FragSTR: OPEN-SOURCE STR MARKER ANALYSIS

BY

Prateek Singh

SUBMITTED

in partial fulfillment of the requirement of the Degree of

MASTER OF TECHNOLOGY

to

Indraprastha Institute of Information Technology, Delhi

Under the guidance of

Dr. Debasis Dash

INDRAPRASTHA INSTITUTE of INFORMATION TECHNOLOGY DELHI
(A State university established by Government of NCT of Delhi)

M.Tech. Thesis evaluation /defense form

Name of the student: _____ Roll No: _____

M.Tech. Specialization: _____

Date of Thesis Submission: _____ Date of Thesis Defense: _____

Thesis Title:

Thesis Evaluation Committee

1. Supervisor _____
2. Internal Examiner _____
3. Internal/External Examiner _____

Recommendation of the Committee:

☐ **Accepted** ☐ **Rejected** ☐ **Accepted with major modifications (Specific suggestions or changes needed)**

Whether the Thesis is recommended for best M.Tech Thesis Award?

☐ Yes

☐ No

If yes, write a few lines in support of your recommendation. _____

_____

_____ _____ _____

Supervisor **Internal Examiner** **External/Internal Examiner**

_____

Approval of the PGC Chair

Date: _____

PGC Chair

Dedicated to,

my beloved parents,

Thanks for always supporting me

Acknowledgement

The first and foremost gratitude goes to my guide and mentor, **Dr. Debasis Dash**, Principal Scientist, CSIR-IGIB, New Delhi. His enthusiasm, motivation and patience inspired my interest in research and guided me in the right direction. I learned a lot in our discussions. He is dedicated to providing the students with a comfortable and pleasant working environment. I feel very fortunate and honored for having the opportunity to work with him. I could not have imagined a better mentor.

I sincerely wish to thank my faculty **Dr. Subhadip Raychaudhary**, **Dr. K. Sriram**, **Dr. Ganesh Bangler**, **Dr. Debajyoti Bera, Dr. Rohit Gupta, Dr. G. P. S. Raghava, Dr. Arnab Bhattacharjee** and **Dr. Dhirendra Kumar Gupta** for providing their valuable guidance throughout the course of my master's programme.

I would like to thank my group members, **Mr. Abhilash Gangadharan** and **Mr. Supratim Chaudhuri** for their valuable suggestions and contributions to the project.

Finally, I would like to thank all my friends and family who, while not directly contributing to the thesis, have provided invaluable support in helping me overcome the more difficult parts of my two years at IIIT Delhi and without whom I would not be where I am today.

Prateek Singh

(Author)

INDEX

List of Figures

List of Tables

Introduction

Genetic diseases are considered rare, with about 7 crore Indians suffering from some form of a rare genetic disease. Typically, they go through 8 physician visits, going through an average of 3 misdiagnoses, with a total of 7 years elapsing before being correctly diagnosed [1]. Since the diagnosis of genetic diseases is technically challenging, emerging research area, most doctors are unaware of the diagnostic procedures in contrast to classical diseases.

If the patient decides to refer to a geneticist, a sequencing or an analysis for polymorphic markers/STR markers may be done for genetic diseases [2,3]. Whole genome sequencing while continually decreasing in cost, is still too expensive for many people. STRs or Short Tandem Repeats are known to be associated with many genetic diseases. So, the STR marker fragment analysis is an equally accurate, cheap and faster method for detecting genetic diseases [4].

STR or microsatellites are short repetitive regions in the genomic DNA with a repeat length of 2 to 13 base pairs. They are highly widespread throughout the genome of the organism. The variation in the number of repeats of a repeat unit can be associated with different forms of alleles and brings variation in the population [5]. By matching specific STR markers between two DNA samples, it is possible to distinguish between them. This property is used for DNA fingerprint, lineage analysis, cell line authentication etc. The combined effect of the variations for various molecular markers makes it very unlikely for two individuals to have similar genomes. For example, the probability of two persons having the same genomic information according to the FBI's CODIS software is one in a billion. STR analysis is very sensitive and accurate, thus it has a high discriminative power. Therefore, it makes it a suitable candidate for determining the genotype of a DNA sample [6].

STR fragment analysis is protected by copyrights and proprietary software limiting its use to labs or research institutions, where they have capillary electrophoretic fragment analysis machines and additional software supplied with a single user license.

Recent advancements in technology have allowed analysis of multiple samples in an automated fashion in capillary electrophoresis machines. Thus, it is possible to do high throughput analysis of fragments using current machinery. Therefore, it increases the accessibility and further decreases the cost of the method.

The process for fragment analysis is simple. Capillary electrophoresis DNA sequencing uses fluorescent dyes which bind to the molecular markers present in the DNA and amplify the samples using PCR i.e. Polymerase Chain Reaction. The amplified fragments differ in their length depending on the number of

STR units present in the markers. They when moving through the strong electric field in capillary electrophoresis tube get separated from each other. The shorter fragments migrate farther as compared to longer fragments. The fluorescent dyes in the primers are excited using a laser source. This makes the dyes fluoresce at well-defined wavelengths. A photometer then records the wavelength for each of the DNA fragments. As the electrophoresis is in progress, the output of the photometer provides an intensity versus elution time data which we call spectra [7]. Each sample is multiplexed with multiple dyes to enable differentiation between similarly sized markers. One of these dyes is fixed for use with the size ladder that contains DNA fragments of known sizes. These size ladders are used for estimating the sizes of the rest of the fragments. Each set of samples is run alongside an Allelic ladder containing all possible alleles for each marker present to calibrate for electrophoretic migration anomalies.

The objective of the thesis is to develop a non-technical interface to analyze fragment data avoiding the use of proprietary software that requires technical training. I and my team at CSIR-IGIB aim to build a web-enabled software for fragment analysis. Due to its web-based nature, its accessibility is far more as compared to commercial software, which work only on the computers connected to the sequencing machine. Using this software, anyone can analyze fragments on his own convenience on his own device with the aid of the internet.

Currently, the commercially available software for STR analysis are Peak Scanner 2 (no longer available at the time of writing) and GeneMapper 5. The software that I and my collaborators at IGIB made is known as FragSTR, found at the address www.fragstr.co.in.

FragSTR is a web-enabled software, completely built in R, that can analyze fragments to call alleles for the molecular markers present in them. The inputs required by the analysis are the ABI FSA files, the Size Standard, the Panel file and the Bin file. The ABI FSA contains multiple tracks out of which one is the size ladder. The size standard contains the sizes for the peaks (local maxima of photo intensity) available in the size ladder for a sample. The Panel and Bin files come as a pair and contain all the information about the molecular markers and the alleles that can be obtained from the data. Further details about the input files will be provided in the next sections.

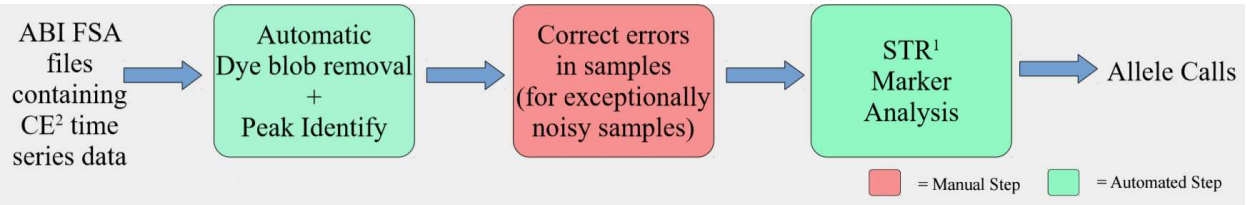


Figure 1 A concise flowchart for the STR analysis system FragSTR.

The red boxes are manual steps in the workflow and the green boxes are the automated steps.

In its essence, the analysis creates a regression model using the size ladder from each sample and the size standard and then sizes the peaks obtained by peak finding to obtain sizes for all the peaks. Then the analysis performs a lookup on the Panel and the Bin files to identify molecular marker alleles nearest to the particular to call alleles. Finally, confidence scores are generated to report deviation of the peaks from the default allele positions.

The output of the analysis is a text file containing all the alleles recognized and a plot displaying all the alleles that are called with their intensity and confidence scores in an at a glance view. An example of the allele call plot is given in the figure below.

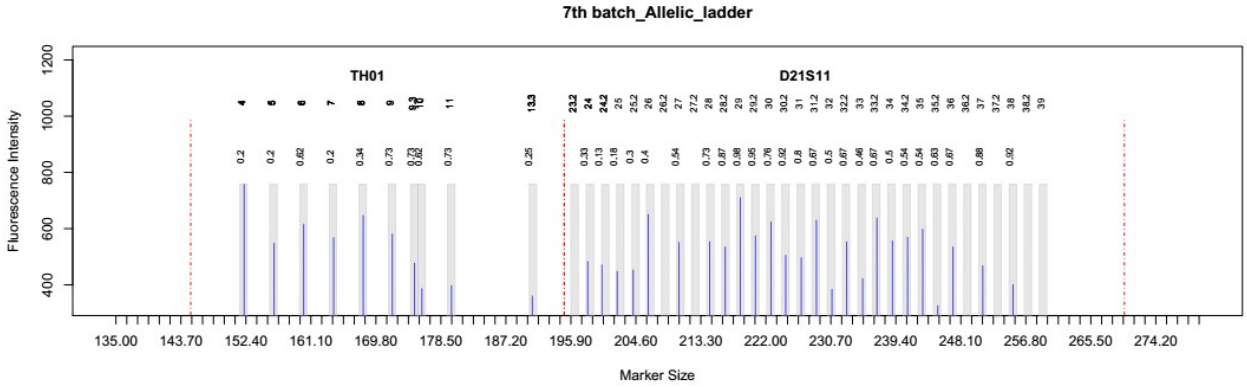


Figure 2 An example of the generated allele call plot.

Each bar represents an allele. Its length represent the photo intnsity of the allele. The number on top of the bar is the confidence score for the allele call. Above the confidence score is text representing the allele number. The text grouping many alleles is the molecular marker those alleles belong to.

Its results are comparatively similar or even better in some cases to GeneMapper 5 results over a wide variety of samples that were tested. Hence, it can be said that FragSTR can be a promising alternative to the GeneMapper 5 and other proprietary software.

2. Inputs required for the analysis

The analysis requires four different types of files which are given as under.

1) **FSA Files** are binary files that contain the capillary electrophoresis elution time series data in the form of tracks. They are named according to the samples they contain information for. The tracks generally contain a few data tracks and a size ladder.

2) **The Panel file** is a text file that contains a space separated table with marker type, the color of the track it is found on and the range on the track where the marker can be found. The analysis later uses this range to decide the range of interest for all data tracks except the size ladder. They can also manually be created in the software itself.

3) **The Bin file** is a text file that contains space separated tables for each molecular marker type containing the alleles and their positions in each marker. The analysis uses this information to assign alleles to the various peak detected in tracks of the FSA files.

4) **The size standard** is an XML file that contains sizes of the peaks that are identified in the Size Ladder in each sample FSA file.

A simple table explaining their need in the analysis is given as under.

Table 1 Inputs required by analysis, their file format, purpose and methods to read.

Input Type	File Format	Contents/Purpose	Method to read
ABI FSA Files	FSA/Binary	Serves as input data for the analysis. Has one Size Ladder and other data tracks	read.abif() function from "seqinr" package in R
Size Standard file	XML/Text	Contains the sizes for the peaks detected in the size ladder of the FSA file	Using the "XML" package in R
Panel file	TXT/Text	Contains molecular marker names and the ranges in which their alleles are found	readBins() function of "seqinr" package in R
Bin file	TXT/Text	Contains positions of various alleles of the molecular markers	readPanels() function of "seqinr" package in R

2.2 FSA Files

What are FSA files?

FSA files, or in the long form ABI FSA files, are files generated by capillary electrophoresis machines built by ABI. ABI here stands for Applied Biosystems, the Life Sciences brand of Thermo Fisher Scientific corporation, that builds sequencing and capillary electrophoresis machines. FSA files are proprietary files, with steps to decode according to the specifications given on the following link(http://www6.appliedbiosystems.com/support/software_community/ABIF_File_Format.pdf Page 7- 19).

How to read it?

FSA files are binary, so you must need to know the way the data is stored in the file to get data out of the file. It is not human readable in its native binary format. But once we know the way the data is stored in the file, it is as simple as getting the required byte position from the file and converting it into the specification mentioned type.

For R statistical programming language, a package named “seqinr” has been developed for exploratory data analysis and visualization of biological data [8]. It enables for easy retrieval of information from biological data, for example FASTA files and ABI FSA files. The analysis uses this method to read the files stored in the FSA files.

What does the file contain?

The FSA file, once read properly, is divided into three fields i.e. the Header field containing format related information about the file, the Directory field containing the offset to the positions of data found in the Data field and the Data field containing the capillary electrophoresis elution time and wavelength data.

How does the analysis use it?

The data field is what we require for the analysis. The capillary electrophoresis elution time series data and the wavelength data is extracted from the data field.

The capillary electrophoresis elution time series data is used to identify the various fragments containing peaks which are later converted to allele calls. This data contains one size ladder containing data from a fragment we know the sizes for and other data tracks we don’t know the sizes for. The data from the size ladder is later used to assign sizes to the rest of the tracks.

The wavelength extracted is used to assign dye colors to each of the tracks including the size ladder. This dye color is later used to select range of interest with respect to the panel file.

