UNIVERSITAT ROVIRA i VIRGILI

Creació d'un programa per a perfilar els STRs d'una mostra a partir d'un arxiu amb el seu genoma complert

Arnau Garcés Baldó

TREBALL DE FINAL DEL MÀSTER EN GENÈTICA, FÍSICA I QUÍMICA FORENSE

Realitzat a: URV, grup de recerca en Quimioinformàtica i Nutrició

Tutor acadèmic/professional:

Santiago García Vallvé, santiago.garcia-vallve@urv.cat

Setembre de 2022

Resum

En els últims anys, els mètodes de seqüenciació massiva han anat evolucionant fins a abaratir els costos a preus que son assequibles fins i tot per al públic general. Tot i així, en genètica forense es segueixen fent els perfilats de la mateixa manera que en els últims anys, amb electroforesi capil·lar, que és un mètode a cegues que només permet distingir llargades diferents.

En aquest treball, es proposa la creació d'un programa que permeti el perfilat per STRs d'un individu a partir d'un arxiu amb la seva seqüència completa, de manera que segueixi sent compatible amb els perfils generats fins ara que estan emmagatzemats a les bases de dades policials.

Paraules clau: STR, seqüenciació, perfilat.

Abstract

In the last years, massive sequencing has been evolving to the point where analysis prices are affordable even to the common public. Nonetheless, in genetic forensics, DNA profiling still uses the same method as in the last few years, with the use of capillary electrophoresis, which only distinguishes between different lengths.

This work pretends to create a program capable of doing the STR profiling of a person using their complete filed sequence, making the output compatible with the profiles done to the date with the old method saved in the police databases.

Keywords: STR, sequencing, DNA profiling.

Índex

1.- Introducció

1.1.- Genètica forense i els STRs

Les mostres biològiques i la seva identificació i individualització sempre han estat claus importants en les investigacions criminals, ja que saber la procedència d'una mostra ens pot portar a deduir si un sospitós va tocar l'arma d'un crim, o saber a qui pertanyen els fluids trobats en una víctima de violació (1). No cal dir, doncs, que l'aparició de les primeres eines genètiques van suposar un gran avanç en aquest camp, i van forjar el que avui dia coneixem com "genètica forense".

Ja l'any 1985, Alec Jeffreys va utilitzar per primer cop regions variables del gen de la mioglobina per tal de discriminar individus (2,3). Aquesta regió conté una seqüència de 33 nucleòtids que es repeteix, fent que amb una sonda marcada i enzims de restricció es pugui obtenir un fragment amb una variació notable de pes entre individus, tècnica anomenada *Restriction Fragment Lenght Polimorphism* o RFLP.

Més tard, amb la introducció de les tècniques de PCR el 1986, es va fer cada cop més senzill analitzar regions amb nombres de repetició variables fins i tot quan la concentració d'ADN en la mostra era molt baixa. Això, junt al descobriment de més polimorfismes, ha portat a l'anàlisi d'STRs tal i com el coneixem avui dia.

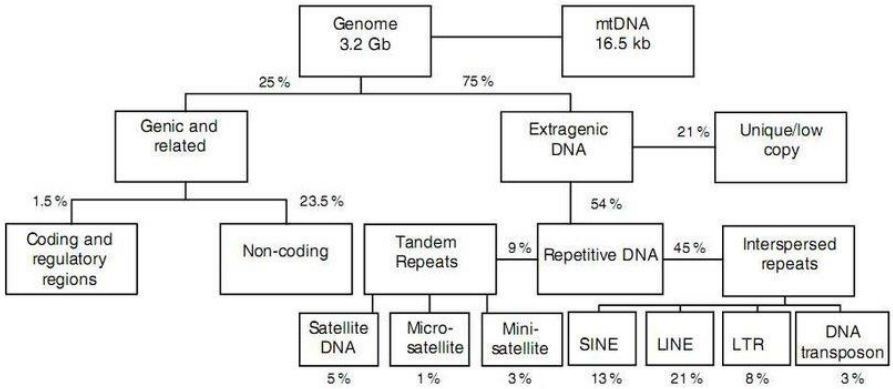


Figura 1. Classificació de l'ADN genòmic segons la seva estructura. Extret de l'article (4).

1.1.1.- Anàlisi d'STRs

Només una petita part de l'ADN són gens (regions que codifiquen per a proteïna) com es veu a la Figura 1. La resta, són regions repetitives i altres.

Per altra banda, també hi ha una petita part d'aquest ADN que sigui variable entre individus, els anomenats polimorfismes. Aquestes zones de l'ADN són l'objectiu dels estudis de genètica forense, i les podem separar en (3,5):

- *Single Nucleotide Polymorphisms* (SNPs): Variació d'un sol nucleòtid. És la més comú, constituint un 90% de les variacions .
- Insercions-delecions (*Indels*): Una petita variació del nombre de nucleòtids deguda a que una part de la seqüència s'ha eliminat o afegit.
- Elements transposables: Són seqüències que poden copiar-se a si mateixes i inserir-se en alguna altra regió de l'ADN. Un bon exemple d'aquest tipus són les repeticions Alu humanes, o *Copy Number Variations* (CNVs), variacions de nombre de còpia en català.
- Altres elements repetitius: Són repeticions d'una certa seqüència. Normalment es divideixen en microsatèl·lits i minisatèl·lits , coneguts en anglès com a *Short Tandem Repeats* (STRs) i *Variable Number Tandem Repeats* (VNTRs) respectivament. L'única diferencia és el nombre de nucleòtids de la repetició, sent més petits els STRs (entre 2 i 7 nucleòtids) que els VNTRs (entre 15 i 100).

Aquestes últimes són les que més han beneficiat a la genètica forense, donada la seva facilitat per a ser analitzades: Un canvi en el nombre de nucleòtids permet veure la diferència entre dues mostres a partir

d'una electroforesi, una tècnica que permet separar fragments d'ADN segons la seva llargada.

En el camp de la genètica forense, per tal d'obtenir un perfil genètic únic d'un individu s'utilitzen els STRs en contraposició als VNTRs i CNVs que s'utilitzaven al principi (els RFLP). Aquests STRs són repeticions d'entre 2 i 7 nucleòtids, que poden estar repetits entre 3 i 60 cops (1,3,5). Es poden classificar de diverses maneres, però al llibre de Barrio i Crespillo (3), utilitzen la següent classificació, que també es pot veure a la Figura 2:

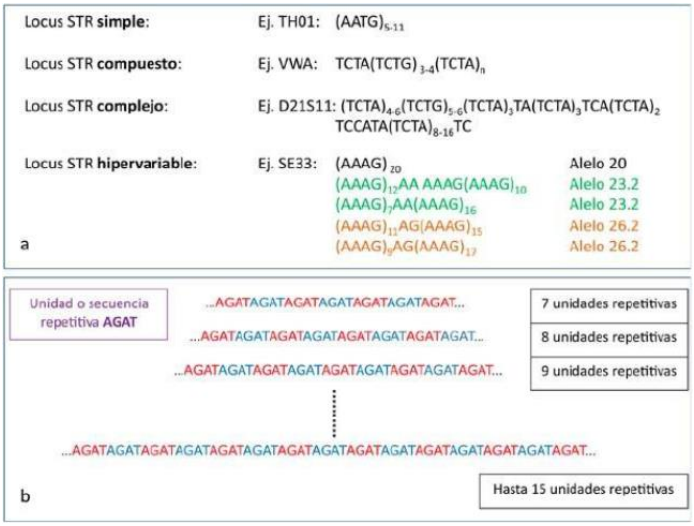


Figura 2. Exemples dels diferents tipus d'STR. Extrets del llibre (3).

- STR simples: Contenen una única unitat repetitiva o motiu, que es repeteix un nombre determinat de vegades sense interrupcions (en tàndem).
- STR composts: Contenen dos o més motius que es repeteixen en tàndem.
- STR complexos: Contenen diversos motius que es repeteixen en tàndem en diverses zones flanquejades per seqüències invariables.
- STR hipervariables: Són STRs tant complexos com simples en els que la distribució dels motius en la zona contigua al STR és altament diversa, creant diversos al·lels amb el mateix nombre de repeticions.

1.1.2.- CoDIS i les bases de dades d'STRs

Un dels primers estàndards en quan a STRs per a la formació d'una base de dades de perfils genètics, va ser creat per l'FBI l'any 1997 i es va anomenar *Combined DNA Indexing System* (CoDIS). Consistia en un conjunt de 13 STRs autosòmics validats estadísticament per a tindre una gran variabilitat entre individus: CSF1PO, FGA, TH01, TPOX, VWA, D3S1358, D5S818, D7S820, D8S1179, D13S317, D16S539, D18S51 y D21S11 (3).

