**Resumen metodológico del estudio del Genotipado de la colección de guisante (IAS)**

1. **Descripción de los datos recibidos de DiversityArray**

* Tras secuenciar nuestra colección de guisante por DArTSeq, DiversityArray analizo los datos brutos de secuenciación para identificar los marcadores moleculares.
* Diversity array ha analizado mis muestras junto con las de Eleonora lo que permitirá comparar sus resultados con los míos (gracias a los nombres que tienen en común).
* Diversity array ha analizado por separado lo que llama los SilicoDArT markers que son los fragmentos de ADN secuenciados (marcadores largos) y los SNP que han sido identificado en cada uno de estos SilicoDarT markers. Un mismo SilicoDarT puede contener varios SNP.
* Después compararon y alinearon nuestros datos con 3 bases de datos diferentes:

1. El genoma de referencia del guisante
2. Genome de referencia de *Medicago truncatula,* del garbanzo y genoma provisional del guisante (incompleto)
3. La base de datos nr y de los procariotas del NCBI para descartar contaminantes

* Al final nos mandaron los siguientes ficheros:
  + Report\_DPea19-4080\_1\_moreOrders\_SilicoDArT\_1.csv del 11/04/2019: SilicoDArT alineados con *Medicago truncatula* y *Cicer arietinum*.
  + Report\_DPea19-4080\_1\_moreOrders\_SilicoDArT\_1.csv del 27/02/2020: SilicoDArT alineados con el genoma de referencia del guisante.
  + DPea19\_4080\_1moreOrder\_SilicoDArTsBLAST.csv del 12/04/2019: resultado del BLAST de nuestros datos con las bases de datos del NCBI para detectar posibles contaminaciones.
  + Report\_DPea19-4080\_1\_moreOrders\_SNP\_2.csv del 27/2/2020: SNP markers alineados con *Medicago truncatula* y *Cicer arietinum* en “2-row format” es decir que cada SNP esta descrito en 2 líneas, 1 con la información del alelo de referencia (el principal) con presencia /ausencia score y otra con la información del alelo alternativo.
  + Report\_DPea19-4080\_1\_moreOrders\_SNP\_3.csv del 27/2/2020: SNP markers alineados con el genoma de referencia del guisante en “2-row format” es decir que cada SNP esta descrito en 2 líneas, 1 con la información del alelo de referencia (el principal) con presencia /ausencia score y otra con la información del alelo alternativo.
  + Report\_DPea19-4080\_1\_moreOrders\_SNP\_mapping\_2: SNP markers alineados con *Pisum sativum* descritas en una sola línea.
  + Report\_DPea19-4080\_1\_moreOrders\_SNP\_mapping\_2\_old: SNP markers alineados con *Medicago truncatula* y *Cicer arietinum* descritas en una sola línea.

1. **Limpieza de las bases de datos SilicoDArT y SNP**

* Como Diversity Array analizó mis secuencias junto con los datos de Eleonora, tengo que eliminar la información de sus genotipos antes de empezar el análisis.
* Además, hay que recalcular las estimaciones de calidad, call, PIC, etc. ya que han sido determinado con los datos conjuntos (Eleonora y mías).
* También sería bueno juntar las diferentes anotaciones en el mismo fichero para tener toda la información a mano.
* La limpieza de la base de datos se realiza con el programa R utilizando el script llamado Matrixpreparation.R.
* En primer lugar, he creado una carpeta para poner todos los ficheros intermedios generado durante la limpieza de la base de datos que he llamado “Database genotipado”.

