

Lecture de fichiers au format ABIF sous

Jean R. Lobry



25 juillet 2017

Essai de lecture de fichiers au format ABIF pour établir un profil génétique. Cas d'un fichier d'empreintes génétiques utilisées en sciences forensiques en 2008. Utilisation du standard de taille interne pour estimer la taille des fragments. Utilisation de l'échelle allélique pour identifier les allèles présents dans l'échantillon.

Table des matières

1	Introduction	2
2	Importation des données	2
3	Utilisation de l'étalon de taille interne	3
4	Exploitation des échelles alléliques	7
5	Confrontation avec l'échelle allélique	9
5.1	Au locus D8S1179	10
5.2	Au locus D21S11	10
5.3	Au locus D7S820	11
5.4	Au locus CSF1PO	11
6	Pour le fluorochrome VIC	12
6.1	Au locus D3S1358	12
6.2	Au locus TH01	12
6.3	Au locus D13S317	13
6.4	Au locus D16S539	13
6.5	Au locus D2S1338	14
7	Pour le fluorochrome NED	14
7.1	Au locus D19S433	15
7.2	Au locus vWA	15
7.3	Au locus TPOX	16
7.4	Au locus D18S51	16
8	Pour le fluorochrome PET	17
8.1	Au locus Amelogenin	17
8.2	Au locus D5S818	18
8.3	Au locus FGA	18

1 Introduction

La société Applied Biosystems¹ commercialise un certain nombre de machines de laboratoire utilisées en particulier pour déterminer des profils génétiques humains basés sur le polymorphisme de longueur des microsatellites. Ces machines génèrent des fichiers de données dans un format binaire appelé ABIF, pour Applied Biosystem Format qui dérive du format TIFF. Chose rare, ce format est documenté de façon transparente [2]. Il est donc relativement facile d'importer dans  ce type de données. Cette fiche en donne un exemple concret en utilisant la fonction `read.abif()`² du paquet  `seqinR` [3].

```
library(seqinr)
```

2 Importation des données

On importe deux fichiers d'exemples distribués avec le paquet `seqinR` :

```
JL0 <- read.abif(system.file("abif/2_FAC321_0000205983_B02_004.fsa", package = "seqinr"))
ECH <- read.abif(system.file("abif/2_0000206138_C01_005.fsa", package = "seqinr"))
```

Ce sont une lecture d'un profil génétique (JL0) et une échelle alléliques (ECH). Chaque objet est une liste de trois éléments :

```
names(JL0)
[1] "Header" "Directory" "Data"
```

Dans la suite nous n'utiliserons que la composante `Data` de ces deux objets. La composante `Data` est elle-même une liste de nombreux éléments qui sont détaillés en annexe page 19.

```
names(JL0$Data)
[1] "ANME.1" "CTID.1" "CTNM.1" "CTOw.1" "CTTL.1" "CpEP.1" "DATA.1"
[8] "DATA.2" "DATA.3" "DATA.4" "DATA.5" "DATA.6" "DATA.7" "DATA.8"
[15] "DATA.105" "DCHT.1" "DSam.1" "DySN.1" "Dye#.1" "DyeB.1" "DyeB.2"
[22] "DyeB.3" "DyeB.4" "DyeB.5" "DyeN.1" "DyeN.2" "DyeN.3" "DyeN.4"
[29] "DyeN.5" "DyeW.1" "DyeW.2" "DyeW.3" "DyeW.4" "DyeW.5" "EPVt.1"
[36] "EVNT.1" "EVNT.2" "EVNT.3" "EVNT.4" "GTyp.1" "HCFG.1" "HCFG.2"
[43] "HCFG.3" "HCFG.4" "InSc.1" "InVt.1" "LANE.1" "LIMS.1" "LNTD.1"
[50] "LsrP.1" "MCHN.1" "MODF.1" "MODL.1" "NAVG.1" "NLNE.1" "OfSc.1"
[57] "PANL.1" "PSZE.1" "PTYP.1" "PXLB.1" "RGNm.1" "RMXV.1" "RMDn.1"
[64] "RMDv.1" "RMDx.1" "RPrN.1" "RPrV.1" "RUND.1" "RUND.2" "RUND.3"
[71] "RUND.4" "RUNT.1" "RUNT.2" "RUNT.3" "RUNT.4" "Rate.1" "RunN.1"
[78] "SCAN.1" "SMED.1" "SMLt.1" "STYP.1" "SVER.1" "SVER.3" "SVER.4"
[85] "Satd.1" "Scal.1" "Scan.1" "SpNm.1" "StdF.1" "TUBE.1" "Tmpr.1"
[92] "UDEF.1" "User.1"
```

Dans la suite nous n'utiliserons qu'un sous ensemble de ces éléments, essentiellement ceux dont le nom commence par `DATA` qui contiennent les intensités de fluorescence mesurées au cours du temps. Nous avons dans ces données 5 fluorochromes.

```
JL0$Data[["Dye#.1"]]
[1] 5
```

1. www.appliedbiosystems.com

2. la fonction `read.abif()` a été introduite dans la version 2.0-1 du paquet `seqinR`. Les jeux de données dans la version 2.0-2. Si vous utilisez une version antérieure, vous ne pourrez pas l'utiliser.

3 Utilisation de l'étalon de taille interne

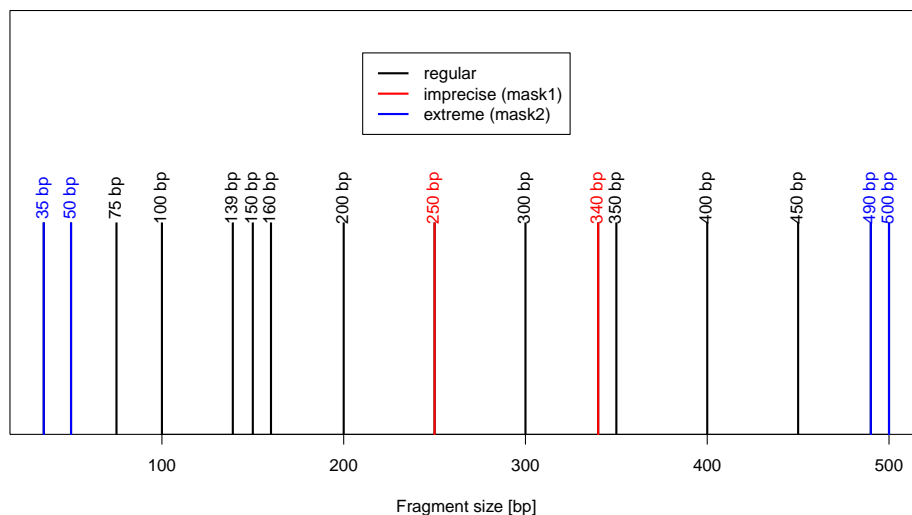
Le nom du standard de taille interne est documenté dans `StdF.1` :

```
JL0$Data[["StdF.1"]]
[1] "GS500LIZ(75-450)"
```

Il s'agit donc du `GS500LIZ(75-450)`. C'est un standard de taille qui contient 16 fragments de 35 à 500 bases. C'est une métadonnée qui n'est pas disponible directement dans les objets `JLO` et `ECH`, mais le jeu de données `gs500liz`³ contient ce qui est nécessaire.

```
data(gs500liz)
str(gs500liz)
List of 3
 $ liz : num [1:16] 35 50 75 100 139 150 160 200 250 300 ...
 $ mask1: logi [1:16] TRUE TRUE TRUE TRUE TRUE TRUE ...
 $ mask2: logi [1:16] FALSE FALSE TRUE TRUE TRUE TRUE ...
example(gs500liz, echo=FALSE)
```

GS500LIZ size standard



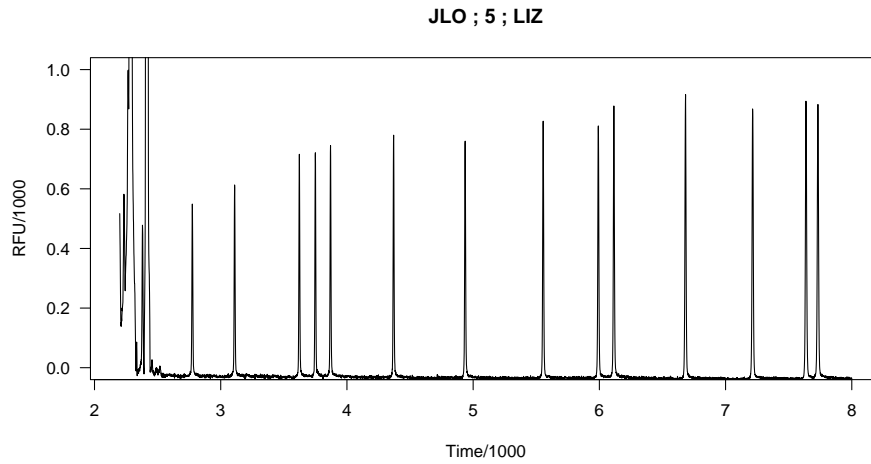
Les fragments à 250 et 340 peuvent migrer de façon anormale à cause de la présence de structures secondaires. Les deux plus petits fragment sont noyés dans le bruit dans nos expériences. On les exclura donc. On enlèvera également les fragments à 490 et 500 pour travailler finalement avec les fragments de 75 à 450 bp.

Les résultats pour l'étalon de taille interne sont lus avec le dernier fluorochrome, la fonction `plotabif()`⁴ nous permet de représenter les données facilement :

```
plotabif(JL0, chanel = 5, tmin = 2.2, tmax = 8, ylim = c(0,1))
```

3. à partir de la version 2.0-2 de `seqinR`.