Des de la creació d'aquest estàndard, s'han anat incorporant nous locus d'STRs per a incrementar la capacitat de individualització d'aquests. A Europa, per exemple, es van afegir 5 STRs que no pertanyien al CoDIS original, i finalment, el 2017, l'FBI va afegir-los junt a dos STRs més (D2S1338, D19S433, D1S1656, D12S391, D2S441, D10S1248 i D22S1045) (3).

1.1.3.- La electroforesi capil·lar

Avui dia, per a la determinació dels STR, es segueix, com ja hem comentat fins ara, l'amplificació dels loci específics amb una PCR multiplex seguida d'un mètode de separació, una electroforesi. Per tal de poder fer tots els STR possibles, s'utilitzen primers marcats amb fluorocroms diferents, i a distàncies diferents del STR. Això genera fragments més pesats que separen les zones on poden aparèixer diferents STRs (3).

Això és possible gràcies a l'electroforesi capil·lar (EC). En una electroforesi normal, s'acostuma a utilitzar un gel de poliacrilamida al qual s'aplica un camp elèctric. Com que l'ADN és una molècula carregada negativament, migra a través del polímer cap al pol positiu. Com més llarga una cadena, més facilitat d'enredar-se amb la xarxa de poliacrilamida, per tant més lent avança.

En el cas d'una EC, aquest gel es troba dins un tub capil·lar d'entre 25 i 75 μm. Això permet nombrosos avantatges entre els que destaquen un augment del voltatge que es pot fer córrer dins el gel (fins a 20 kV), fet que fa possible una major discriminació de llargades, fins i tot arribant al nivell de poder distingir fragments amb un sol nucleòtid de diferència (3). El tub després passa per un detector que incideix llum per activar els fluorocroms i detectar la llum produïda per la mostra per a mostrar-la posteriorment en un electroferograma.

Aquest mètode, al ser purament basat en la llargada dels fragments, no ens permet saber, dels possibles al·lels d'un locus d'un STR hipervariable amb la mateixa llargada, quin s'està analitzant.

1.2.- Seqüenciació

Des del descobriment de les funcions de l'ADN, els investigadors han intentat trobar maneres d'aconseguir la seqüència d'aquest. Els primers intents es basaven en exonucleases específiques, però el 1997 va aparèixer el mètode de Sanger (6,7).

El mètode consisteix en l'ús de nucleòtids terminadors. Això genera fragments amb pesos creixents acabats en el nucleòtid en concret, que es poden separar amb una electroforesi per a després poder escriure l'ordre dels fragments.

Amb l'aparició de la piroseqüenciació durant la dècada dels 90, va sorgir una nova era sovint anomenada "seqüenciació de nova generació" (NGS, en anglès). Això va portar al creixement de marques com 454, Solexa, Illumina i Ion Torrent (6). Cada sistema utilitzava una tecnologia lleugerament diferent, però en general, eren mètodes que encara que no poguessin seqüenciar fragments molt llargs, podien fer-ho en temps rècord, amb Illumina afirmant que podien seqüenciar 16 genomes en 3 dies.

Les NGS, però, presenten un problema alhora de generar seqüències i és que com que generen fragments curts (anomenats reads) que després es "munten" en assemblatges, aquests han de trobar la seqüència consens de tots els reads que se sobreposen. És a dir, si hi ha molta variació entre aquests, la seqüència perd qualitat. A més, aquests mètodes només generen una lectura per cada cromosoma, és a dir, que la seqüència consens barreja els dos cromosomes.

1.2.1.- La tercera generació

Amb aquest objectiu, va néixer una nova onada de mètodes de "seqüenciació de molècula simple". Aquests es basen en seqüenciar sense amplificar amb PCR prèviament, és a dir, evitant les mutacions que es poden anar acumulant per errors de la polimerasa. Són exemples d'aquesta generació, els mètodes *Single Molecule Real-Time* (SMRT) i *Nanopore* (6).

Aquests mètodes, no només permeten una seqüenciació més filedigne, sinó que a més permeten separar els dos cromosomes (el patern i el matern), ja que el sistema utilitzat fa que els reads d'un mateix pouet en SMRT o d'un mateix porus en *Nanopore* fossin de la mateixa molècula.

Això, juntament a l'abaratiment de les tècniques de seqüenciació i la facilitat d'execució d'aquests, fa que sigui més pràctic i econòmic realitzar una seqüenciació de tot el genoma que no realitzar l'electroforesi capil·lar a la mostra.

Tot i així, com que la seqüenciació pot aportar més informació sobre els STRs que la EC i la majoria de bases de dades de l'actualitat estan creats amb una nomenclatura adaptada a aquesta tècnica, el canvi haurà de ser progressiu i la nomenclatura haurà de ser compatible amb la que s'utilitza en EC.

2.- Hipòtesi i objectius

El sistema actual per a la creació de perfils no ha canviat gaire en els últims anys, i la seva eficiència i velocitat tampoc.

Per altra banda, els sistemes de seqüenciació són cada cop més ràpids i accessibles. Per aquest motiu, la hipòtesi d'aquest treball és:

- Si podem obtenir fàcilment la seqüència del genoma complert d'un individu, el perfilat dels STRs es podria obtenir a partir d'aquesta mitjançant eines informàtiques.

D'aquesta manera, es plantegen els següents objectius per al treball:

- Familiaritzar-se amb l'entorn informàtic i els paquets necessaris per a la creació dels scripts.
- Crear un programa que, a partir d'una llibreria de seqüències patró trobi els al·lels d'una sèrie de loci.
    - Dissenyar un algoritme que obtingui l'al·lel d'un STR a partir de la seqüència.
    - Crear un script que permeti la creació de la llibreria de seqüències patró de cada loci.
    - Adaptar el programa per a que el pugui utilitzar qualsevol investigador.
- Validar aquest programa amb genomes de la base de dades.

3.- Metodologia

3.1.- Format de fitxers

Per tal d'acotar el treball, es va decidir que només s'utilitzaria arxius en format FASTA (extensió ".fasta" o ".fna"). Aquest format es caracteritza per la seva senzillesa. Un arxiu FASTA es pot llegir com un arxiu de text, i cada seqüència conté una línia de capçalera amb la informació necessària, seguida de tota la seqüència en les línies necessàries. Aquesta informació (nom, teixit, procedència, estudi, etcètera) ve precedida pel símbol ">".

Els scripts generats per als diferents programes, escrits en Python, tenen la extensió ".py", i es van obtindre a base de la exportació dels arxius ".ipynb" generats per l'entorn de treball jupyter. Pel que fa a la resta d'arxius necessaris per al programa es va utilitzar el format estàndard de text (.txt) i en casos de llistes i taules, el format de variables separades per coma (.csv). També es van utilitzar per tal de generar els arxius de sortida. Com bé diu el seu nom, en aquest format es tabulen els valors utilitzant el símbol de coma ",".

3.2.- Llenguatges i entorns

Per tal de realitzar el treball es va utilitzar principalment el llenguatge Python, ja que té una sintaxis molt senzilla i està altament estès en la comunitat bioinformàtica. A més, nombroses llibreries de funcions s'han realitzat fins la data com ara el paquet principal en el que es basa el programa: Biopython.

Python es pot utilitzar directament des de la terminal, però es va decidir que per tal de facilitar el seu ús es faria servir un entorn de treball. En aquest cas, es va utilitzar l'entorn Jupyter. Aquest utilitza un navegador per tal de mostrar una interfície que permet escriure el codi, seccionar-lo, i fins i tot activar només una part.

3.2.1.- Biopython

Aquesta llibreria en qüestió, es una molt bona recopilació de funcions útils per a la bioquímica. Amb aquest paquet és possible guardar les variables en forma d'un objecte nou anomenat "sequence" o "seq()", el qual permet aplicar transcripció, traducció, o fins i tot fer la cadena complementaria inversa d'una seqüència.

com alineaments de seqüències tant de manera local com global, llegir arxius amb seqüències en diferents formats (FASTA, GenBank),

Les funcions més destacables d'aquest paquet són els següents:

- SeqIO: Permet llegir arxius en format FASTA o GenBank i generar un "SeqRecord", un objecte que inclou la seqüència, i tota la informació del gen.
- Entrez: Permet, si l'ordinador té xarxa, accedir a la base de dades del GenBank i descarregar seqüències.
- Align: Permet generar alineaments de dues seqüències tant de manera global com de manera local.