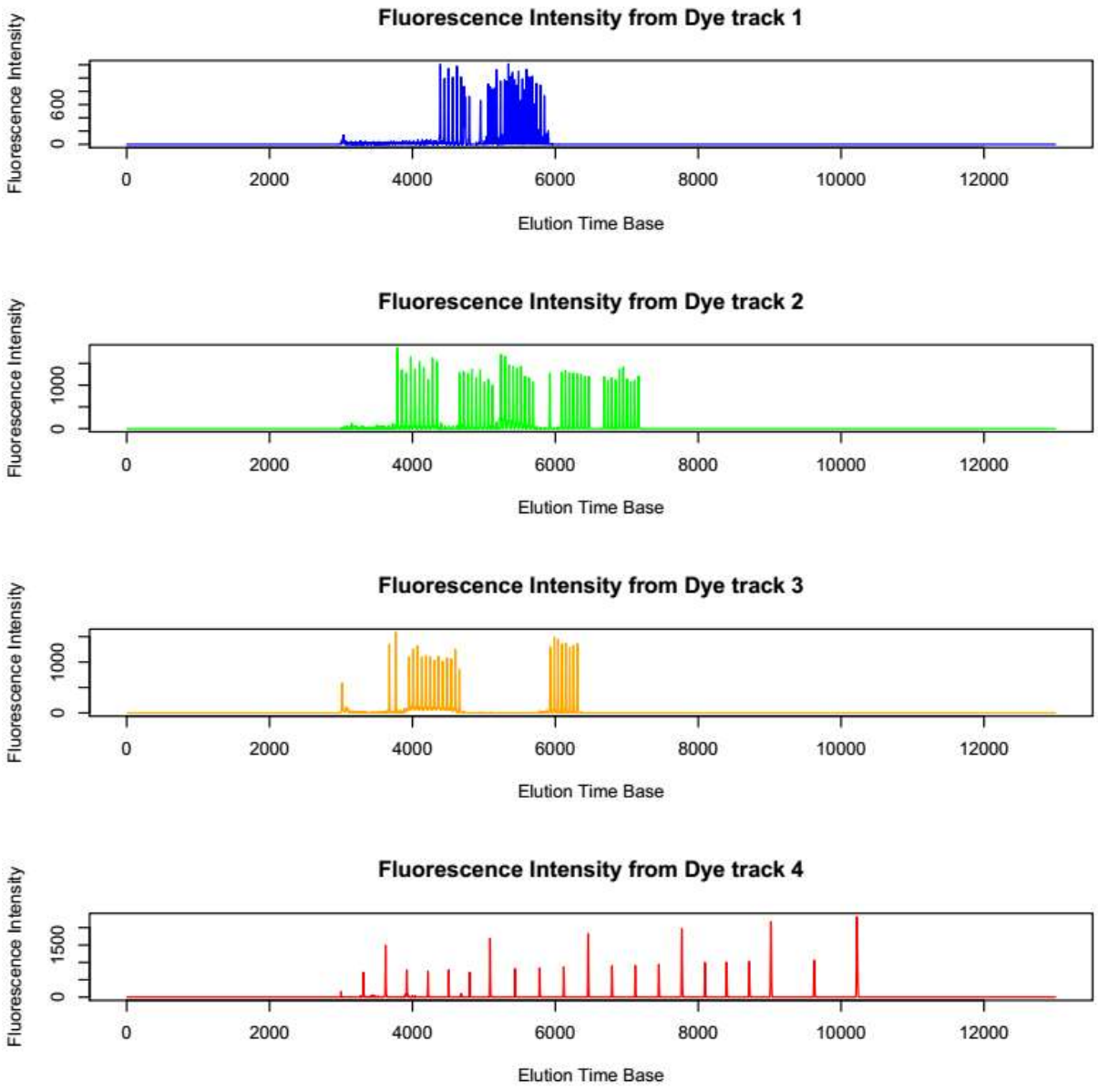


Figure 3 Preview of contents of FSA files

The FSA files contains capillary electrophoresis elution time series data stored as different tracks. The data can contain from 2 to 199 tracks according to the format standard. In the above figure, the top three plots show the data tracks, containing unknown fragments, and the last plot is the size ladder.

2.3 Size Standard

What is the size standard?

The size standard is an XML file that contains sizes for the size ladder.

How is the Size Standard read?

To read the Size Standard, an XML parser is used and content from every occurrence of the node named "sizeStdDefinition" is extracted. This content can be stored in some data structure on any programming language.

In R programming language, "XML" package can be used to parse the Size Standard.

How does the analysis use it?

The size ladder is run with the other fragments to get a file containing the capillary electrophoresis elution time series data. The data gives out peaks equal in number to the sizes collected from the size standard. These peaks are directly matched to each of the sizes found in the size standard. Interpolating between these sizes can give us the size for a peak at any other position.

```
<?xml version="1.0"?>

<SizeStandardContainer>

<xmlSizeStandard>

<sizeStdDefinition>60.0</sizeStdDefinition>

<sizeStdDefinition>80.0</sizeStdDefinition>

<sizeStdDefinition>100.0</sizeStdDefinition>

<sizeStdDefinition>120.0</sizeStdDefinition>

</xmlSizeStandard>

</SizeStandardContainer>
```

2.4 Bin File

What is the Bin file?

A panel file is a text file that contains information about the molecular markers present in the dye such as the dye color on which they are found and the range of sizes within which they are found.

How are they read?

Since it is a text file with tab separated values, it can be easily read by the read table function provided in most new programming languages.

In R programming language, using readLines(<filename>, sep=" ") will allow the file to be read. Similar methods can be found in other programming language. There is also a function named "readBins()" available in "seqinr" package that enables reading the bin files[8]

What does it contain?

The bin file contains a table for each marker. Each table contains the following four columns, 1) Allele name/ type (allele.name), 2) Size of the allele (size.bp), 3) negative relaxation to the size (minus.bp), 4) positive relaxation to the size (plus.bp). A preview of the contents of the file is given in Table 2.

Table 2: Contents of the Bin file

```
$GenePrint_10_v1.1$sTA
   allele.name size.bp minus.bp plus.bp
1         10.0  122.17      0.5     0.5
2         11.0  126.19      0.5     0.5
3         12.0  130.16      0.5     0.5
4         13.0  134.25      0.5     0.5
5         14.0  138.26      0.5     0.5
$GenePrint_10_v1.1$TPX
   allele.name size.bp minus.bp plus.bp
1            5  257.00      0.5     0.5
2            6  261.04      0.5     0.5
3            7  265.09      0.5     0.5
4            8  269.03      0.5     0.5
5            9  273.05      0.5     0.5
```

How does the analysis use the Bin file?

The Bin file provides the positions of the alleles according to the size calls.

2.5 Panel file

What is a panel file?

A panel file is a text file that contains information about molecular markers, the dye color on which they can be found on and the range of sizes within which they are found.

How are they read?

The panel file is a text file with space separated values, it can be easily read by the read function provided in popular programming languages.

In R programming language, using readLines(<filename>,sep=" ") will allow the file to be read. Similar methods can be found in other programming language. There is also a function named "readPanels()" available in "seqinr" package that enables reading the panels files.[8]

What does it contain?

The bin file contains a table containing the following eight columns out of which only first four columns are used. The columns that are used are 1) Marker name, 2) color of the dye it is found on, 3) minimum size at which it is found, 4) maximum size at which it is found. The other columns have still unknown purposes. A preview of the contents of the file is given in Table 3.

Table 3: Contents of the Panel file

```
$GCG_Panel
    marker dye.col min.bp max.bp exp.pcg repeat.bp stutter.pc uknw
1     AA01    blue 145.01    195   6,9.3         4       0.06 none
7   DSA5D3   green 310.01    361      12         4       0.10 none
8     AMEL  yellow  92.00    112     X,Y         9       0.00 none
9      STA  yellow 112.01    222   16,19         4       0.14 none
10    THYR  yellow 222.01    305      11         4       0.06 none
```

How does the analysis use it?

The bin file is used to pick out loci of interest from the various fragments. This enables accurate finding of the regions of interest. These regions are used for finding marker alleles. It also helps with the removal of the dye blob in all the dye tracks except the size ladder.

3. Methods

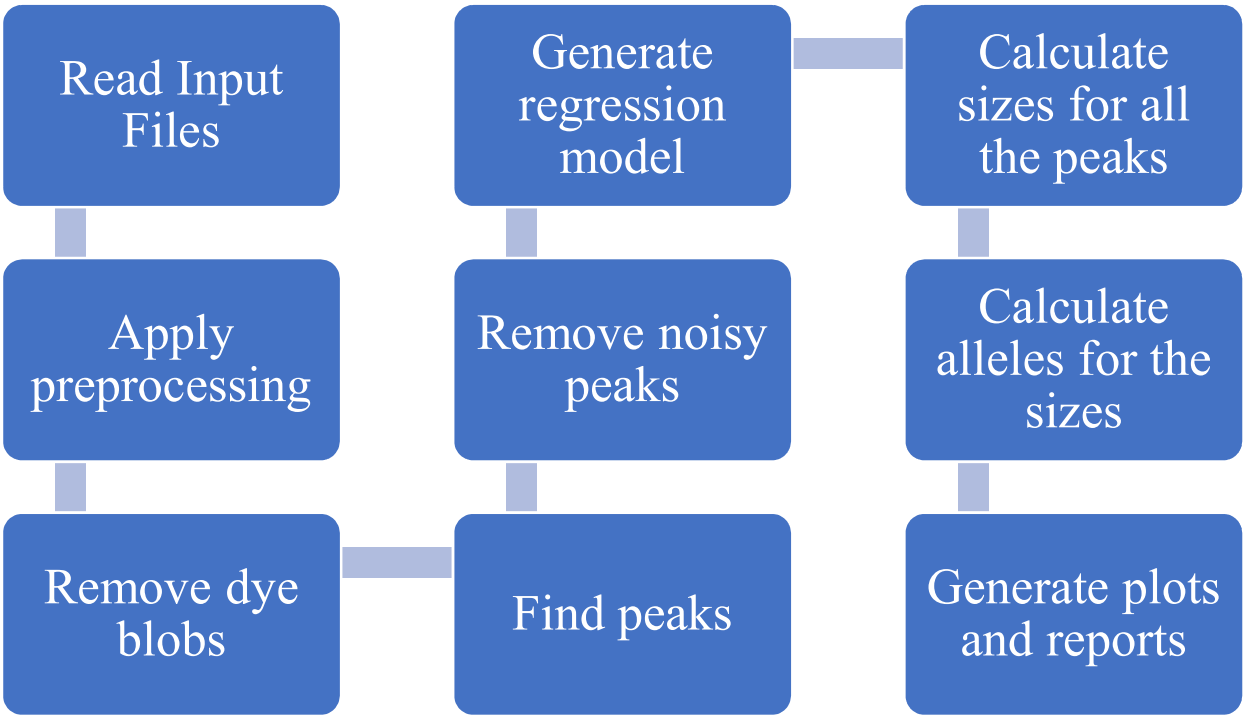


Figure 4 Workflow of the analysis

The FragSTR analysis consists of the following distinct steps that make up the entire analysis.

1) FSA files are read. Spectrum is extracted and is processed.
2) Peaks are identified. Peak cleaning for noisy size ladders is performed.
3) A regression model is generated.
4) The peak positions are converted into sizes.
5) The alleles corresponding to sizes are identified and confidence score is calculated.
6) A textual and a graphical report of the allele calls is generated.

3.1 Reading the FSA files

The FSA files are read using the "read.abif()" function from the package "seqinr". Before using the function, the file uploaded is checked for adherence to the ".fsa" file format and specifications. There are two checks to ensure that. The first check is, verifying the file extension. The extension is checked using "file_ext()" function from the "tools" package natively available in R. The second check is, converting the first 4 bytes of the file to character to see if they form the "ABIF" string. If the file passes these two checks, it is deemed as an FSA file, and can be read.

The data that we need to fetch are as given below.

1) **Capillary electrophoresis elution time series data**

   The capillary electrophoresis elution time series data is contained inside the Data compartment of the FSA file. The data element is also a named list. The elements inside Data element named DATA.1 to DATA.4 and DATA.105 to DATA.199 can contain the capillary electrophoresis elution time series data for each of the tracks of the sample. The number of these elements to query is decided according to the number of tracks present.

2) **Wavelength data for each track (converted later to colors)**

   The wavelength data is contained inside the Data compartment of the FSA file. The elements inside Data element named DYEW.1 to DYEW.4 and DYEW.105 to DYE.199 can contain the wavelength for each of the tracks of the sample. Similar to the above time series data, the number of these elements to query is decided according to the number of tracks present.

3) **Number of tracks of data present in the sample**

   The number of tracks present in the sample can be obtained from the element named Dye# inside the Data element.