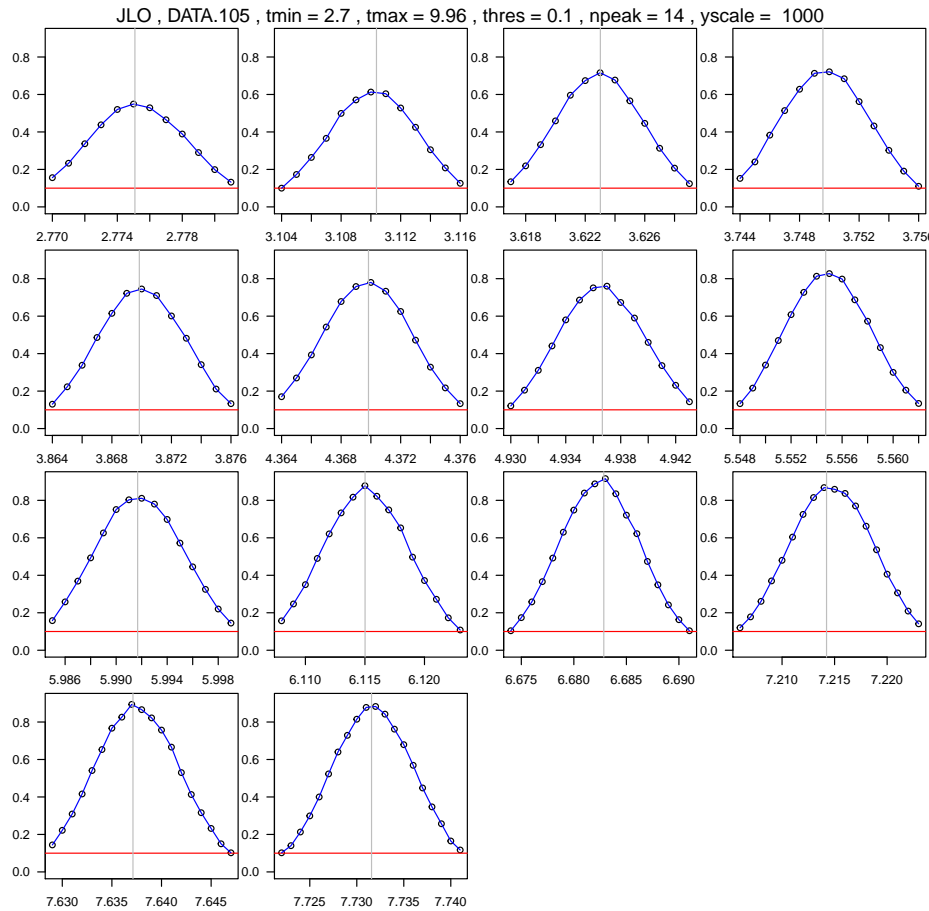
4. à partir de la version 2.0-2 de `seqinR`.



Les observations sont cohérentes avec ce qui est attendu. Il nous faut maintenant convertir les temps en paires de bases. Utilisons la fonction utilitaire `peakabif()`⁵ pour localiser les 14 pics utiles ici.

```
JLO.maxis <- peakabif(JLO, 5, npeak = 14, tmin = 2.7, thres = 0.1)
```

5. à partir de la version 2.0-2 de seqinr.



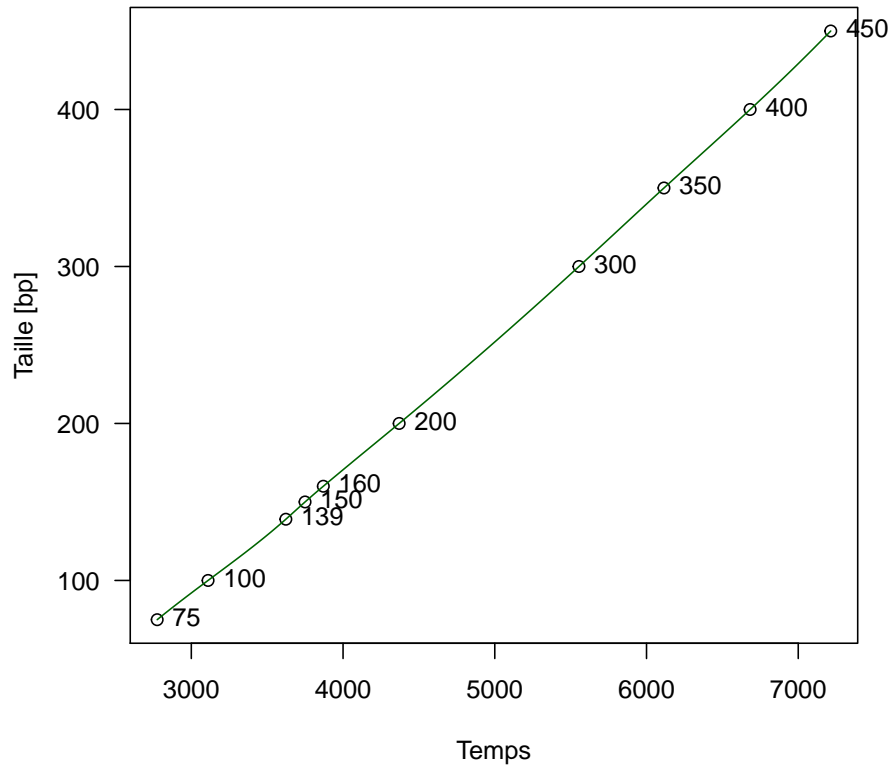
La position des 14 pics est donnée par :

```
JLO.maxis
$maxis
[1] 2775.084 3110.375 3623.017 3749.569 3869.843 4369.838 4936.663 5554.716 5991.684
[10] 6115.012 6682.851 7214.235 7637.124 7731.569
$heights
[1] 549.1890 616.3047 716.0125 724.7610 745.4973 780.9986 766.1858 828.2333 812.8576
[10] 878.0145 917.9073 870.4230 895.2032 887.1197
$surfaces
[1] 4105.003 4681.915 5348.001 5515.069 5619.746 5962.827 6170.331 7141.333 7317.085
[10] 7819.401 8831.741 9026.993 9513.500 9705.669
$baseline
[1] -0.03492254
```

La position des pics n'est pas forcément un entier parce que nous avons interpolé ici entre les observations. Calibrons avec une spline cubique :

```
lizok <- gs500liz$liz
lizok[!gs500liz$mask1 | !gs500liz$mask2] <- NA
lizok <- lizok[-c(1,2)]
y <- lizok[!is.na(lizok)]
x <- JLO.maxis$maxis[!is.na(lizok)]
plot(x, y, las = 1, xlab = "Temps", ylab = "Taille [bp]", main = "Calibration")
JLO.calibr <- splinefun(x, y)
xseq <- seq(min(x), max(x), le = 255)
lines(xseq, JLO.calibr(xseq), col = "darkgreen")
text(x,y,lizok[!is.na(lizok)], pos = 4, xpd = NA)
```

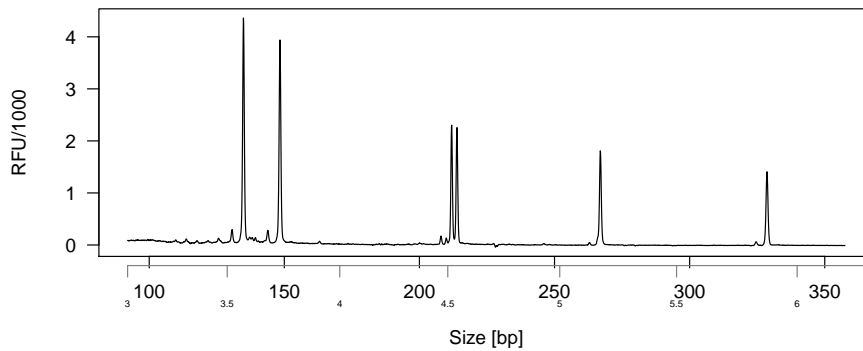
Calibration



On a donc une fonction utilitaire qui nous permet de convertir facilement le temps de migration en taille des fragments. Par exemple :

```
plotabif(JL0, 1, 3, 6.2, calibr = JL0.calibr)
```

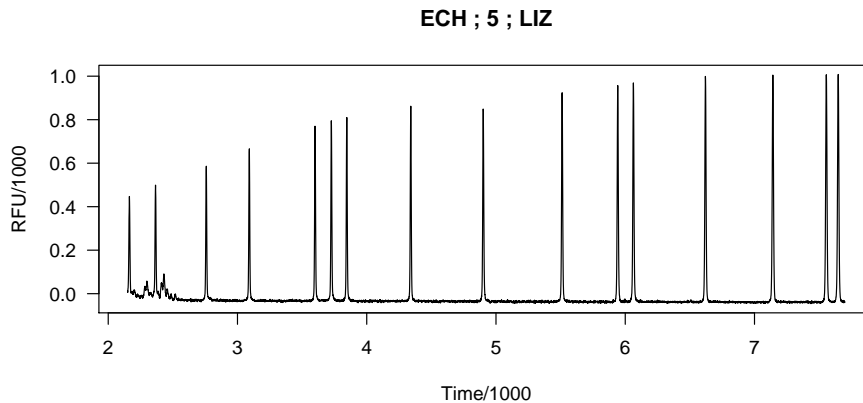
JLO ; 1 ; 6-FAM



4 Exploitation des échelles alléliques

C'est l'objet `ECH` qui contient l'échelle allélique. On regarde l'étalon interne sur le dernier fluorochrome :

```
plotabif(ECH, 5, 2.15, 7.7)
```