3.2.2.- Pandas

Com ja s'ha comentat, es farà servir arxius separats per coma, i això ens fa treballar amb *dataframes*. Un *dataframe* és com una taula, amb files i columnes que tenen nom (índex). El paquet Pandas permet llegir els documents ".csv" i convertir la informació de la taula en un *dataframe*, a part de incloure diverses funcions que permeten treballar amb aquests objectes.

3.2.3.- re

La llibreria de Python "re" (de l'anglès *regular expressions*) conté eines de cerca. Aquest, però, no és el motiu pel qual s'utilitza al treball: la funció *search()* es pot iterar, podent generar fàcilment una llista amb tots els índex en els que apareix la cadena de caràcters.

3.3.- Bases de dades

3.3.1.- GenBank

Com ja hem comentat, el paquet Biopython permet treballar amb la base de dades del GenBank. És una base de dades del *National Institute of Health* (NIH) i amb seu electrònica al *National Center of Biotecnology and Information* (NCBI), que conté seqüències d'àcids nucleics de multitud d'éssers vius. Els arxius extrets d'aquesta base de dades per a fer el treball s'expliquen en la següent taula:

Nom	ID accés Assembly	Tipus	Observacions
GRCh38	GCA_000001405.15	Pseudohaploide (principal)	Projecte genoma de referència
HG002	GCA_021951015.1	Pseudohaploide (principal)	Utilitzat per fer proves
HG00733	GCA_003634875.1	Diploide	Utilitzat per fer proves

3.3.2.- STRbase i STRbase 2.0

STRbase és una base de dades on estan recopilats diferents tipus de microsatèl·lits, tant humans com d'altres espècies d'interès. Per a aquest treball hem utilitzat tant la versió original de la web, com la nova, ja que aquesta última encara està en la fase de prova i li falta informació, però conté algunes dades d'interès.

D'aquesta base de dades es van extreure els noms dels loci, els primers, la informació sobre el patró de repeticions (sobretot en STRs complexos), l'al·lel del genoma de referència GRCh38.p12 i moltes coses més que es poden veure a l'annex 1.

Per a aquest treball, només es van utilitzar els STR del CoDIS originals.

4.- Resultats i discussió

4.1.- Descripció del programa

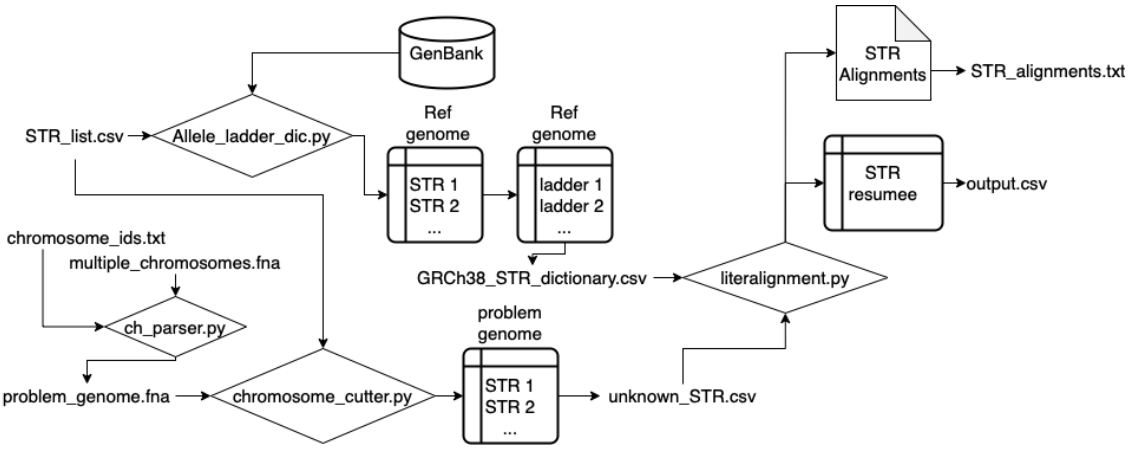


Figura 3. Esquema de la relació entre tots els fitxers i com cada script genera els fitxers de sortida.

El programa que es va consolidar, consta de tres executables Python com es veu en la Figura 3. El programa inclou un fitxer de text, README.txt amb les instruccions, com sol ser comú.

Com s'explica al fitxer, cada executable requereix que a la mateixa carpeta s'hi posin els inputs necessaris amb els mateixos noms. Aquests executables són els següents:

- allele_ladder_dic.py: Aquest script genera biblioteques de tots els al·lels dels STRs d'una llista (input STR_list.csv). Aquesta biblioteca s'emmagatzema en un nou fitxer csv (GRCh38_STR_dictionary.csv).
- chromosome_cutter.py: Permet analitzar un fitxer FASTA amb tot el genoma per tal de generar un altre csv amb les seqüències dels STRs de STR_list.csv tallades amb els mateixos primers que la biblioteca del punt anterior, informació que es guarda en l'arxiu unknown_STR.csv.

- literalignment.py: A partir dels fitxers generats en els dos programes anteriors, es pot executar aquest tercer script per a obtenir dos fitxers de sortida:
    - Un csv amb els al·lels assignats al comparar amb les biblioteques tal com s'obtindrien en un EC, però amb observacions sobre possibles mutacions dins la seqüència.
    - Un arxiu de text amb els alineaments realitzats per a poder analitzar les observacions de l'output anterior.
- FASTA parser: És una carpeta que conté un quart script (ch_parser.py) amb dos inputs (un FASTA amb més elements dels que calen i una llista dels IDs que volem). Aquest programa extra es va fer ja que tot i que el programa només pot processar seqüències haploides, és a dir, que per a un genoma diploide, s'hauran de generar dos fitxers FASTA un amb cada meitat del genoma complert. El programa funciona llegint els IDs de les seqüències i generen un FASTA amb només aquells. També resulta útil si el fitxer FASTA inclou o no té ordenades les seqüències.

4.1.1.- Fitxers d'input del programa

Com ja hem dit, els executables requereixen que dins la carpeta existeixin certs fitxers que serveixen d'input per al programa.

Per una banda, calen els genomes: el genoma de referència i el de la mostra que es vol perfilar. El primer dels dos, s'obté d'internet gràcies al mòdul Bio.Entrez ja que així el programa és més lleuger al fer que la carpeta no contingui un FASTA de 3 GB. El genoma a perfilar, ha de constar d'un únic arxiu FASTA (problem_genome.fna) amb una entrada per a cada cromosoma, amb IDs (allò que segueix el símbol ">") alfanumèricament creixents, és a dir, que els cromosomes estiguin ordenats.

Per altra banda, està el llistat d'informació dels STRs. STR_list.csv consisteix en una taula (com podeu veure a l'annex 1) amb les següents entrades:

- ID del STR: Merament informatiu, per a produir la taula final.
- Cromosoma: El cromosoma al que pertany. Permet descarregar nomes els cromosomes que calen.
- Posició: No aporta res, ja que es volia utilitzar en cas que cap dels primers funcioni com s'explica al apartat 4.1.2.
- GRCh38: L'al·lel de l'STR al genoma de referència.
- Motiu: Els nucleòtids que conformen el motiu de repetició.
- Primer *Forward* i *Reverse*: Han de ser els que tallen la seqüència del motiu que s'ha escrit, i no la complementaria (com s'explica a la Figura 4).
- Forma: Defineix l'estructura del STR, de manera que només amb l'al·lel i el motiu, es pugui generar qualsevol dels al·lels possibles (veure apartat 4.1.3).
- Rang: Defineix el rang dels al·lels que s'han trobat fins ara. Es poden utilitzar tant els rangs dels *allelic ladders* dels kits com la informació de STRbase.

4.1.2.- Algoritme per a tallar els loci patró i problema

El programa chromosome_cutter.py consisteix en un conjunt de tècniques per a tallar les seqüències dels STRs. Aquest, però, també l'utilitza el programa allele_ladder_dic.py. El motiu de treballar amb zones delimitades, és purament tècnic, ja que és més

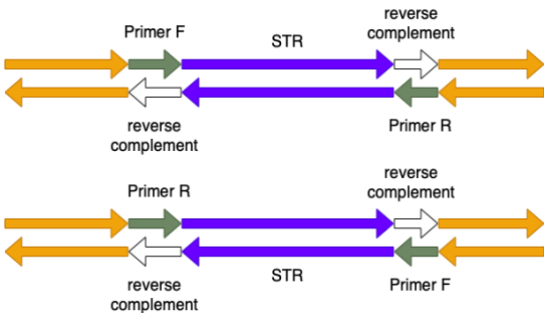


Figura 4. Esquema que representa la seqüència escrita (de dreta a esquerra) i les dues possibilitats en que es pot trobar l'STR.

senzill treballar amb aquestes que amb les entre 250 i 50 Mbp d'un arxiu FASTA cromosòmic.

