Procedure to create a dataset made from diluted DNA samples

Before to run the capillary electrophoresis analysis

→ All the sample names need to follow this format:

SampleName-DNAquantity-Replica

Example 1) DNA007-31.25pg-1
2) PersX-125pg-4

We will use for the following document, the denomination “sample” to design each different profile, even if obtain by the replication of the amplification.

In GeneMapper

→ Once the electropherogram analysis finished, control the results as usual. You can applied a threshold and clean the alleles.

→ Export the table of results (all samples and **one** ladder) from GeneMapper to a given folder in **.csv** format. The data separator is the comma (,) and the decimal separator the point (.)

We will refer to this file along this procedure as **initial file**. This file needs some adjustment before to be imported in R.

Export the following data from GeneMapper:

Sample File	Sample Name	Run Name	Panel	Marker	Allele 1	Size 1	Height 1	Peak Area 1

Table 1: Name of the data to export from GeneMapper. You can export as much as 40 alleles and their parameters (size-height-peak Area).

C++ script to process the initial file

We developed a script in C++ language to process the initial file into a clean file for R (we wanted to keep, in this initial file, all the alleles analyzed without threshold for further potential exploration).

This script will align in column the alleles present for each marker based on the ladder and remove the OL alleles. This step gives a table easier to clean by removing manually the alleles which are not part of the profile.

Example:

	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Sample Name	Marker	Allele 1	Size 1	Height 1	Peak Area 1	Allele 2	Size 2	Height 2	Peak Area 2	Allele 3	Size 3	Height 3	Peak Area 3
49	Ladder_NGMSElect	D10S1248	8	75.09	1272	8655	9	79.36	1184	7848	10	83.65	1128	7447
50	007-31.25pg-1	D12S391	OL	228.05	25	156	18	245.06	237	1968	19	249.25	316	2428
51	007-31.25pg-9	D12S391	OL	230.39	15	65	18	245.17	410	2863	19	249.03	347	2376
52	007-15.63pg-7	D12S391	OL	236.23	25	124	16	237.38	25	153	18	244.95	164	1048
53	007-31.25pg-2	D12S391	OL	235.07	15	74	16	237.37	29	178	17	240.96	47	341
54	007-31.25pg-10	D12S391	OL	228.18	16	80	OL	231.54	21	150	OL	236.29	21	88
55	007-15.63pg-8	D12S391	OL	235.04	34	154	OL	238.2	25	155	OL	242.29	19	104
56	007-31.25pg-3	D12S391	14	228.68	24	98	17	240.96	97	653	18	245.01	1372	9392
57	007-15.63pg-1	D12S391	14	229.12	31	189	17	241.29	44	296	OL	242.91	26	66
58	007-15.63pg-9	D12S391	17	240.86	54	437	18	245.06	515	3759	18.3	247.85	17	42
59	007-31.25pg-4	D12S391	OL	225.18	18	134	16	237.49	31	188	17	241.08	83	669
60	007-15.63pg-2	D12S391	OL	226.58	23	134	OL	239.26	39	198	OL	240	16	82
61	007-15.63pg-10	D12S391	OL	225.55	19	90	OL	226.5	20	117	OL	227.97	27	153
62	007-31.25pg-5	D12S391	OL	226.74	20	97	OL	237.8	25	111	17	241.08	60	526
63	007-15.63pg-3	D12S391	OL	237.58	29	188	18	245.06	442	3020	19	249.14	641	4355
64	007-15.63pg-11	D12S391	16	237.45	27	117	18	245.12	259	1767	20.3	255.64	18	129
65	007-31.25pg-6	D12S391	OL	234.23	25	139	OL	234.96	28	169	17	240.98	50	280
66	007-15.63pg-4	D12S391	OL	228.05	29	194	14	229.42	13	110	OL	231.63	27	172
67	007-7.81pg-1	D12S391	OL	229.96	24	117	OL	231.97	20	109	18	245.06	236	1877
68	007-31.25pg-7	D12S391	OL	233.7	28	157	OL	236.01	19	47	18	245.12	154	1004
69	007-15.63pg-5	D12S391	OL	236.13	21	55	18	245.06	76	533	19	249.14	129	1027
70	007-7.81pg-2	D12S391	OL	238.63	32	77	19	249.25	69	514	OL	266.16	23	119
71	007-31.25pg-8	D12S391	OL	238.53	18	88	17.3	243.98	27	139	18	245.06	134	980
72	007-15.63pg-6	D12S391	OL	226.64	27	159	OL	239.58	22	130	17.3	244.52	28	194