**B.1. Limpieza de la base de Datos de marcadores Silico DArT**

* Para empezar limpio la base de datos de SilicoDArT. Para ello, he pegado los 3 ficheros .csv generado por Diversity Array sobre estos marcadores.
* Para diferenciar los ficheros les he renombrado así:
  + DPea19-4080\_SilicoDArT\_Pisum.csv: base de datos alineada con el genoma de referencia del guisante = Report\_DPea19-4080\_1\_moreOrders\_SilicoDArT\_1.csv del 27/02/2020
  + DPea19-4080\_1\_SilicoDArT.csv: SilicoDArT alineados con *Medicago truncatula* y *Cicer arietinum* = Report\_DPea19-4080\_1\_moreOrders\_SilicoDArT\_1.csv del 11/04/2019
  + DPea19\_4080\_1\_SilicoDArt\_BLAST.csv: comparación de nuestros datos con las bases de datos del NCBI = DPea19\_4080\_1moreOrder\_SilicoDArTsBLAST.csv

**B.1.1 Eliminar las secuencias de Eleonora**

* Todos los genotipos de las muestras de Eleonora empiezan con “Gen” seguido de un número de identificación mientras que los míos son identificados por su Numero de referencia en la colección IAS.
* En consecuencias, para eliminar de la base de datos todo lo relativo a las secuencias de Eleonora:
  + Abrimos el fichero DPea19-4080\_SilicoDArT\_Pisum.csv
  + filtramos la base de datos todas las columnas que contienen “Gen” con la función “select” del paquete de R “dplyr”.
  + Usando los comandos:

# abrir la base de datos y elimina los genotipos de Eleonora (todos empiezan por "Gen")

Gen<-read.csv ("DPea19-4080\_SilicoDArT\_Pisum.csv", skip=6, header=TRUE) %>%

select (-c(starts\_with("gen"))) %>% arrange (CloneID) %>% select (2:3,1, 4:7,8:339)

**B.1.2. Juntar las tres anotaciones en un solo fichero**

* Para ello abrimos los otros dos ficheros (DPea19-4080\_1\_SilicoDArT.csv y DPea19\_4080\_1\_SilicoDArt\_BLAST.csv) en R.
* Seleccionamos las columnas que contienen la información de los marcadores y las anotaciones.
* Se junta las 3 bases de datos con la función “full-join” del paquete de R “dplyr”.
* Como algunos marcadores no tienen el mismo nombre en los diferentes ficheros por una razón desconocida utilizamos la secuencia de cada marcador contenido en la columna “Allele\_Sequence” para fusionar las bases de datos.
* Tras estos pasos se guarda la base de datos en formato .csv (genotipos\_SilDArT\_coleccionGWAS.csv).

# abre y selecciona los datos utiles del archivo con los alineamientos contra el genoma de Medicago y garbanzo

P1<-read.csv ("DPea19-4080\_1\_SilicoDArT.csv", skip=6, header=TRUE) %>%

select (2,1,4,5,7,12,13,15) %>%

rename (Chr\_Medicago=Chrom\_Medicago\_v4.0.2, Pos\_Medicago=ChromPos\_Medicago\_v4.0.2, Eval\_Medicago=AlnEvalue\_Medicago\_v4.0.2, Chr\_Chickpea=Chrom\_Chickpea\_cdcfrontier\_v03, Pos\_Chickpea=ChromPos\_Chickpea\_cdcfrontier\_v03, Eval\_Chickpea=AlnEvalue\_Chickpea\_cdcfrontier\_v03) %>%

arrange (CloneID)

# abre y selecciona los datos utiles del archivo con los BLASTs contra la base de datos nr y de procriota

Blast<- read.csv ("DPea19\_4080\_1\_SilicoDArt\_BLAST.csv", skip=3, header=TRUE) %>%

select (2,1,4,7,8,11) %>%

arrange (CloneID) %>%

rename (Blast\_nr=Chrom\_nt, Eval\_nr=AlnEvalue\_nt,Blast\_proc=Chrom\_ref\_prok\_rep\_genomes, Eval\_proc=AlnEvalue\_ref\_prok\_rep\_genomes)

# Union de les tres bases de datos...

# Las tres tienen el mismo numero de secuencias pero no siempre con el mismo nombre

# asi que se hace la union según la secuencia (AlleleSequence) y no el nombre (CLoneID)

P<- P1 %>% full\_join (Blast, by="AlleleSequence") %>% full\_join (Gen, by="AlleleSequence") # Une las tres bases de datos

write.csv (P,"genotipos\_SilDArT\_coleccionGWAS.csv", row.names=FALSE) # guarda la base de datos un archivo .csv