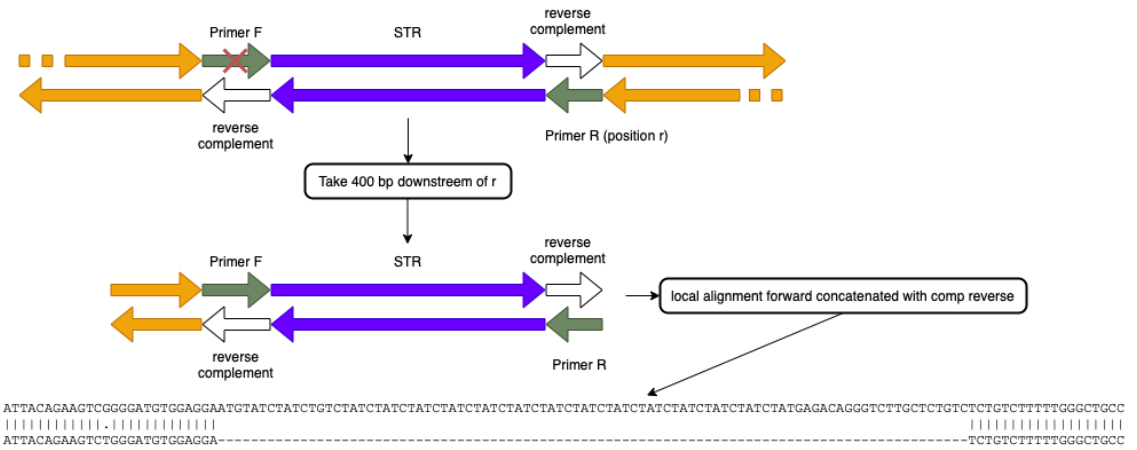


Figura 5. Esquema del funcionament del mètode de tall per alineament. Com es pot veure, el primer forward té una mutació, però això no evita que, a partir del primer reverse l'alineament funcioni. També es pot veure com apareix la mutació a l'alineament (simbolitzat com un punt ".").

Els STR tal i com estan tipificats a la base de dades de STRbase, tot i ser seqüències repetitives en les dues cadenes del cromosoma, no sempre es troben en la cadena que es troba seqüenciada a GenBank.

A més, al disposar només d'una de les dues cadenes, com es pot comprovar a la Figura 4, tampoc trobarem allà la seqüència dels dos primers alhora. Si trobem el *forward*, del primer *reverse* trobarem la seqüència complementaria inversa, i viceversa.

Per a acotar la zona de l'STR, es van pensar diferents mètodes, però l'ús dels primers *forward* i *reverse*, es la millor opció pel que fa a velocitat de processament. Tot i així, en cas que hi hagués una mutació en la zona o algun primer aparegui en alguna altra zona del cromosoma, el programa, que treballa amb seqüències literals, no pot trobar la seqüència exacta i falla. En aquest cas es pot fer un alineament, tal com es mostra a la Figura 5.

L'alineament es fa entre el cromosoma i una concatenació del primer *forward* + el complementari revers del *reverse*. Teòricament, el millor alineament hauria de ser aquell on les zones de primer facin *match* i la

zona del mig (la seqüència de l'STR) com un gap. Per a propiciar aquest tipus de resultat, l'alineament es fa local (sense gaps externs) i amb un preu més punitiu per la obertura de gap (-4) que per l'extensió (-1).

Com ja hem explicat, aquesta opció no només és extremadament lenta, sinó que pot portar al congelament del ordinador. Per aquest motiu, abans de realitzar aquest alineament, s'ha d'acotar la zona.

Amb la intenció de fer el programa el més eficient possible, es va pensar en l'algoritme de la Figura 6. Tot i així, encara queden forats (senyalats en vermell a la figura) en el programa. Aquestes situacions que no estan contemplades, son les que son menys probable de que passin.

Pel que fa a velocitat, que el programa contacti amb la plataforma Entrez del NCBI, faci una petició per a descarregar un arxiu de mida genòmica, esperi a que la petició s'accepti, i descarregui entre 250 i 50 MB a través del portal. Aquest problema es podria solucionar, en futures modificacions del programa, amb la implementació de l'eina Blast.NCBIWWW que permet fer un BLAST fora de la web, la qual no s'ha explorat en aquest treball però podria ser una bona solució a la diversificació de la Figura 6.

4.1.3.- Biblioteques d'al·lels o *allelic ladder*

Com ja s'ha comentat a la introducció, d'STRs n'hi ha 3 tipus. La majoria dels utilitzats per al perfilat genètic són simples o compostos, però també hi ha algun de complex.

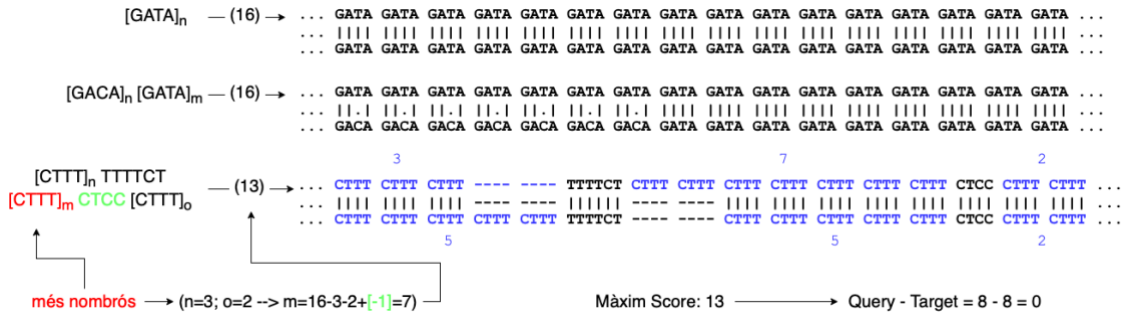


Figura 7. Exemples de formació d'un STR a partir de la columna "form" de STR_list.csv per a cada tipus de STR. En els exemples, s'ensenya com es forma l'al·lel artificial 16 d'STRs simples i compostos i l'al·lel 13 d'STRs complex i a sota una seqüència problema alineada.

Per a la creació de la biblioteca, es va utilitzar de patró el genoma de referència GRCh38.p14, del qual coneixem els al·lels dels loci de la llista STR_list.csv. A partir d'aquest, es va obtindre la seqüència pre-STR, la post-STR, i entremig es van anar inserint diferents repeticions del motiu, generant una biblioteca de seqüències d'ordre creixent, com es pot veure a la Figura 7 i a la Figura 8.

Pel que fa als STR simples o compostos, la manera en la que funciona l'algoritme permet determinar l'al·lel amb una biblioteca que només tingui un motiu, és a dir, interpretant els STR compostos com a simples per a generar la biblioteca.

En el cas de les simples, si hi ha per exemple 9 repeticions, l'alineament ja donarà *score* màxim amb l'element de la biblioteca que tingui 9

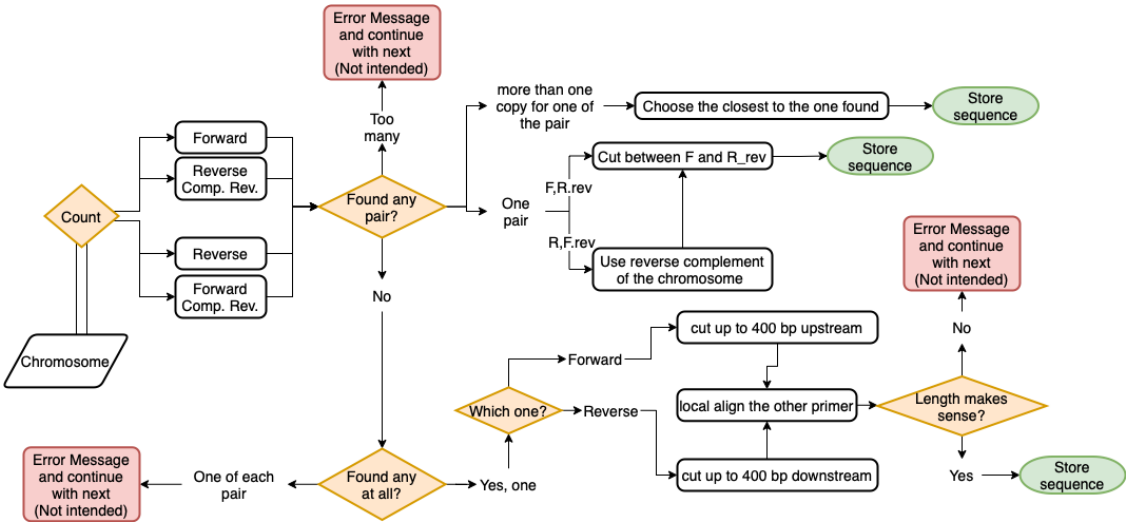


Figura 6. Esquema de l'algoritme per a decidir com acotar la seqüència.

repeticions, però en el cas dels compostos, la biblioteca només té repeticions del primer motiu. En l'exemple de la figura, hi ha dos motius (GACA i GATA, dels que s'utilitza GATA).