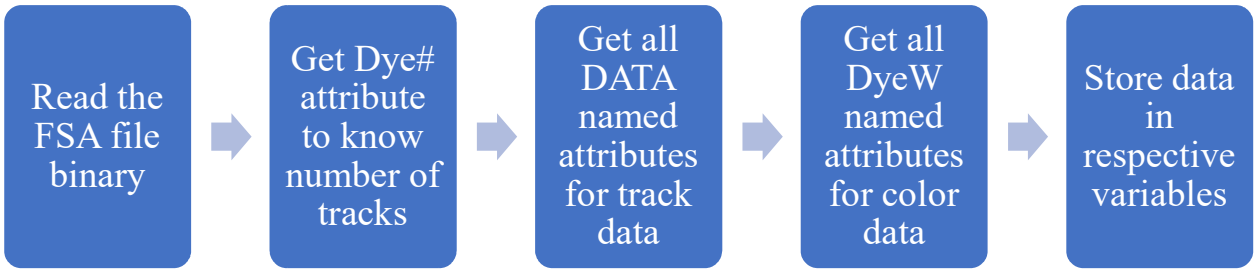


Figure 5 Fact sheet for FSA file reading method

3.2 Explaining Sample Names and Identifiers

All sample files contain two types of tracks. These are of the following two types.

1) Size ladder tracks – These are fragments that are run with known sizes. Associated to the sizes contained in the Size Standard, the size ladder tracks are used to create regression models that calculated sizes for other tracks.
2) Other tracks – These are tracks for which sizes are unknown. Regression models generated from size ladders is used to size the peaks that are found in these tracks

The size ladders are treated as one of input for the analysis and other remaining tracks in the sample constitute the output.

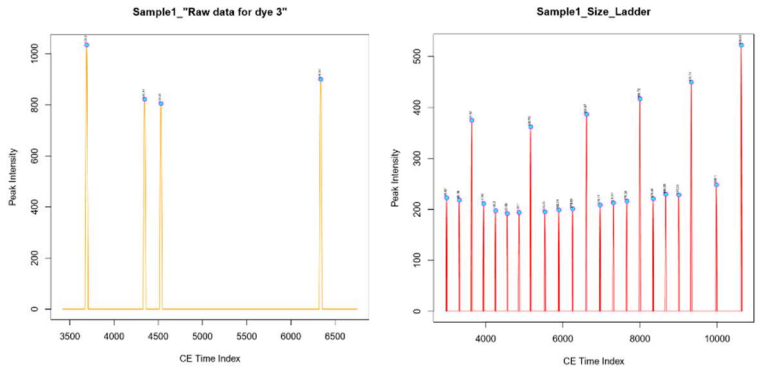


Figure 6 Comparison between a Normal Track (left) and the Size Ladder Track(Right)

The sample files themselves can be classified into two categories. The two categories are as follows.

1) Allelic ladders – These are samples regarded as positive control that contain all the alleles that can be detected using a pair of bin and panel files. These also serve as a metric to infer the correctness of an analysis.
2) Other sample – These are samples containing the usual test data

The allelic ladder samples are also useful when calculating confidence scores for the allele calls generated by the analysis.

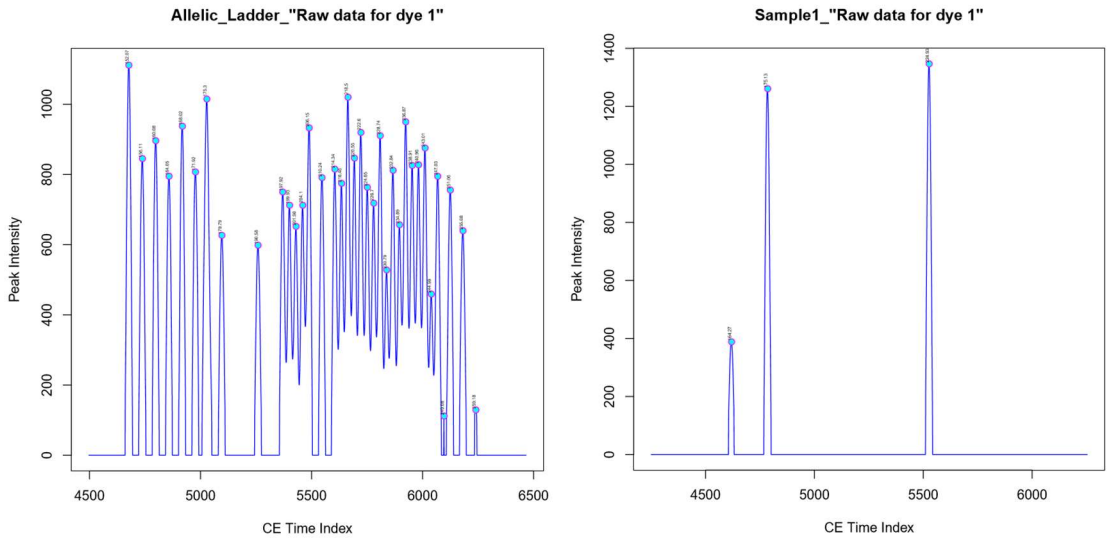


Figure 7 Comparing Allelic ladder(left) and normal sample(right)

3.3 Manual methods

There are two processes that need user input. The two processes are as follows.

1) **Blob removal** – The blob is an artifact created due to accumulation of the dye. It is present in most samples. It poses problems when performing the analysis. So, it must be removed. To remove it, the user needs to manually decide the end of the blob.
2) **Extra peak removal** – Sometimes, there are more peaks in the size ladder when compared to the sizes specified in the size standard. In such a case, the number of peaks need to be made equal to the number of sizes in the Size Standard so that direct peak to size assignment can be done.

Currently, the blob removal and extra peak removal processes have been automated, but it is necessary to include them to demonstrate the evolution of the software over time. Even though they have been automated, the analysis still contains an option to choose to go through the manual route.

3.3.1 Blob Removal (Manual)

The blob is an artifact created by accumulation of dyes while running the samples in the capillary electrophoresis machines. It needs to be removed from the samples. There are a few steps that need to be performed by the user to remove the blob. Those steps are follows.

1) Once the sample starts loading, the screen will be as given in the below picture.

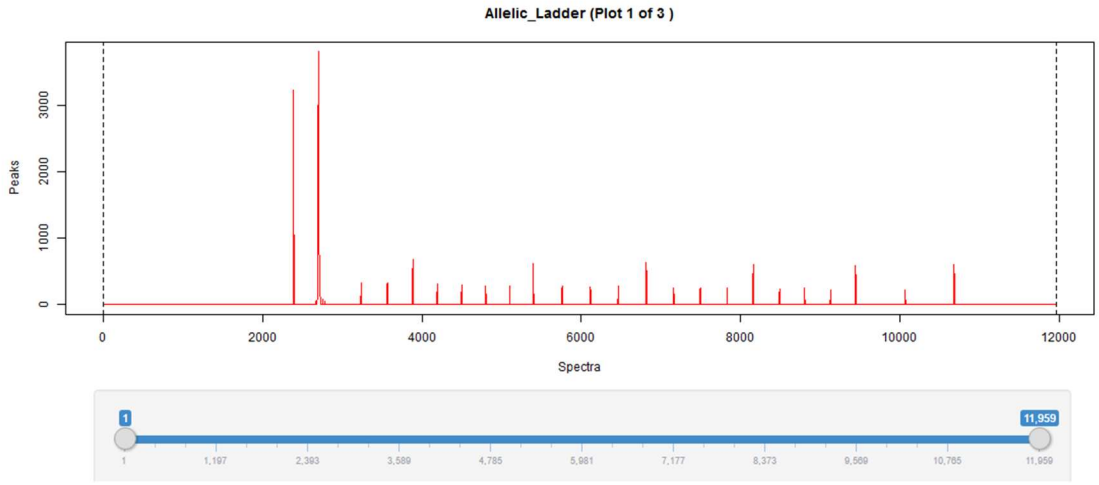


Figure 8 Screen displayed for blob removal

2) Move the slider to just after the end of the blob as shown below.

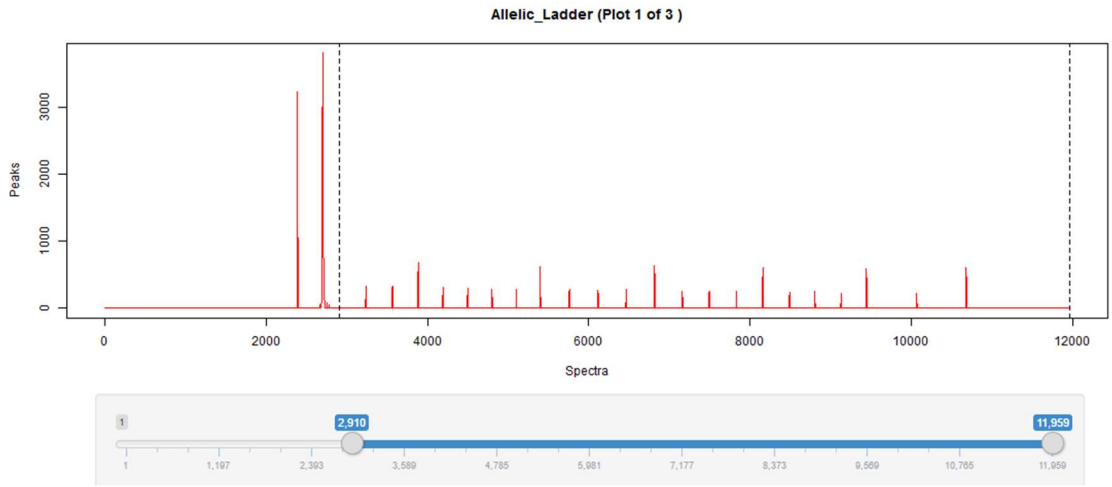


Figure 9 Range slider has been shifed to after the blob. It will be removed after the process completes

3) The UI will automatically remove all points in the sample before this position.

3.3.2 Noisy Peak Removal(Manual)

Due to interference or noise while running the size ladder, there are more peaks detected than the sizes specified in the size standard. These extra peaks need to be removed before continuing as the peaks need to have a one-to-one map to the size standard. The process for peak removal given below.

1) Once the page loads, the following prompt will show up on the screen.

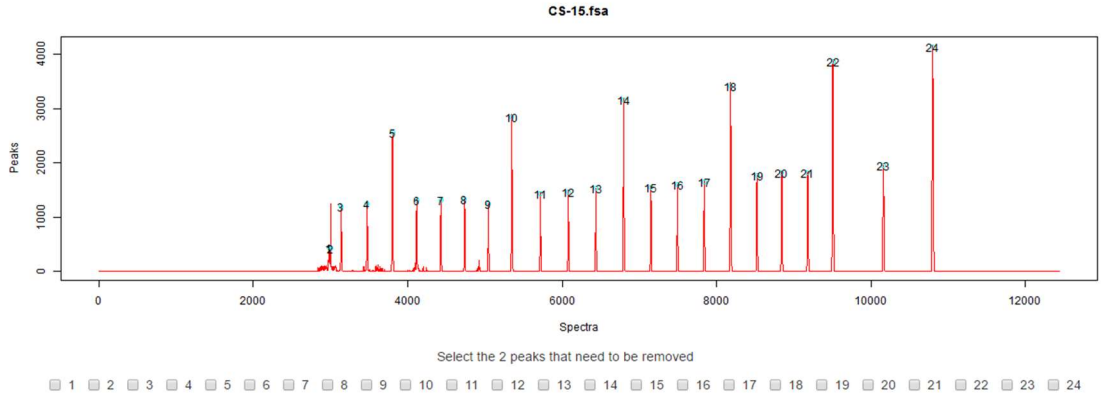


Figure 10 Screen displayed for removal of extra peaks

2) The user then needs to select peaks that she needs to remove.

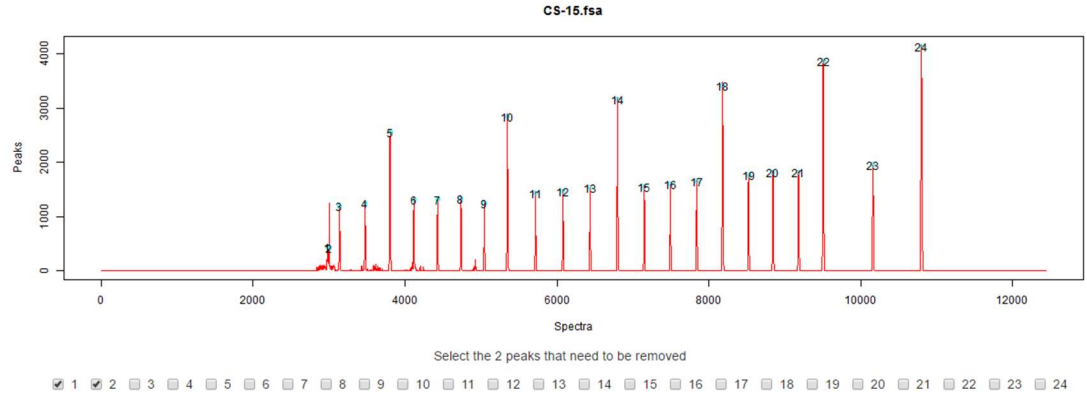


Figure 11 Two peaks are selected for removal.

3) On continuing using the Next Plot button, the process will remove the selected peaks from the size ladders

3.4 Automated Methods

Except the manual steps, all other steps of the analysis are fully automatic. These steps include the following.

1) **Data preprocessing**

   Negative values from the data are removed and the data is smoothed to remove any stutter peaks.

2) **Selecting required loci from tracks**

   Loci of interest is selected for allele calling according to the panel files. Rest of the area is removed.

3) **Peak detection**

   Finding peaks in the data tracks provided. These peaks are transformed into sizes and allele calls.

4) **Building a regression model**

   Building a regression model using the size standard and the peaks detected. This model is later used to size tracks other than the size ladder.