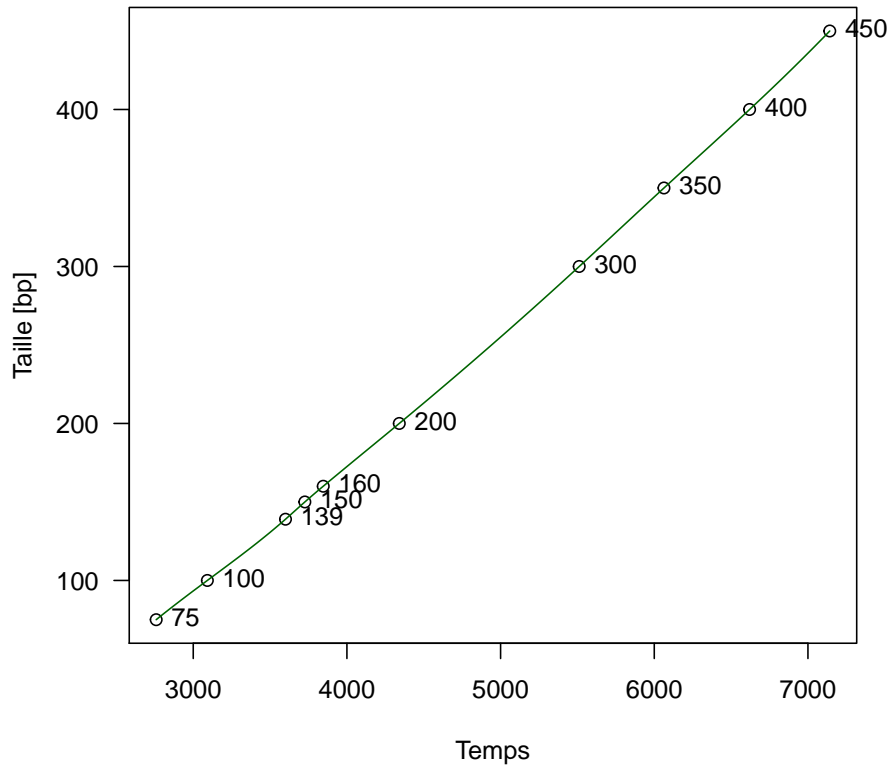
On pourrait essayer d'exploiter les pics à 35 et 50 mais on décide de n'utiliser que les mêmes pics que ceux utilisés pour l'échantillon. On utilise la même fonction pour localiser les pics.

```
(ECH$maxis <- peakabif(ECH, 5, npeak = 14, tmin = 2.7, thres = 0.1, fig = F))
$maxis
[1] 2758.838 3091.735 3600.230 3726.214 3845.808 4340.700 4901.292 5511.538 5941.890
[10] 6063.043 6620.257 7141.904 7555.415 7647.834
$heights
[1] 587.2207 670.0109 773.7607 798.0366 812.8600 865.9735 853.1504 930.1775
[9] 957.8616 969.0959 1002.9166 1005.2178 1011.9813 1008.8912
$surfaces
[1] 4406.837 5115.581 6055.417 6254.913 6392.913 6954.579 7416.246 8394.257
[9] 8945.418 9303.249 10192.742 10831.416 11424.486 11530.084
$baseline
[1] -0.03803393
```

On calibre :

```
y <- lizok[!is.na(lizok)]
x <- ECH$maxis$maxis[!is.na(lizok)]
plot(x, y, las = 1, xlab = "Temps", ylab = "Taille [bp]", main = "Calibration")
ECH.calibr <- splinefun(x, y)
xseq <- seq(min(x), max(x), le = 255)
lines(xseq, ECH.calibr(xseq), col = "darkgreen")
text(x,y,lizok[!is.na(lizok)], pos = 4, xpd = NA)
```

Calibration



Nous avons donc une fonction de calibration `ECH.calibr()` pour notre échelle allélique qui transforme les temps en longueur exprimées en paires de bases. Il nous faut donc récupérer les temps d'élution des échelles alléliques. On essaye avec la même fonction. C'est pareil sauf qu'il faut récupérer la position de beaucoup plus de pics. Encore des métadonnées non disponibles directement dans les objets `JLO` ou `ECH`. Les échelles alléliques sont décrites dans la figure 1. Le jeu de données `identifier` de `seqinR`⁶ donne le nom des allèles. La fonction `plotladder()` donne une représentation schématique de l'échelle allélique :

```
ECH6FAM.maxisbp <- plotladder(ECH, 1, ECH.calibr, tmin = 3.1, thres = 0.3, fig = FALSE)
```

6. à partir de la version 2.0-2

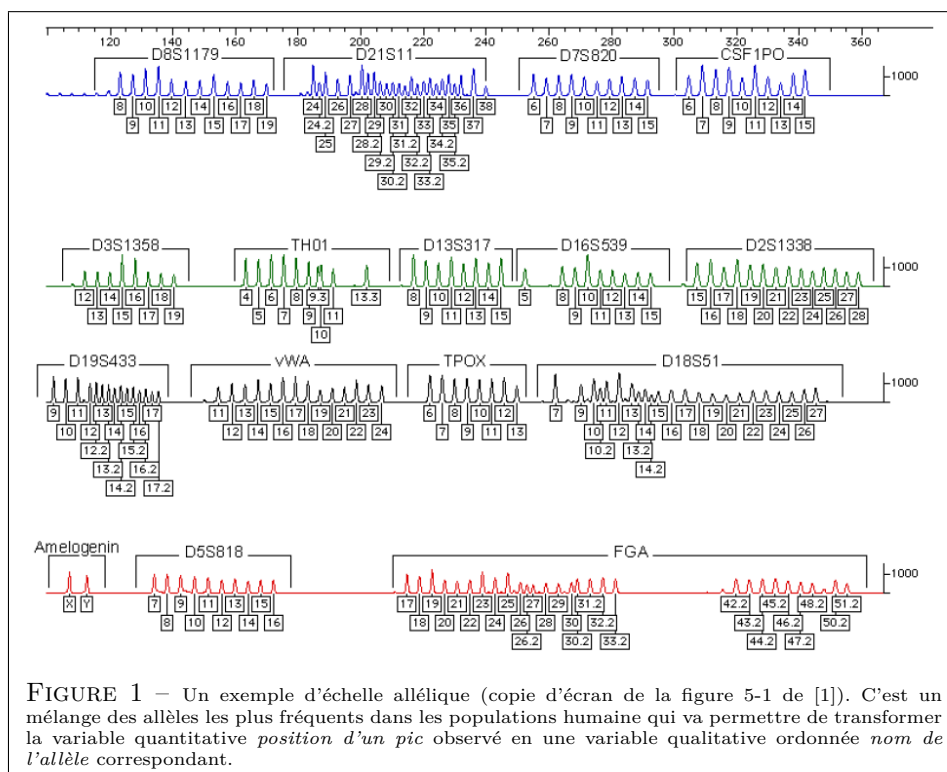
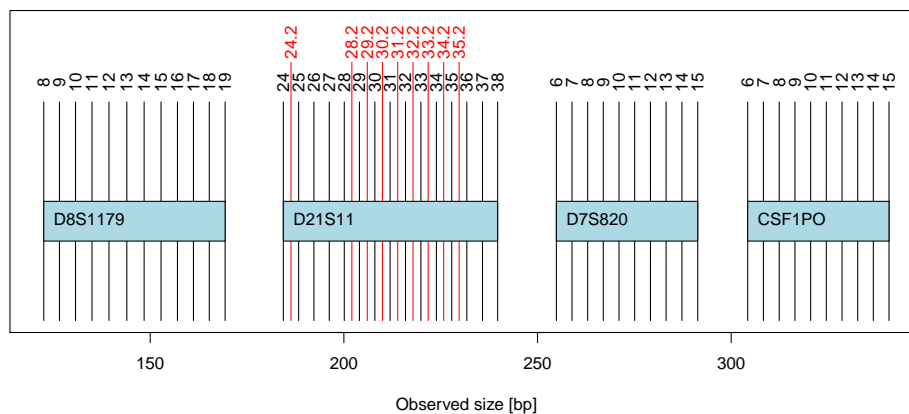


FIGURE 1 – Un exemple d'échelle allélique (copie d'écran de la figure 5-1 de [1]). C'est un mélange des allèles les plus fréquents dans les populations humaine qui va permettre de transformer la variable quantitative *position d'un pic* observé en une variable qualitative ordonnée *nom de l'allèle* correspondant.

Observed allelic ladder for Run_3130xl_2008-11-06_14-03_5134
with fluorochrome 6-FAM



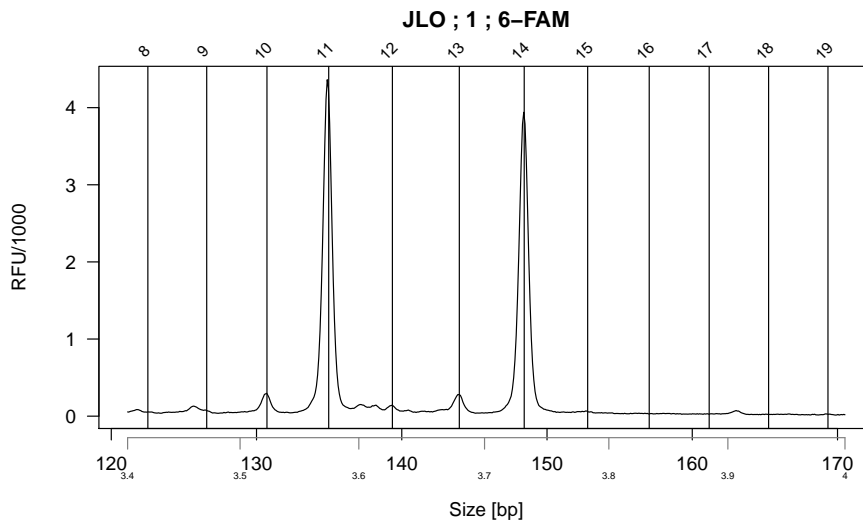
5 Confrontation avec l'échelle allélique

Il s'agit de superposer les résultats de l'échantillon avec ceux de l'échelle allélique, les deux ayant été au préalable convertis avec le standard de taille interne en paires de bases. C'est ceci qui nous permettra d'identifier les allèles. Pour faire une identification précise il faut travailler locus par locus. Dans la

série de ceux qui sont lus avec le fluorochrome 6-FAM, nous attaquons avec le locus D8S1179.

