```
library(dplyr)

rladies_global %>%
  filter(city == 'Your city')
```

R en Genética: Clusterización de Haplotipos

¡Hola!

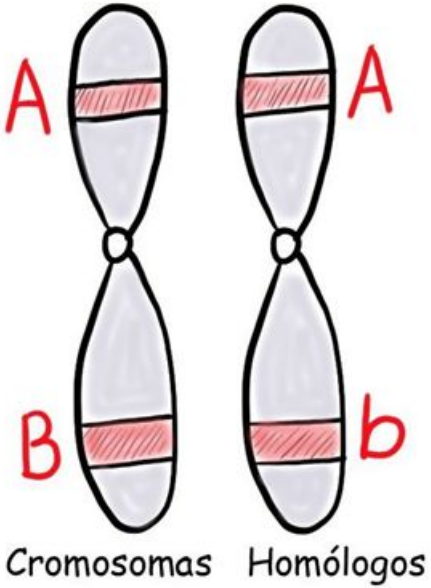
Soy Tatiana Parlanti

Estudiante avanzada de Lic. en Ciencias Básicas
con orientación en Matemática (FCEN-UNCUYO)

tsparlanti@gmail.com

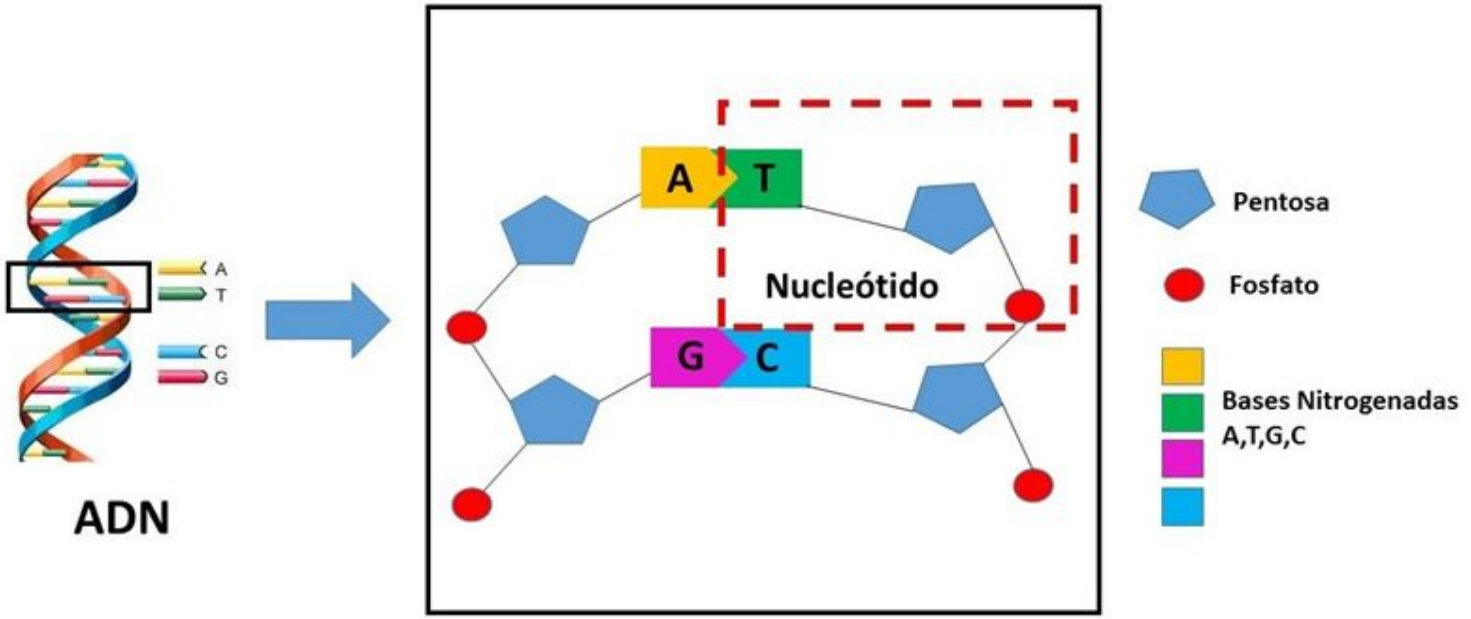
1. Conceptos básicos de Genética

Conceptos básicos
de Genética



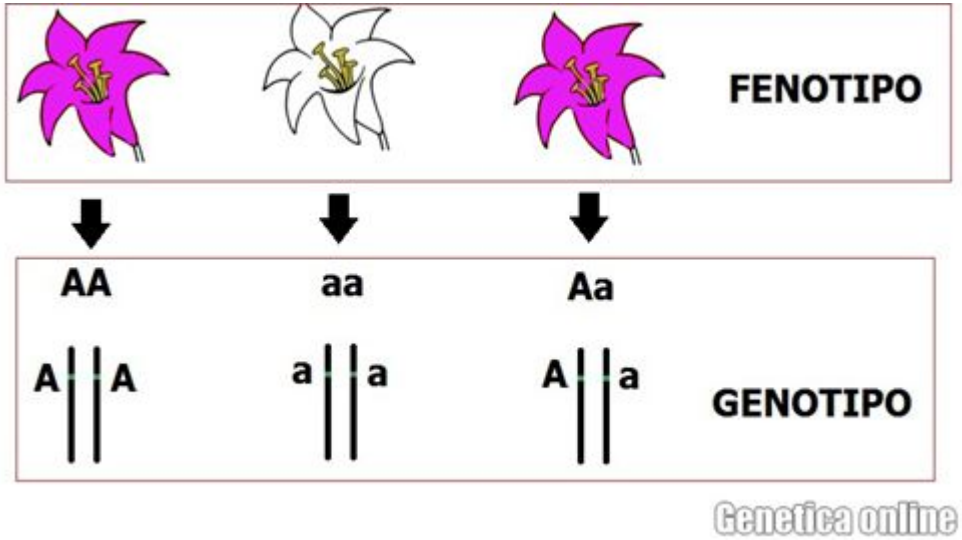
- Gen
- Cromosoma
- Locus y loci
- Alelo