Image 1: Before the script, the alleles are not align on the ladder

	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Sample Name	Marker	Allele 1	Size 1	Height 1	Peak Area 1	Allele 2	Size 2	Height 2	Peak Area 2	Allele 3	Size 3	Height 3	Peak Area 3
377	Ladder_NGMSElect	D12S391	14	229.13	1412	9944	15	233.1	1560	11007	16	237.07	1425	9886
378	007-31.25pg-1	D12S391												
379	007-31.25pg-9	D12S391												
380	007-15.63pg-7	D12S391									16	237.38	25	153
381	007-31.25pg-2	D12S391									16	237.37	29	178
382	007-31.25pg-10	D12S391												
383	007-15.63pg-8	D12S391												
384	007-31.25pg-3	D12S391	14	228.68	24	98								
385	007-15.63pg-1	D12S391	14	229.12	31	189								
386	007-15.63pg-9	D12S391												
387	007-31.25pg-4	D12S391									16	237.49	31	188
388	007-15.63pg-2	D12S391												
389	007-15.63pg-10	D12S391					15	232.7	21	88	16	237.14	22	117
390	007-31.25pg-5	D12S391												
391	007-15.63pg-3	D12S391												
392	007-15.63pg-11	D12S391									16	237.45	27	117
393	007-31.25pg-6	D12S391												
394	007-15.63pg-4	D12S391	14	229.42	13	110								
395	007-7.81pg-1	D12S391												
396	007-31.25pg-7	D12S391												
397	007-15.63pg-5	D12S391												
398	007-7.81pg-2	D12S391												
399	007-31.25pg-8	D12S391												
400	007-15.63pg-6	D12S391												

Image 2: After the script, the alleles are aligned on the ladder

Put the script in the same folder as the initial files. We compile and run this script through a terminal window.
Open the terminal and access the folder where the script and the initial files are. Use Tab button for automatic completion. Write the command line after the $ symbol. If you copy the text from below, don't copy the $:

```
$ cd ~/
```

Command 1: Access the folder where the data and the script C are.

```
$ g++ ScriptC.cpp -o CompileScript.out
```

Command 2: Compile the script

Change manually the initial file name that you want to process to “initial_file.csv”. Return under the terminal and run this command:

```
$ ./CompileScript.out
```

Command 3: Transform the data file which has for name “initial_file.csv”

A new csv file named “arranged.file.csv” is created in the folder. It can be opened with Excel, LibreOffice or other spreadsheet maker programs.

Under LibreOffice or Excel

Select the whole table except the first line with the column name and sort the data. It will removed the blank line between the different markers. Sort the table by marker then to clean the profile if it was not done earlier. Save the document in csv, make sure that the field delimiter is a comma “,” and the decimal separator a point “.”.

Import the csv file in R

Script 0.DataImportation.R

We recommend the use of an R program with a user friendly interface, such as RStudio (free and open-source), as it makes R easier to handle and allows a good overview of the script, the data and even the graph.

Load the script on RStudio and go through it to complete the missing information before to run it.

This script imports the arranged.file.csv above mentioned, extracts the theoretical quantities from the sample name and allow the user to create a new column with the quantities measured (called real quantity). For the following scripts, it makes sure that all data imported as factor are transformed into a character chain. At the end of the script, the data for the amelogenin are removed and a copy of the data is created (called data.copy0).

