STRyper: a macOS application for microsatellite genotyping and chromatogram management

Jean Peccoud

Laboratoire Écologie et Biologie des Interactions, Équipe Écologie Évolution Symbiose, Université de Poitiers, UMR CNRS 7267, Bât. B31, 3 rue Jacques Fort, TSA 51106, 86073 Poitiers Cedex 9, France

jeanpeccoud@gmail.com

Running title: An application for microsatellite genotyping

Statements relating to ethics and integrity policies

- STRyper and its source code are available at https://github.com/jeanlain/STRyper/
- The author declares no conflict of interest.
- Part of this work was funded by intramural funds from the CNRS and the University of Poitiers, and by Agence Nationale de la Recherche grant ANR-20-CE02-0004 (SymChroSex).

Abstract

Microsatellite markers analyzed by capillary sequencing remain useful tools for rapid genotyping and low-cost studies. This contrasts with the lack of a free application to analyze chromatograms for microsatellite genotyping that is not restricted to human genotyping. To fill this gap, I have developed STRyper, a macOS application whose source code is published under the General Public License. STRyper only uses macOS libraries, making it very lightweight, responsive, and behaving like a modern application. Its three-pane window enables easy management and viewing of chromatograms imported from .fsa and .hid files, the creation of size standards and of microsatellite marker panels (including bins). STRyper features powerful search capabilities (with smart folders) and a modern graphical user interface allowing, among others, the edition of DNA ladders and of individual genotypes by drag-and-drop. It also introduces a new way to mitigate the effect of variations in electrophoretic conditions on estimated allele sizes.

Keywords: microsatellites, capillary electrophoresis, chromatograms, population genetics, graphical user interface

Introduction

More than three decades after their first use, microsatellites markers, also known as short tandem repeat (STR) loci, remain popular DNA markers to assess gene flow, population history, structure and membership, ancestry, or the integrity of laboratory breeding lines, among other uses [1, 2]. When locus-specific variation is not the focus of a study, a limited number of microsatellite markers are sufficient to assess evolutionary processes affecting the whole genome and to genetically identify an individual [3]. This ability stems for the sheer number of alleles per marker, which often counts in the dozens, leading to a per-locus information amount that exceeds that of single-nucleotide polymorphisms (SNPs) [4].

Due to frequent indels in the microsatellite repeat motive, microsatellites alleles essentially differ in their length, which can be estimated by simple electrophoresis of amplicons. Amplicon sequencing by the Illumina technology has however emerged as relatively affordable and more reliable alternative to capillary electrophoresis (e.g. De Barba, Miquel (5), Barbian, Connell (6), Suez, Behdenna (7)). These pipelines of microsatellite analysis via amplicon sequencing forgo the definition and optimization of microsatellite multiplexes and allow the analysis of more markers. In species for which tried and tested microsatellite multiplexes exist, microsatellite genotyping via electrophoresis still offers a compelling money- and time-saving solution. At a few dollars per individual in terms of consumables (for a couple of multiplexes typically combining 10-20 loci) genotyping can be performed locally in one day, as it amounts to DNA extraction, PCR, amplicon dilution and placing a plate in a capillary sequencer. When a quick answer is needed or when only few individuals need analyzing, typically for simple genotype checking, this traditional technique remains the cheapest and easiest one.

However, the difficulty sharply rises when it comes to analyzing the results of capillary electrophoresis. As opposed to genotyping via NGS, which is generally done via fully- or partially automated free tools (e.g., [8, 9]), traditional microsatellite genotyping requires inspecting fluorescence curves, hence applications with a complex graphical user interface (GUI), which are rarely free. To various degrees, these applications are focused on human identification by genotyping and forensics. A such, they are packed with features and safeguards that are of little relevance to most researchers, which somewhat complicate their use, and which may come at a high price.

This is the case of GeneMapper by ThermoFisher Scientific, a commercial application running on the Windows operating system, and which remains, to my knowledge, the most widely used for microsatellite genotyping. A Google scholar search for “genemapper”, excluding references, patents and review articles, and limited to 2023 and 2024, returned 2820 results as of July 20th 2024. Most results pertained to medicine and forensics or may correspond to preprints, but the first 130 results comprised 15 English-written studies on non-human species using traditional microsatellite analyzes, indicating that this technique is far from abandoned.

GeneMarker by Softgenetics is a similar commercial application. The price of a license of either software may restrict its installation to a single computer per research laboratory. A free alternative from ThermoFisher Scientific, Peak Scanner, has limited functionalities. Complementary command-line tools [10, 11] provide missing features such as allele scoring via binning, but may dissuade those who seek to conduct fragment analyses from chromatogram import to the export of individual genotypes in a single user-friendly application. In that regard, Geneious Prime and its microsatellite analysis plugin may represent an interesting tradeoff between price and features. The cost of a subscription to Geneious Prime may still appear excessive to users who do not need the features that this product offers for the analysis of DNA sequences.

Osiris [10, 11], stands out as being a free, feature-rich and multi-platform (Windows and macOS) tool for STR analysis. Yet, this software is, as far as I know, rarely used by population geneticists, possibly because it is highly specialized in human identification.

Researchers, especially population geneticists, would therefore benefit from a free application enabling quick microsatellite genotyping and management of thousands of samples. To meet this need, I have developed STRyper, an open-source, lightweight and user-friendly application that can analyze chromatogram files for STR genotyping. STRyper is published under the GNU General Public License v. 3 and its name is a portmanteau of "STR" and "Genotyper". As described below, STRyper features a modern GUI allowing, among others, unconstrained chromatogram management via nested folders, advanced and dynamic metadata-based chromatogram search with "smart" folders, easy folder import/export, chromatogram and genotype filtering based on multiple criteria, the definition of microsatellite multiplexes and custom size standards, fast and responsive visualization of fluorescence curves with animated zooming and automatic vertical scaling, the edition of DNA ladders and of individual genotypes by drag-and-drop, and a new way to mitigate the effect of variations in electrophoretic conditions on estimated allele sizes. The application and its codebase are available at https://github.com/jeanlain/STRyper.

Methods

The efficiency of an application designed for microsatellite genotyping mostly relies on its GUI, as chromatograms must be visually checked, and genotypes validated without a command line. However, underlying methods of fluorescence data analysis for automatic genotype calling are described first and in greater details than design choices, which are less of a scientific mater. Those are covered in the results section, which describe the GUI.

Fluorescence data analysis

The analysis of fluorescence data in chromatograms starts with the delineation of peaks, whose horizontal positions represent the lengths of DNA fragments. For this, a simple algorithm was developed. This algorithm (detailed in the supplementary text) determines whether a fluorescence data point (a "scan") is elevated enough, both relative to neighbor scans and in absolute fluorescence level. Peak delineation serves as a basis to subtract baseline fluorescence level, which helps peak visualization. The method developed for this task adjusts the height (fluorescence level) of a curve such that the start and end point of each peak are placed at level zero (figure S1). Although this adjustment cannot be applied on signals that are too faint to contain meaningful peaks, it has the benefit of offering two baseline subtraction modes: one that preserves absolute peak height, and one that maintains relative peak elevation compared to the baseline (Supplementary Text and Figure S1). As this method reduces background noise (Figure S1), no smoothing algorithm was implemented.

Chromatograms always contain fluorescence data from several wavelengths (channels). In multichannel fluorescence analysis, it is crucial to determine whether a peak represents a DNA fragment or interference from another channel (i.e., "crosstalk"). The method developed for this task compares the position, shape and relative size of peaks between channels, accounting for saturation of the sequencer camera (supplementary text). While certain applications alter fluorescence data to correct for pull-up due to crosstalk [11], flagging peaks resulting from crosstalk and leaving the source signal untouched was considered sufficient. These peaks are simply ignored in automatic detection of alleles and DNA ladder fragments (detailed below), although the user can manually assign these peaks, should they wish to.

Peak assignment

Two types of peaks in chromatograms must be assigned: those that correspond to DNA ladder fragments, in the context of sample sizing, and those corresponding to alleles in the context of genotyping.

The method used to detect DNA ladder fragments and assign them to sizes of a known size standard is based on relative peak positions and accounts for non-linear relationship between fragment size and migration speed (supplementary text). Peaks resulting from crosstalk or whose height are unusual compared to others are ignored. To account for non-linearity, a polynomial of the first, second, or third degree (depending on the user choice) is used to estimate fragment size, where the response variable is the size of a fragment specified in the size standard, and the explanatory variable is the scan numbers at the tip of the corresponding peak (representing migration speed). This principle is also implemented in other applications such as GeneMapper. Fitting is achieved via the Cholesky decomposition implemented in the Linear Algebra Package (https://netlib.org/lapack/). Fitting parameters are used to draw fluorescent curves ("traces") by computing the size in base pairs corresponding to every scan. Traces are therefore drawn on a plot whose horizontal axis represents the size in base pairs rather than the scan numbers. The horizontal distance between successive scans varies unless a polynomial of the first degree (linear regression) is used for the sizing.

To evaluate the quality of the sizing, a score from 0 to 1 was developed, based on the residuals of the fitted model (differences between fragment sizes as defined in the size standard, and fragment sizes estimated by the model). This score involves computing the difference in residuals for every pair of adjacent peaks and is computed as follows. If ΔR is the difference between residuals of every pair of adjacent peaks, ΔS the difference in scan number of these peaks, n_p the number of peaks and n_s is number of sizes in the size standard, the quality score is:

$$1 - \max\left(\frac{\Delta R^2}{|\Delta S|}\right)\frac{10}{3} - \frac{n_s - n_p}{10}$$

Any negative score is set to zero. This formula was tuned by testing many chromatograms to ensure that the score is greatly reduced (often to zero) by a single assignment error, forcing the user to rectify it. The score is reduced less drastically if certain sizes of the size standard are not assigned to any peak, as this generally reflects problems during electrophoresis rather than fixable errors.

