*IFator*, *YIFator* and *XIFator* are three R programs that capture the sequence variation present in Illumina ForenSeq™ Genotype reports (in autosomal, Y and X STRs, respectively) and assign sequence-based (rather than length-based) genotypes for each locus and individual. Sequence-based alleles are coded as variants of the length-based alleles; i.e., a 13-repeat allele can have sequence variants designed as 13a, 13b, etc. The actual sequence (as provided in the Illumina ForenSeq™ Genotype report file) corresponding to each sequence allele can be found in the Excel files *allele sequence variation.xlsx* (autosomal STRs), *YSTRallestructure.xlsx* (Y STRs) and *XSTRallestructure.xlsx* (X STRs), along with the rationale used, particularly in the more complex cases, to assign suffix letters to each sequence variant.

The input files needed to run each program have the same structure:

i) *seqaldefinitions.csv*, *yseqaldefinitions.csv*, and *xseqaldefinitions.csv* (provided in this folder) contain one line per each different locus and allele, and each line consists of (comma-separated), the locus name, the length allele name, the sequence allele name, and the actual sequence:

CSF1PO,8,8a,AGATAGATAGATAGATAGATAGATAGATAGAT

CSF1PO,9,9a,AGATAGATAGATAGATAGATAGATAGATAGATAGAT

CSF1PO,10,10a,AGATAGATAGATAGATAGATAGATAGATAGATAGATAGAT

CSF1PO,11,11a,AGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGAT

CSF1PO,12,12a,AGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGAT

CSF1PO,13,13a,AGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGAT

CSF1PO,14,14a,AGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGAT

D10S1248,11,11a,GGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAA

D10S1248,12,12a,GGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAA

.

.

.

These files need not be edited unless new alleles are discovered (see below).

ii) The actual data, as it appears in the “Automal STR coverage”, “Y STR coverage”, and “X STR coverage” sheets of the Illumina ForenSeq™ Genotype Excel report file. Each should be formatted into csv files without row or column header, with one allele per row and four fields: individual ID, locus name, allele length, allele sequence (see, respectively, testdata.csv, ytestdata.csv, and xtestdata.csv):

ALE011,CSF1PO,10,AGATAGATAGATAGATAGATAGATAGATAGATAGATAGAT

ALE011,CSF1PO,12,AGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGAT

ALE011,D10S1248,13,GGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAA

ALE011,D10S1248,17,GGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAAGGAA

ALE011,D12S391,22,AGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGACAGACAGACAGACAGACAGACAGACAGACAGACAGAT

ALE011,D12S391,23,AGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGACAGACAGACAGACAGACAGACAGACAGACAGAC

.

.

.

Data should be sorted by individual and then locus; check that no more than two alleles are present in an individual for each locus. A different file name can be used, as long as the *inputfile="testdata.csv"* command in the program file is appropriately edited to assign the new file name.

Once the input files are prepared, the program can be run from the R command line or by using interfaces such as R Studio. Before running the program, it is advisable to check the initial lines

*inputfile="testdata.csv"*

*filenameforallelesnotfound="allelesnotfound.csv"*

*filenameforseqgenotypes="seqgenotypes.csv"*

*filenameforseqallelefrequencies="seqfreq.csv"*

*filenameforlenallelefrequencies="lenfreq.csv"*

*filenameforstats="stats.csv"*

*setwd("c:/docum180612/Francesc/Illumina Forense")*

to ensure that the working directory is correct and that the input and output files are given their correct names.

*IFator*, *YIFator* and *XIFator* examine each allele sequence in the input data and confront it with the allele definitions in the *seqaldefinitions.csv*, *yseqaldefinitions.csv*, and *xseqaldefinitions.csv* files. New alleles (which are not defined in these files) can be present in your sample. If that is the case, these programs will stop and produce this error message: *"new allele found. Check the filenameforallelesnotfound"*. If that is the case, the file name attributed to *filenameforallelesnotfound* (*allelesnotfound.csv* by default) will contain the alleles present in the sample but not in the allele definition files. Each line is an instance of a new allele, with individual ID, locus ID, length allele and actual sequence. These alleles should be then incorporated to the allele definition files, and be given a name (allele length plus a letter suffix). We recommend following the rationales provided in the *allele sequence variation.xlsx* (autosomal STRs), *YSTRallestructure.xlsx* (Y STRs) and *XSTRallestructure.xlsx* (X STRs)files, although they should not be taken as official allele designations. It may be the case that a new length allele appears in a locus without any sequence variation; in that case, it should be given the “a” suffix. For instance, if an STR had a defined range of 9-15 repeats without any sequence variation, and the allele 16 appears in your sample, then it will prompt the error message above, and it should be incorporated into the allele definition file as sequence allele 16a.

All output files are in the semicolon-separated format that Excel recognizes, so they can be opened directly from Excel. Not, on the contrary, that the .csv input files are comma-separated, and that an Excel file saved as .csv will cause IFator to crash. Such a file should be first opened with the Windows Notebook or any text file editing program, and the semicolons (“;”) must be replaced with commas (“,”).

*IFator* produces the following output files:

i) *filenameforseqgenotypes* (*seqgenotypes.csv* by default) contains the sequence-based genotypes for each individual

ii) *filenameforseqallelefrequencies* (*seqfreq.csv* by default)contains the sequence allele frequencies for each locus.

iii) *filenameforlenallelefrequencies* (*lenfreq.csv* by default)contains the length allele frequencies for each locus.

iv) *filenameforstats* (*stats.csv*)contains the a priori statistics for each locus: sample size, number of different length alleles (K(len)), number of different sequence alleles (K(seq)), expected length heterozygosity (het(len)), expected sequence heterozygosity (het(seq)), power of discrimination based on length alleles (POD(len)), power of discrimination based on sequence alleles (POD(seq)), chance of excluding a false father in a paternity trio based on length alleles (CE(len)), or on sequence alleles (CE(seq)).

*YIFator* produces the following output files:

i) *filenameforseqhaplotypes* (*yseqhaplotypes.csv* by default)contains the sequence-based haplotypes of each individual

ii) *filenameforseqghapfreq* (*yhapseqfreq.csv* by default)contains the definition and frequencies of all different sequence-based haplotypes.

iii) f*ilenameforlenghapfreq* (*yhaplenfreq.csv* by default)contains the definition and frequencies of all different length-based haplotypes.

iv) *filenameforseqallelefrequencies* (*yseqfreq.csv* by default)contains the sequence allele frequencies at each locus

v) *filenameforlenallelefrequencies* (*ylenfreq.csv* by default)contains the length allele frequencies at each locus

vi) *filenameforstats* (*ystats.csv*) gives the number of haplotypes (without missing alleles), the number of different sequence haplotypes and the sequence haplotype diversity, and number of different length haplotypes and the length haplotype diversity.

Finally, XIFator produces only one output file (*filenameforseqgenotypes*, *xseqgenotypes.csv* by default), containing the sequence-based genotypes of each individual at each X-STR locus.