5.1 Au locus D8S1179

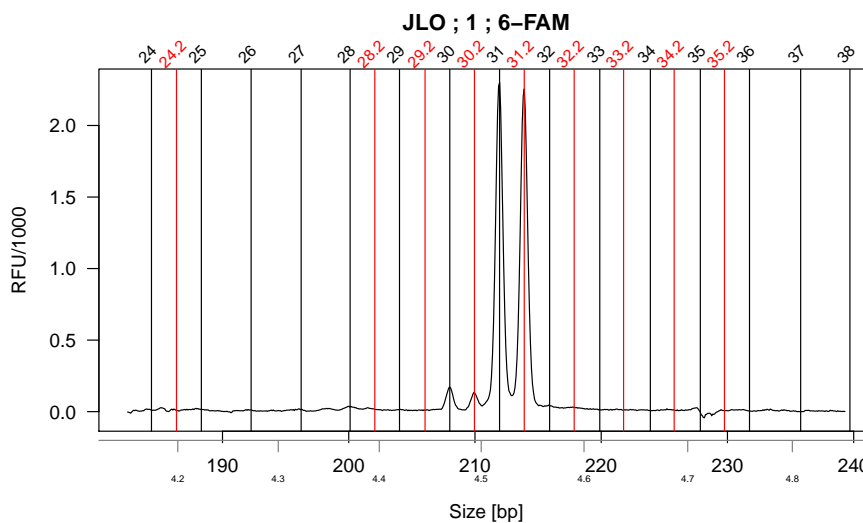
```
plotabif(JL0, 1, 3.4, 4, calibr = JL0.calibr, ladder.bp = ECH6FAM.maxisbp )
```



Donc il n'y a pas photo, cet individu est hétérozygote (11, 14) pour ce locus.

5.2 Au locus D21S11

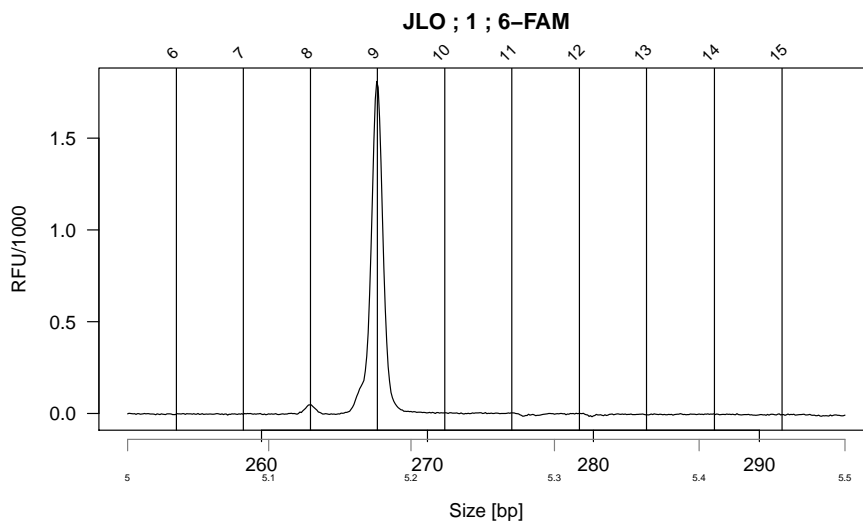
```
plotabif(JL0, 1, 4.15, 4.85, calibr = JL0.calibr, ladder.bp = ECH6FAM.maxisbp )
```



Intéressant, présence d'un microvariant. L'individu JLO est hétérozygote (31, 31.2) pour ce locus. À noter la taille des "stutter" ici, celui du micro-variant est plus faible.

5.3 Au locus D7S820

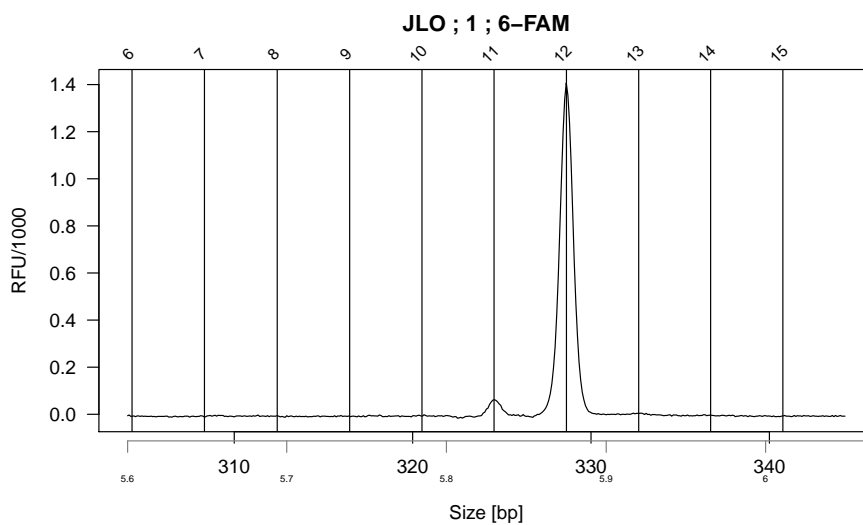
```
plotabif(JLO, 1, 5, 5.5, calibr = JLO.calibr, ladder.bp = ECH6FAM.maxisbp )
```



Homozygote 9 sur ce coup là.

5.4 Au locus CSF1PO

```
plotabif(JLO, 1, 5.6, 6.05, calibr = JLO.calibr, ladder.bp = ECH6FAM.maxisbp )
```

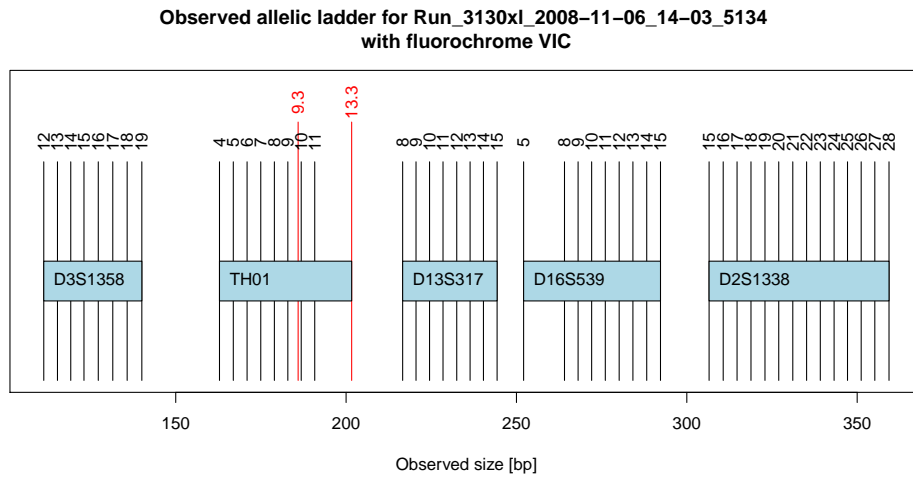


Homozygote 12.

6 Pour le fluorochrome VIC

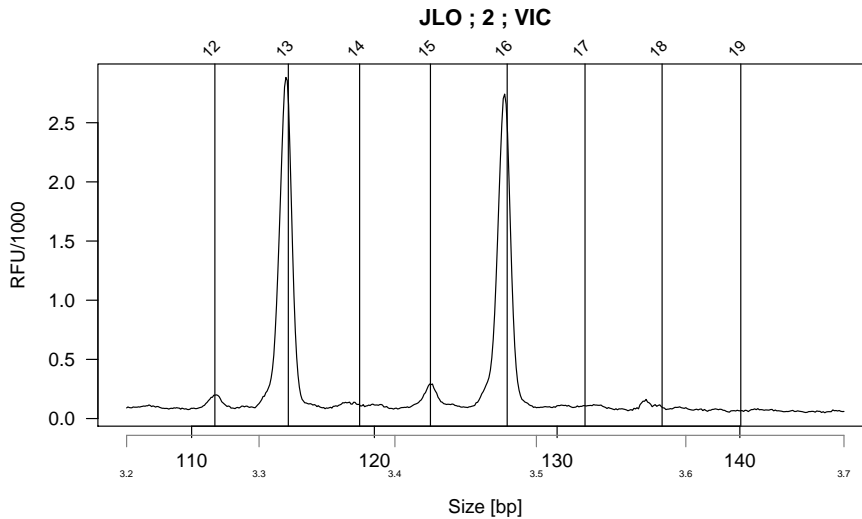
Voici à quoi ressemble l'échelle allélique :

```
ECHVIC.maxisbp <- plotladder(ECH, 2, ECH.calibr, tmin = 3.1, thres = 0.35, fig = FALSE)
```



6.1 Au locus D3S1358

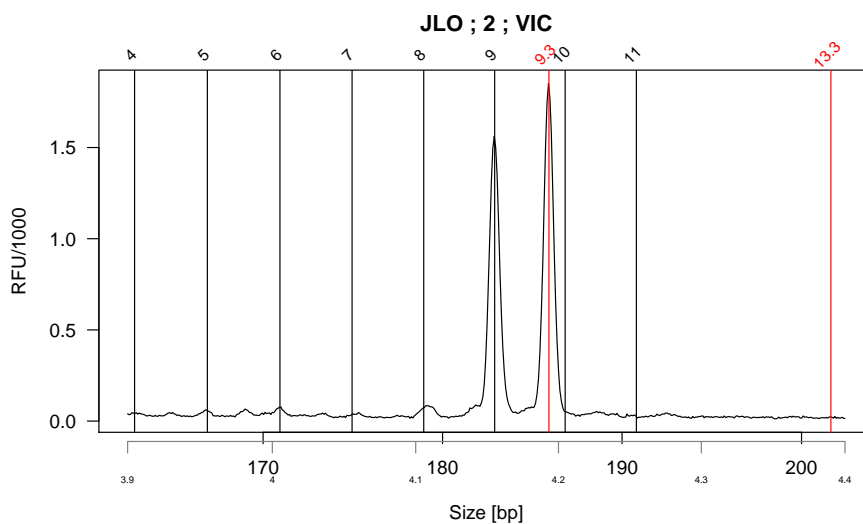
```
plotabif(JLO, 2, 3.2, 3.7, calibr = JLO.calibr, ladder.bp = ECHVIC.maxisbp )
```



Hétérozygote 13, 16.