Regarding allele calling, automatic genotyping must account for two main biochemical processes producing DNA fragments of different lengths. One is the addition of a non-template nucleotide to the 3' end of the new DNA strand by the DNA polymerase during PCR [12]. Because the added nucleotide is generally an adenosine, this process is referred to as “adenylation”. If adenylation affects only a portion of the amplicons, they may differ in length by one nucleotide, generating two peaks. The other process is “slippage” during replication, causing indels in the repeated region [13]. Slippage may result in a range of different amplicons whose differ by the size of the repeat, a pattern known as “stuttering”. These considerations served as a basis to develop a method for allele calling that first identifies peak clusters resulting from these processes (detailed in the Supplementary Text), and which accounts for the length of the repeat motive. In each delineated cluster, the most intense peak is considered as that representing the allele. Estimation of peak intensity accounts for clipping due to saturation of the fluorescence signal, in that the width of the saturated region is used when peak height/area may not reflect the quantity of DNA material. Stuttering and adenylation are managed internally by the application, the user remains free to manually assign an allele to any peak.

Importantly, the method does not consider the absolute height or shape of a peak to call the first allele, beyond the fact that a minimal fluorescence level is required to delineate a peak (see Supplementary Text). If a peak is detected in the marker range (see below) and is not interpreted as crosstalk, at least one allele will be called. It was considered that the assessment of peak quality was better left to the user, who is expected to visually inspect every genotype.

Genotyping

Identifying peaks representing alleles in a chromatogram requires a user-defined range of expected allele sizes at a microsatellite marker. For a diploid individual, the number of different alleles detected within that range determines the individual’s genotype: homozygous if one allele is detected, heterozygous otherwise. Because this inference is invalid for polyploid markers, it was decided that only haploid and diploid markers could be defined in the application, constraining the maximum number or alleles per locus to 2. To cope with this constraint, the ability to annotate additional DNA fragments of interest, either automatically or manually, was implemented. Additional fragments may inform on the presence of paralogs, polyploidy, insufficient specificity of the PCR or contamination between samples. The application therefore distinguishes two types of peaks: those that are interpreted as alleles and whose number is limited to the ploidy of the marker, and others representing these additional DNA fragments. Because neither should comprise fragments produced by stuttering or adenylation, additional peaks are detected like alleles are (i.e., by identifying peak clusters). The relative height of peaks is used to categorize alleles (higher peaks) and additional peaks (smaller peaks).

The second crucial phase of microsatellite genotyping is the characterization of alleles based on estimated amplicon sizes. Because the estimated size of an amplicon slightly varies between electrophoreses and never exactly match its true length [14] amplicons are assigned to alleles via “binning” [15], where bins are non-contiguous intervals delimiting the expected sizes of fragments corresponding to alleles of a marker. Proper bin definition must account for factors affecting amplicon mobility during electrophoresis [16], which often cause the estimated distance between consecutive microsatellite alleles (in base pairs) to slightly differ from the repeat motive length [15]. Binning can be left to specialized programs like Tandem [17], which can work on fragment sizes estimated by other programs like STRyper. The management of bins within STRyper was still considered a necessity. Indeed, visualizing bins behind traces helps to characterize alleles that do not conform to the periodicity of the repeat motive, and to mitigate variations in fragment sizes between sequencer runs (further discussed below). Methods to import bin sets as text files (produced by other tools like Tandem), to generate bin sets within the application, and to modify bins individually were therefore implemented. Given a set of bins for a marker, the principle of binning is simple: if the size of a fragment falls within a bin, the fragment takes the bin name. By default, a bin name is the rounded size of its midpoint when the bin was created, but it can be changed to any Unicode string.

The width of a bin might not cover the full range of estimated sizes of amplicons from given allele over all electrophoretic conditions. Regularly, a peak representing an allele would fall outside the corresponding bin, although identical fragments that migrated in other sequencer runs were properly binned. To circumvent the issue, a mixture of amplicons of known sizes for each marker, known as “allelic ladder” or “inter-lane standard”, can be added alongside samples for each run or sequencing plate. Allelic ladders are however only available for model species.

To mitigate the issue, a novel approach was conceived. Rather than managing multiple bin sets per marker, this approach considers that it is the sizes of alleles, not the position of bins, which should be adjusted. The method thus correct fragment sizes using the formula $y = a +$

bx, where x is the observed size of a DNA fragment, as estimated by the fitted model mentioned earlier, y is the size that will be used for fragments identified in the marker range, and a and b are constants (hereafter called “offset parameters”). This approach assumes that the effect of varying electrophoresis conditions can be compensated by this linear combination. If there is no correction, a = 0 and b = 1. Good offset parameters are those that minimize the distance (in base pairs) between peaks and their corresponding bins. Because automatically determining which bins and peaks to associate might have been error-prone, a manual GUI-based method was developed. The application lets the user move and/or resize a rectangle representing the range of the bin set such that bins coincide with peaks (Figure 1). To infer offset parameters a and b from this operation, we let s represents the start of a bin and e its end, in base pairs. If s' and e' represent the corresponding boundaries after the user has moved the bin set appropriately, the offset parameters can be computed by solving

$$\begin{cases} s' = a + bs \\ e' = a + be \end{cases}$$

Hence $b = \frac{e'-s'}{e-s}$ and $a = s' - s\frac{e'-s'}{e-s}$.

Since the user moves the bin set as a whole, the operation yields same offset parameters for all bins. These parameters are then associated to the chromatograms involved in the procedure (e.g., those displayed in Figure 1) and a given marker. Bin boundaries are internally unchanged (they remain e and s) such that no new bin set is created. However, bins are *displayed* using *s'* and *e'*.

Development of the application

The principle described above were incorporated in a chromatogram management application controlled by a GUI. The application needed to implement chromatogram file parsing, importing and organization, the display and editing of metadata, performant fluorescence curve drawing, the definition of microsatellite markers and bins, genotype editing, the exportation of results, etc.

GUI development relies on application programming interfaces and frameworks that depend on the target operating system and development tools. These were dictated by my use of the Mac operating system (macOS) and by the fact that developing STRyper was a hobby project of an evolutionary biologist, not the effort of a team of professional developers.

Being unencumbered by cross-platform development gave me the freedom to choose the right tools to program a GUI that was intuitive, responsive and consistent with “native” macOS applications. STRyper was thus developed using Xcode and frameworks provided by Apple (https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/OSX_Technology_Overview/SystemFrameworks/SystemFrameworks.html). These frameworks include “Core Data”, which was used to define and manage objects representing chromatograms, marker panels (multiplexes), bins, alleles, genotypes and size standards, and to save them in a persistent relational database (Figure S2). Internally, Core Data uses the SQLite database engine to manage the persistent store. GUI elements (windows, views, controls and so on) were implemented using “AppKit”. “Core Graphics” functions were used to draw fluorescent curves. “Core Animation” layers accelerate compositing via the graphical processing unit (GPU) and provide fluid animation of the interface. These object-oriented frameworks (expect Core Graphics) required the use of the Objective-C programming language (a superset of C) when the project started. The application code was written in the latest version (2.0) of this language.

Evaluating the application

STRyper was developed to facilitate the genotyping of numerous individuals (see results) from chromatogram import, management and viewing, to genotyping editing and data export. How well it performs at these tasks cannot be evaluated without a part of subjectivity.

The ability of an application to assign the right peaks to a DNA ladder fragments or alleles (i.e., allele calling) can be quantified more objectively by comparing these assignments to a reference, which is the assignments that an experienced user would have made by visually inspecting the chromatograms. Another reference, which could be used to evaluate the allele caller specifically, is the genotypes obtained at the same markers from an independent and more reliable method, typically amplicon sequencing. Such reference would allow detecting errors that even an experienced user would not detect. These errors may arise from variations in the amplification of alleles (e.g., null alleles) and motility of amplicons (leading to migration speed not being proportional to fragment length), due to the intrinsic properties of these fragments or variations in experimental conditions, including instruments and operators (reviewed in [18]). Mitigating errors that are visually indetectable should not reasonably be expected from this application. Comparing genotypes called by STRyper to those obtained by sequencing would therefore not constituting a fair evaluation of the allele caller, even if sequence data were available for the same individuals and markers (I am not aware of the public availability of such dataset). The frequency of manual corrections that an experienced user must apply to automatic peak assignment was therefore used as a metric of the application performance, even though it is partly user dependent.

Given these limitations, it was considered more valuable to evaluate STRyper as part of an ongoing study (Vucić et al., in prep) instead of reanalyzing previously published data. Chromatograms were obtained from 314 individuals of the freshwater fish *Phoxinus lumaireul* (Teleostei, Cypriniformes), each amplified at two 6-plexes of microsatellite markers developed by Vucic, Jelic (19). Amplicons were submitted to electrophoresis in an SeqStudio sequencer (Applied Biosystems) after addition of the GeneScan 500-LIZ size standard. After importing the 648 chromatograms (628 from amplified samples and 20 negative controls) into STRyper, the GeneScan 500 size standard was applied to each using the 3rd degree polynomial as sizing method. Assignments of DNA ladder fragments to sizes were then checked visually and corrected if necessary. For each marker, a set of bins was generated in one step by a specifying a bin width of 1 base pair and setting bin spacing according to the length of the microsatellite repeat motives [19]. For samples of a reference sequencing plate, the bin set was moved and resized as a whole, such that bins position matched peaks corresponding to alleles. In other plates, when bins of a marker and peaks appeared slightly misaligned, a correction factor was applied according to the procedure described in Figure 1. Individual bins were added at locations indicating the presence of alleles that did not strictly follow the repeat pattern (probably due to mutations in microsatellite flanking regions). Then genotypes were called, visually checked, and manually corrected if necessary. The complete project was then exported as an archive (supplementary file S1).