**B.1.3. Calcular parámetros de polimorfismos (PIC, Call rate, % heterocidad,etc.)**

* La evaluación de los marcadores Silico\_DArTs en cada genotipo se estimó con:
  + “1”: presencia del marcador.
  + “0”: ausencia del marcador.
  + “-“: secuencia encontrada en el genotipo pero con un numero de copia demasiado bajo para asignarle un “1” con seguridad – suele representar heterocigoto.
* Para determinar los diferentes parámetros básicos hay que contar el numero de “1” (presencia), “0” (ausencia) y “-“ (heterocigoto).
* No he conseguido hacer esto en R así que lo hago en Excel con la función “CONTAR.SI”.
* Tras contar el numero de veces que aparece “1”, “0” y “-“ en los genotipos por cada marcador en Excel, he salvado el fichero en formato .xlsx y .csv con el nombre “gen\_SilDArT\_coleccionGWAS.csv”.
* Para calcular los diferentes parámetros importantes se utiliza las siguientes formulas:
  + CallRate =
  + One Ratio = MAF =
  + % Heterocigocidad = Het=
  + PIC =
* Los cálculos de estos parámetro se hizo en R con las fórmulas anteriores con la función “mutate” del paquete de R “dplyr”.
* Después se salvó el fichero en formato .csv con el nombre “gen\_SilDArT\_coleccionGWAS.csv”
* Se utilizó los siguientes comandos en R:

G<-read.csv2 ("gen\_SilDArT\_coleccionGWAS.csv", header =TRUE, row.names=NULL) %>%

mutate (CallRate= ((Presencia + Ausencia)/Total),OneRatio= Presencia/Total, Missing=Missing/Total,PIC=(2\*OneRatio\*(1-OneRatio))) %>%

rename (Het=Missing,MAF=OneRatio)

write.csv (G, "gen\_SilDArT\_coleccionGWAS.csv", row.names=FALSE)

**B.1.4. Filtrar marcadores de mala calidad y alelos raros**

* El filtrado se hace en R con la función “filter” del paquete de R “dplyr”
* Los parámetros de filtrados fueron los siguientes:
  + Presencia ≥ 1 (marcadores detectado al menos en uno de mis genotipos)
  + MAF ≥ 0.05 (se conserva solo los marcadores que están presentes en al menos 5% de la colección)
  + %Het ≤ 0.1 (se conserva solo los marcadores con menos de 10% de heterocigosis)
* Con este filtrado se elimina por completo (o casi) todas las secuencias contaminantes de humano, y otros animales encontrado según Diversity Array ya que la contaminación viene de una o dos muestras de Eleonora.
* Tras filtrar se guarda el fichero en formato .csv con el nombre de “gen\_SilDArT\_colGWAS\_Filtered.csv”.
* El filtrado se hizo con estos comandos:

Db<- G %>% filter (Presencia >=1, MAF>=0.05, Het<0.1)

write.csv (Db,"gen\_SilDArT\_colGWAS\_Filtered.csv", row.names=FALSE)

* En los diferentes pasos de la limpieza obtuvimos:
  + Antes de limpiar cada base de datos contenían: **66643** marcadores
  + Tras fusionar las bases de datos obtuvimos: **66685** marcadores (no se porque hay más marcadores…)
  + Tras el filtrado tenemos: **27616** marcadores de buena calidad
* De estos marcadores **6259** (22% de los marcadores) no se encontraron en el genoma del guisante (criterio de BLAST de Diversity array E-value= 5e-05 y min %identify=80%)
* Antes de seguir con el análisis de los datos voy a intentar mapear estos marcadores

**B.1.5. Mapeo de los marcadores sin localización sobre el genoma de referencia**

* Para mapear estos marcadores voy a realizar un Blast frente al genoma de referencia del guisante
* Para ello primero hay que crear un fichero con el nombre y la secuencia de todos los marcadores no mapeado. Esto lo hizo en R con los siguientes comandos:

Db<- read.csv ("gen\_SilDArT\_colGWAS\_Filtered.csv", header=TRUE)

Nomap<- Db %>% filter (Chrom\_Pisum\_sativum\_v1a =="")