6.2 Au locus TH01

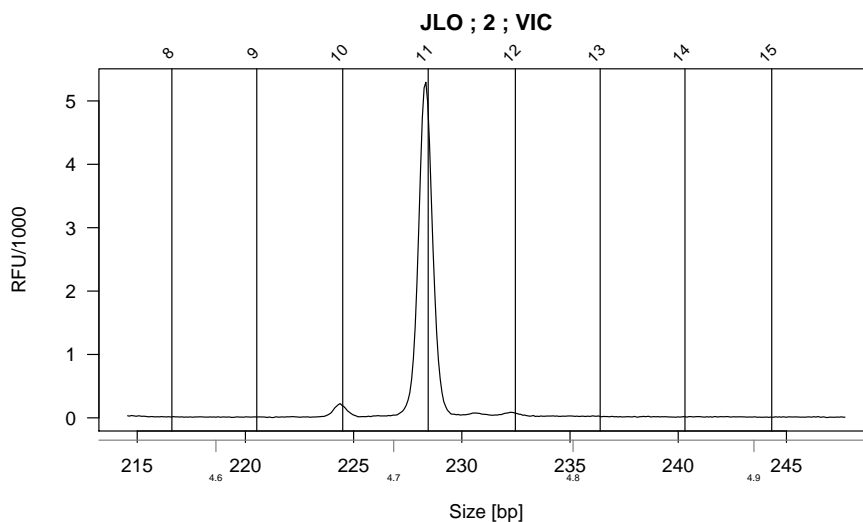
```
plotabif(JLO, 2, 3.9, 4.4, calibr = JLO.calibr, ladder.bp = ECHVIC.maxisbp )
```



Hétérozygote 9, 9.3

6.3 Au locus D13S317

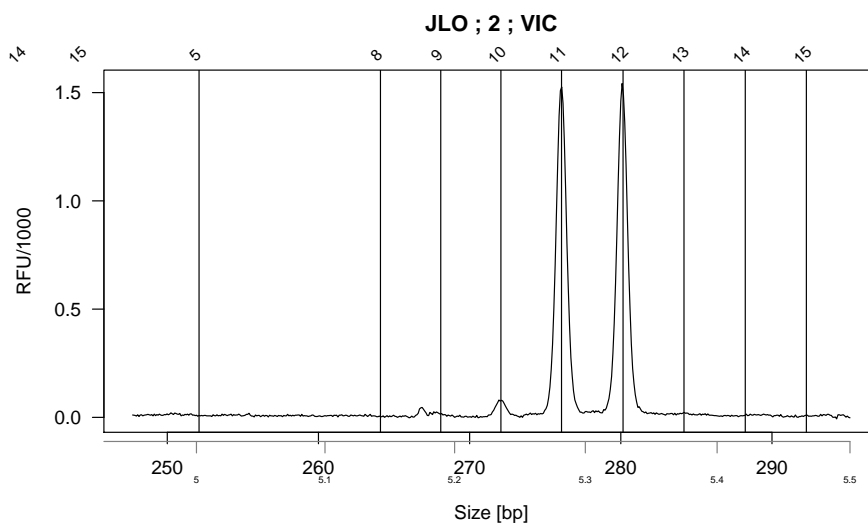
`plotabif(JL0, 2, 4.55, 4.95, calibr = JL0.calibr, ladder.bp = ECHVIC.maxisbp)`



Homozygote 11.

6.4 Au locus D16S539

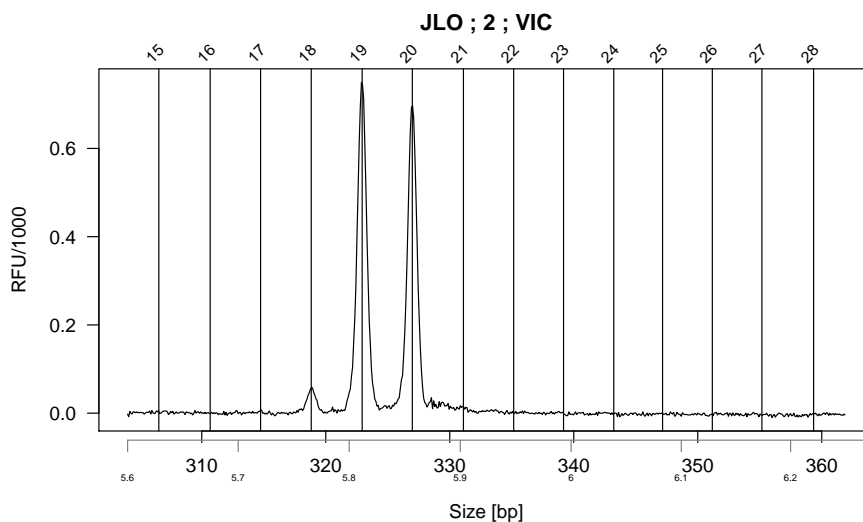
`plotabif(JL0, 2, 4.95, 5.5, calibr = JL0.calibr, ladder.bp = ECHVIC.maxisbp)`



Hétérozygote 11, 12.

6.5 Au locus D2S1338

```
plotabif(JLO, 2, 5.6, 6.25, calibr = JLO.calibr, ladder.bp = ECHVIC.maxisbp )
```

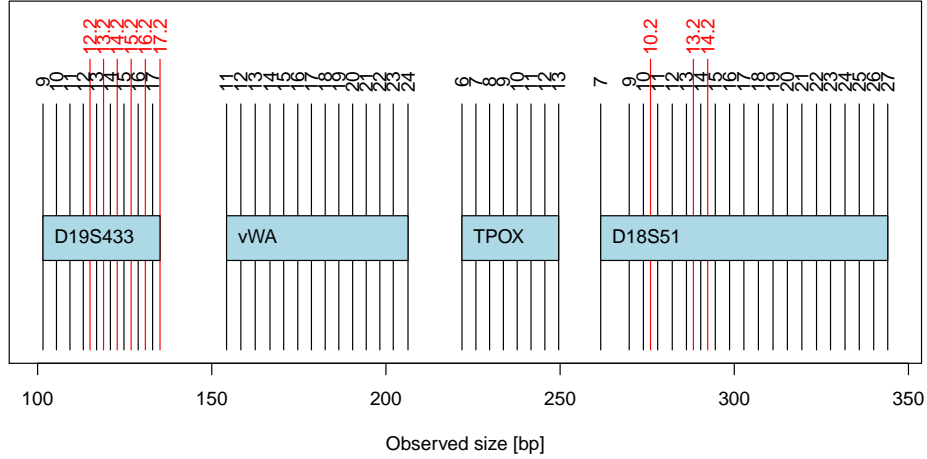


Hétérozygote 19, 20.

7 Pour le fluorochrome NED

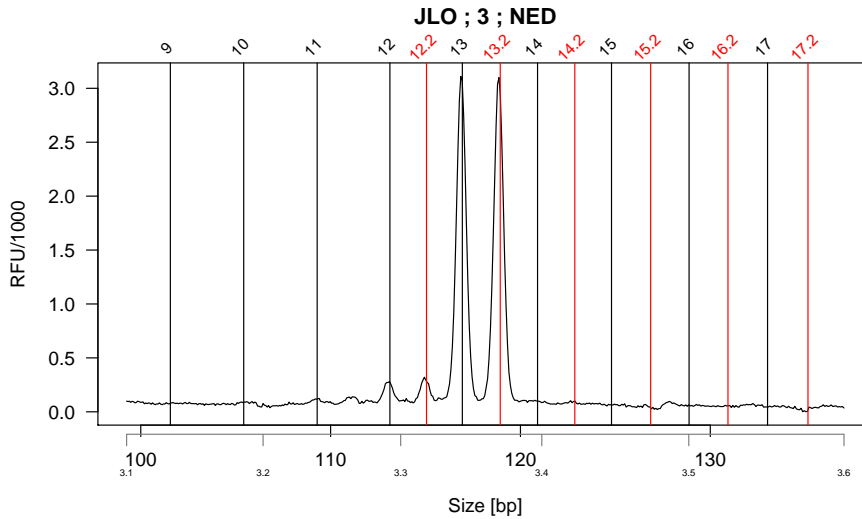
```
ECHNED.maxisbp <- plotladder(ECH, 3, ECH.calibr, tmin = 3.1, thres = 0.2, fig = FALSE)
```