During this analysis, performance metrics were measured. Memory usage was monitored in Xcode 15, and debugging code was added to measure the time taken by the three operations that were found to cause a perceivable delay: chromatogram import, application of the size standard, and allele calling.

Results

General characteristics of STRyper

STRyper runs under macOS version 10.13 or higher. The application does not include third-party libraries and does not require special installation steps. Its bundle contains binaries compiled for the X86 and arm64 architectures and weighs less than 15 Megabytes, including the user guide.

The application comprises a main window (Figure 2) composed of three panes; a design paradigm used by several database-management applications like email clients. The left collapsible sidebar is a hierarchical list of folders and subfolders containing samples. Folder and samples can be organized freely by drag and drop. A middle pane shows the content of the selected folder (samples and associated genotypes) and comprises tabs to manage size standards and markers. The right pane shows the traces (fluorescent curves) of selected samples and genotypes.

STRyper uses very few modal panels or dialogs to validate user actions and all actions that affect the database can be undone. Most are at a couple of clicks away or less as they do not require opening and closing windows. Drag and drop can be used throughout: from importing samples to applying size standards, markers, and to manually attributing alleles or size molecular ladder fragments to peaks.

STRyper can import FSA files (HID file support is experimental, as the HID format specifications are not public) containing data for 4 or 5 channels (fluorescent dyes). Samples are imported into folders, and they can be moved or copied between folders at any time. A folder and all its content, including subfolders, samples, genotypes at microsatellite markers, associated marker panels (including bins) and custom size standards, can be archived and transferred between instances of the application. Upon importing an archived folder, any marker panel and size standard encoded in the archive is imported unless is it already in the database. The imported folder therefore shows the same content as the original one.

Since samples are not constrained to compartmentalized projects, the application provides search tools to find and gather samples from the whole database. Users can define various search criteria, including run date, sizing quality, well identifier, plate name, marker panel name, etc. Search results appear in “smart folders” which dynamically update their contents as new samples meet the search criteria.

Chromatogram viewing

Selecting a folder of the database shows all its samples, and associated genotypes if a panel of microsatellite markers have been applied to the samples. Samples can be filtered and sorted by various metadata items constituting columns that can be hidden and reordered. An inspector panel dynamically updates to show information about selected samples, including sizing information (Figure 3).

Upon selecting samples in the table, their chromatograms are instantaneously displayed on the right pane (Figure 2). As the application fully supports the dark theme of macOS (version 10.14 or more recent), it can display traces on a dark background to alleviate eye strain. Any region in which a peak statured the sequencer camera is shown behind curves as a rectangle whose color reflects the channel that likely caused saturation. Traces can be scrolled and zoomed in/out horizontally via trackpad gestures such as swipe, pinch and double tap, via the scroll wheel, or by dragging the mouse over horizontal rulers to define a size range. Dragging the mouse over the vertical ruler sets the fluorescence level at the top of the view, hence the vertical scale. Zooming is animated, which helps users keep track of the range (in base pairs) that is displayed.

Viewing options include automatic vertical scaling to the highest visible peaks, synchronizing the vertical scales and horizontal positions, showing/hiding bins and region of fluorescence saturation, stacking curves from several samples or channels in the same view, and subtracting the baseline fluorescence level (see Methods). An original option fills the areas under peaks resulting from crosstalk with the color of the channel that was inferred to induce crosstalk. This feature helps users avoid considering these peaks as alleles or DNA ladder fragments and makes clear why they were ignored during automatic genotyping.

Applying size standards and checking molecular ladders

To apply size standards to samples, STRyper comes with several widely used standards, namely those from the GeneScan brand. Users can easily edit these size standards within the application and make their own.

STRyper displays the trace of the molecular ladder like any other trace, letting users switch spontaneously between genotype and molecular ladder editing. Sizes attributed to molecular ladder fragment can be changed by dragging and dropping size labels onto peaks. Any change to the molecular ladder automatically updates the sizing of the sample without user validation. The red component of the color used for size labels is proportional to the difference between the computed size of a peak and its theoretical size, making size assignment errors easy to spot. The sample inspector (Figure 3) also helps to find such errors if points deviate from the curve representing the relationship between scan number and peak size.

Genotyping

Users can define their own panels of haploid or diploid microsatellite markers within STRyper and organize them in folders. Markers are defined by their fluorescent dye, ploidy, length of repeat motive, name, and the size range of their alleles. These attributes can be changed after a marker is created, except for the first two. Markers can be copied between panels. Users can export marker panels, which contain bins, to text files conforming to simple specifications described in the user guide. These text files can be imported back as marker panels. STRyper can also import panel and bin description text files exported from Genemapper.

A set of automatically named bins for a marker can be added by specifying the width and spacing of bins. To accommodate the fact that the observed distance between microsatellite alleles slightly differs from the repeat motive [15], the position and width of the whole be set can be adjusted by clicking and dragging, in a fashion similar to that described in Figure 1. Individual bins can also be added and modified via click and drag. These actions do not involve dedicated windows or panels, they can be performed at any time on the trace views where bins are displayed (Figure 2, right).

All genotypes from samples of the current folder are listed in a table that can be sorted and filtered according to various criteria (including allele names and sizes). This table lets users quickly scan genotypes, as corresponding peaks and allele labels of the selected genotype(s) appear on the right-pane. Correcting errors in allele call typically takes a single step: the user can simply drag the mouse from a peak to a bin, drag an allele label from one peak to another (Figure 4), or double-click a peak, which removes/attaches an allele from/to the peak. Double clicking allele labels lets users enter arbitrary allele names directly above peaks.

Genotypes and associated sample metadata can be exported as text files, or simply copied from selected table rows to a text editor or a spreadsheet application.

Evaluation of the software

STRyper was used to genotype 314 *Phoxinus limaireul* individuals amplified at two multiplexes of six markers each (see Methods). Ignoring electrophoresis failures that made 25 samples unusable, visual inspection of peak assignments to DNA ladder fragments found issues in height chromatograms. In all cases, a size was not assigned to the appropriate peak or the DNA ladder because the peak was missing or abnormally short. As issues due to missing peaks cannot be fixed, manual corrections were applied to only four chromatograms. Overall, the verification of the DNA ladder for all chromatograms took less than five minutes.

For certain sequencing plates and markers (PHOX4, PHOX11, PHOX29, PHOX33 and CtoA-247 [19]), peaks and bins appeared slightly misaligned (by less than 0.5 base pairs). Offset parameters for peak sizes were thus defined according to the procedure described in Figure 1. This procedure made bins coincide neatly with peak locations for all regularly spaced alleles. I therefore saw no evidence that the use of linear relationship to estimate offset parameters was inappropriate.

In cases of PCR failures, the application assigned relatively faint peaks amounting to noise as alleles, which was expected. These cases were easily detected by visual inspection.

The only common source of error resulted from varying degrees of adenylation at certain markers. The most intense peak or a cluster, which the application assigns to an allele, may sometime represent adenylated fragments and sometimes non-adenylated fragments. The estimated size of the same allele will therefore vary between individuals by approximately one base pair. This type of variation was much more rarely induced by stuttering, the degree of which is more constant.

In rare instances, peaks representing alleles had the same position as taller peaks in other channels and were erroneously considered as resulting from crosstalk. These errors were detected because neighboring peaks of similar shapes were present (indicating stuttering or heterozygosity) despite the absence of peaks in other channels at their position. More frequently, small artefactual peaks were not interpreted as crosstalk because their shape was irregular and/or their position was slightly shifted from that of the peaks that induced interference. This issue rarely affected genotyping as these peaks were generally too small to be considered as alleles. In a few cases though, an artifactual peak (not identified as such) was taken as an allele instead of the correct peak. This occurred in very specific situations where the length of the true alleles differed by only one base pair, such that one of the peaks they induced was considered to result from adenylation. Only comparison with other genotypes showed that adenylation was unlikely. The genotype caller does implement such check (Supplementary Text) and tries to correct genotypes that were initially considered homozygous. In the cases described here however, the genotypes were considered heterozygous, since an artefactual peak was taken as the second allele, and were therefore not checked.

Finally shorter allele dominance in heterozygotes [20], causing the peak representing the longer allele to be much smaller due to a very large difference in length between alleles (> 60 bp), was not always properly managed. Admittedly, whether such peak should be considered as an allele is difficult to determine even for experienced users.

Speed, memory usage and responsiveness

When it came to execution speed, importing the 648 chromatograms took 2.45 s, i.e., 264 chromatograms were imported per second on average. Application of the size standard (which involves peak assignment to DNA ladder fragment) took less than 0.12 s (~5400 chromatograms per second). Allele calling of the 3600 genotypes took 0.24 s (~15000 genotypes called per second). Since chromatograms/genotypes are processed successively in a single execution thread, the runtime of these tasks is proportional to the number of chromatograms or genotypes processed.

Memory usage was measured at 132 Megabytes (MB) after chromatogram import. It peaked at 250 MB after selecting the 3600 called genotypes and scrolling the 3600 traces from top to bottom and back. Memory usage peaked at 460 MB after selecting the 648 samples to display the stacked traces at the five channels (2000 traces displayed at once, as the application does not display more than 400 stacked traces per row).

All tasks other than those timed above were essentially instantaneous. Only the display of the 3600 genotypes and the 648 samples in the right pane induced a noticeable delay of about 1 second. On a laptop with a high-resolution 120 Hz display and an M1 Pro chipset, zooming and scrolling traces was generally achieved without noticeable frame drops, except when zooming in/out more than about 500 traces (stacked in several rows) near their full range (about 600 base pairs).