5) **Calculating sizes for peaks**

   Using the regression model, sizes for each peak detected is calculated. These sizes are later used to call alleles according to the bin file.

6) **Allele calling**

   The alleles are called as specified in the bin file. These called alleles are the result of the analysis.

7) **Automated blob removal**

   An automated alternative to the manual method for removing dye blob.

8) **Automated peak choosing**

   An automated alternative to the manual method for removing extra peaks or choosing the correct peaks.

3.4.1 Data Preprocessing

Before the data is fed into the pipeline, it first needs to be modified to suit the needs of the algorithms. This improves the quality of the results. These preprocessing steps are as follows.

1) **Negative removal**

   As intensity values are always positive, the negative values need to be removed from the data as shown in the data. Figure 12 show the effects of the negative removal on the data. The red line signifies zero photo intensity. The left plot shows the data with positive as well as negative values and the right plot shows the data with only positive values.

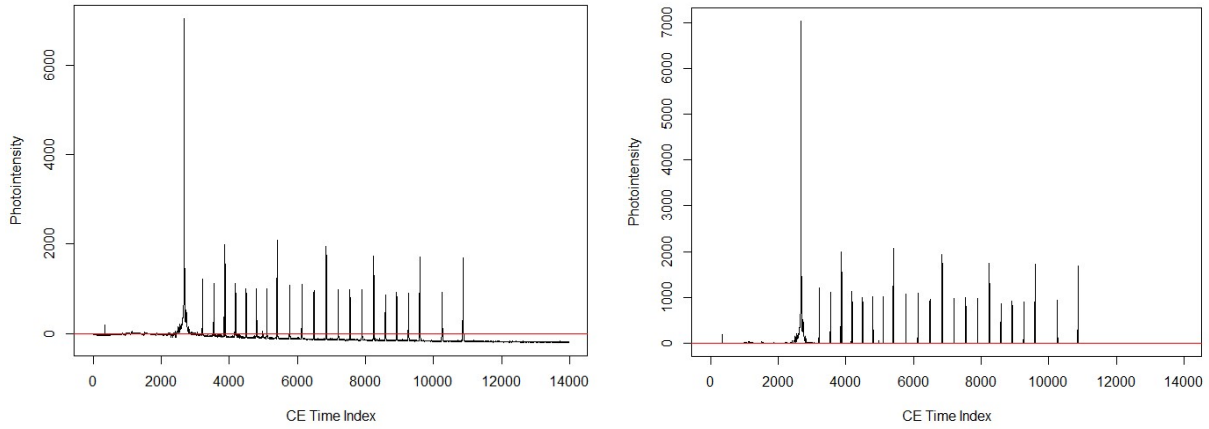


Figure 12 Example of Negative Removal.

Left is the data without negative removal. Right side is the data after negative removal

2) **Data smoothing**

   Smoothing is required to remove the problem of duplicate allele calls cause due to stutter peaks. Stutter peaks are small peaks that are part of large peaks but still show up as independent peaks in peak detection. These small peaks lead to duplicate allele calls. Smoothing eliminates these stutter peaks, hence duplicate allele calls. The Figure 13 shows the effects of smoothing on a peak. The left plot shows the peak with stutter peak and the right plot shows the same peak after smoothing.

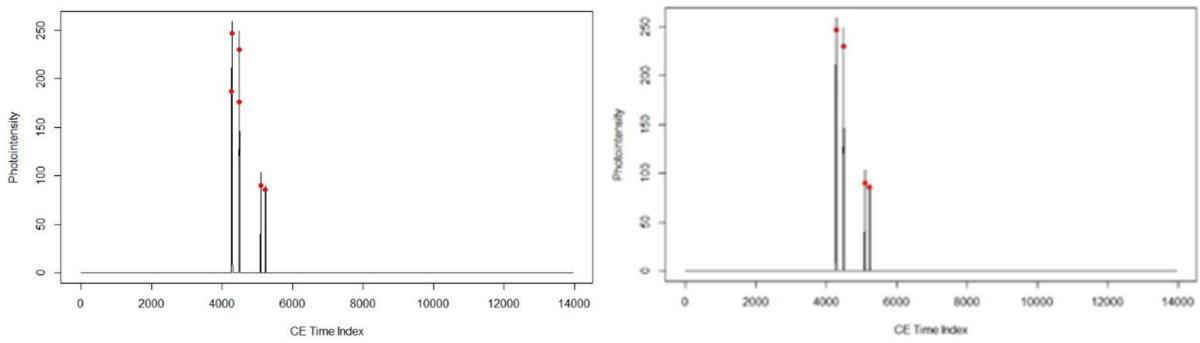


Figure 13 Example of Stutter removal after smoothing.

Left side is non-smoothed showing stutter peaks. Right side is smoothed; hence the stutter peaks are not present

3.4.2 Extracting area of interest

This method picks out the loci required for allele calling from the tracks. In the process, it also removes dye blobs from the data. The function works in five distinct steps.

Step 1- Build a regression model using the size ladder of the selected sample and the size standard. This regression model is used to convert peak position in data to peak sizes

Step 2 – Select a track from the sample and acquires its color attribute. The color attribute is needed to query the markers on the panel file.

Step 3– Searches for all markers available for the particular color and records their size ranges. The data inside these size ranges is the one that is required.

Step 4– Convert the sizes into data positions to acquire ranges of interest.

Step 5– Remove all data outside the ranges. The remaining data will be used to call alleles.

A flowchart explaining the functioning of the method is as given in Figure 14.

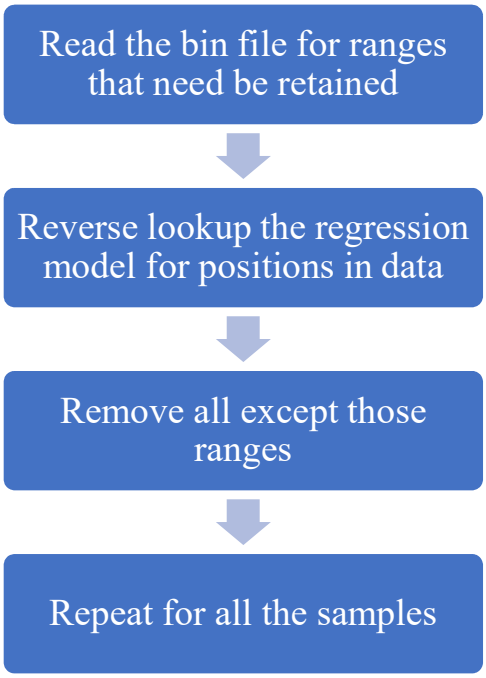


Figure 14 Flowchart for Extracting region of interest

3.4.3 Peak detection

Peak detection is a very important step in the analysis. Correct detection of peaks is essential to the accuracy of the analysis.

The initial method to peak detection used a function called "Spectrum Search" in the "Peaks" package in R programming language [9]. It used Markov chain modelling to determine all the local maxima which it labelled as peaks.

This method had two problems associated with it.

1) **Stutter peaks –** Stutter peaks are peaks that are too close to a parent peak to be claimed as an independent peak. This causes duplication of allele calls in the analysis. They have been taken care of by smoothing the data before detecting peaks.
2) **Missed peaks –** There were also cases where peaks that were important but comparatively small in photo intensity were missed by the Peak Finder. The solution to it was building an iterative peak finder which is described as below.

Iterative peak finder

Iterative peak finder allows the system to find peaks that would have normally been missed due to being too small in photo intensity. It thus allows the system to have an accuracy competitive to other proprietary software. Iterative peak finder works through the following steps:

Step 1 – Scan the data for peaks using "Spectrum search" function mentioned before. It provides a list of all the peaks detected by it.

Step 2 – Find the peak with the highest photo intensity. High intensity peaks are sure to be detected.

Step 3 – Trace the start and end of the peak to remove it. Removing the peak with the highest intensity lowers the relative difference in photo intensity between the remaining peaks, hence allowing them to be detected by the original peak finder.

Step 4 – If there are peaks left in the fragment data, store the current highest peak in a variable and go to Step 1

A flowchart showcasing the functionality of "Iterative Peak Finder" is given in Figure 15.

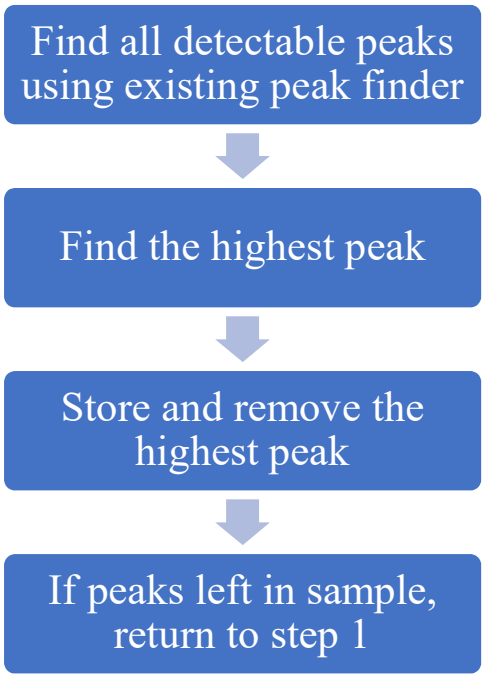


Figure 15 Flowchart for Iterative Peak Finder

3.4.4 Building a regression model

A regression model is also a very important step in the analysis. Its correctness improves the accuracy of the analysis. A regression model provides a function to convert the data points of peaks on the data into the sizes of the peaks. Multiple regression models have been tested for the analysis. A comparison of the regression models tested is shown in the figure below.

The three regression techniques used are as follows.

1) **Linear model –** Linear modelling provides a straight line to represent all the data in the sample. But the relation between size standard and the size ladder peaks detected is not linearly distributed so it has very high SSE and thus cannot be used for the regression.
2) **Loess model -** Loess divides the data into overlapping parts like a sliding window and then performs local regression on them. It has very low SSE when modelling the peaks to the size standard, but a method with even lower SSE was required which lead to the building of the piece-wise regression modeler by the team.
3) **Piece-wise method –** Piece-wise modelling divides the data into intervals and then performs loess curve modelling on each interval of data. Since an independent model is built for each interval of data, piece-wise modelling satisfies each change in the modelling curve. It thus provides lower SSE scores than a Loess model.

All these methods have their advantages. The linear model is the simplest of the techniques but also has the most error rate. The loess model has lower error rate but piece-wise provides the lowest error rates among all the methods tried. The figure 16 below explains the model built by each of the regression methods and how they correlate to the actual size standard.

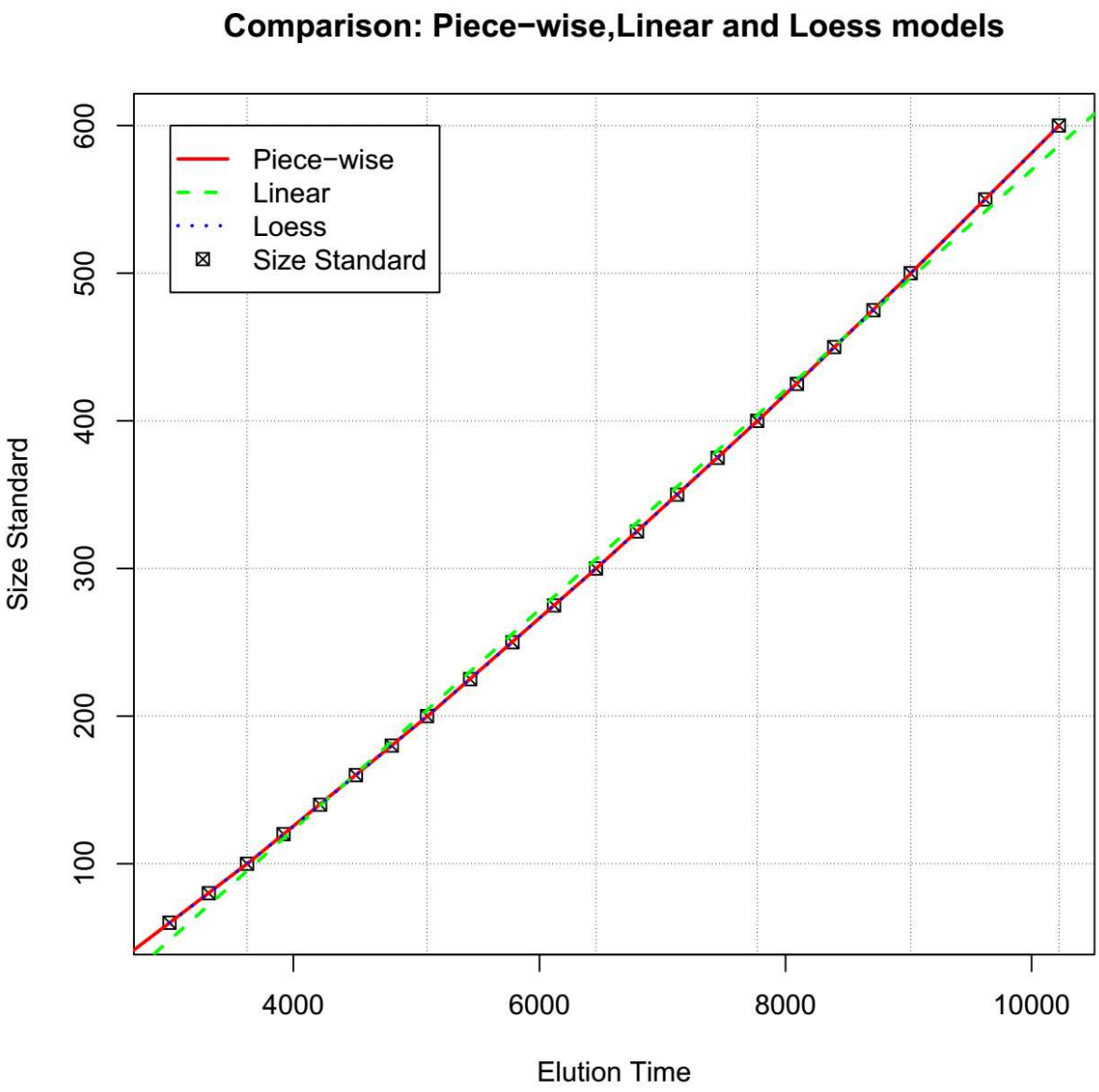


Figure 16 A comparison of the various modellig techniques tested.