**Observed allelic ladder for Run_3130xl_2008-11-06_14-03_5134
with fluorochrome NED**



7.1 Au locus D19S433

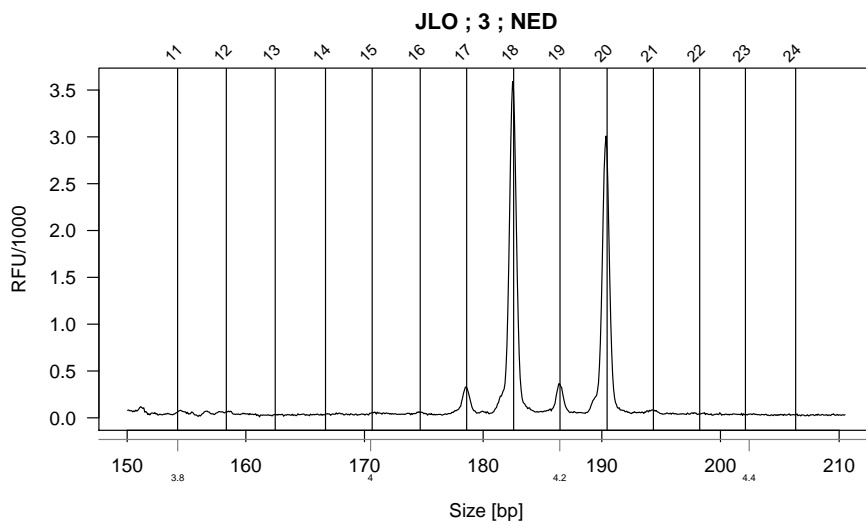
`plotabif(JL0, 3, 3.1, 3.6, calibr = JL0.calibr, ladder.bp = ECHNED.maxisbp)`



Hétérozygote 13, 13.2.

7.2 Au locus vWA

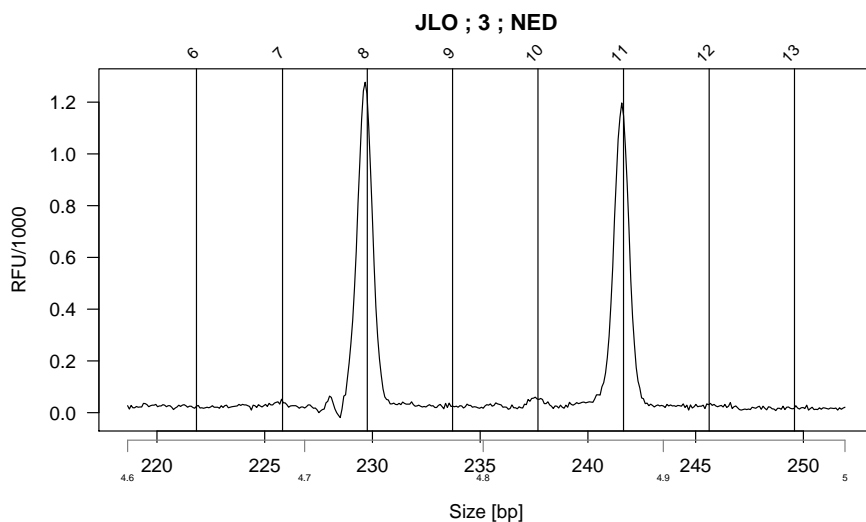
`plotabif(JL0, 3, 3.75, 4.5, calibr = JL0.calibr, ladder.bp = ECHNED.maxisbp)`



Hétérozygote 18,20.

7.3 Au locus TPOX

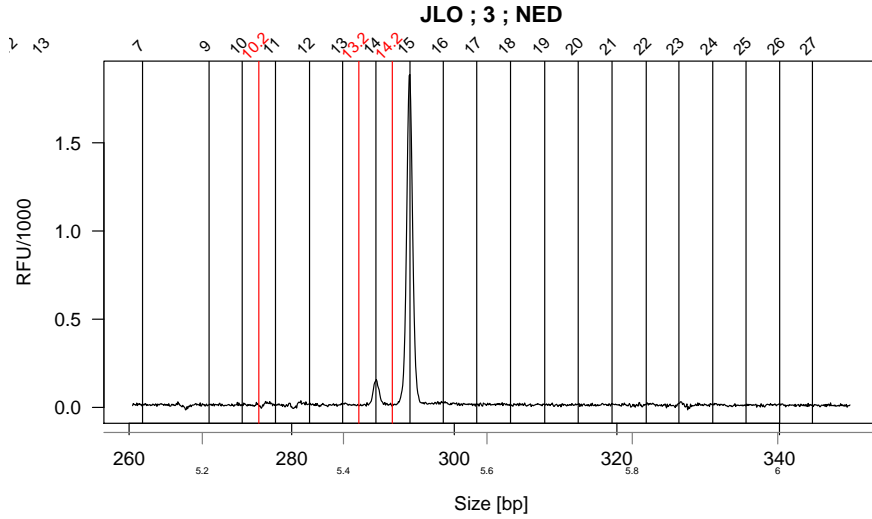
```
plotabif(JLO, 3, 4.6, 5, calibr = JLO.calibr, ladder.bp = ECHNED.maxisbp )
```



Hétérozygote 8, 11.

7.4 Au locus D18S51

```
plotabif(JLO, 3, 5.1, 6.1, calibr = JLO.calibr, ladder.bp = ECHNED.maxisbp )
```

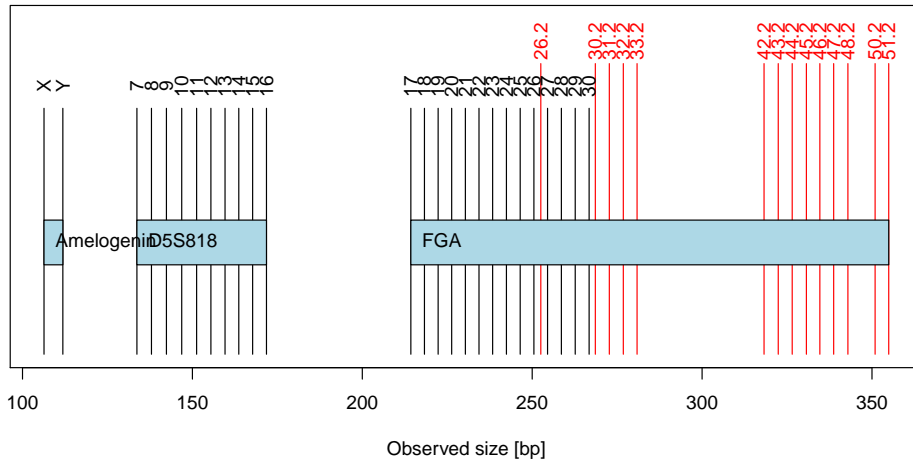



Homozygote 15.

8 Pour le fluorochrome PET

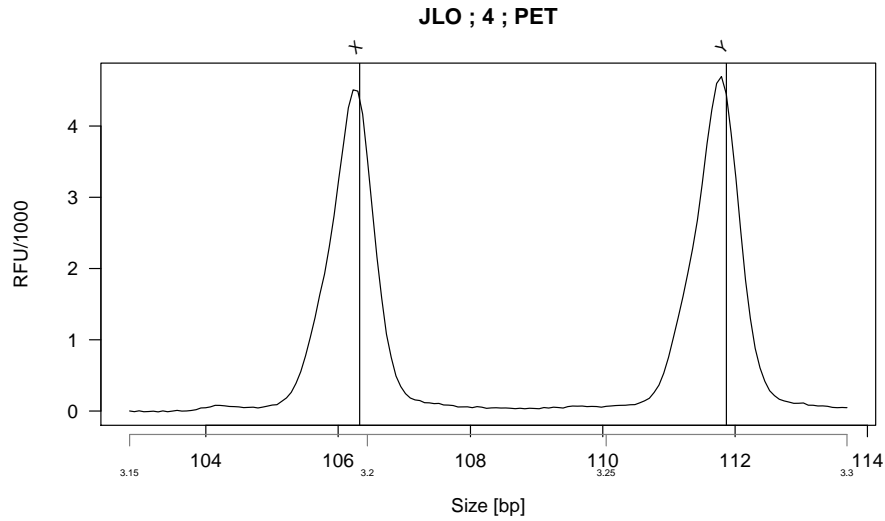
```
ECHPET.maxisbp <- plotladder(ECH, 4, ECH.calibr, tmin = 3.1, thres = 0.2, fig = FALSE)
```

**Observed allelic ladder for Run_3130xl_2008-11-06_14-03_5134
with fluorochrome PET**



8.1 Au locus Amelogenin

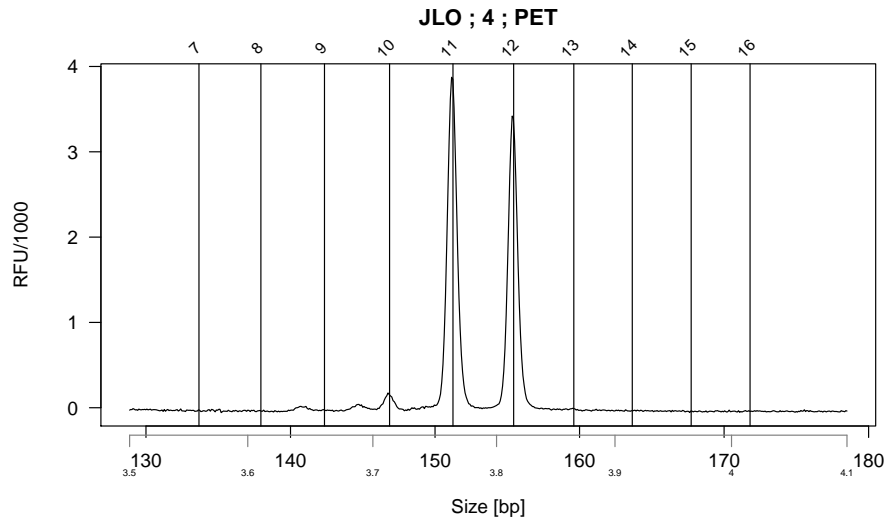
```
plotabif(JLO, 4, 3.15, 3.3, calibr = JLO.calibr, ladder.bp = ECHPET.maxisbp )
```



C'est un mâle.