Discussion

Based on its design, features and performance, STRyper should be a valuable tool for researchers who use traditional microsatellite markers. Genotyping hundreds of *Phoxinus* individuals at 12 markers with STRyper proved much faster than any of my previous genotyping jobs on similar data, keeping in mind that I cannot afford a comparison with recent versions of commercial competing applications. This test also showed that crosstalk detection and genotype calling was reasonably efficient, although improvable. While the underlying methods can surely be refined, I believe that substantial improvements in these areas require comparisons between samples. Ideally, trained artificial intelligence could be employed to analyze chromatograms [21], assuming STRyper development benefits from the contribution of AI specialists. As STRyper is not immune to genotyping errors (reviewed in [18]) one should always visually review genotypes and perform downstream corrections on exported results (e.g., [22-24]).

Independently of the performance of its allele caller, the main benefits of STRyper lie in its streamlined user interface that is optimized for the management and inspection of hundreds of chromatograms. This optimization is essential to population geneticists, who cannot spend as much time on individual genotypes as forensic researchers can. Since STRyper is not designed for diagnostics and must not be used for this task (it comes with no guaranty), it does not assume that allele calls are reviewed by several users. Therefore, it does not record the history of manual corrections applied to genotypes (but still allows adding comments on genotypes). Such feature would have cluttered the user interface for very little benefits for most researchers.

Based on the reported metrics, users should not be concerned about the performance and responsiveness of STRyper. The size of the database and the number of samples contained in the selected folder should have little effect on the application performance and memory usage. The application essentially shows tables (including its right pane), for which only the visible rows, and a few others kept in cache for performance, are allocated in system memory (a feature provided by the NSTableView class of the AppKit framework). Rows that are not yet visible are not allocated, and those that move out view during scrolling eventually become deallocated. When chromatograms are fetched using textual metadata (sample name, plate well, plate name, run date, etc.), for example during a search though the whole database, only that piece of data is fetched from the store and allocated in memory (a feature of the Core Data framework). Fluorescence data is stored in separate objects (Supplementary Text) and is only fetched and allocated in memory when traces are displayed.

As the application only uses about 460 MB when displaying 2000 traces at once – the most that is allowed – memory usage should not be a concern either. The use of Apple-provided frameworks (mainly AppKit, Core Data and Core Animation) contributes to the low memory footprint and responsiveness of STRyper but would require a major rewrite of the GUI and database-management code if the application were to be ported to non-Apple platforms. However, methods related to chromatogram parsing, peak assignments (genotype calling and sizing) and drawing of fluorescence curves do not heavily depend on these frameworks (they mostly use functions written in plain C) and can be reused with only minor modifications.

From a GUI standpoint, several features of STRyper should be particularly useful to users. The first is the distinction between alleles and additional peaks. Since the number of peaks assigned to alleles never exceeds the marker ploidy, users should rarely need to remove peaks to correct a genotype that was called, a repetitive task that proved rather tedious in my previous genotyping jobs. The detection of additional peaks is optional, and these peaks can be reviewed, added manually, removed, or simply ignored as they are not part of an individual’s genotype (they are listed and exported in a dedicated column). Theoretically, additional peaks should allow genotyping polyploid species, but I have not tested STRyper for this usage.

The second feature to underline is the implementation of fragment binning. The possibly to assign off-bin peaks to alleles (bins) via drag-and-drop (Figure 4, left) is certainly a time saver compared to typing allele names or selecting them among a long list. This task can even be avoided by minimizing the offset between peak and bin locations (Figure 1) prior to binning, in case variations in electrophoretic conditions have shifted the position of peaks relative to bins. This is currently done manually by the user, but a fully automatic, or user-assisted, procedure that minimizes the offset between peaks and bins (or theoretical fragment sizes) could be the goal of future developments. Granted, binning can be performed automatically by downstream programs like Tandem [17]. However, minimizing the offset between bins and peak representing “standard” alleles should help to distinguish alleles whose size do not follow the periodicity of the microsatellite repeat motive, and which may justify the creation of specific bins. Tandem alerts the user about problematic alleles but does not create new bins.

When it comes to database management, STRyper distinguishes itself by advanced search and filtering capabilities, which help reviewing problematic cases, among other benefits. For instance, all samples showing a particular allele at a marker can easily be retrieved across the whole database and displayed. To this end, samples can be gathered in a smart folder according to the name of the marker panel applied to them. Then, the list of their genotypes can be filtered based on the marker name, and the allele name or size. Any new genotyped sample presenting this allele would automatically appear in the smart folder.

Finally, the set of chromatograms contained in a folder (or a smart folder) with all its related data (marker panels and bin sets, custom size standard(s), genotypes…) is easy to share, as it can be transferred between instance of STRyper with a few mouse clicks and no option to set. Making folder archives available alongside any publication using STRyper should help to review results and to standardize the analysis of the same microsatellite markers by different researchers.

Acknowledgements

I thank Dr. Douglass Hoffman from the United States National Institute of Health for his advice on decoding HID files, and numerous colleagues for testing STRyper. I also thank Drs. Matej Vucić and Frédéric Grandjean for giving access to *Phoxinus* chromatogram files, and Dr. Romain Pigeault for reading a draft of the manuscript. Part of this work was funded by intramural funds from the CNRS and the University of Poitiers and by Agence Nationale de la Recherche grant ANR-20-CE02-0004 (SymChroSex).

References

1. Hauser SS, Athrey G, Leberg PL. *Ecol. Evol.* 2021;11(22):15800-14. http://doi.org/https://doi.org/10.1002/ece3.8250

2. Hodel RG, Segovia-Salcedo MC, Landis JB, Crowl AA, Sun M, Liu X, et al. *Appl Plant Sci*. 2016;4(6). http://doi.org/10.3732/apps.1600025

3. Selkoe KA, Toonen RJ. *Ecol Lett*. 2006;9(5):615-29. http://doi.org/10.1111/j.1461-0248.2006.00889.x

4. Verbiest M, Maksimov M, Jin Y, Anisimova M, Gymrek M, Bilgin Sonay T. *J Evol Biol*. 2023;36(2):321-36. http://doi.org/10.1111/jeb.14106

5. De Barba M, Miquel C, Lobréaux S, Quenette PY, Swenson JE, Taberlet P. *Mol. Ecol. Resour.* 2017;17(3):492-507. http://doi.org/https://doi.org/10.1111/1755-0998.12594

6. Barbian HJ, Connell AJ, Avitto AN, Russell RM, Smith AG, Gundlapally MS, et al. *Ecol. Evol.* 2018;8(16):7946-63. http://doi.org/https://doi.org/10.1002/ece3.4302

7. Suez M, Behdenna A, Brouillet S, Graça P, Higuet D, Achaz G. *Mol. Ecol. Resour.* 2016;16(2):524-33. http://doi.org/https://doi.org/10.1111/1755-0998.12467

8. Liu P, Wilson P, Redquest B, Keobouasone S, Manseau M. *Mol Ecol Resour*. 2024;24(3):e13929. http://doi.org/10.1111/1755-0998.13929

9. Zhan L, Paterson IG, Fraser BA, Watson B, Bradbury IR, Nadukkalam Ravindran P, et al. *Mol. Ecol. Resour.* 2017;17(2):247-56. http://doi.org/https://doi.org/10.1111/1755-0998.12561

10. Goor RM, Forman Neall L, Hoffman D, Sherry ST. *Bull Math Biol*. 2011;73(8):1909-31. http://doi.org/10.1007/s11538-010-9598-0

11. Goor RM, Hoffman D, Riley GR. *Forensic Science International: Genetics*. 2021;51. http://doi.org/10.1016/j.fsigen.2020.102410

12. Clark JM. *Nucleic Acids Res*. 1988;16(20):9677-86. http://doi.org/10.1093/nar/16.20.9677

13. Hauge XY, Litt M. *Hum Mol Genet*. 1993;2(4):411-5. http://doi.org/10.1093/hmg/2.4.411

14. Rosenblum BB, Oaks F, Menchen S, Johnson B. *Nucleic Acids Res*. 1997;25(19):3925-9. http://doi.org/10.1093/nar/25.19.3925

15. Idury RM, Cardon LR. *Genome Res*. 1997;7(11):1104-9. http://doi.org/10.1101/gr.7.11.1104

16. AMOS W, HOFFMAN JI, FRODSHAM A, ZHANG L, BEST S, HILL AVS. *Mol. Ecol. Notes*. 2007;7(1):10-4. http://doi.org/https://doi.org/10.1111/j.1471-8286.2006.01560.x

17. Matschiner M, Salzburger W. *Bioinformatics*. 2009;25(15):1982-3. http://doi.org/10.1093/bioinformatics/btp303

18. Pompanon F, Bonin A, Bellemain E, Taberlet P. *Nature Reviews Genetics*. 2005;6(11):847-59. http://doi.org/10.1038/nrg1707

19. Vucic M, Jelic M, Klobucar G, Jelic D, Gan HM, Austin C, et al. *J. Fish Biol.* 2022;101(5):1225-34. http://doi.org/https://doi.org/10.1111/jfb.15194

20. Walsh PS, Erlich HA, Higuchi R. *PCR Methods Appl*. 1992;1(4):241-50. http://doi.org/10.1101/gr.1.4.241

21. Taylor D, Powers D. *Forensic Science International: Genetics*. 2016;25:10-8. http://doi.org/10.1016/j.fsigen.2016.07.013

22. Wang C, Schroeder KB, Rosenberg NA. *Genetics*. 2012;192(2):651-69. http://doi.org/10.1534/genetics.112.139519

23. De Meeûs T, Noûs C. *Peer Community Journal*. 2022;2. http://doi.org/10.24072/pcjournal.165

24. Van Oosterhout C, Hutchinson WF, Wills DPM, Shipley P. *Mol. Ecol. Notes*. 2004;4(3):535-8. http://doi.org/https://doi.org/10.1111/j.1471-8286.2004.00684.x

Data accessibility

STRyper and its source code are available at https://github.com/jeanlain/STRyper/

The archive of the folder containing the analyzed data from the 314 *Phoxinus sp* individuals is available as supplementary file S1.

Author Contributions

JP developed the application, analyzed data and wrote the paper.