Com es veu a la figura, el programa farà l'alineament entenent les repeticions que siguin del segon motiu com a mutacions, fent que l'*score* màxim sigui el que tingui el mateix nombre d'al·lel que la seqüència problema.

Això és possible gràcies a que l'alineador no penalitza les no-coincidències, i és un alineament global, de manera que necessita que els extrems siguin el més alineat possible. Això fa que el primer que pensi el programa en fer sigui una mutació i no una *indel*. Tot i així, seguim amb el problema de crear biblioteques per a al·lels complexos.

4.1.4.- Biblioteques per a un STR complex

En el cas de que no només hi hagi més d'un motiu, sinó que apareguin nucleòtids lliures entre mig de les repeticions, fa que la creació d'una biblioteca amb tots els al·lels sigui impossible. Tot i així, es pot aconseguir amb dues premisses i un seguit d'aproximacions:

- Per una banda, els STR complexos casi mai tenen poques repeticions, de manera que es pot agafar la zona més constant, i agregar-hi les repeticions apart. Per exemple, si el loci conté tres zones de repetició aïllades per seqüències no repetitives, es pot agafar la més nombrosa i agafar des de l'al·lel més petit possible al més gros (a poder ser dins el rang).

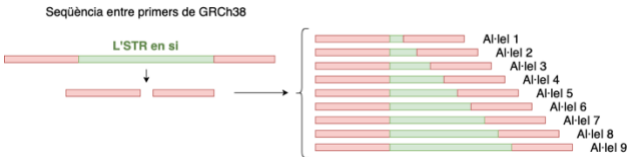


Figura 8. Esquema del funcionament de la formació de la biblioteca.

- El programa penalitza els *indels*, però els pot posar, de manera que si el patró i el problema tenen les repeticions en altres llocs,

el programa ho interpreta com a *indel* dins el propi STR, com es veu al exemple de la Figura 7.

Amb aquestes premisses, es pot de manera deliberada generar una expressió que el programa pugui fer variar des del al·lel més baix trobat per a un loci, i afegir a partir d'aquest repeticions al final o a una zona indicada del mig, sense que això afecti massa al output del programa.

4.1.5.- Algoritme per a trobar l'al·lel

Com ja hem explicat, un STR consisteix en una seqüència de nucleòtids que es repeteix. Amb aquesta premissa, es va pensar en diferents maneres d'obtenir l'al·lel a partir de la seqüència. Els mètodes que es van provar van ser els que s'expliquen a continuació.

Al final, es va decidir utilitzar un mètode bastant més efectiu que els anteriors i que no tenia perill de fallar cas que la seqüència problema a determinar contingui alguna mutació, com es veurà a l'apartat 4.2.1.

El mètode, consisteix en crear una biblioteca de seqüències artificial creades a partir d'un patró d'al·lel conegut, amb un nombre creixent de repeticions. Un cop creada la biblioteca, fer un alineament amb cada una de les seqüències. Aquest alineament es fa amb les següents especificacions:

- Mode: Global (algoritme Needleman-Wunsch)
- Puntuació per coincidència: +1 per bp
- Puntuació per no-coincidència: 0
- Puntuació per *gap* intern: -1 per bp

Aquest format és punyent amb les obertures i extensions de forats(*gaps*) dins la seqüència, cosa que fa que per a al·lels de la biblioteca inferiors al problema a alinear, les poques coincidències fan que la puntuació dels alineaments sigui baixa, mentre que en el cas

dels al·lels superiors, la creació de *gaps* en la *query* baixarà la puntuació, fent que l'al·lel més proper al de la seqüència problema sigui el que major puntuació tingui.

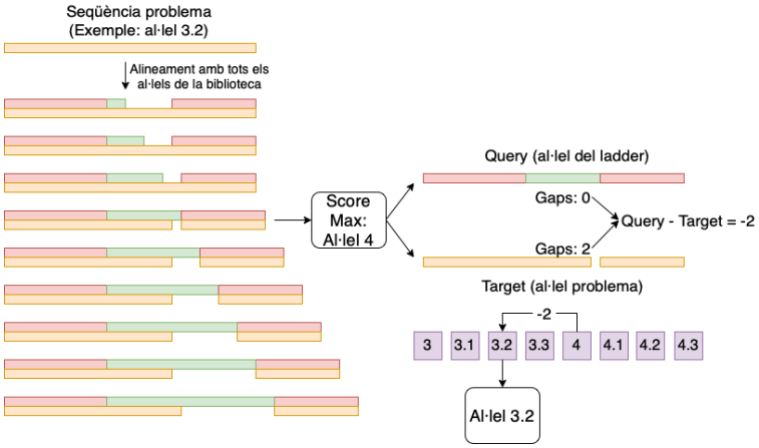


Figura 9. Esquema de com el programa tria el millor alineament i com d'aquest es realitza l'ajust final. En l'exemple, tenim l'al·lel 3.2, d'on treiem l'al·lel 4, al que es resten 2 gaps *per a trobar que es tracta del 3.2.*

Un cop trobat l'al·lel més probable, el programa fa una sèrie de comptatges tal com s'explica a continuació, que determinen si la seqüència és l'al·lel escollit (per exemple 12), o si s'ha generat algun *gap* que el converteix en l'al·lel 11.3 o el 12.2.

Per tal de fer aquest ajust final, es compten els *gaps* de la *query* (patró de la biblioteca) i se li resten els del *target* (seqüència problema) per determinar com de diferent es una de l'altra en llargària i sumar el resultat a l'al·lel escollit tal com es mostra a la Figura 9. En cas que hi hagués *gaps* en totes dues cadenes, els *gaps* d'una contraresten els de l'altra.

Taula 1. Sortida del programa al analitzar el genoma de referència.

STR id	Allele	Observations
TPOX	8	
D3S1358	16	There are 1 mutations in STR.
FGA	22	
CSF1PO	13	
D5S818	11	
D7S820	13	
D8S1179	13	There are 1 mutations in STR.
TH01	7	
VWA	17	There are 5 mutations in STR.
D13S317	11	
D16S539	11	
D18S51	18	
D21S11	29	There are 6 mutations in STR.

4.2.- Validació del programa

L'ideal per a poder validar aquest programa seria un genoma del qual sabéssim els STR segons el mètode tradicional. Això, però, no

és possible ja que les seqüències del GenBank que contenen tot el genoma solen vindre d'estudis on només s'ha fet la seqüenciació massiva de tot un individu. Tot i així, com veurem més endavant, si que s'ha aconseguit validar el funcionament de l'algoritme.

De totes maneres, es va pensar en utilitzar el propi genoma de referència GRCh38 per a comprovar el funcionament del programa. Gràcies a la base de dades STRBase2.0, se saben els al·lels d'aquest, de manera que es va poder comprovar, com es veu a la Taula 2, que els al·lels trobats per al programa coincideixen amb els recollits de la base de dades.

4.2.1.- Validació de l'algoritme determinant d'al·lel

Per tal de comprovar la capacitat del mètode de determinació al·lèlica, es va agafar la seqüència del cromosoma 7 del genoma de referència, així com, del genoma HG002 d'un estudi de genoma complert (WGA), un altre cromosoma 7 (ID de GenBank: CM039017.1) per tal de provar el programa amb l'STR D7S820.

Un cop obtingudes les seqüències de la zona utilitzant els mateixos primers, es va copiar la seqüència problema (CM039017.1) per a generar artificialment mutacions, insercions, delecions, etcètera, i comprovar el potencial de l'algoritme. Totes les seqüències basades en la seqüència problema es troben al annex 2.