The x-axis represents the actual x- coordinates of all the peaks in the data. The y-axis represents the size values in the size standard. The boxes represent the sizes in the size ladder. The green dashed line represents the Linear Regression model which fits a striaght line for the data. Its fitting quality is for the data. The blue dotted line is represents the Loess Curve Modelling. It has very low SSE. The red line is the Piece-wise curve modelling. Its fitting is similar and better to the Loess curve modelling.

3.4.5 Peak Size calculation

Once the regression model is built, all that is needed is to predict the sizes of all the peaks using the regression model.

Prediction can be done In R using “predict()” function in the “stats” package, where inputs being the model and the value for which a prediction is needed.

The steps for the process are as follows:

Step 1 – Get all peak positions provided in the tracks

Step 2 – Predict their sizes using the predict() function as stated earlier

Step 3 – Repeat the same process for other samples/tracks

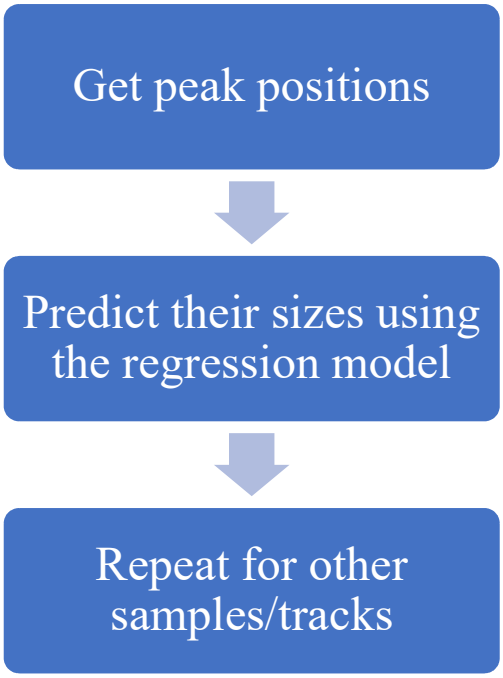


Figure 17 Flowchart for finding sizes for peaks

3.4.6 Allele Calling

Allele calling is the final step of the analysis. It analyses the size calls generated from the regression model and reports if any allele is associated with it. It does this using the following steps.

Step 1 – Choose a size belonging to a peak.

Step 2 – Check the Bin file for the closest entry to the size given.

Step 3 – Report the allele with a confidence score

A flowchart representing the algorithm is given in Figure 18.

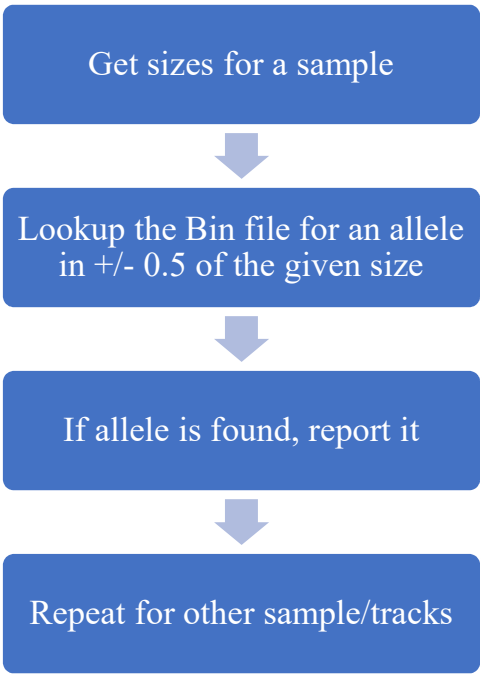


Figure 18 Flowchart for allele calling

Using this method, alleles are called for all the peaks detected in the data and reported by the analysis on the last step of the analysis.

3.4.7 Automated Blob Removal

Automated blob removal uses a combination of smoothing and peak boundary finding to remove the blob from the dye data. The steps taken to remove the blob are as follows.

Step 1 – Smooth the track data with loess smoothing with a large span

Step 2 – Find the first peak

Step 3 – Calculate its boundaries

Step 4 – Set all the data before end of first peak to zero.

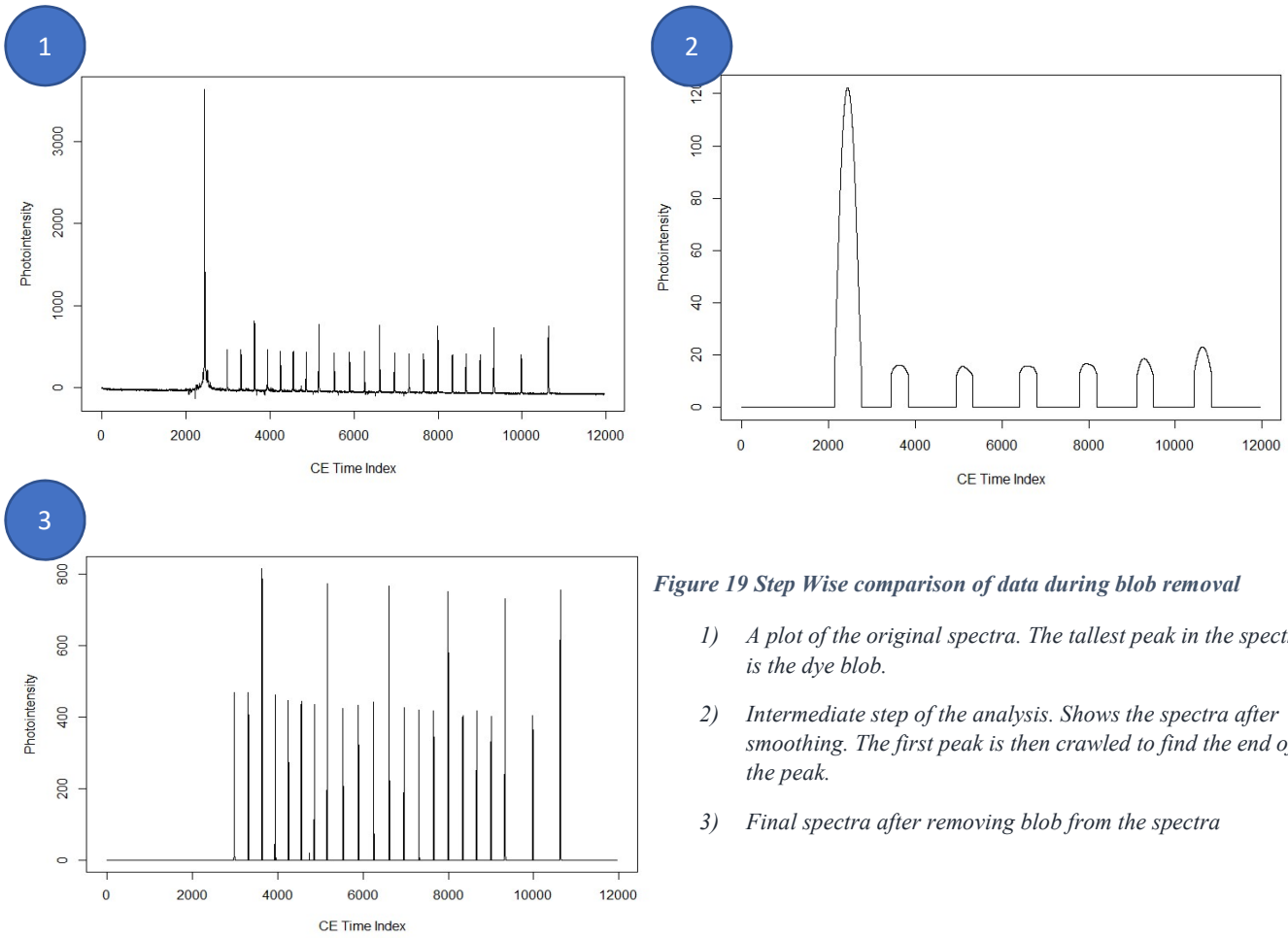


Figure 19 Step Wise comparison of data during blob removal

1) *A plot of the original spectra. The tallest peak in the spectra is the dye blob.*
2) *Intermediate step of the analysis. Shows the spectra after smoothing. The first peak is then crawled to find the end of the peak.*
3) *Final spectra after removing blob from the spectra*

Flowchart for the Automated Blob Removal method is given in Figure 20.

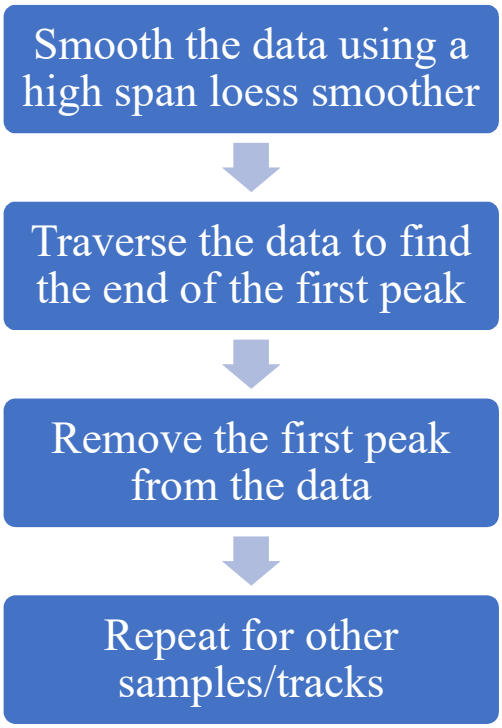


Figure 20 Flowchart for Automated Blob Removal

The automated blob removal method has been able to remove dye blobs accurately in all the samples it was tested on. Therefore, it can reliably remove dye blobs present in the samples.

3.4.8 Automated Noise Peak Removal

For noisy data, this method that can accurately select peaks that match the size ladder. It works on the principle that the proportion of data distance and size difference will be approximately equal.

Flowchart for the working of the algorithm is as follows

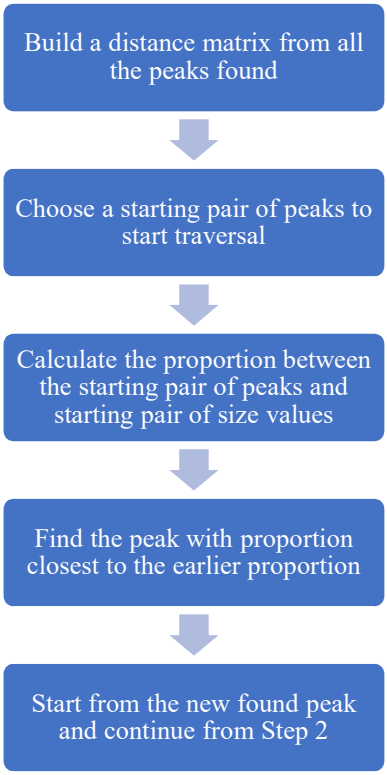


Figure 21 Flowchart for Automated Noise Peak Removal

An example of its working has been explained in the below segment.

For the given size standard with sizes
70, 85, 150, 185, 225, 250, 275, 350, 375, 435, 475, 530, 550

And the distance matrix as given below, this is how the system will choose peaks. Each cell of the given matrix is the distance between two peak numbers, arranged and numbered according to position.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	128	673	971	1267	1307	1365	1531	1791	2389	2608	3116	3445	3886	4041
2	-128	0	545	843	1139	1179	1237	1403	1663	2261	2480	2988	3317	3758	3913
3	-673	-545	0	298	594	634	692	858	1118	1716	1935	2443	2772	3213	3368
4	-971	-843	-298	0	296	336	394	560	820	1418	1637	2145	2474	2915	3070
5	-1267	-1139	-594	-296	0	40	98	264	524	1122	1341	1849	2178	2619	2774
6	-1307	-1179	-634	-336	-40	0	58	224	484	1082	1301	1809	2138	2579	2734
7	-1365	-1237	-692	-394	-98	-58	0	166	426	1024	1243	1751	2080	2521	2676
8	-1531	-1403	-858	-560	-264	-224	-166	0	260	858	1077	1585	1914	2355	2510
9	-1791	-1663	-1118	-820	-524	-484	-426	-260	0	598	817	1325	1654	2095	2250
10	-2389	-2261	-1716	-1418	-1122	-1082	-1024	-858	-598	0	219	727	1056	1497	1652
11	-2608	-2480	-1935	-1637	-1341	-1301	-1243	-1077	-817	-219	0	508	837	1278	1433
12	-3116	-2988	-2443	-2145	-1849	-1809	-1751	-1585	-1325	-727	-508	0	329	770	925
13	-3445	-3317	-2772	-2474	-2178	-2138	-2080	-1914	-1654	-1056	-837	-329	0	441	596
14	-3886	-3758	-3213	-2915	-2619	-2579	-2521	-2355	-2095	-1497	-1278	-770	-441	0	155
15	-4041	-3913	-3368	-3070	-2774	-2734	-2676	-2510	-2250	-1652	-1433	-925	-596	-155	0

Figure 22 The distance matrix for peaks found in the data.