Figures

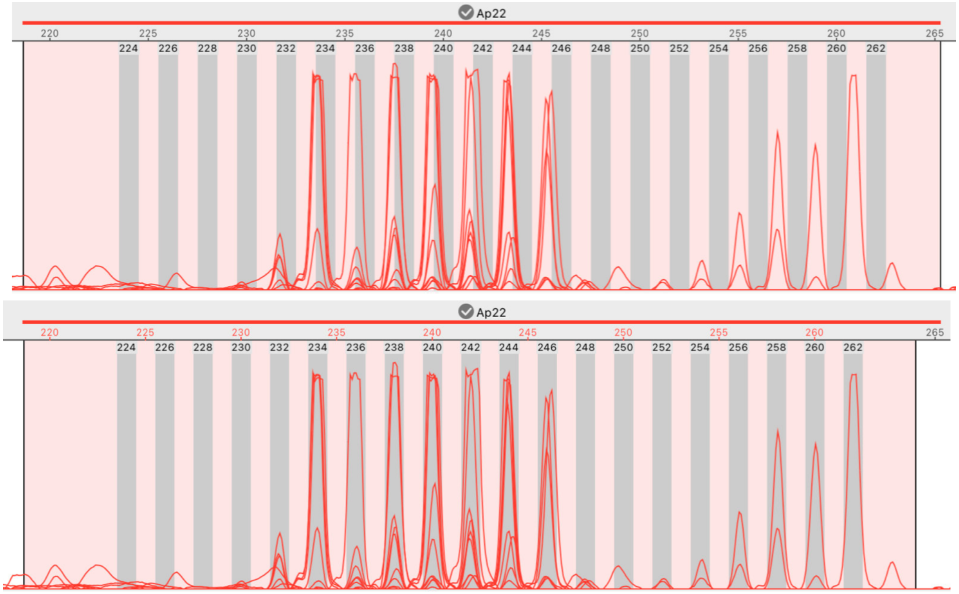
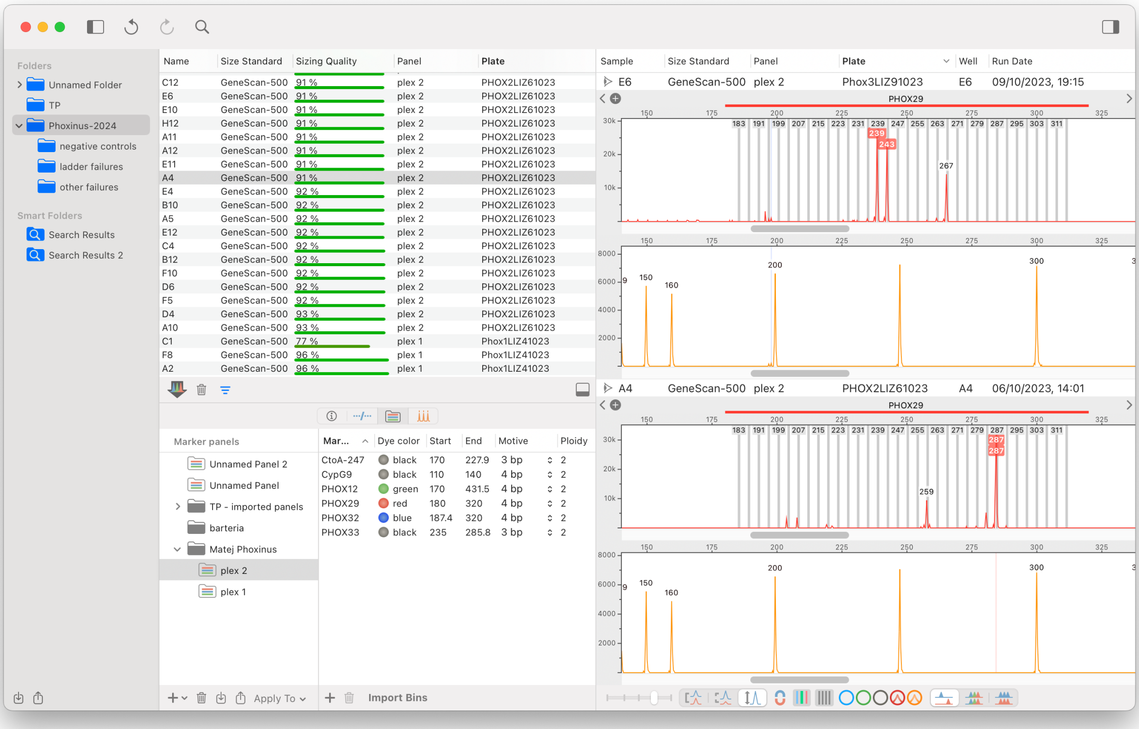


Figure 1. A case of out-of-bin alleles that is solved. Both images show the stacked traces from 8 samples of the same sequencer run. Peaks represent amplicons of a dinucleotide marker called “Ap22”. Its range is represented by a horizontal red segment above the ruler showing graduations in base pairs (bp). Bins appear as grey rectangles. Top: peaks are shifted to the left with respect to bins, and more so for longer alleles, although bins are separated by exactly two base pairs. Bottom: the user has moved and narrowed the light-pink rectangle representing the range of the marker, such that bins coincide with peaks. This move translates into offset parameters a = –6.40 and b = 1.029 (see main text). As a result, the estimated size of peaks overlapping bin 258 (bottom image) has changed from ~257 bp to ~258.1 bp.



Name	Size Standard	Sizing Quality	Panel	Plate
C12	GeneScan-500	91 %	plex 2	PHOX2LIZ61023
E6	GeneScan-500	91 %	plex 2	PHOX2LIZ61023
E10	GeneScan-500	91 %	plex 2	PHOX2LIZ61023
H12	GeneScan-500	91 %	plex 2	PHOX2LIZ61023
A11	GeneScan-500	91 %	plex 2	PHOX2LIZ61023
A12	GeneScan-500	91 %	plex 2	PHOX2LIZ61023
E11	GeneScan-500	91 %	plex 2	PHOX2LIZ61023
A4	GeneScan-500	91 %	plex 2	PHOX2LIZ61023
E4	GeneScan-500	92 %	plex 2	PHOX2LIZ61023
B10	GeneScan-500	92 %	plex 2	PHOX2LIZ61023
A5	GeneScan-500	92 %	plex 2	PHOX2LIZ61023
E12	GeneScan-500	92 %	plex 2	PHOX2LIZ61023
C4	GeneScan-500	92 %	plex 2	PHOX2LIZ61023
B12	GeneScan-500	92 %	plex 2	PHOX2LIZ61023
F10	GeneScan-500	92 %	plex 2	PHOX2LIZ61023
D6	GeneScan-500	92 %	plex 2	PHOX2LIZ61023
F5	GeneScan-500	92 %	plex 2	PHOX2LIZ61023
D4	GeneScan-500	93 %	plex 2	PHOX2LIZ61023
A10	GeneScan-500	93 %	plex 2	PHOX2LIZ61023
C1	GeneScan-500	77 %	plex 1	Phox1LIZ41023
F8	GeneScan-500	96 %	plex 1	Phox1LIZ41023
A2	GeneScan-500	96 %	plex 1	Phox1LIZ41023

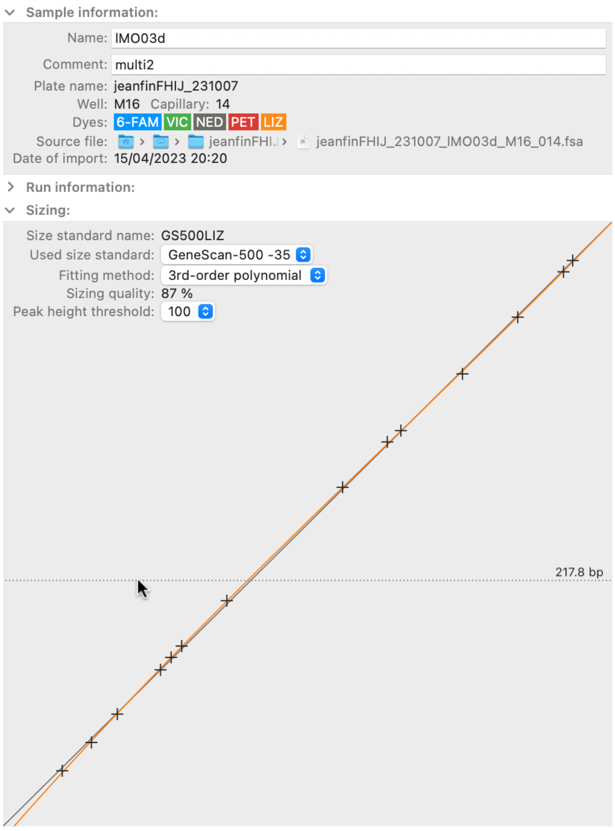
Marker panels

- Unnamed Panel 2
- Unnamed Panel
- TP - imported panels
- barteria
- Matej Phoxinus
  - plex 2
  - plex 1

Mar...	Dye color	Start	End	Motive	Ploidy
CtoA-247	● black	170	227.9	3 bp	2
CypG9	● black	110	140	4 bp	2
PHOX12	● green	170	431.5	4 bp	2
PHOX29	● red	180	320	4 bp	2
PHOX32	● blue	187.4	320	4 bp	2
PHOX33	● black	235	285.8	3 bp	2

Sample	Size Standard	Panel	Plate	Well	Run Date
E6	GeneScan-500	plex 2	Phox3LIZ91023	E6	09/10/2023, 19:15
A4	GeneScan-500	plex 2	PHOX2LIZ61023	A4	06/10/2023, 14:01

Figure 2. The main window of STRyper. The left pane contains the list of folders and smart folders (search results) containing samples. The middle pane is a split view comprising a top pane listing the samples of the selected folder. Its bottom pane has four tabs, which are from left to right: an inspector showing data on selected samples (Figure 2), a table of genotypes from the samples shown on the top pane, the marker library (currently shown) and the size standard library. The right pane shows the traces of selected samples in a scrollable view that can display thousands of traces. The red channel currently shows the range of a diploid DNA marker ("PHOX29") that contain bins shown as vertical grey rectangles. Alleles are annotated with rectangular labels colored after the channel. Supplementary peaks in bins "267" and "259" have been annotated. The orange channel shows the molecular ladder.