8.2 Au locus D5S818

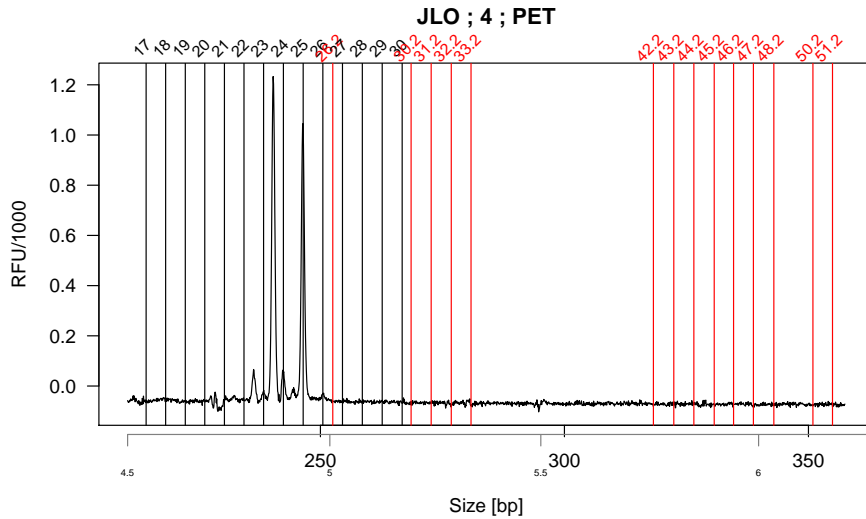
```
plotabif(JLO, 4, 3.5, 4.1, calibr = JLO.calibr, ladder.bp = ECHPET.maxisbp )
```



Hétérozygote 11, 12.

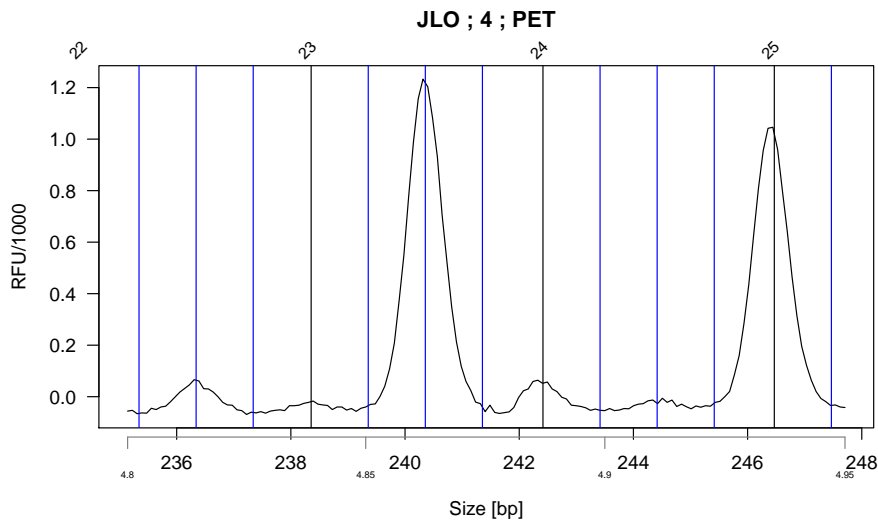
8.3 Au locus FGA

```
plotabif(JLO, 4, 4.5, 6.2, calibr = JLO.calibr, ladder.bp = ECHPET.maxisbp )
```



Il y a un allèle 25 mais l'autre est hors échelle allélique. On zoome un peu et on ajoute les bp.

```
par(plotabif(JLO, 4, 4.8, 4.95, calibr = JLO.calibr, ladder.bp = ECHPET.maxisbp ))
x <- ECHPET.maxisbp
abline(v=c(x+1, x+2, x+3),col="blue")
```



C'est donc un hétérozygote 23.2, 25.

9 Annexe : description des données

Matériel et méthodes

Cette section a été générée automatiquement à partir des données disponibles dans les objets JLO et ECH. On peut la passer en première lecture, mais

elle donne un résumé de ce qui suit. Le nom et le numéro de série de l'instrument utilisé pour effectuer les électrophorèses capillaires était 3130xl-1473-021, c'est un modèle 3100. Il possède 16 capillaires. Les numéros de capillaires étaient 4 et 5, les noms des tubes étaient B2 et C1, les tags d'identification étaient a2de3f04ac0211dd9a0600065bddb0ed et a2de3efdac0211dd9a0600065bddb0ed, les noms des échantillons étaient FAC321_0000205983 et 0000206138, les types des échantillons étaient `Sample` et `Allelic Ladder`, pour JLO et ECH, respectivement. L'expérience a été initiée en 2008, le 6 November à 14 h 46 mn 57 s. L'expérience a été terminée en 2008, le 6 November à 15 h 28 mn 37 s. Au total, 9960 mesures ont été faites pour chacun des 16 capillaires. Le nom du jeu de fluorochromes utilisé était G5, c'était un jeu de 5 fluorochromes. Les noms des fluorochromes utilisés étaient 6-FAM, VIC, NED, PET et LIZ. Les longueurs d'ondes exprimées en nm utilisées pour la lecture étaient de 522, 554, 575, 595 et 655, respectivement. La puissance du laser pour l'excitation était de 15000 microWatts. Le détecteur était à une distance de 36 cm de la source. Pour l'électrophorèse la température était de 60 °C, le voltage de 15000 volts et le gel de type POP4 . Le numéro du lot du polymère était 0802043 et sa date d'expiration Aug 22, 2008. Les paramètres pour l'électro-injection étaient de 10 s et 3000 volts tels que définis dans le fichier `module_3kV_10s`. Les plaques utilisées étaient de type 96-Well.

```
names(JLO$Data)
[1] "ANME.1" "CTID.1" "CTNM.1" "CTOw.1" "CTTL.1" "CpEP.1" "DATA.1"
[8] "DATA.2" "DATA.3" "DATA.4" "DATA.5" "DATA.6" "DATA.7" "DATA.8"
[15] "DATA.105" "DCHT.1" "DSam.1" "DySN.1" "Dye#.1" "DyeB.1" "DyeB.2"
[22] "DyeB.3" "DyeB.4" "DyeB.5" "DyeN.1" "DyeN.2" "DyeN.3" "DyeN.4"
[29] "DyeN.5" "DyeW.1" "DyeW.2" "DyeW.3" "DyeW.4" "DyeW.5" "EPVt.1"
[36] "EVNT.1" "EVNT.2" "EVNT.3" "EVNT.4" "GTyp.1" "HCFG.1" "HCFG.2"
[43] "HCFG.3" "HCFG.4" "InSc.1" "InVt.1" "LANE.1" "LIMS.1" "LNTD.1"
[50] "LsrP.1" "MCHN.1" "MODF.1" "MODL.1" "NAVG.1" "NLNE.1" "OfSc.1"
[57] "PANL.1" "PSZE.1" "PTYP.1" "PXLB.1" "RGNm.1" "RMXV.1" "RMdN.1"
[64] "RmV.1" "RmX.1" "RPrN.1" "RPrV.1" "RUND.1" "RUND.2" "RUND.3"
[71] "RUND.4" "RUNT.1" "RUNT.2" "RUNT.3" "RUNT.4" "Rate.1" "RunN.1"
[78] "SCAN.1" "SMED.1" "SMLt.1" "STYP.1" "SVER.1" "SVER.3" "SVER.4"
[85] "Satd.1" "Scal.1" "Scan.1" "SpNm.1" "StdF.1" "TUBE.1" "Tmpr.1"
[92] "UDEF.1" "User.1"
```

```
JLO$Data[["ANME.1"]]
[1] "method FTA Id v3"
```

Nom de la méthode d'analyse utilisée pour GeneMapper.

Container related

```
JLO$Data[["CTID.1"]]
[1] "JAA313"
JLO$Data[["CTNM.1"]]
[1] "JAA313"
```

Tag d'identification (container identification) et nom d'identification (container name), en général identiques.

```
JLO$Data[["CTOw.1"]]
[1] "[%currentuser%]"
```

Container Owner.

```
JLO$Data[["CTTL.1"]]
[1] "Comment:"
```

Un titre ou commentaire pour la manip.

Est-ce une machine capillaire ?

```
JLO$Data[["CpEP.1"]]  
[1] "\001"  
class(JLO$Data[["CpEP.1"]])  
[1] "character"
```

Is Capillary Machine? *devrait être booléen, non. Donné comme byte dans la doc*

Les DATA

```
str(JLO$Data[["DATA.1"]])  
int [1:9960] -1 -4 -4 -2 -2 -3 -7 -3 -3 -1 ...
```

De DATA.1 à DATA.4 on a les mesures de fluorescence pour chaque fluorochrome. Le standard de taille interne est dans DATA.105. DATA.5 donne le voltage (decaVolts), DATA.6 l'intensité du courant (milliAmps), DATA.7 la puissance (milliWatts), DATA.8 la température en Celcius. La fréquence d'échantillonnage n'est pas la même pour les mesures physiques et la fluorescence.