#crear un fichero con las secuencias de los marcadores sin información de mapa

FASTA <- Nomap %>% select (1:2)

write.table (FASTA,"nomapseq.txt", row.names=FALSE)

#crear un archivo con las secuencias trimmed de los marcadores sin información de mapa de guisante

Ftrim<- Nomap %>% select (1,3)

write.table (Ftrim, "nomapseqtrimmed.txt", row.names=FALSE)

* Con estos comandos creamos 2 ficheros con la información de los marcadores no mapeado con su secuencia completa o trimmed (elimina cebadores, y parte de baja calidad)
* Después abrimos estos ficheros .txt en Excel y Edit+ para transformarlos en formato fasta para que pueda servir para el BLAST.
* El análisis BLAST se hará en local así que me descargue la base de datos genómica del guisante de la página del INRA así como el programa MAGICBLAST y BLAST del NCBI. El MAGICBLAST es un nuevo programa del NCBI especialmente diseñado para mapear Secuencia de RNAseq y GBS sobre genomas de referencias
* Tras descargarme la base de datos y los programas se trabaja en el “símbolo de sistema” de Windows (antiguo MS-DOS)
* Para utilizar el magicblast se hace
  + Formatear la base de Datos con el comando

>makeblastdb -in <nombre base de datos.fa> -out <nombre de la base de datos> -parse\_seqids -dbtype nucl

* + Hacer el BLAST

>magicblast -query <nombre fichero con las secuencias a “blastear”> -db <nombre de la base de datos de referencia> -out<nombre del fichero output aquí:MagicBlast\_nomap.txt>

* También hice un local blast con la función blastn-short specifico por secuencias cortas de ADN.
* Para ello use como valor límites: E-value 1e-4 y min%Identity 80
* La línea de comando fue la siguiente:

>blastn -task blastn-short -query <nombre fichero query> -out <nombre fichero output aqui: Blastn\_short\_nomap.csv> -perc\_identity 80 -evalue 0.0001 -outfmt 10 (=formato .csv) -max\_target\_seqs 1 -max\_hsps 1

* Una vez realizado los blast con las secuencias completas y Trimmed con los dos métodos, he comparado los resultados juntándolos todos en un mismo fichero en R
* Tras compararlo se ve que los resultados de los diferentes BLAST son idénticos o casi.
* Tras eliminar las secuencias repetidas conservo los datos del Blastn\_short\_trimed que encuentran más hits.
* Finalmente hay que añadir los datos de mapa nuevo a la base de datos.
* Esto lo hago en R y tengo que seguir estos pasos:
  + Seleccionar los marcadores que no estaban mapeado
  + A estos marcadores se añade la información obtenido por los BLAST que he hecho
  + Añadir esta información a la base de datos completos.
  + Finalmente guardo la base de datos en formato .csv con el nombre “gen\_SilDArT\_colGWAS\_Filteredmapped.csv”
  + Estos pasos se hicieron con los siguientes comandos:

#abre base de datos completa filtrada y crear 2 base de datos uno con los datos mapeados y otros con los datos sin info de mapa

Db<- read.csv ("gen\_SilDArT\_colGWAS\_Filtered.csv", header=TRUE) %>%

rename (Chrom\_Pisum=Chrom\_Pisum\_sativum\_v1a,Pos\_Pisum=ChromPos\_Pisum\_sativum\_v1a,Eval\_Pisum=AlnEvalue\_Pisum\_sativum\_v1a)

DbMap<- Db %>% filter (Chrom\_Pisum !="")

Dbnomap<- Db %>% filter (Chrom\_Pisum =="")%>% arrange (CloneID)

#Abre el archivo con el resultado BLASTn-short-trimmed

Bt<-read.csv("Blast-short\_nomaptrimmed.csv",header=TRUE)%>%

rename (CloneID=queryacc,Identity=X.identity,Bs\_Pisumchr=reference.acc, BS\_Position=reference.start)%>%

select (1:2,9,11) %>% arrange (CloneID)

# junta los dos archivos

DbnomapBt<- Dbnomap %>% select (-c(4:6)) %>%

left\_join (Bt, by="CloneID") %>%

select (1:3,350:352,4:349) %>%

rename (Chrom\_Pisum=Bs\_Pisumchr,Pos\_Pisum=BS\_Position,Eval\_Pisum=e\_value)