Sample information:

Name: IMO03d
Comment: multi2
Plate name: jeanfinFHIJ_231007
Well: M16 Capillary: 14
Dyes: 6-FAM VIC NED PET LIZ
Source file: › › jeanfinFHIJ › jeanfinFHIJ_231007_IMO03d_M16_014.fsa
Date of import: 15/04/2023 20:20

Run information:

Sizing:

Size standard name: GS500LIZ
Used size standard: GeneScan-500 -35
Fitting method: 3rd-order polynomial
Sizing quality: 87 %
Peak height threshold: 100

Figure 3. The sample inspector of STRyper. This panel with three collapsible sections dynamically updates to display information on samples that are selected in the sample table (Figure 2). The plot at the bottom shows the relationship between the time at which DNA fragments of the molecular ladder (black crosses) were detected by the sequencer camera (the X axis) and their observed sizes in base pairs (the Y axis). The relationship used to estimate fragment sizes is established by fitting a polynomial (here, of the third degree) to the points shown on the plot. This polynomial is represented by the orange curve.

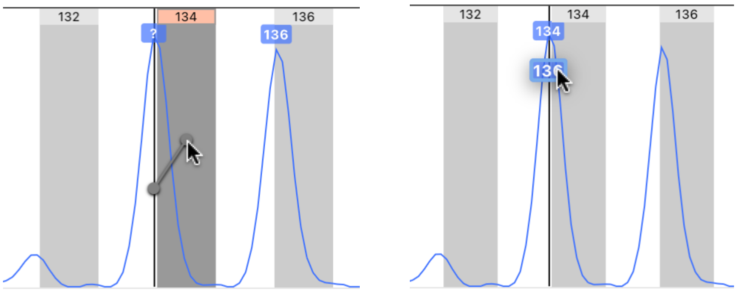


Figure 4. Genotype editing by drag and drop in STRyper. Vertical grey rectangles represent bins that define expected ranges of microsatellite alleles. Each bin has a name displayed on

top. Allele names are represented by colored labels above peaks. Left-hand screen capture: the user is dragging the mouse from a peak to a bin. This will assign the peak an allele named after the bin, thereby replacing the question mark used for alleles that are out of bins. During the operation, a grey-colored handle connects the mouse location to another point horizontally located at the peak tip and vertically located at the clicked point. Right-hand screen capture: the user has decided that only the peak on the left should represent an allele and is dragging an allele label from the right-hand peak to the other. These actions are assisted by “magnetism” to lock the handle or allele label to the closest suitable destination, which triggers haptic feedback on the trackpad.

Supporting information for "STRyper: a macOS application for microsatellite genotyping and chromatogram management"

Peak delineation

To delineate peaks in the fluorescence data, STRyper uses a simple method that enumerates fluorescence levels from the first to the last recorded scan. A scan is a data point that is denoted by an integer index varying from 0 to the total number of data points.

The method records the lowest fluorescence level (l) and the highest level (h), and their respective scan numbers (s_l, s_h), observed up to the current scan number (s_f) whose fluorescence level is denoted as f. A peak is delineated if $h > t$, $l/h \leq r$ and $f/h \leq r$, t being the minimal fluorescence level to consider a peak (by default, 100 fluorescence units) and r being a parameter denoting the minimum peak elevation above the background. Horizontally, the peak starts at scan s_l, and its tip is at scan s_h. Its right boundary will correspond to the left boundary of the next peak. This method thus generates contiguous peaks.

For best results, it was found that three rounds of peak detection should be applied to the data, each round being followed by one pass of baseline fluorescence level subtraction (see next section). The first two rounds use a value of 0.7 for r, a modest peak elevation that allows the detection of faint peaks. The last iteration uses r = 0.5, which means that a peak must be at least twice higher than the background level, considering that baseline fluorescence level subtraction makes peak stand-out more.

After these three rounds, the left and right boundaries of each peak are delineated by the closest scan from each side of the peak's tip that has a fluorescence level of 0, using fluorescence levels with baseline level subtracted. This produces non-contiguous peaks.

Baseline fluorescence level subtraction

STRyper subtracts the baseline fluorescence level of a trace after peaks are delineated (see previous section) as follows. A virtual line segment is drawn from the start to the end of each peak (Figure S1). For a given scan number s_f, the height of the segment is denoted as y and is considered the "baseline fluorescence". The recorded fluorescence level for the scan is denoted as f and the fluorescence level at the peak tip is denoted as h.

For each value of s_f within the peak, a value v to subtract to the fluorescence level depends on whether the absolute height of peaks should be preserved. If so, $v = y(h - f)/(h - y)$. Otherwise, $v = y$. If v is negative, it is set to 0. The new value for the fluorescence level is $f - v$. After this operation, each peak starts and ends at a fluorescence level of 0.

The same operation is performed between peaks to reduce the background noise. Between peaks, $v = y$.

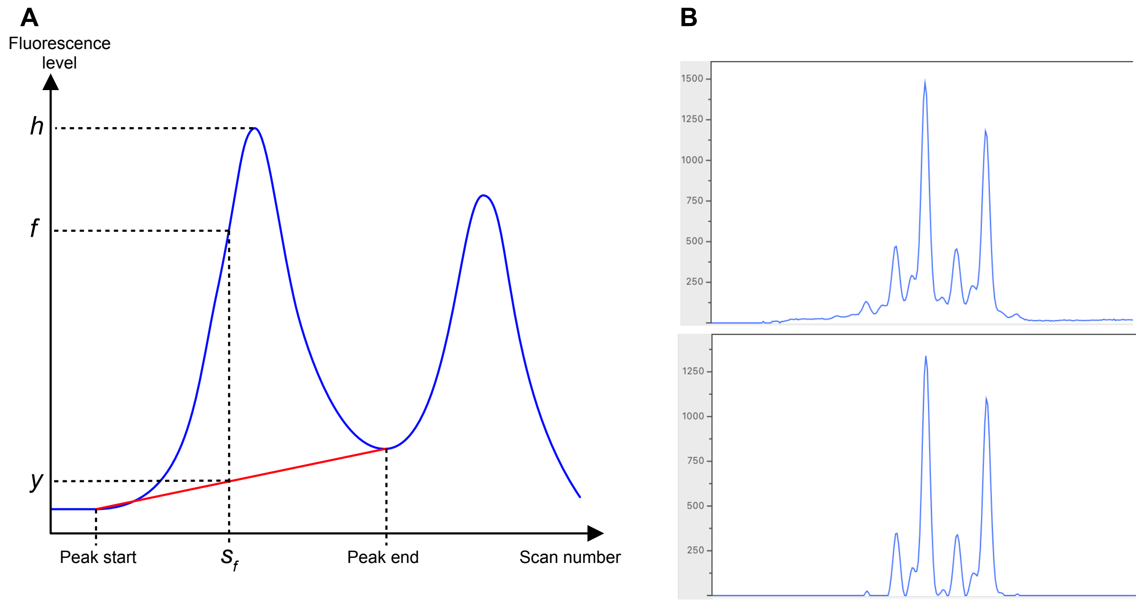


Figure S1. Subtraction of baseline fluorescence level. A) Principle of the method. Symbols are defined in the supplementary text. B) Effect of the three passes of the method (see supplementary text) on fluorescence curves. Top: raw fluorescence data. Bottom: fluorescence data after baseline fluorescence level was subtracted.

Determination of crosstalk

STRyper determines whether a peak in fluorescence results from interference between channels, i.e., crosstalk. This inference relies on the presence of saturation, or of higher peak of similar shapes, in other channels.

A chromatogram file lists each scan number for which the signal saturated the sequencer camera but does not specify which channel caused the saturation. STRyper determines this channel by first delineating regions composed of consecutive scan numbers where saturation occurred.

For each region, the channel that is considered to have caused saturation is the one whose fluorescence level is the highest at the first scan of the region. This criterion does not compare maximum/average fluorescence levels over the region between channels, because the peak at the channel that caused saturation if often clipped and may be smaller than peaks of other channels in the region. However, this peak has the highest fluorescence level at the point where saturation began.

A peak is considered to results from crosstalk if the following conditions are met: (i) its tip lies within a region where saturation is caused by another channel, and (ii) the fluorescence level at the peak tip is at least twice those recorded at the scan preceding the start and the scan after the end of the region. Criterion (ii) accounts for the fact that several DNA fragment may have migrated at the same speed, such that legit peaks appear at the same locations. However, the fluorescence level at a peak resulting from crosstalk should not be high before the saturation from another channel is recorded.

Alternatively, crosstalk may cause a “crater” in other channels, that is, sharp peaks at the edges of the saturated region. If a small peak lies near such edge and sharply decreases within the saturated region, the peak is considered to result from crosstalk.

If a focus peak is not considered are resulting from crosstalk based on these criteria, the program inspects other channels to find the one with highest fluorescence level at the peak tip, and for which the fluorescence level is at least 1.6 times that at the peak tip. If it finds one, it then evaluates how much peaks of both channels overlap, using two criteria. The program first scales down the higher peak such that is elevation corresponds to the smaller. It then measures the peak areas by summing fluorescence levels. The first criterion is considered passed if the area representing the intersection between peaks is at least 30% of the area representing the union of the peaks. The second criterion precisely evaluates how much the peak horizontal positions are aligned. For that, the difference in fluorescence level (curve height) between channels is computed at each scan along the range encompassing both peaks. The sign of the difference is reversed if the scan is greater than the scan of a given peak’s tip. For each peak, these differences are summed across all scans of the range. The second criterion is considered passed if the absolute value of each sum is less than 30% the combined areas of the peaks. If both criteria are met, the program checks if other peaks in the channel that may have induced crosstalk also induced crosstalk in the focus channel.