Si ens fixem en les dades obtingudes al passar totes les variants (anar a l'annex 3), podem veure que l'algoritme interpreta tots els *gaps*, tant els que es formen dins com fora de l'STR, com els de dins, fent que el resultat també contingui només la informació que obtindríem del mètode tradicional (EC).

Per exemplificar tot això, podem comparar l'al·lel 11.3 amb l'al·lel 12 amb una deleció d'un nucleòtid fora de la zona de l'STR. Tant si fem aquesta deleció abans com si la fem després, a les taules es pot veure

com els *gaps* finals en tots tres casos donen "-1", el que significa que en tots tres casos donen "11.3", cosa que seria compatible amb el que donaria una EC. A més, podem veure que en cada cas, les posicions del *gap* es troben abans, dins o després del rang de posicions de l'STR (columna "STR postGaps" a la taula de l'annex). El programa final permet trobar les zones on es generen diferències entre la seqüència patró i la problema, permetent posar informació extra al output com ara "hi ha *indels* dins l'STR" o "els *indels* de la seqüència es contraresten". D'aquesta manera, es manté la nomenclatura original d'una EC. Tot i així, en futures edicions del programa es podria considerar una nomenclatura que fos retrocompatible i que la comunitat científica hagués estandarditzat.

4.2.2.- Funcionalitat del programa sencer

Per a comprovar la funcionalitat del programa, es va descarregar la seqüència FASTA de tot el genoma utilitzat a la validació 4.2.1 (HG002). Aquest *assembly*, però, inclou també fragments que no s'han ubicat dins de cap cromosoma (els *scaffolds*).

Un cop descarregat l'arxiu i copiat a la carpeta, es va analitzar amb el programa i es van obtenir: Els alineaments que es poden trobar a l'annex 4, i la taula de resultats que es pot veure a la Taula 2.

Aquesta taula ens ensenya, com es pot comprovar a l'annex, l'al·lel que hagués trobat una EC, i com en alguns loci aquesta

Taula 2. Sortida del programa al analitzar la seqüència del genoma de HG002.

STR id	Allele	Observations
TPOX	8	
D3S1358	16	There are 1 mutations in STR.
FGA	20	
CSF1PO	10	
D5S818	11	
D7S820	12	
D8S1179	13	
TH01	9	Some indel inside STR zone. There are 3 mutations in STR.
VWA	16	There are 4 mutations in STR.
D13S317	11	
D16S539	11	
D18S51	16	
D21S11	31.2	Some indel inside STR zone. There are 5 mutations in STR.

informació es veu complementada amb mutacions i *indels* que fan que al llegir els alineaments busquem directament aquests canvis.

Les mutacions de VWA i D21S11 són d'esperar, ja que són part d'un altre motiu (STRs compostos), mentre que les del loci TH01 són per alguna inserció. Per a aquestes, i pensant de nou en una possible millora del programa, es podria buscar una manera de fer que només aquelles mutacions que no es produeixin arran del mètode. Això es podria fer mitjançant un llistat dels diferents motius que es poden trobar en aquest loci i buscar-les a la "zona del STR".

5.- Conclusió

Aquest treball ha permès l'aprenentatge de les opcions que dóna Python i el paquet de biopython alhora de realitzar un script que compleixi amb el propòsit que plantejàvem als objectius.

Per altra banda, s'ha creat un programa que permet tant generar la biblioteca d'al·lels a partir d'una llista d'STRs, com analitzar un genoma haploide/pseudohaploide a partir d'aquesta biblioteca per a obtenir l'al·lel de manera anàloga al mètode original, a més d'informació addicional.

Així doncs, el programa resultant ha arribat a les expectatives, permetent no només obtenir el mateix resultat teòric al que s'arribaria amb el mètode tradicional de la electroforesi capil·lar, sinó que a més pot donar informació valuosa alhora d'afegir més variants que les normalment obtingudes.

6.- Referències

1. Kowalczyk M, Zawadzka E, Szewczuk D, Gryzińska M, Jakubczak A. Molecular markers used in forensic genetics. Med Sci Law. 2018;58(4):201-9.
2. Bright J, Kelly H, Kerr Z, Mcgovern C, Buckleton JS, Kelly H, et al. The interpretation of forensic DNA profiles : an historical perspective. J R Soc New Zeal [Internet]. 2019;0(0):1-15. Disponible a: https://doi.org/10.1080/03036758.2019.1692044
3. Crespillo Márquez M, Barrio Caballero PA. Genética forense: del laboratorio a los tribunales. Madrid: Diaz de Santos; 2019.
4. Ismaail B, Al-Awadi, Zahra M, Al-Khafaji Z. Study of Genetic Variations of 15 Autosomal short Tandem Repeats (STRs) and Amelogenin Loci to Establish Database for Iraqi Population. 2014.
5. A.F. Al-Koofee D, M.H. Mubarak S. Genetic Polymorphisms. En: The Recent Topics in Genetic Polymorphisms [Internet]. IntechOpen; 2020. p. 271. Disponible a: https://www.intechopen.com/books/the-recent-topics-in-genetic-polymorphisms/genetic-polymorphisms
6. Giani AM, Gallo GR, Gianfranceschi L, Formenti G. Long walk to genomics: History and current approaches to genome sequencing and assembly. Comput Struct Biotechnol J [Internet]. 2020;18:9-19. Disponible a: https://doi.org/10.1016/j.csbj.2019.11.002
7. Sanger F, Nicklen S, Coulson R. DNA sequencing with chain-terminating inhibitors. Proc Natl Acad Sci USA. 1977;74(12):5463-7.
8. Hammond HA, Jin L, Zhong Y, Thomas Caskey C, Chakraborty R. Evaluation of 13 short tandem repeat loci for use in personal identification applications. Am J Hum Genet. 1994;55(1):175-89.

9. Urquhart A, Oldroyd NJ, Kimpton CP, Gill P. Highly discriminating heptaplex short tandem repeat PCR system for forensic identification. Biotechniques. gener 1995;18(1):116-118,120-121.

10. Mills KA, Even D, Murray JC. Tetranucleotide repeat polymorphism at the human alpha fibrinogen locus (FGA). Hum Mol Genet. 1992;1(9):779.

11. Krenke BE, Tereba A, Anderson SJ, Buel E, Culhane S, Finis CJ, et al. Validation of a 16-locus fluorescent multiplex system. J Forensic Sci. juliol 2002;47(4):773-85.

12. Barber MD, Parkin BH. Sequence analysis and allelic designation of the two short tandem repeat loci D18S51 and D8S1179. Int J Legal Med. 1996;109(2):62-5.

13. Urquhart A, Kimpton CP, Downes TJ, Gill P. Variation in Short Tandem Repeat sequences -a survey of twelve microsatellite loci for use as forensic identification markers. Int J Legal Med. 1994;107(1):13-20.