Each cell contains distance of the peaks. The green marking indicates peaks with correct proportion to the starting pair. The red show the peaks with incorrect proportion to the starting pair.

The Automated Peak Choosing algorithm has been able to reliably choose the correct peaks on each sample provided to it as a test. So, it can be said to accurately removal noise peaks in the fragment data. It also has the added bonus of ignoring the dye blob in the process. Hence removing the dye blob from the process.

4 Outputs

There are two formats in which the analysis outputs allele calls. The two methods are as follows.

1) **Text/XLS format**

   The analysis outputs a table containing four columns as an XLS file in comma separated values. The four columns are 1) Sample name, 2) Allele found (marker name + allele index), 3) Size (in base pairs) of the allele, and 4) the Confidence Score (represents deflection of allele from ideal position). An example of the output is as given below in the table.

Table 4 Textual allele call report of the analysis.

Sample	**Allele Call**	**Size Call**	**Confidence Score**
Allelic_Ladder	TH01 4	152.07	0.74
Allelic_Ladder	TH01 5	156.11	0.85
Allelic_Ladder	TH01 6	160.08	0.71
Sample1	D13S317 12	190.63	0.88
Sample1	D13S317 14	198.55	0.78
Sample1	D7S820 11	231.43	0.41

2) **Plot/PDF format**

   The analysis also outputs a PDF of a specialized report of the alleles. It also conveys the same information as above table but in a graphical format, making alleles easier to identify. A sample of the graphical output is as given in figure below.

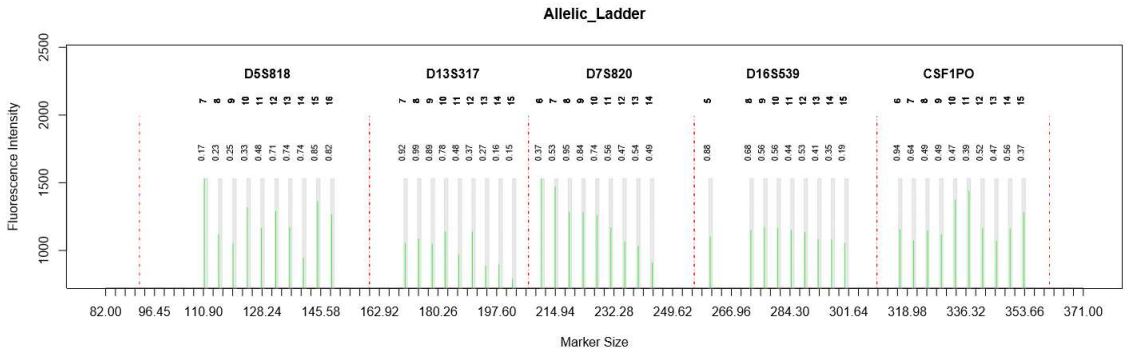


Figure 23 Graphical allele call report of the analysis.

The green bars represent an allele. Their height represents the photo intensity of the alleles. The number above the green bars is confidence scores for allele calls. The color of the bars is denoted by the color of the dye track. The indication above confidence calls is the name of the allele called. The overarching label is the molecular marker the allele call belongs to. The dotted red lines represent limits to the range of each molecular marker.

5. Results

The results of FragSTR compare well to GeneMapper 5. FragSTR can identify alleles to the same extent as GeneMapper 5, even better in some cases. Even if we consider the alleles GeneMapper 5 missed as errors in the results, we still get an accuracy of 99.02% for across all markers and batch of samples of Allelic Ladder, which is a positive control sample and 96.54% for all samples.

The process of calculation of accuracy is straight forward. For a sample, scoring follows a scheme where If for a marker,

1) the results of FragSTR and GeneMapper are same, then the score = 100%
2) FragSTR missed the particular marker, then the score is 0%
3) If FragSTR has more alleles in result than GeneMapper, then
   the score = (1 - Number of extra alleles/Total number of alleles FragSTR detected) * 100%

Using the above criterion, the scores for each marker in each sample in each batch were calculated by averaging all the scores for a marker across the all samples. The tests only show a comparison to GeneMapper. The correctness of GeneMapper results can also be questioned in case of alleles missing from GeneMapper results but found by FragSTR. A detailed table explaining the scores obtained is given below.

Table 5 Accuracy calculation for Allelic ladders for all batches of samples available. *B1 – B16 are the batches of samples available. Only the allelic ladders for each batch are considered. It serves as positive control for the analysis. Y-axis contains all the markers from the samples. The numbers show accuracy compared to GeneMapper in percentage. NA implies an Allelic Ladder was not available for the batch.*

ALM	Batch	B1	B2	B3	B4	B5	B7	B8	B9	B10	B11	B12	B14	B15	B16	Over All % Match
No of Alleles	Markers															Average
10	TH01	NA	NA	100	100	100	100	100	100	100	100	85	100	95	100	**98.33**
32	D21S11	NA	NA	92.86	93.10	86.21	96.55	98.28	98.28	94.83	96.55	91.38	93.10	96.55	100	**94.81**
10	D5S818	NA	NA	100	100	95	100	100	100	100	100	100	100	100	100	**99.58**
9	D13S317	NA	NA	100	100	100	100	100	100	100	100	94	100	100	100	**99.54**
9	D7S820	NA	NA	100	100	83.33	100	100	100	100	100	100	100	100	100	**98.61**
9	D16S539	NA	NA	100	100	100	100	100	100	100	100	100	100	100	100	**100.00**
10	CSF1PO	NA	NA	100	100	100	100	100	100	100	100	100	100	100	100	**100.00**
2	AMEL	NA	NA	100	100	100	100	100	100	100	100	100	100	100	100	**100.00**
14	vWA	NA	NA	100	100	92.31	100	100	100	100	100	100	100	100	100	**99.36**
10	TPOX	NA	NA	100	100	100	100	100	100	100	100	100	100	100	100	**100.00**
	Average			**99.29**	**99.31**	**95.68**	**99.66**	**99.83**	**99.83**	**99.48**	**99.66**	**97.08**	**99.31**	**99.16**	**100**	
								Overall Accuracy = 99.02%								

Table 6 Accuracy calculation for all samples for all batches of samples available. *B1 – B16 are the batches of samples available. All samples for each batch are considered. It serves as real world usage for the analysis. Y-axis contains all the markers from the samples. The numbers show accuracy compared to GeneMapper in percentage. NA implies an Allelic Ladder was not available for the batch.*

	Batch	B1	B2	B3	B4	B5	B7	B8	B9	B10	B11	B12	B14	B15	B16	match
No of Alleles	Marker															Average
10	TH01	100	100	100	100	100	100	100	100	94.44	92.86	100	100	100	100	**99.09**
25	D21S11	91.67	100	100	92.42	95.83	100	93.06	100	83.33	100	100	90	100	100	**96.17**
10	D5S818	100	100	100	100	81.25	100	100	100	91.67	100.00	100	100	18.75	100	**92.26**
9	D13S317	8.33	100.00	92.86	95.45	81.25	100.00	95.83	100.00	66.67	92.86	100.00	100.00	81.25	87.50	**85.86**
9	D7S820	87.5	100.0	100.0	90.9	89.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	**97.71**
9	D16S539	25	100	100	100	100	100	100	96.67	100.00	100.00	100	100	100	100	**94.40**
10	CSF1PO	75	100	100	96.97	100	100	100	95	100.00	100.00	100	100	100	100	**97.64**
2	AMEL	100	100	100	100	100	100	100	100	100.00	100.00	100	100	100	100	**100.00**
13	vWA	83.33	100.00	100.00	93.94	95.83	100.00	95.83	95.00	100.00	100.00	100.00	100.00	100.00	100.00	**97.42**
8	TPOX	79.16667	100	100	100	100	95.83	95.83	100	100.00	92.86	94.44	100	100	93.75	**96.56**
	Overall % match	**75**	**100**	**99.29**	**96.9697**	**94.375**	**99.58**	**98.05556**	**98.66667**	**93.61111**	**97.85714**	**99.44444**	**99**	**90**	**98.13**	
								Overall Accuracy = 95.71%								

6. Software walkthrough

The software experience is divided into four views

1) **the Home page view** – Contains links to the analysis and all the static content on the web page.
2) **the Data Upload view** – Allows uploading of files and manipulation of tracks
3) **the Progress view** – Shows progress indicator for the background process
4) **the Analysis Results view** – Displays the results of the analysis and contains links to download the report

Home page view

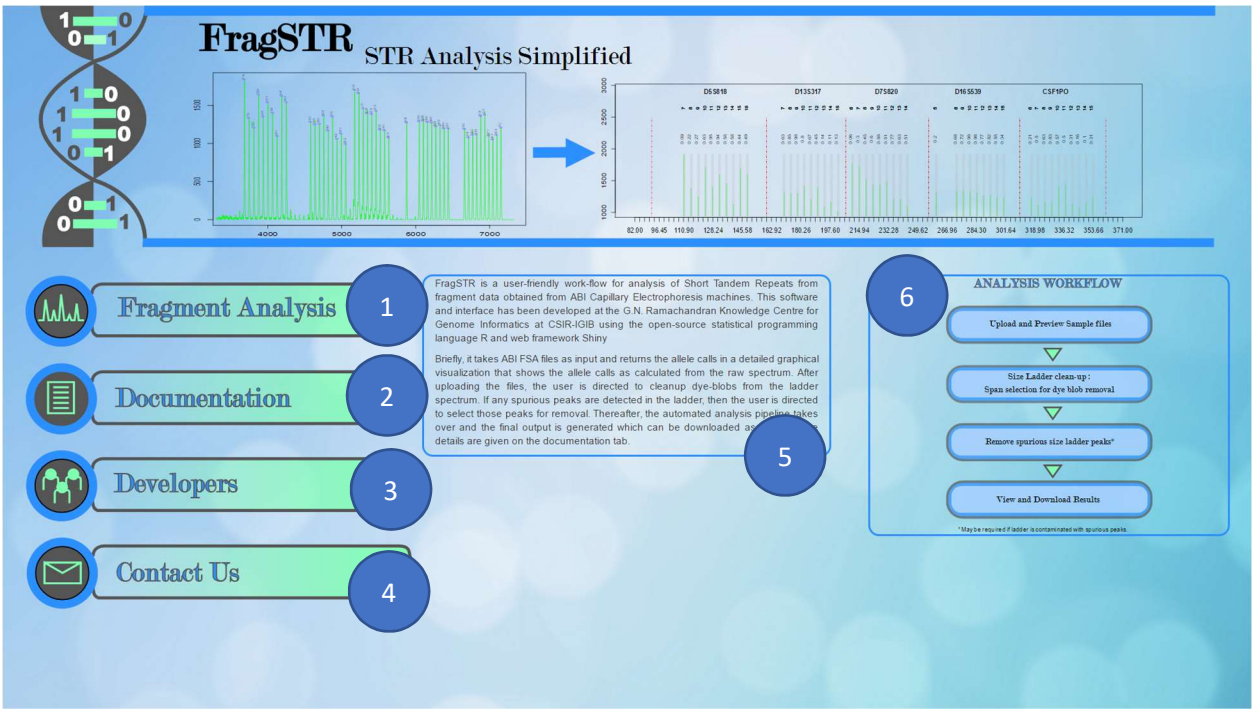


Figure 24 Screenshot of the Home Page

Legend for Home Page view:

1) Link to Data Upload page
2) Documentation of the website
3) Developer Information
4) Contact Information
5) General Information about the analysis
6) Minimalistic workflow of the analysis