This inspection relies on the expected ratio of peak heights between the two channels, which should be rather constant in the case of crosstalk and in the absence of saturation. If another peak does not appear to have induced crosstalk, then the peak under consideration is not considered to result from crosstalk.

Size assignment of molecular ladder fragments

The algorithm conceived to assign sizes to molecular ladder fragments inspects peak in the appropriate channel, ignoring those resulting from crosstalk (see previous section). In the following, the “scan number” of a peak refers to the scan at its tip.

Peaks are first enumerated by decreasing scan numbers, and the average peak height is computed at each step. Any peaks whose height is at least twice the current average and whose scan number is less than 1/3 total number of scans in the trace is discarded. This eliminates high-intensity peaks of short size (in base pairs) resulting from degradation of the molecular ladder.

The algorithm then discards weak peaks amounting to “noise”, which sometimes affect the data. To do so, remaining peaks are enumerated by decreasing height. The enumeration stops when the number of enumerated peaks corresponds to the number of sizes specified in the size standard, or when a peak is at least three times smaller than the previous one. Any peak that is at least twice as small as the least enumerated peak is discarded.

To assign remaining peaks to sizes defined in the size standard, peaks are ordered by increasing scan number. The method assigns the lowest size to the first peak, and the largest size to the last peak. To understand the process, picture a straight line of equation $y = a + bx$ passing through these two peaks on a plot where the x axis represents scan numbers, and the y axis sizes in base pairs.

Peaks are then enumerated in decreasing order, starting from the second-to-last. The size of the fragment causing a peak is estimated as $a + bx$, x being the peak scan number. The size of the size standard that is the closest to the observed size is assigned to the peak, only if the difference between both sizes is less than 15 bp in absolute value.

The next peak is evaluated in the same fashion. If it is assigned to the same size as a previous peak, both peaks are confronted to retain the one whose predicted size is the closest. The a

and b parameter are updated to correspond to the line connecting the two peaks that were assigned last. Hence, the size/scan relationship dynamically changes to account for non-linearity.

At the end of the procedure, the shortest size of the size standard may be assigned to a different peak than the one of lowest scan number. This is not the case for the longest size, which remains assigned to the peak of largest scan number, although this assignment might be erroneous (this is addressed using subsequent iterations, as described below).

A quality index is computed to evaluate the assignments. This index relies on the residuals of the linear regression between scan number and size in base pairs, using ordinary least squares. For each pair of successive points (peaks), the difference between residuals is divided by the difference between scan numbers, both in absolute value. The mean of these ratios is computed. The inverse of this mean, multiplied by the percentage of sizes that were assigned to peaks, constitutes the quality index. If this index is higher than a certain value (chosen at 100), the number of assigned sizes is recorded as a reference.

Further iterations of assignments are performed by decrementing the longest assignable size (to consider the possibility that electrophoresis failed or stopped before the last fragment was detected), then by decrementing the last assignable peak. Assignments are not recorded if the number of assigned sizes is lower than the reference, and iterations stop when the number of assignable sizes/peaks is lower than the reference.

In the end, the set of assignments that yielded the best quality index is retained.

Detection of microsatellite alleles

To identify microsatellite alleles at a marker in a chromatogram, peaks found in the marker range are first sorted by decreasing number of saturated scans they induced, then by decreasing height (fluorescence level). This sorting accounts for clipping due to saturation of the fluorescence signal. Hereafter, a peak position/size (in base pairs) refers to the position of its tip, hence the estimated length of the DNA fragment that induced the peak.

For each peak (hereafter called a “reference” peak), neighboring peaks are successively inspected at increasing distance to identify peak clusters. Neighbors that are at the left (lower scan numbers) are inspected before those at the right. Briefly, the inspection first evaluates if a neighbor resulted from stuttering: the distance between the neighbor and the reference peak (of from a previously inspected neighbor already considered as a stutter) must be the motive length of the marker ± 0.5 bp. In addition, the neighbor must be smaller than the reference peak or than a previously inspected neighbor already considered as a stutter. If these requirements are met, the neighbor is flagged as a “child” of the reference peak, that is, both are considered part of the same cluster and have arisen from amplification of the same allele. If the distance between a neighbor and the reference peak (or a previously inspected neighbor) lies between 0.5 and 1.5 bp, and if the neighbor is smaller than the reference peak, the neighbor is interpreted as resulting from adenylation of the amplicon, hence as a child peak. This is also the case if the neighbor is distant from the motive length ± 0.5 bp from a previous peak considered as resulting from adenylation.

The application stops the inspection of neighbors (at the left or at the right or a reference peak) if a neighbor is already flagged as a child from a reference peak inspected prior, or if its distance from the last inspected neighbor exceeds the motive length + 0.5 pb, in which case it the neighbor is not considered as a child.

Additional checks based on peak heights are implemented to avoid considering alleles differing by only one repeat motive as part of the same cluster. One check accounts for short allele dominance in heterozygotes: the fact that the longer allele is almost always amplified with lower yield during PCR. The resulting peak is therefore smaller, but it must not be considered as a result a stuttering. To account for dropout, a neighboring peak at the right of the reference peak is considered as resulting from stuttering only if its height is <30% that of the reference peak. Conversely, the method also considers rare cases where the shorter allele has amplified with lower yield than the longer. If allele lengths differ by just one repeat motive, the left peak may be erroneously considered as a stutter. To avoid this, the method computes the ratio of peak heights (left peak / right peak). If the ratio is ≥ 0.7, the program looks for an additional stutter peak at the left, at distance that is the motive length ± 0.5. This check assumes that a first peak arising from stuttering is followed by others with similar height ratio between neighboring peaks. If no such peak is found, the left peak is not considered as a child peak, and the inspection of neighbors that are at the left of the reference peak stops. Note that if the shorter allele amplifies with a much lower yield than the longer allele, distinguishing it from a stutter would require comparing individuals, which the application does not.

After neighbors are inspected for all reference peaks, all peaks are processed by decreasing height. The first peak is always considered as an allele and any subsequent peak will be as well if the following conditions are fulfilled. First, the number of alleles must not exceed the marker ploidy and the peak must not be a child peak. Then, a ratio is computed by dividing its height with that of the last peak considered as an allele. If the focus peak is at the left of the allele and the ratio is ≥ 0.7, it is considered as another allele. If the subsequent peak is at the right, the ratio must exceed 0.3, or the ratio multiplied by the absolute difference in peak positions (in bp) must exceed 4. This second condition allows alleles that are much longer to yield peaks that are much smaller, accounting for short allele dominance. A peak not considered as an allele is flagged as a "supplementary" peak (see main text) if it is not a child peak and if its height is ≥20% of the height of last peak considered as an allele (or ≥12% of the allele height if the focus peak itself has child peaks).

The method also considers a special case where two alleles of a heterozygote differ by only one nucleotide in length. In this case, one allele may be wrongly considered as a peak resulting from adenylation and a homozygous genotype would be called. Only comparison with other genotypes may indicate whether the degree of adenylation of the marker is compatible with this assessment. Therefore, during alle call, the application records the ratio between the height of a given peak and the height of a peak inferred as an allele, if the peak is within 1.5 bp from the allele. This ratio is called l if the fragment represented by the peak is longer than the allele or s if the fragment is shorter. $L = \max(l)$ and $S = \max(s)$ are computed for a given genotype. In the absence of detectable adenylation, these maxima would be zero. The peak that has the highest ratio is considered as a "possible allele" of the genotype.

If, during the same allele call, at least two genotypes were considered as heterozygous and at least one as homozygous for a given diploid marker, the application computes arithmetic means $M(L)$ and $M(S)$ over all genotypes that are heterozygous. The application then inspects the possible allele of every homozygous genotype. If a possible allele fulfills ($l > 6 \times M(L)$ and $l \geq 0.5$) or ($s > 6 \times M(S)$ and $s \geq 0.5$), it is promoted as allele.

Overview of the database managed by STRyper

STRyper manages a database composed of objects of different classes that are represented in Figure S2.

An object of class *Marker* describes a microsatellite marker by specifying its *name*, and the *start* and *end* of the range of expected alleles expressed in base pairs (these attributes are inherited from a superclass called *Region*). *motiveLength* specifies the length of the repeat motive, *channel* the channel (wavelength) used to reveal amplicons (blue, green, black/yellow, or red) and *ploidy* is self-explanatory. The set called *bins* points to the bins defined for the marker. The *start* and *end* attributes of a Bin (inherited from the Region class) specify the expected range of an allele. A marker also points to the *Panel* (i.e., the multiplex) it belongs to, which reciprocally points to its markers via a set called *markers*.

An object of the *Chromatogram* class stores data imported from a .fsa or .hid file into various attributes, including the sample name (*sampleName* attribute), the *plate* name, the *well* identifier, the time of the end of sequencer run (*runStopTime*), the number of scans recorded (*nScans*), the indices of scans for which the camera was saturated (*offScaleScans*), etc. The *panel* relationship points to the panel of markers that were amplified to generate the chromatogram. Reciprocally, a panel points to all chromatograms that use it for genotyping, via a set called *samples*.

Via the *traces* set, a chromatogram points to four or five objects of class *Trace*, each of which encodes the raw fluorescence data (in the *rawData* attribute) measured at a *channel*. Peaks identified in each trace are stored in the *peaks* attribute, which is an array of structures composed of three integers: the scan at the start of the peak, the number of scans from the start to the tip, and the number of scans from the tip to the end of the peak. The *dyeName* attribute stores the name of dye that emitted the recorded fluorescence used (e.g., “6-FAM”, “LIZ”) and *isLadder* tells whether the trace represents the molecular ladder that was added to the sample before electrophoresis.