7.- Webs i Bases de dades

NCBI: https://www.ncbi.nlm.nih.gov/genbank/

STRbase: https://strbase.nist.gov/index.htm

STRbase 2.0: https://strbase-b.nist.gov

Annex 1: Llista d'informaicó CoDIS

id	chromosome	position	GRCh38	motif	primer F	primer R	form	allele range	Ref.
CSF1PO	5	1E+11	13	AGAT	AACCTGAGTCTGCCAAGGACTAGC	TTCCACACACCACTGGCCATCTTC	s*n	5-17	(8)
FGA	4	2E+11	22	CTTT	GCCCCATAGGTTTTGAACTCA	TGATTTGTCTGTAATTGCCAGC	'TTTC'*3 + 'TTTTTTCT' + s*(n-8) + 'CTCC' + 'TTCC'*2	12-53	(9)
TH01	11	2E+09	7	AATG	GTGGGCTGAAAAGCTCCCGATTAT	ATTCAAAGGGTATCTGGGCTCTGG	s*n	1-14	(8)
TPOX	2	1E+09	8	AATG	CACTAGCACCCAGAACCGTC	CCTTGTCAGCGTTTATTTGCC	s*n	4-18	(10)
VWA	12	6E+09	17	TCTA	CCCTAGTGGATGATAAGAATAATCAGTATG	GGACAGATGATAAATACATAGGATGGATGG	s*n	10-26	(9)
D3S1358	3	5E+10	16	TCTA	ATGAAATCAACAGAGGCTTGC	ACTGCAGTCCAATCTGGGT	s*n	8-22	(11)
D5S818	5	1E+11	11	AGAT	GGTGATTTTCCTCTTTGGTATCC	AGCCACAGTTTACAACATTTGTATCT	s*n	4-20	(11)
D7S820	7	8E+10	13	GATA	ATGTTGGTCAGGCTGACTATG	GATTCCACATTTATCCTCATTGAC	s*n	5-22	(11)
D8S1179	8	1E+11	13	TCTA	TTTTTGTATTTCATGTGTACATTCG	CGTAGCTATAATTAGTTCATTTTCA	s*n	5-20	(12)
D13S317	13	8E+10	11	TATC	ATTACAGAAGTCTGGGATGTGGAGGA	GGCAGCCCAAAAAGACAGA	s*n	5-17	(11)
D16S539	16	8E+10	11	GATA	GGGGGTCTAAGAGCTTGTAAAAAG	GTTTGTGTGTGCATCTGTAAGCATGTATC	s*n	4-17	(11)
D18S51	18	6E+10	18	AGAA	GAGCCATGTTCATGCCACTG	CAAACCCGACTACCAGCAAC	s*n	6-40	(9)
D21S11	21	2E+10	29	TCTA	ATATGTGAGTCAATTCCCCAAG	TGTATTAGTCAATGTTCTCCAG	s*13 + 'TA'+s*3 + 'TCA' + s*2 + 'TCCATA' + s*(n-18)	18-42	(13)

Annex 2: Seqüències artificials de validació

STR D7S820

Al·lel 12:
GAGTTATTTTAAGGTTAATATATATAAAGGGTATGATAGAACACTTGTCATAGTTTAGAACGAACTAAC
GATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGACAGATTGATAGTT
TTTTTTTATCTCACTAAATAGTCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT

Al·lel 11.3:
GAGTTATTTTAAGGTTAATATATATAAAGGGTATGATAGAACACTTGTCATAGTTTAGAACGAACTAAC
GATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATGACAGATTGATAGTTT
TTTTTTATCTCACTAAATAGTCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT

Al·lel 12 amb deleció post STR:
GAGTTATTTTAAGGTTAATATATATAAAGGGTATGATAGAACACTTGTCATAGTTTAGAACGAACTAAC
GATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGACAGATTGATAGTT
TTTTTTTATCTCAC-AAATAGTCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT

Al·lel 12.1:
GAGTTATTTTAAGGTTAATATATATAAAGGGTATGATAGAACACTTGTCATAGTTTAGAACGAACTAAC
GATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGA**T**CAGATTGATAGT
TTTTTTTTATCTCACTAAATAGTCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT

Al·lel 12 amb mutació interna:
GAGTTATTTTAAGGTTAATATATATAAAGGGTATGATAGAACACTTGTCATAGTTTAGAACGAACTAAC
GATAGC**TAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATA**GACAGATTGATAGTT
TTTTTTTATCTCACTAAATAGTCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT

Al·lel 12 amb deleció pre STR:
GAGTTATTTTAAGGTTAATATATATAAAGGGTATGATAGAACACTTGTCATAGTT-AGAACGAACTAAC
GATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGACAGATTGATAGTT
TTTTTTTATCTCACTAAATAGTCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT

Al·lel 12 amb deleció interna:
GAGTTATTTTAAGGTTAATATATATAAAGGGTATGATAGAACACTTGTCATAGTTTAGAACGAACTAAC
GATAGATAGATAGAT-GATAGATAGATAGATAGATAGATAGATAGATAGATAGACAGATTGATAGTTT
TTTTTTATCTCACTAAATAGTCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT

Al·lel 12 amb inserció post STR:
GAGTTATTTTAAGGTTAATATATATAAAGGGTATGATAGAACACTTGTCATAGTTTAGAACGAACTAAC
GATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGACAGATTGATAGTT
TTTTTT**T**ATCTCACTAAATGCAGTCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT

Al·lel 12 amb inserció pre STR:
GAGTTATTTTAAGGTTAATATATATAAAGGG**CG**TATGATAGAACACTTGTCATAGTTTAGAACGAACTA
AC**GATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATA**GACAGATTGATAG
TTTTTTTTTATCTCACTAAATAGTCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT

Al·lel 12 amb inserció interna:
GAGTTATTTTAAGGTTAATATATATAAAGGGTATGATAGAACACTTGTCATAGTTTAGAACGAACTAAC
GATAGATAGATAGATAGATAGATAGATAGAGA**GATAGATAGATAGATAGATA**GACAGATTGATAG
TTTTTTTTTATCTCACTAAATAGTCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT

Annex 3: Dades validació algoritme

STR D7S820

Al·lel 12:

Alelle	Score	STR postGaps	Gap count	Q-T	Mut pos	Gap pos
10	177.0	69-109	8	8	[138]	[111, 112, 113, 114, 115, 116, 117, 118]
11	185.0	69-113	4	4	[138]	[115, 116, 117, 118]
12	193.0	69-117	0	0	[138]	[]
13	189.0	69-121	4	-4	[142]	[119, 120, 121, 122]
14	185.0	69-125	8	-8	[146]	[119, 120, 121, 122, 123, 124, 125, 126]

Your sequence is probably: 12

Al·lel 11.3:

Alelle	Score	STR postGaps	Gap count	Q-T	Mut pos	Gap pos
10	178.0	69-109	7	7	[137]	[111, 112, 113, 114, 115, 116, 117]
11	186.0	69-113	3	3	[137]	[115, 116, 117]
12	191.0	69-117	1	-1	[138]	[116]
13	187.0	69-121	5	-5	[142]	[116, 119, 120, 121, 122]
14	183.0	69-125	9	-9	[146]	[116, 119, 120, 121, 122, 123, 124, 125, 126]

Your sequence is probably: 11.3

Al·lel 12 amb deleció post STR:

Alelle	Score	STR postGaps	Gap count	Q-T	Mut pos	Gap pos
10	175.0	69-109	9	7	[138]	[111, 112, 113, 114, 115, 116, 117, 118, 146]
11	183.0	69-113	5	3	[138]	[115, 116, 117, 118, 146]
12	191.0	69-117	1	-1	[138]	[146]
13	187.0	69-121	5	-5	[142]	[119, 120, 121, 122, 150]
14	183.0	69-125	9	-9	[146]	[119, 120, 121, 122, 123, 124, 125, 126, 154]

Your sequence is probably: 11.3

Al·lel 12.1:

Alelle	Score	STR postGaps	Gap count	Q-T	Mut pos	Gap pos
10	176.0	69-109	9	9	[139]	[111, 112, 113, 114, 115, 116, 117, 118, 119]
11	184.0	69-113	5	5	[139]	[115, 116, 117, 118, 119]
12	192.0	69-117	1	1	[139]	[119]
13	191.0	69-121	3	-3	[142]	[120, 121, 122]
14	187.0	69-125	7	-7	[146]	[120, 121, 122, 123, 124, 125, 126]

Your sequence is probably: 12.1

Al·lel 12 amb mutació interna:

Alelle	Score	STR postGaps	Gap count	Q-T	Mut pos	Gap pos
10	177.0	69-113	8	8	[138]	[74, 75, 77, 78, 115, 116, 117, 118]
11	185.0	69-117	4	4	[138]	[74, 75, 77, 78]
12	192.0	69-117	0	0	[74, 138]	[]
13	188.0	69-121	4	-4	[74, 142]	[119, 120, 121, 122]
14	184.0	69-125	8	-8	[74, 146]	[119, 120, 121, 122, 123, 124, 125, 126]

Your sequence is probably: 12

Al·lel 12 amb deleció pre STR:

Alelle	Score	STR postGaps	Gap count	Q-T	Mut pos	Gap pos
10	175.0	69-109	9	7	[138]	[55, 111, 112, 113, 114, 115, 116, 117, 118]
11	183.0	69-113	5	3	[138]	[55, 115, 116, 117, 118]
12	191.0	69-117	1	-1	[138]	[55]
13	187.0	69-121	5	-5	[142]	[55, 119, 120, 121, 122]
14	183.0	69-125	9	-9	[146]	[55, 119, 120, 121, 122, 123, 124, 125, 126]

Your sequence is probably: 11.3

Al·lel 12 amb deleció interna:

Alelle	Score	STR postGaps	Gap count	Q–T	Mut pos	Gap pos
10	178.0	69–112	7	7	[137]	[84, 86, 87, 114, 115, 116, 117]
11	186.0	69–116	3	3	[137]	[84, 86, 87]
12	191.0	69–117	1	−1	[138]	[84]
13	187.0	69–121	5	−5	[142]	[84, 119, 120, 121, 122]
14	183.0	69–125	9	−9	[146]	[84, 119, 120, 121, 122, 123, 124, 125, 126]

