewer2.org/.

[17] P. Gill, R. L. Sparkes, and C. P. Kimpton. Development of guidelines to designate alleles using an STR multiplex system. *Forensic Sci. Int.* 89(3):185–97, Oct. 1997.

[18] T. M. Clayton, J. P. Whitaker, R. L. Sparkes, and P. Gill. Analysis and interpretation of mixed forensic stains using DNA STR profiling. *Forensic Sci. Int.* 91(1):55–70, Jan. 1998.

[19] K. J. van der Gaag et al. Massive Parallel Sequencing of Short Tandem Repeats — Population data and mixture analysis. In preparation.