The Data Upload view

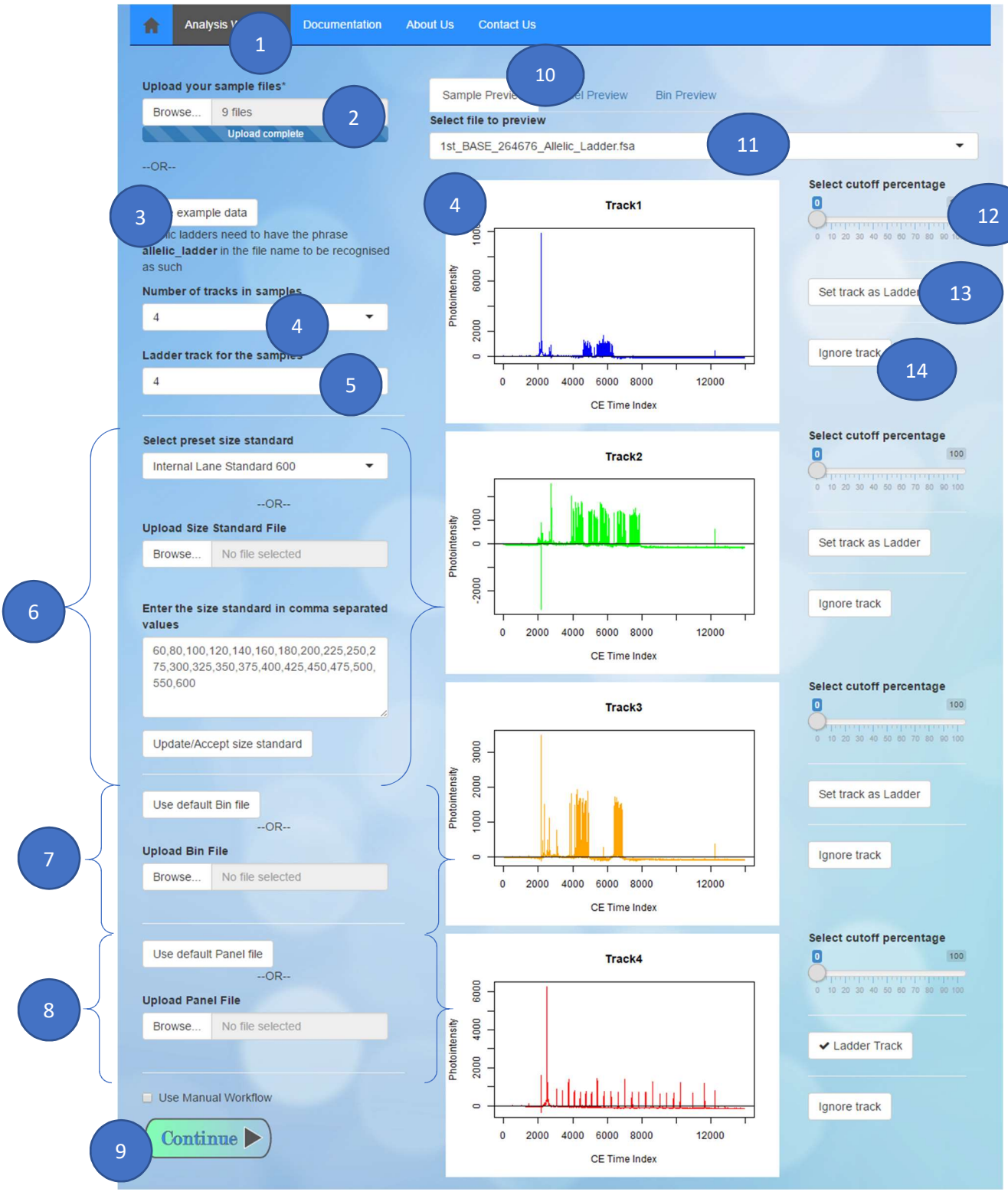


Figure 25 Screenshot of the Data Upload page

Legend for Data Upload view:

1) Tab Bar to switch between Home, Analysis, Documentation, Contact Info
2) Button to upload files
3) Choose preloaded example data for testing
4) Override for number of tracks in sample
5) Override for ladder track in sample
6) Input Size Standard file
7) Input Bin file
8) Input Panel file
9) Continue with the analysis if all files are correctly uploaded
10) Tab Bar to View various input files
11) Select input file to view its tracks
12) Select horizontal cutoff for the tracks
13) Button to set track as Size Ladder
14) Button to ignore track from analysis

The Progress view

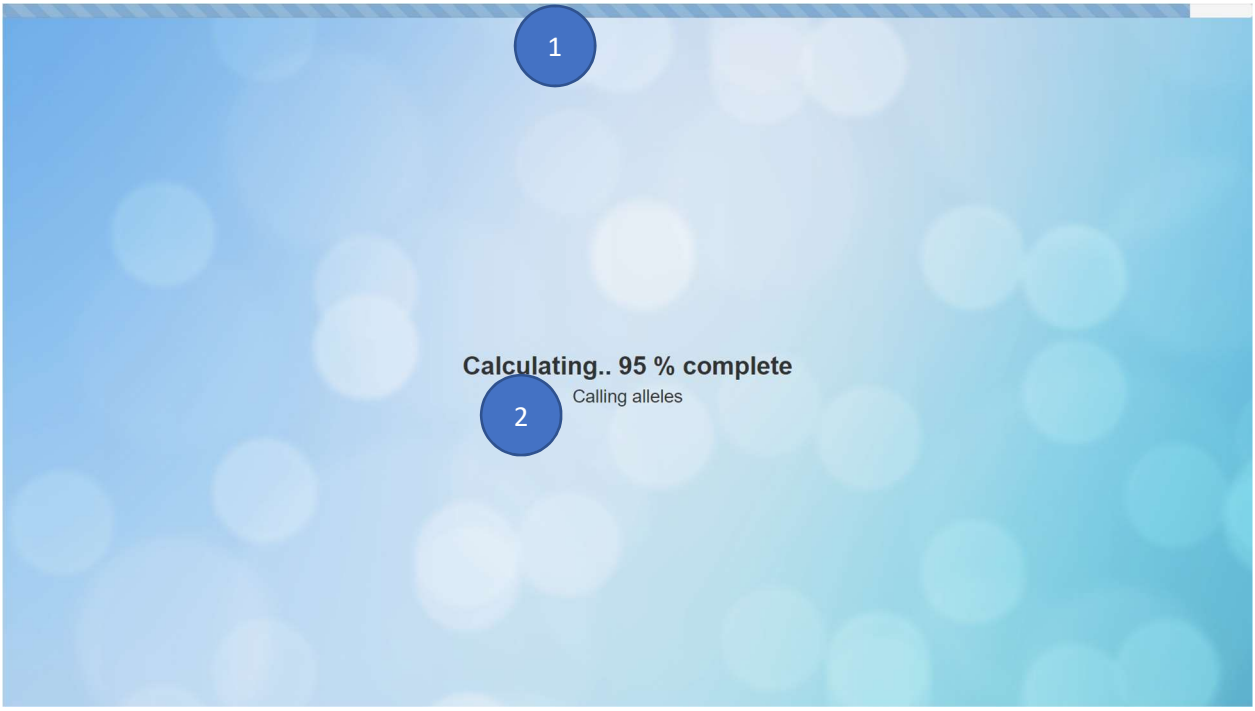


Figure 26 Screenshot of the Progress view

Legend for Progress View:

1) Progress percentage indicator
2) Current activity in the workflow

There are 9 such activities in the workfow.

1) Removing negative values
2) Removing dye blobs
3) Preparing ladders/tracks
4) Applying cutoff
5) Finding peaks
6) Selecting required range from tracks
7) Sizing the peaks
8) Calculating confidence scores
9) Calling alleles

Calculating takes 15 seconds for each sample of 4 dye tracks

The Results view

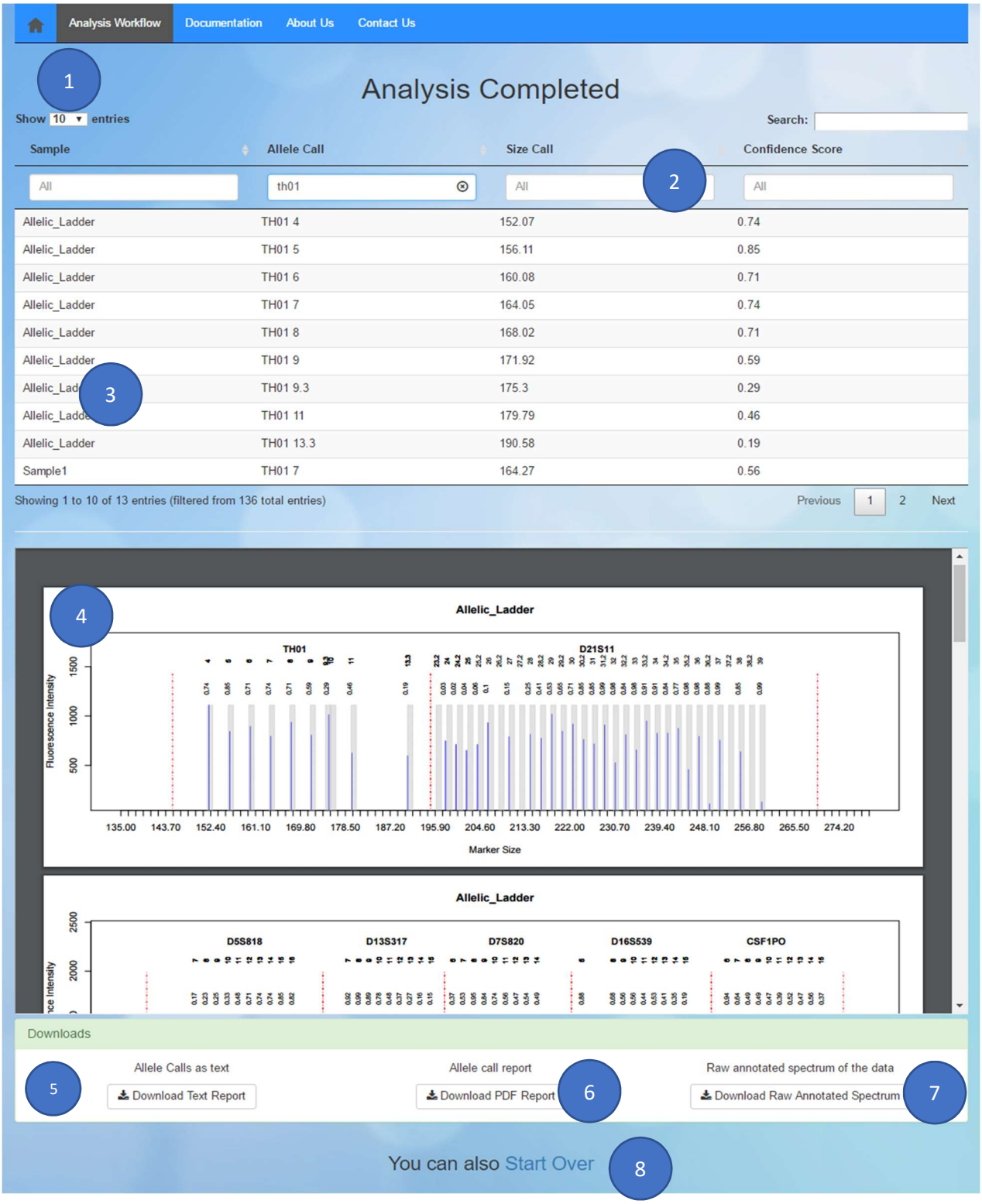


Figure 27 Screenshot of Results view

Legend for Results View:

1) Number of rows of Textual report to display
2) Column filters for the Textual report
3) Textual Report
4) Graphical Report
5) Download Textual Report as XLS
6) Download Graphical Report as PDF
7) Download plots displaying peak alignment as PDF
8) Refresh the web page

7. Future Scope

STR markers have the capacity to differentiate between two types of DNA, but can also determine similar DNA. This property of the STR markers can be used in application for many purposes. A few applications for this STR analysis workflow include.

1) **Cell line authentication –** comparing the STR markers of known cell lines to authenticate the cell (verify that the cell line is what we think it is)
2) **Pedigree DNA analysis –** determination of breed of canines using STR markers obtained from the DNA of the dog.
3) **Heredity analysis –** comparing STR of two or more people to identify whether they have a common heritage
4) **Human identification from sample, e.g. Using CODIS Panels –** identifying people by comparing obtained STR Markers with a database of STR Markers
5) **Animal, human and plant disease classification –** using previously known information about the a STR Markers' correlation to the occurrence of a disease in humans, plants and animals

The basic infrastructure for fragment analysis is already present. To add a new capability, a module must be built for the kind of output that is required. There is also the possibility of adding plugins to the software to activate specialized functions. Hence, the analysis can be used for any type of fragment based analysis and provide any type of outputs as required just by addition of modules over the currently available methods.

8 Discussion

8.1 Issue of noise in size ladders

In an ideal case, the size ladder should contain peaks equal in number to the sizes in the size standard. A regression model is then created from the positions of the peaks of the size ladder and the sizes in the size standard. Using this regression model, rest of the tracks are sized and alleles are called.

Deviating from the ideal case, samples may contain artifacts due to interference that cause the appearance of more than the ideal number of peaks. A few examples of size ladders containing artifacts are given in Figure 28.