The size standard defining the molecular ladder is referred to by the chromatogram, via its *sizeStandard* relationship. A SizeStandard object has a *name* (e.g., “GeneScan 500”) and a set of *SizeStandardSize* objects (called *sizes*), each of which has a *size* attribute specifying a fragment size in base pairs. This set facilitates adding, removing or changing sizes. The *editable* attribute of a size standard tells whether its *sizes* can be edited by the user.

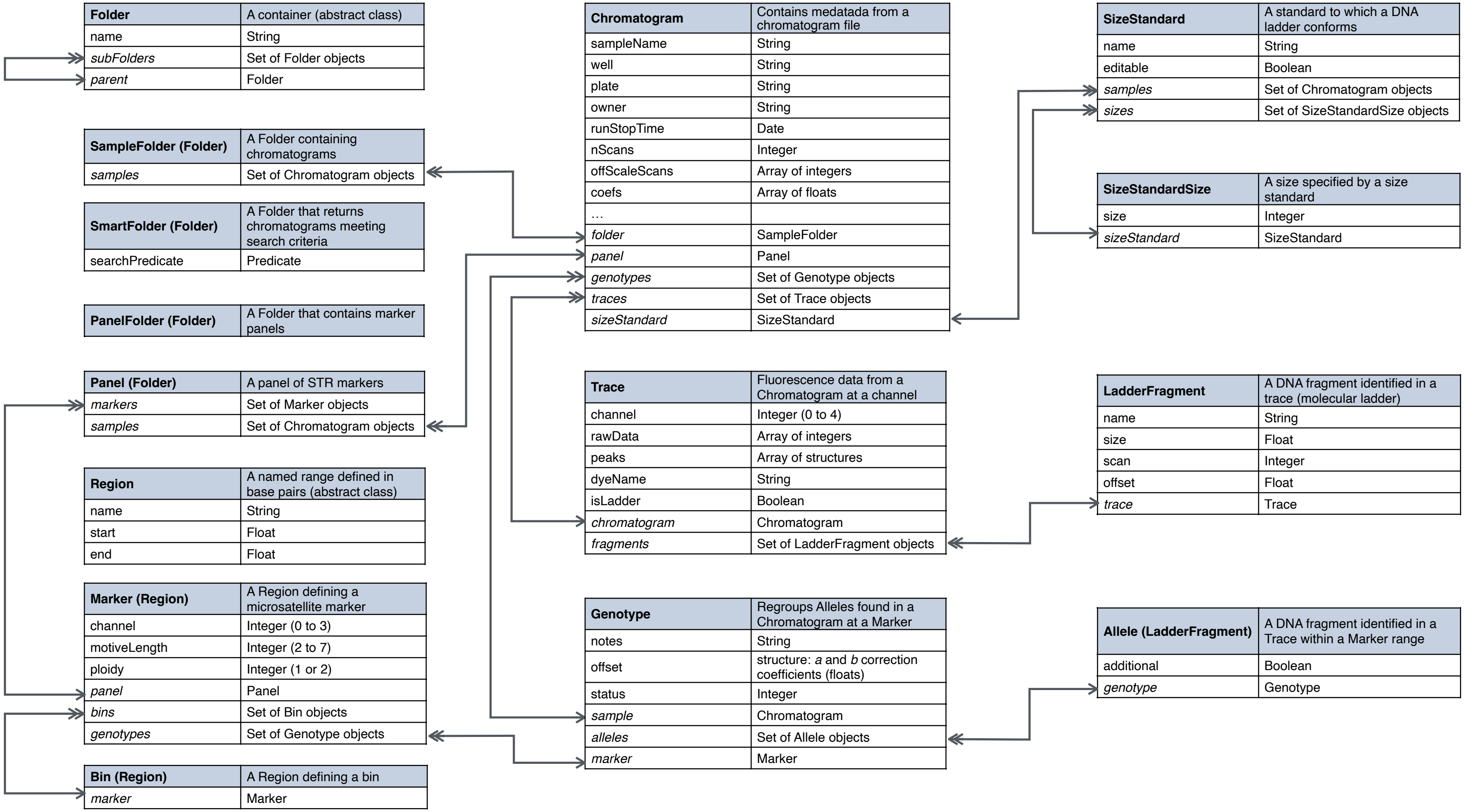
The *fragments* relationship of a trace points to *LadderFragment* objects. Such object defines a DNA fragment that produced a peak identified in the trace. Its *scan* attribute refers to the scan at the tip of the peak (hence the location of the fragment in the trace), and its *size* attribute is the size (in base pairs, taken from the size standard) that was attributed by the method described in section "Size assignment of molecular ladder fragments". The *name* of the fragment (shown to the user) is its *size* encoded as characters. The *offset* attribute is the difference (in base pairs) between the location of the *scan* (computed via the *coefs* attribute of the chromatogram, see below) and the *size*. This helps users detect size assignment errors. The coefficients of the polynomial that was fitted based on detected ladder fragments (see main text section "**Erreur ! Source du renvoi introuvable.**") are stored as an array of floating-point numbers in the *coefs* attribute of the chromatogram to which the trace belongs.

For a trace that is not a molecular ladder, the *fragments* set can only contain *Allele* objects. An Allele defines a DNA fragment that produced a peak in the range of a marker (from the panel applied to the trace's chromatogram) that has the same *channel* as the trace. The *size* attribute of an allele is computed from its *scan*, and its *name* is independent of its *size*. The *additional* attribute tells whether the allele represents an additional fragment (see main text section "**Erreur ! Source du renvoi introuvable.**"). An allele does not use the *offset* attribute that it inherits from the LadderFragment class.

A *Genotype* object regroups alleles that were found in an individual (chromatogram) at a marker. It therefore points to these objects using relationships called *alleles*, *sample* and *marker* respectively. A genotype stores the *a* and *b* parameters used to correct for allele sizes (see main text section "**Erreur ! Source du renvoi introuvable.**") in an attribute called *offset*. Indeed, these offset parameters are specific to an individual analyzed at a marker, hence of a genotype. The genotype *status* attribute tells the user whether the genotype has been called, whether alleles were found, the genotype has been manually edited, etc.

The Folder class lets users organize chromatograms and marker panels. A folder has a name and points to subfolders via a set called subfolders. Reciprocally, each subfolder points to its parent folder via its parent relationship. A nested hierarchy of folders can therefore be constructed. As a subclass of Folder, a Panel has a name and a parent, which is an object of the PanelFolder class. The subfolders of such object may be panels and/or other panel folders. As a Folder, a Panel can technically have subfolders, but the application code prevents this.

A SampleFolder is a Folder that can contains Chromatogram objects in a set called samples. A SmartFolder specifies a search predicate in its *searchPredicate* attribute. When it is accessed, a smart folder returns chromatogram objects found across the whole database via the search predicate. A smart folder has a parent that is the same for all smart folders, it but is not allowed to contain subfolders.



Folder	A container (abstract class)
name	String
subFolders	Set of Folder objects
parent	Folder

SampleFolder (Folder)	A Folder containing chromatograms
samples	Set of Chromatogram objects

SmartFolder (Folder)	A Folder that returns chromatograms meeting search criteria
searchPredicate	Predicate

PanelFolder (Folder)	A Folder that contains marker panels

Panel (Folder)	A panel of STR markers
markers	Set of Marker objects
samples	Set of Chromatogram objects

Region	A named range defined in base pairs (abstract class)
name	String
start	Float
end	Float

Marker (Region)	A Region defining a microsatellite marker
channel	Integer (0 to 3)
motiveLength	Integer (2 to 7)
ploidy	Integer (1 or 2)
panel	Panel
bins	Set of Bin objects
genotypes	Set of Genotype objects

Bin (Region)	A Region defining a bin
marker	Marker

Chromatogram	Contains medatada from a chromatogram file
sampleName	String
well	String
plate	String
owner	String
runStopTime	Date
nScans	Integer
offScaleScans	Array of integers
coefs	Array of floats
…	
folder	SampleFolder
panel	Panel
genotypes	Set of Genotype objects
traces	Set of Trace objects
sizeStandard	SizeStandard

Trace	Fluorescence data from a Chromatogram at a channel
channel	Integer (0 to 4)
rawData	Array of integers
peaks	Array of structures
dyeName	String
isLadder	Boolean
chromatogram	Chromatogram
fragments	Set of LadderFragment objects

Genotype	Regroups Alleles found in a Chromatogram at a Marker
notes	String
offset	structure: *a* and *b* correction coefficients (floats)
status	Integer
sample	Chromatogram
alleles	Set of Allele objects
marker	Marker

SizeStandard	A standard to which a DNA ladder conforms
name	String
editable	Boolean
samples	Set of Chromatogram objects
sizes	Set of SizeStandardSize objects

SizeStandardSize	A size specified by a size standard
size	Integer
sizeStandard	SizeStandard

LadderFragment	A DNA fragment identified in a trace (molecular ladder)
name	String
size	Float
scan	Integer
offset	Float
trace	Trace

Allele (LadderFragment)	A DNA fragment identified in a Trace within a Marker range
additional	Boolean
genotype	Genotype

Figure S2. Overview of the database managed by STRyper. Each table describes a class defined in the application. The top left cell shows the class name followed by the name of its superclass in parentheses, if relevant. The top-right cell shows a brief description of the class. Below the table header, the left column lists the names of attributes and relationships of the class. Relationship names are italicized. The right column specifies the type of each attribute/relationship. A class inherits attributes and relationships from its superclass, but it may not use them in the application code. All relationships are reciprocal, and reciprocity is represented by arrows. Single arrows point to to-one relationships (pointers to a single object) and double arrows point to to-many relationships (sets of pointers to several objects). Only attributes and relationships saved in the database are shown. For a complete definition of these classes, see header files at https://github.com/jeanlain/STRyper/tree/main/STRyper/Entities/