Calculate the H, proxy for the DNA quantity

Script 1.Determination_H.R

This script needs to be run after the script **0.DataImportation.R** as it uses the dataset imported and modified.

Firstly set a threshold. If you already defined a threshold in GeneMapper, use the same here or define a new one (higher). The script will replace all the peak height falling below this threshold by “NA”, and will consider them later as dropout.

We implemented the model describes by Tvedebrink *et al*. [REF] In a temporary matrix, the script will extract each sample, one by one, determined the sum of the peak heights and the number of alleles present (the homozygous allele are multiplied by two). Then it divides the sum by the number of allele and rounded it two decimals after the comma. This is our H value.

Finally, we merge the temporary matrix with our dataset “data” and add a new column with the information about the type of allele (0 if an allele is homozygote and 1 if the allele is heterozygote). A copy of the data is created (called data.copy1).

Calculate dropout

Script 2.Calcul_DO.R

This script needs to be run after the script **1.Determination_H.R** as it will take in account the threshold defined previously and all the allele peak heights falling below this threshold and replaced by “NA”, and called also dropout.

The dropout calculation is defined in the script as follows; for heterozygous alleles, we counted one dropout for each allele below the threshold. They are saved in two new columns named DO.H1 and DO.H2 (respectively for the allele 1 and the allele 2). For homozygous alleles, we counted two dropouts when the allele was below the threshold. With the capillary electrophoresis used during the analysis, it is not possible to define if one allele is completely missing and the other one fall below the threshold, or if both are very low and the sum falls then below the threshold. The count is saved in the column names DO.H1. The column DO.H2 is filled with “NA”.
The initial dataset “data” is split into heterozygous data (data.het) and homozygous data (data.hom), towards the script and the two datasets are bound together at the end.
We added also a new column “Locus.DO”, which contains 1 when both heterozygous allele are missing or homozygous allele below the threshold, 0 otherwise.
The last part of the script is summing up all the dropout alleles for each profile. A copy of the data is created (called data.copy2).

Attribution of the dye corresponding to marker

Script 3.DyeAttributionPerMarker.R

This script can be run at any time during the process of data variable acquisition, but we advise to follow the script order numbering. In the first part, we define four vectors, named by dye (Blue, Green, Black, Red) and containing the different markers present on the kit NGMSElect from Applied Biosystems™. A temporary empty vector will store the dye information of each marker obtained through a loop. This vector is added to our dataset “data” into a new column called “Dye”. A copy of the data is created (called data.copy3).

Calculate base pair mean

Script 4.Mean_bp.R

This script can be run at any time during the process of data variable acquisition, but we advise to follow the script order numbering. As we can see when we make replication from a sample, on the same electropherogram or on a different run, the size may vary slightly, within a tolerance range given by the ladder and the GeneMapper parameters. This script has for purpose to calculate for each allele of each marker, based on all alleles present on the dataset, an average size in base pairs, called estimated base pair (Est.bp). The results are merged with our dataset “data” in new columns called “Est.bp1” and “Est.bp2” regarding if they belong to the Allele 1 or Allele 2. A copy of the dataset is created (called data.copy4).

Reshaping the data

Script 5.Data_formatting.R

This script is formatting the data to allow each allele to have a row. Under the “reshape” function, it will first copy the general information valid for both allele (Run.Name, Sample.File, Panel, Sample.Name, Marker, RealQuantity, TheoreticalQuantity, H, Dye, Type, Locus.DO, DO.tot). As a result, the data frame will double the initial number of row. Then the informations about Allele 2 will be transferred in these new rows, below Allele 1.
We add also new columns scaling the data (subtracting the mean and dividing by the standard deviation) for future investigations (H.scale, Est.bp1.scale, Size.1.scale).