Conceptos básicos
de Genética



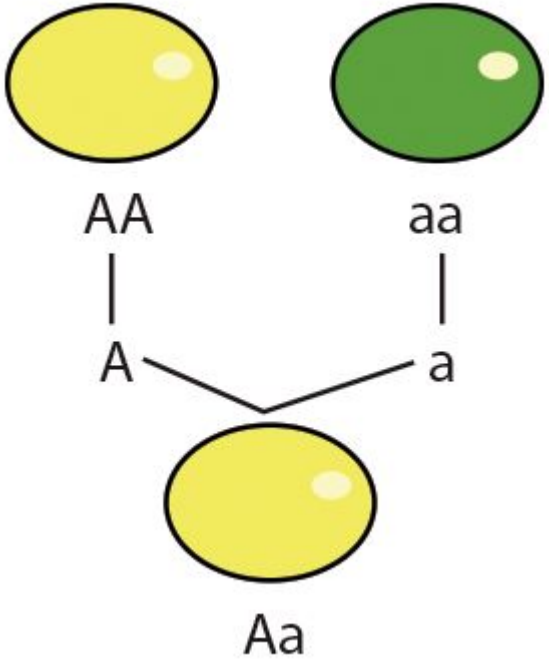
Estructura de ADN y Nucleótidos

Conceptos básicos
de Genética

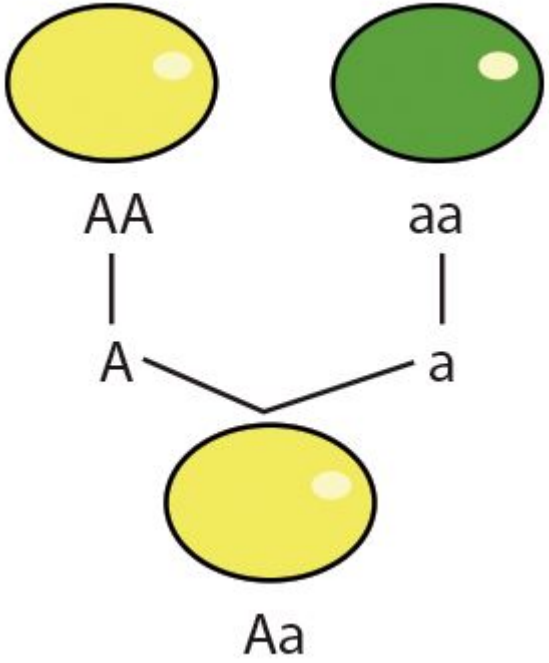


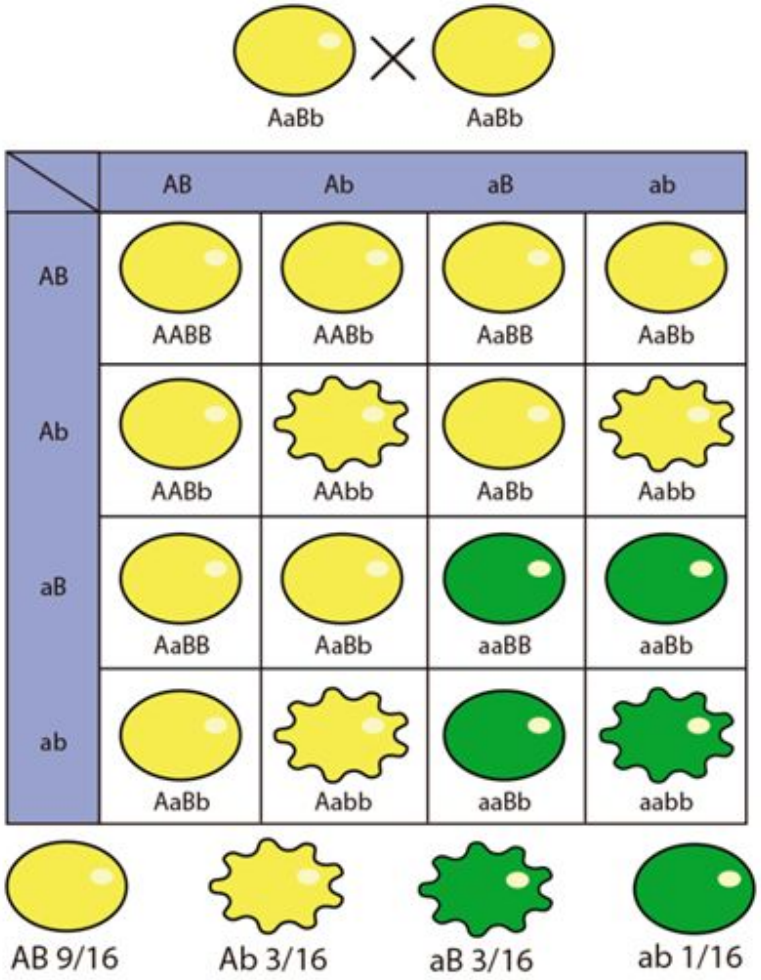
2. Clusterización de haplotipos

Mendel