#añadir la base de datos mapeado nuevamente a la base de datos mapeado por DiversityArray

Dbnew<- rbind(DbMap,DbnomapBt) %>% arrange(CloneID)

#guardar la base de datos completa

write.csv (Dbnew,"gen\_SilDArT\_colGWAS\_Filteredmapped.csv", row.names=FALSE)

* Tras este intento de mapearlo, queda **2104** marcadores sin localizar en el genoma del guisante lo que representa unos 7.6% de los marcadores. Por lo que hemos conseguido mapear a unos 4000 marcadores.
* Con esto tendríamos ya la base de datos limpiada definitiva (a falta de quitar los marcadores con Pic>0.01 es decir los no polimórficos).
* Para poder usarlo en los diferentes programas TASSEL, Plink, etc… hay que cambiar el formato del fichero.
* En total y después de filtrar en TASSEL hay un total de **24279** markers de los cuales **19514** SilicoDArT maped to pea chromosome (80.4%), **2703** mapped to unanchored contigs (11.1%) and **2062** unmapped markers (8.5%).

**B.2. Limpieza de la base de datos de marcadores SNPs**

* Tras limpiar la base de datos de los marcadores tipo SilicoDArT, hago lo propio con las bases de datos SNP
* Para ello he creado una nueva carpeta llamada “SNPs” dentro de la carpeta “Database Genotipado”.
* Dentro de esta carpeta he pegado los 2 archivos .csv generado por Diversity Array con los datos SNP en 1 linea.
* Para diferenciar y simplificar estos dos archivos les he renombrado así:
  + DPea19-4080\_SNP\_Pisum.csv: base de datos alineada con el genoma de referencia del guisante = Report\_DPea19-4080\_1\_moreOrders\_SNP\_mapping\_2 del 27/02/2020
  + DPea19-4080\_SNP.csv: SilicoDArT alineados con *Medicago truncatula* y *Cicer arietinum* = Report\_DPea19-4080\_1\_moreOrders\_SNP\_mapping\_2\_old del 11/04/2019

**B.2.1 Eliminar las secuencias de Eleonora de la base de datos SNP**

* Todos los genotipos de las muestras de Eleonora empiezan con “Gen” seguido de un número de identificación mientras que los míos son identificados por su Número de referencia en la colección IAS.
* Para eliminar de la base de datos SNP todo lo relativo a las secuencias de Eleonora editamos el script creado para los marcadores silicoDArT para cambiar el nombre del archivo a abrir y modificar.
* Como los dos archivos tienen en realidad la misma información y la única información importante es la localización de los marcadores sobre el genoma de guisante, solo uso el archivo DPea19-4080\_SNP\_Pisum.csv y me salto la etapa de juntar las anotaciones de los dos ficheros.
* Guardo el archivo filtrado con el nombre “DbSNP\_coleccionGWAS.csv”. Este archivo contiene **54269** SNP y 325 genotipo.
* Nota: por una razón que desconozco hay menos marcadores de tipo SNP que de tipo SilicoDArT.

**B.2.2. Calcular parámetros de polimorfismos (Call rate, AlleleFrequency, % heterocidad,etc.) y filtrado de la base de datos**