Divers

```
JLO$Data[["DCHT.1"]]  
[1] 0  
JLO$Data[["DSam.1"]]  
[1] 1
```

Detector Cell Heater temperature et Downsampling factor. *C'est quoi ?*

Les fluorochromes

```
JLO$Data[["DySN.1"]]  
[1] "G5"
```

Nom du jeu de fluorochromes.

```
JLO$Data[["Dye#.1"]]  
[1] 5
```

Nombre total de fluorochromes utilisés.

```
JLO$Data[["DyeB.1"]]  
[1] " "
```

Toujours un espace pour les échantillons.

```
JLO$Data[["DyeN.1"]]  
[1] "6-FAM"  
JLO$Data[["DyeN.2"]]  
[1] "VIC"  
JLO$Data[["DyeN.3"]]  
[1] "NED"
```

```
JLO$Data[["DyeN.4"]]
[1] "PET"
JLO$Data[["DyeN.5"]]
[1] "LIZ"
```

Le nom des fluorochromes utilisés.

```
JLO$Data[["DyeW.1"]]
[1] 522
```

Les longueur d'onde utilisées pour les fluorochromes (nm)

Divers

```
JLO$Data[["EPVt.1"]]
[1] 15000
```

Le réglage du voltage pour l'électrophorèse en volts.

Chaines de caractères des événements

```
JLO$Data[["EVNT.1"]]
[1] "Run Started"
JLO$Data[["EVNT.2"]]
[1] "Run Stopped"
JLO$Data[["EVNT.3"]]
[1] "Collection Started"
JLO$Data[["EVNT.4"]]
[1] "Collection Stopped"
```

On ne voit pas trop à quoi ça sert.

Type du gel utilisé

```
JLO$Data[["GTyp.1"]]
[1] "POP4"
```

Hardware

```
JLO$Data[["HCFG.1"]]
[1] "CE"
JLO$Data[["HCFG.2"]]
[1] "31XX"
JLO$Data[["HCFG.3"]]
[1] "3130x1"
JLO$Data[["HCFG.4"]]
[1] "UnitID=7;CPUBoard=EPCU500;ArraySize=16;SerialNumber=1473-021;"
```

Injection

```
JLO$Data[["InSc.1"]]
[1] 10
JLO$Data[["InVt.1"]]
[1] 3000
```

Temps (s) et voltage (Volts) de l'injection.

Capillaire

```
JLO$Data[["LANE.1"]]
[1] 4
```

Numéro du capillaire utilisé.

ID de l'échantillon

```
JLO$Data[["LIMS.1"]]
[1] "a2de3f04ac0211dd9a0600065bddb0ed"
```

C'est de l'hexadécimal visiblement.

Distance au détecteur en cm

```
JLO$Data[["LNTD.1"]]
[1] 36
```

Puissance du laser en microWatts

```
JLO$Data[["LsrP.1"]]
[1] 15000
```

Nom de l'instrument et numéro de série

```
JLO$Data[["MCHN.1"]]
[1] "3130x1-1473-021"
```

Data collection module file

```
JLO$Data[["MODF.1"]]
[1] "module_3kV_10s"
```

C'est le nom du fichier qui contient les paramètres de la manip, par exemple le temps d'injection et le voltage utilisé.

Numéro du modèle

```
JLO$Data[["MODL.1"]]  
[1] "3100"
```

Pixel averaged per lane

```
JLO$Data[["NAVG.1"]]  
[1] 1
```

Nombre de capillaires

```
JLO$Data[["NLNE.1"]]  
[1] 16
```

Pics hors échelle

```
str(JLO$Data[["OfSc.1"]])  
int [1:702] 1624 1625 1626 1627 1628 1629 1630 1631 1645 1646 ...
```

GeneMapper

```
JLO$Data[["PANL.1"]]  
[1] "Identifiler_v2"
```

Taille et type de la plaque

```
JLO$Data[["PSZE.1"]]  
[1] 96  
JLO$Data[["PTYP.1"]]  
[1] "96-Well"
```

Pixel Bin size

```
JLO$Data[["PXLB.1"]]  
[1] 3
```

Nom du groupe de résultats

```
JLO$Data[["RGNm.1"]]  
[1] "RG96_Individus"
```


Run Module

```
JLO$Data[["RMXV.1"]]
[1] "1"
JLO$Data[["RMdN.1"]]
[1] "module_3kV_10s"
JLO$Data[["RMdV.1"]]
[1] "1"
JLO$Data[["RMdX.1"]]
[1] "<?xml version=\"1.0\"?>\n<Run-Module-Instance><Name>module_3kV_10s</Name><Version>2</Version><Method-Name>
```

Run Protocol

```
JLO$Data[["RPrN.1"]]
[1] "Y_Identifier_3kV_10s"
JLO$Data[["RPrV.1"]]
[1] "1"
```

Données temporelles

Les RUND donnent les dates, les RUNT donnent les heures. Il y a une date de départ de la manip et une date de départ de la collecte des données. L'heure de début et de fin de collecte des données est ici :

```
unlist(JLO$Data[["RUNT.3"]])
  hour minute second hsecond
   15     1     48         0
unlist(JLO$Data[["RUNT.4"]])
  hour minute second hsecond
   15    28     28         0
```

Soit 26 minutes et 40 secondes, donc 1566 secondes.
L'heure de début et de fin de la manip est ici :

```
unlist(JLO$Data[["RUNT.1"]])
  hour minute second hsecond
   14    46     57         0
unlist(JLO$Data[["RUNT.2"]])
  hour minute second hsecond
   15    28     37         0
```

Soit 42 minutes et 37 secondes, soit 2557 secondes.
Je ne vois pas la relation avec les 9960 valeurs des données brutes.

Fréquence d'échantillonnage

```
JLO$Data[["Rate.1"]]
[1] 00 00 00 00 00 00 00 00 9e 00 00 00 01 20
```

Il faudrait savoir comment c'est codé.

Nom de la manip

```
JLO$Data[["RunN.1"]]  
[1] "Run_3130x1_2008-11-06_14-03_5134"
```

Nombre de lectures

```
JLO$Data[["SCAN.1"]]  
[1] 9960
```

C'est cohérent avec la longueur des vecteurs de données brutes. On utilisait aussi dans le temps `Scan.1`, mais c'est gardé uniquement pour compatibilité.

```
JLO$Data[["Scan.1"]]  
[1] 9960
```

Date d'expiration et numéro du lot de polymère

```
JLO$Data[["SMED.1"]]  
[1] "Aug 22, 2008"  
JLO$Data[["SMLt.1"]]  
[1] "0802043"
```

Heu ? C'est normal que la date d'expiration soit dépassée au moment du début de la manip ?

```
unlist(JLO$Data[["RUND.1"]])  
year month day  
2008 11 6
```

SoftWare

Type de données pour GeneMapper :

```
JLO$Data[["STYP.1"]]  
[1] "Sample"  
ECH$Data[["STYP.1"]]  
[1] "Allelic Ladder"
```

Autres données logiciel :

```
JLO$Data[["SVER.1"]]  
[1] "3.0"  
JLO$Data[["SVER.3"]]  
[1] "6286200-02 6286202-01 6286103-04 6258005-00"  
JLO$Data[["SVER.4"]]  
[1] "Fragment Format 2.0"
```

Lectures saturées

C'est la même chose que `OfSc.1`, pourquoi est-ce dupliqué ?

```
all(JLO$Data[["OfSc.1"]] == JLO$Data[["Satd.1"]])
[1] TRUE
all(JLO$Data[["OfSc.1"]] == JLO$Data[["Satd.1"]])
[1] TRUE
all(ECH$Data[["OfSc.1"]] == ECH$Data[["Satd.1"]])
[1] TRUE
all(ECH$Data[["OfSc.1"]] == ECH$Data[["Satd.1"]])
[1] TRUE
```

Facteur d'échelle pour les couleurs

```
JLO$Data[["Scal.1"]]
[1] 9.10844e-44
```

Cette valeur est suspecte.

GenScan

Le nom de l'échantillon utilisé pour construire les fichiers `*.fsa`.

```
JLO$Data[["SpNm.1"]]
[1] "FAC321_0000205983"
JLO$Data[["SpNm.1"]]
[1] "FAC321_0000205983"
ECH$Data[["SpNm.1"]]
[1] "0000206138"
ECH$Data[["SpNm.1"]]
[1] "0000206138"
```

Le standard de taille interne

Le nom du fichier qui contient la taille des fragments de taille connue pour l'étalonnage interne.

```
JLO$Data[["StdF.1"]]
[1] "GS500LIZ(75-450)"
```

Le nom du tube

```
JLO$Data[["TUBE.1"]]
[1] "B2"
JLO$Data[["TUBE.1"]]
[1] "B2"
ECH$Data[["TUBE.1"]]
[1] "C1"
ECH$Data[["TUBE.1"]]
[1] "C1"
```

Le nom du tube utilisé pour construire les fichiers `*.fsa`.

Température

Réglage de la température en Celcius.

```
ECH$Data[["Tmpr.1"]]
[1] 60
```

Données utilisateur

```
JLO$Data[["UDEF.1"]]
[1] "FAC321"
ECH$Data[["UDEF.1"]]
NULL
JLO$Data[["User.1"]]
[1] "[%currentuser%]"
```

Références

- [1] Anonymous. *AmpF ℓ STR[®] Identifiler[®] PCR Amplification Kit. User's Manual*. Applied Biosystems, Foster City, CA, USA, 2006. PN 4323291D.
- [2] Anonymous. Applied biosystems genetic analysis data file format. Technical report, Applied Biosystems, 2006.
- [3] D. Charif and J.R. Lobry. SeqinR 1.0-2 : a contributed package to the R project for statistical computing devoted to biological sequences retrieval and analysis. In H.E. Roman U. Bastolla, M. Porto and M. Vendruscolo, editors, *Structural approaches to sequence evolution : Molecules, networks, populations*, Biological and Medical Physics, Biomedical Engineering, pages 207–232. Springer Verlag, New York, USA, 2007. ISBN 978-3-540-35305-8.