Your sequence is probably: 11.3

Al·lel 12 amb inserció post STR:

Alelle	Score	STR postGaps	Gap count	Q–T	Mut pos	Gap pos
10	175.0	69–109	10	10	[138]	[111, 112, 113, 114, 115, 116, 117, 118, 151, 152]
11	183.0	69–113	6	6	[138]	[115, 116, 117, 118, 151, 152]
12	191.0	69–117	2	2	[138]	[151, 152]
13	187.0	69–121	6	−2	[142]	[119, 120, 121, 122, 155, 156]
14	183.0	69–125	10	−6	[146]	[119, 120, 121, 122, 123, 124, 125, 126, 159, 160]

Your sequence is probably: 12.2

Al·lel 12 amb inserció pre STR:

Alelle	Score	STR postGaps	Gap count	Q–T	Mut pos	Gap pos
10	175.0	71–111	10	10	[140]	[31, 32, 113, 114, 115, 116, 117, 118, 119, 120]
11	183.0	71–115	6	6	[140]	[31, 32, 117, 118, 119, 120]
12	191.0	71–119	2	2	[140]	[31, 32]
13	187.0	71–123	6	−2	[144]	[31, 32, 121, 122, 123, 124]
14	183.0	71–127	10	−6	[148]	[31, 32, 121, 122, 123, 124, 125, 126, 127, 128]

Your sequence is probably: 12.2

Al·lel 12 amb inserció interna:

Alelle	Score	STR postGaps	Gap count	Q–T	Mut pos	Gap pos
11	183.0	69–115	6	6	[140]	[99, 100, 117, 118, 119, 120]
12	191.0	69–119	2	2	[140]	[99, 100]
13	193.0	69–121	2	−2	[142]	[99, 100]
14	189.0	69–125	6	−6	[146]	[99, 100, 123, 124, 125, 126]
15	185.0	69–129	10	−10	[150]	[99, 100, 123, 124, 125, 126, 127, 128, 129, 130]

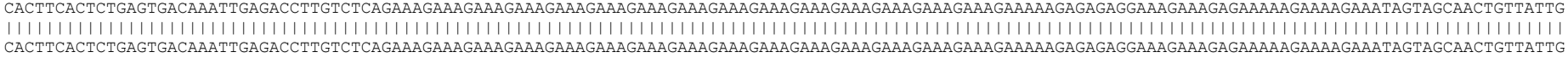
Your sequence is probably: 12.2

Annex 4: Alineaments exemple (HG002)

```
TPOX:
(8)
GACTGGCACAGAACAGGCACTTAGGGAACCCTCACTGAATGAATGAATGAATGAATGAATGAATGTTTG
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
GACTGGCACAGAACAGGCACTTAGGGAACCCTCACTGAATGAATGAATGAATGAATGAATGAATGTTTG


D3S1358:
(16)
ATGTATCTATCTGTCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATGAGACAGGGTCTTGCTCTGTC
||||||||||||.||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
ATGTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATGAGACAGGGTCTTGCTCTGTC


FGA:
(20)
CAGATTAAACTGTAACCAAAATAAAATTAGGCATATTTACAAGCTAGTTTCTTTCTTTCTTTTTTCTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTCCTTCCTTCCTTTCTTCCTTTCTTTTTT
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
CAGATTAAACTGTAACCAAAATAAAATTAGGCATATTTACAAGCTAGTTTCTTTCTTTCTTTTTTCTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTTTCTCCTTCCTTCCTTTCTTCCTTTCTTTTTT


CSF1PO:
(10)
AGGTTGCTAACCACCCTGTGTCTCAGTTTTCCTACCTGTAAAATGAAGATATTAACAGTAACTGCCTTCATAGATAGAAGATAGATAGATTAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGGAAGTACTTAGAACAGGGTC
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
AGGTTGCTAACCACCCTGTGTCTCAGTTTTCCTACCTGTAAAATGAAGATATTAACAGTAACTGCCTTCATAGATAGAAGATAGATAGATTAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGGAAGTACTTAGAACAGGGTC

TGACACAGGAAATGCTGTCCAAGTGTGCACCAGGAGATAGTATCTGAGAAGGCTCAGTCTGGCACCATGTGGGTTGGGTGGGAACCTGGAGGCTGGAGAATGGGCT
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
TGACACAGGAAATGCTGTCCAAGTGTGCACCAGGAGATAGTATCTGAGAAGGCTCAGTCTGGCACCATGTGGGTTGGGTGGGAACCTGGAGGCTGGAGAATGGGCT


D5S818:
(11)
TTACGTAATATTTTGAAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGAGGTATAAATAAGGATACAGATAA
|||.||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
TTATGTAATATTTTGAAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGAGGTATAAATAAGGATACAGATAA
```

```
D7S820:
(12)
GAGTTATTTTAAGGTTAATATATATAAAGGGTATGATAGAACACTTGTCATAGTTTAGAACGAACTAACGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGACAGATTGATAGTTTTTTTTTATCTCACTAAATAG
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||.||||||||||||||
GAGTTATTTTAAGGTTAATATATATAAAGGGTATGATAGAACACTTGTCATAGTTTAGAACGAACTAACGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGACAGATTGATAGTTTTTTTTAATCTCACTAAATAG

TCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT
|||||||||||||||||||||||||||||||||||||||||
TCTATAGTAAACATTTAATTACCAATATTTGGTGCAATTCT


D8S1179:
(13)
TATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATTCCCCACAGTGAAAATAATCTACAGGATAGGTAAATAAATTAAGGCATATTCACGCAATGGGATACGATACAGTGA
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
TATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATTCCCCACAGTGAAAATAATCTACAGGATAGGTAAATAAATTAAGGCATATTCACGCAATGGGATACGATACAGTGA


TH01:
(9)
CCAGCCTGGCCCACACAGTCCCCTGTACACAGGGCTTCCGAGTGCAGGTCACAGGGAACACAGAC--TCCATGG-TGAATGAATGAATGAATGAATGAATGAATG-AATGAATGAGGGAAATAAGGGAGGAACAGGCCAATGGGAATCACC
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||--|..|||.-||||||||||||||||||||||||||||||-|||||||||||||||||||||||||||||||||||||||||||||
CCAGCCTGGCCCACACAGTCCCCTGTACACAGGGCTTCCGAGTGCAGGTCACAGGGAACACAGACAATGAATGAATGAATGAATGAATGAATGAATGAATGAATGGAATGAATGAGGGAAATAAGGGAGGAACAGGCCAATGGGAATCACC


VWA:
(16)
TGACTTGGATTGATCTATCTGTCTGTCTGTCTGTCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCCATCTAT
||||||||||||||||||||.|||.|||.|||.|||||||||||||||||||||||||||||||||||||||||||||||||||||
TGACTTGGATTGATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCCATCTAT


D13S317:
(11)
GAGTTCATTTCTTTAGTGGGCATCCGTGACTCTCTGGACTCTGACCCATCTAACGCCTATCTGTATTTACAAATACATTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCAATCAATCATCTATCTATCTTTCTG
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
GAGTTCATTTCTTTAGTGGGCATCCGTGACTCTCTGGACTCTGACCCATCTAACGCCTATCTGTATTTACAAATACATTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCTATCAATCAATCATCTATCTATCTTTCTG


D16S539:
(11)
TGTACAAGTGCCAGATGCTCGTTGTGCACAAATCTAAATGCAGAAAAGCACTGAAAGAAGAATCCAGAAAACCACAGTTCCCATTTTTATATGGGAGCAAACAAAGGCAGATCCCAAGCTCTTCCTCTTCCCTAGATCAATACAGACAGACAG
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
TGTACAAGTGCCAGATGCTCGTTGTGCACAAATCTAAATGCAGAAAAGCACTGAAAGAAGAATCCAGAAAACCACAGTTCCCATTTTTATATGGGAGCAAACAAAGGCAGATCCCAAGCTCTTCCTCTTCCCTAGATCAATACAGACAGACAG

ACAGGTGGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATATCATTGAAAGACAAAACAGAGATGGATGATA
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
ACAGGTGGATAGATAGATAGATAGATAGATAGATAGATAGATAGATAGATATCATTGAAAGACAAAACAGAGATGGATGATA
```

D18S51:
(16)



D21S11:
(31.2)