* La evaluación de los marcadores SNP en cada genotipo se estimó con:
  + “0”: alelo de referencia homocigoto.
  + “1”: alelo alternativo homocigoto.
  + “2”: alelo heterocigoto
  + “-“: ausencia del marcador SilicoDArT que contiene este SNP
* Para determinar los diferentes parámetros básicos hay que contar el número de “0” (presencia), “1” (ausencia) y “2“ (heterocigoto).
* No he conseguido hacer esto en R así que lo hago en Excel con la función “CONTAR.SI”.
* Tras contar el número de veces que aparece “1”, “0”, “2” y “-“ en los genotipos por cada marcador en Excel, he salvado el fichero en formato.csv con el mismo nombre (“DbSNP\_coleccionGWAS.csv”).
* Para calcular los diferentes parámetros importantes se utiliza las siguientes formulas:
  + CallRate =
  + One RatioRef = AFRef =
  + One RatioSNP = AFSNP =
  + % Heterocigocidad = Het=
* Los cálculos de estos parámetros se hicieron en R con las fórmulas anteriores con la función “mutate” del paquete de R “dplyr” modificando el script preparado para los marcadores SilicoDArT.
* En R he podido calcular las frecuencias alélicas para el alelo de referencia y el alelo alternativo, pero no siempre el de referencia es el principal por lo que no sé cómo calcular el MAF (frecuencia alélica del alelo minoritario) para filtrar.
* Por lo tanto, voy a filtrar la base de datos en R para quitar los marcadores con >20% missing data (call rate>0.8) y con una tasa de heterocigozidad ≤ 0.1.
* Después quitaré los marcadores con MAF<0.05 en TASSEL.
* El filtrado por missing data y heterocigozidad deja **28911** SNP.
* Después se guardó el fichero en formato .csv con el nombre “DbPea\_SNP\_coleccionGWAS\_filtered.csv”
* El ultimo filtro (MAF<0.05) se hará en TASSEL tras transformar el archivo en formato hapmap. Esto deja **11511** SNP (salvado como “PisumSNP\_filtered.hmp.txt”)

**B.2.3. Mapear los marcadores sin alinear sobre el genoma de guisante**

* De los **11511** SNP marcadores obtenidos tras filtrar, del ellos, **9363** SNP mapearon sobre cromosoma del guisante (81.3% marcadores), **1054** SNP (9,2% de los marcadores) mapearon sobre unaligned scaffolds and **1094** (9,5% de los marcadores) no se encontraron en el genoma del guisante (criterio de BLAST de Diversity array E-value= 5e-05 y min %identify=80%)
* Antes de seguir con el análisis de los datos voy a intentar mapear estos marcadores
* Como los SNP son contenido dentro de los Silico-DArT, para mapear los marcadores no-mapeado voy a comparar la matrix de marcadores SNP con la info que tenemos de los Silico-DArT para completar con los datos obtenidos para ello.
* De esta forma he añadido 206 marcadores mapeado sobre los cromosomas de guisante y 78 marcadores mapeados sobre super contigs. El resto sigue sin información probablemente porque estos marcadores no eran polimórficos a nivel Silico-DArT.
* Para el resto voy a hacer un blast sobre el genoma del guisante del mismo modo que hice con los Silico-DArT.
* El primer paso es obtener las secuencias de los marcadores sin info de mapa y transformarlos en formato fasta. Para ello he comparado la lista con los datos incluido en el archivo “DbPea\_SNP\_coleccionGWAS\_filtered.csv” y filtrado para tener solo una copia de cada clone junto con su secuencia.
* Los codigos utilizados fueron:

list2<- read.delim ("SNPlist\_unmapped2.txt", header=TRUE)

ref2<- read.csv("DbPea\_SNP\_coleccionGWAS\_filtered.csv", header=TRUE) %>% select (1:4)

listmap2 <- left\_join (list2,ref2,by="CloneID") %>% group\_by (CloneID) %>% slice\_sample(n=1) # con estos comendos he juntado la lista sin mapa con las secuencias de los clones y conservado solo uns instancia en caso de clones con varios marcadores

Symbol= ">"

Nomap<- listmap2 %>% select (1,3) %>% mutate (S1=Symbol) %>% select (3,1,2)

write.table (Nomap,"SNP\_nomapseq.txt", row.names=FALSE)

Nomaptrim<- listmap2 %>% select (1,4) %>% mutate (S1=Symbol) %>% select (3,1,2)

write.table (Nomaptrim,"SNP\_nomapseq\_trim.txt", row.names=FALSE)

* Después he hecho el local BLAST frente a las secuencias de referencias del guisante igual que para los SilicoDArT.
* Al final tras mapear tengo
  + 10125 SNP mapeado sobre cromosomas del guisante (88%de los marcadores)
  + 1155 SNP mapeado sobre contigs (10% de los marcadores)
  + 231 no mapeados (2% de los marcadores)