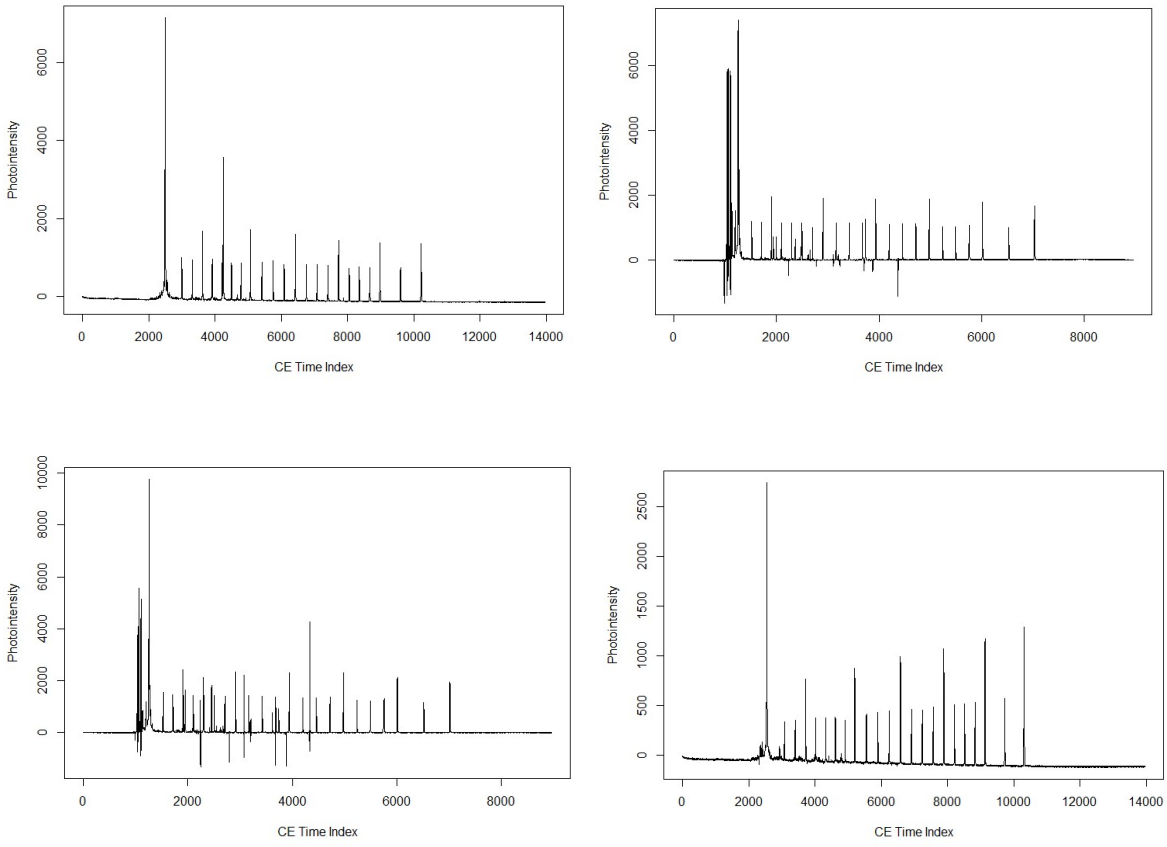


Figure 28 Examples of noisy ladders

When such size ladders are provided, the analysis has great difficulty in picking which of the peaks belongs to the size ladder and which are artifacts. But with careful observations, we can figure out two phenomena which might help us in identifying size ladder peaks. The phenomena are as follows.

1) The distances between the two size ladder peaks is proportionate to the size difference between their corresponding sizes for any pair of peaks. This may help us eliminate any peaks that are too far from the ideal position.
2) The distances between size ladders for the same size difference decreases when going further on the ladder. This is usually applicable, but falls short in some cases.

Using these principles, an algorithm can be built to choose peaks from noisy size ladders using proportions of size differences. This method requires a pairwise distance matrix of all the peaks detected. This matrix is then parsed to recognize valid size ladder peaks. The details of the algorithm can be found in Automatic Peak Selection method in the Methods section.

The algorithm works for some cases but for other cases, some noise peak is at a better position to be selected as a size ladder peak. This can cause issues with the sizing of the peaks. Even one base pair size difference in the sizes can cause an entirely different allele to be called.

There may be two ways to overcome these.

1) Try to remove any outlier peaks that could be removed. But detecting any peak to be outlier poses different problem.
2) Consider the Y- coordinates of the peaks when taking a decision. But any phenomena to consider y-axis values is still unknown.

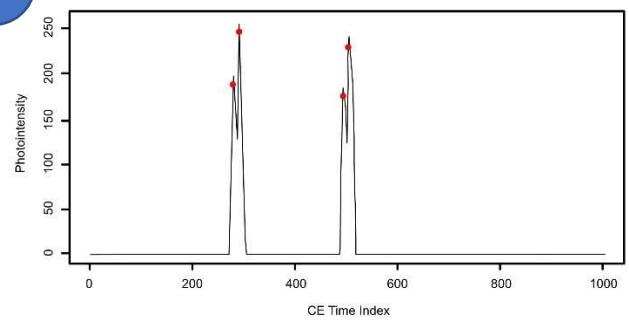
A guess would be trying to draw a curve across the samples and keep the ones closest to it. But this may cause problems with noise peaks which are almost the same height of the size ladder peaks.

An algorithm for deselection of noise peaks has been developed, explained in Section 3.4.8 of the document. But the algorithm has high time complexity. New ways need to be found for elimination of noise in the size ladders.

8.2 Discussion – Stutter peaks

Stutter peaks are peaks that appear as smaller peaks attached to larger peaks, with about 1-4 base pairs difference in sizes. These peaks can lead to duplicate allele calls and extra allele calls. These can attribute to false positives which reduce the accuracy of the analysis.

1



2

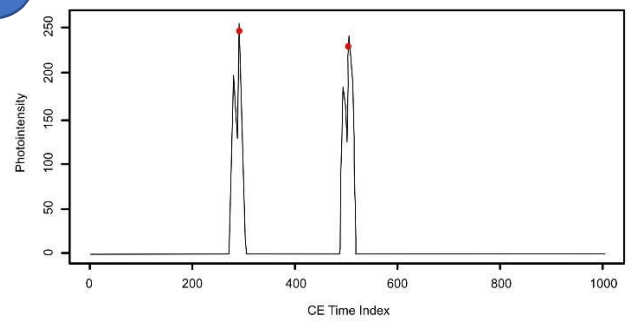


Figure 29 Difference between peaks involving stutter peaks and peaks without stutter peaks.

The plot shows the spectra for capillary electrophoresis data. The red circles show the peaks that are found in the spectra.

Image 1 – The peaks found in the spectra before smoothing the data. The smaller peak before each large peak is the stutter peak. This peak is not required as they are caused due to slippage errors while creation of the data.

Image 2 – The peaks found in the spectra after smoothing of the data. The stutter peaks are not detected as peaks after smoothing the data. Hence, the process of smoothing is effective in eliminating stutter peaks.

A way to solve this is by smoothing the data before applying peak finding. The smoothing eliminates the stutter peak by averaging the cone that forms the peak. Figure 30 provides an illustration explaining the process of stutter peak removal using smoothing.

Smoothing eliminates stutter peaks, but there are two main problems associated which need to be solved to make the workflow better.

1) The process of smoothing the data takes time causing the workflow to have a bottleneck.
2) The peaks positions may shift by an insignificant amount causing an extra position correction algorithm to be run causing delay in the workflow.

Hence, smoothing removes stutter peaks accurately, but not optimally. A new algorithm will have to be built to eliminate the need for smoothing the data to remove stutter peaks.

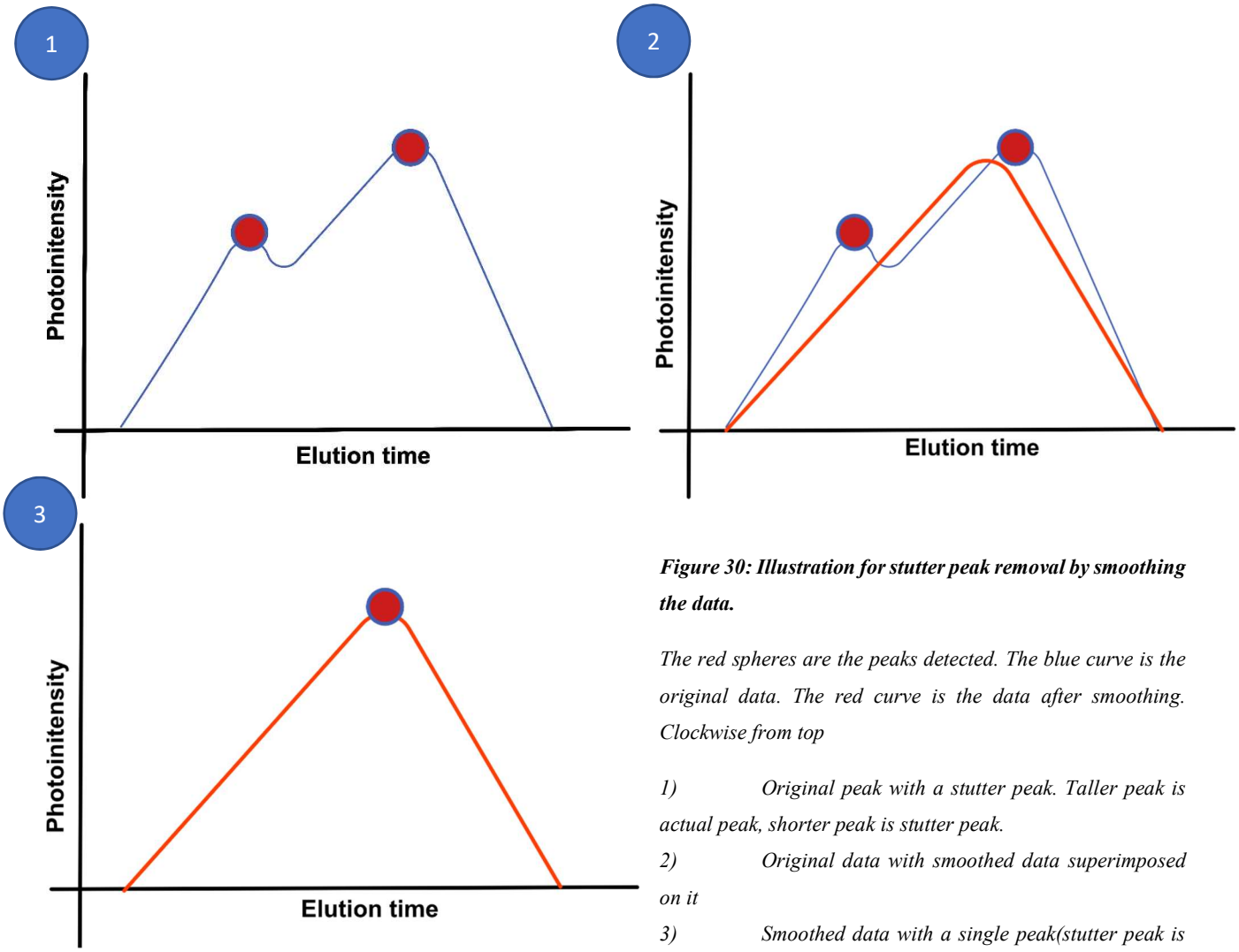


Figure 30: Illustration for stutter peak removal by smoothing the data.

The red spheres are the peaks detected. The blue curve is the original data. The red curve is the data after smoothing. Clockwise from top

1) Original peak with a stutter peak. Taller peak is actual peak, shorter peak is stutter peak.

2) Original data with smoothed data superimposed on it

3) Smoothed data with a single peak(stutter peak is removed)

.

9. Conclusion

The resulting software of the thesis will allow the community to analyze fragment data without the use of commercial software. The software currently implements two types of analysis which are 1) Marker allele analysis and 2) Triplet Repeat Expansion analysis (Though not mentioned in the thesis). The base for all sorts of STR related analysis is ready with the completion of the thesis. Adding a new type of analysis to the process only requires using the available pipeline and modifying it to provide desired results.

The process has been tested against widely used commercial software such as GeneMapper, and it performed at par with them. Thus, it is highly reliable to produce results that are accurate and consistent.

The issue with noisy data and blob removal has also been solved with the use of automated blob removal and peak choosing algorithms, that have proven themselves to be highly resistant to error. Hence, they reliably produce accurate results.

These methods present in the analysis pipeline/software have the capability to make it as an improvement over commercial software. It also has the added benefit of being web-enabled, thus can be accessed anywhere with an internet connection.

Therefore, it will be reasonable to assume that “FragSTR”, the result of the thesis, will be able to compete in accuracy against other commercial software and will provide the community the necessary resources to analyze fragments from the comfort of their own computers.

10. References

[1] http://guardian.meragenome.com Dr. Vinod Scaria, IGIB

[2] http://www.nature.com/nrg/journal/v11/n6/abs/nrg2779.html Cirulli, Elizabeth T., and David B. Goldstein. "Uncovering the roles of rare variants in common disease through whole-genome sequencing." *Nature Reviews Genetics* 11.6 (2010): 415-425.

[3] http://genome.cshlp.org/content/24/11/1894.short Willems, Thomas, et al. "The landscape of human STR variation." *Genome research* 24.11 (2014): 1894-1904.

[4] http://hmg.oxfordjournals.org/content/9/16/2403.short Gray, Ian C., David A. Campbell, and Nigel K. Spurr. "Single nucleotide polymorphisms as tools in human genetics." *Human molecular genetics* 9.16 (2000): 2403-2408.

[5] https://www.thermofisher.com/us/en/home/life-science/sequencing/fragment-analysis/microsatellite-marker-analysis.html Thermofisher Scientific Pvt. Ltd.

[6] http://www.pnas.org/content/91/24/11348.short Woolley, Adam T., and Richard A. Mathies. "Ultra-high-speed DNA sequencing using capillary electrophoresis chips." *Analytical chemistry* 67.20 (1995): 3676-3680.

[7] http://www.sciencedirect.com/science/article/pii/S0021967301902043 Swerdlow, Harold, et al. "Capillary gel electrophoresis for DNA sequencing: laser-induced fluorescence detection with the sheath flow cuvette." *Journal of Chromatography A* 516.1 (1990): 61-67.

[8] http://seqinr.r-forge.r-project.org Charif, D. and Lobry, J.R. (2007) seqinr: Biological Sequences Retrieval and Analysis

11. Appendix

Web-application location

The website is located on the address www.fragstr.co.in.

NOTE: If the website displays white screen i.e. blank, please contact your IT Department to allow the site through firewall.

Web-application framework

Frontend – Shiny Web framework for R, Node.js, JS

Backend – R statistical programming language