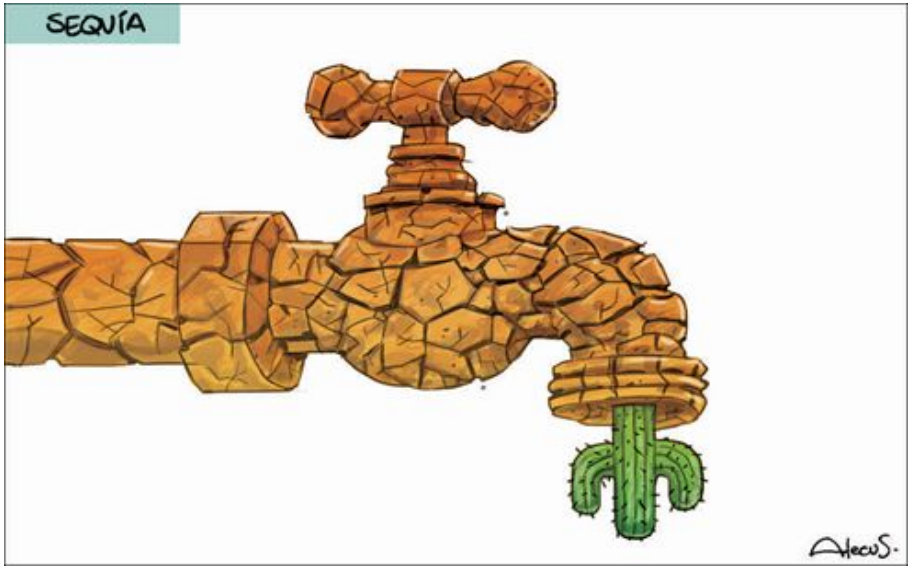
Mendel


AA
aa
A
a
Aa

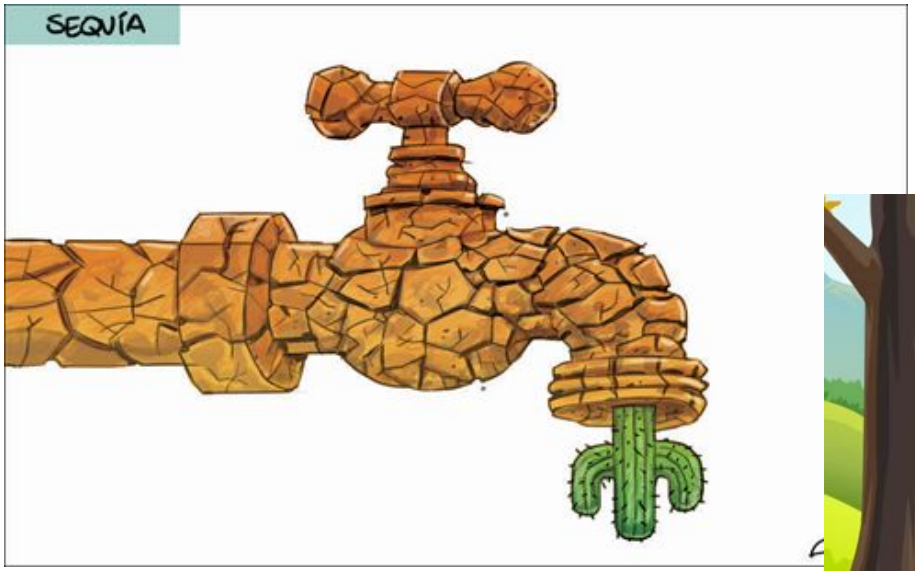

AaBb
AaBb
AB
Ab
aB
ab
AABB
AABb
AaBB
AaBb
AABb
AAbb
AaBb
Aabb
AaBB
AaBb
aaBB
aaBb
AaBb
Aabb
aaBb
aabb
AB 9/16
Ab 3/16
aB 3/16
ab 1/16

Caracteres de tipo
Cuantitativo

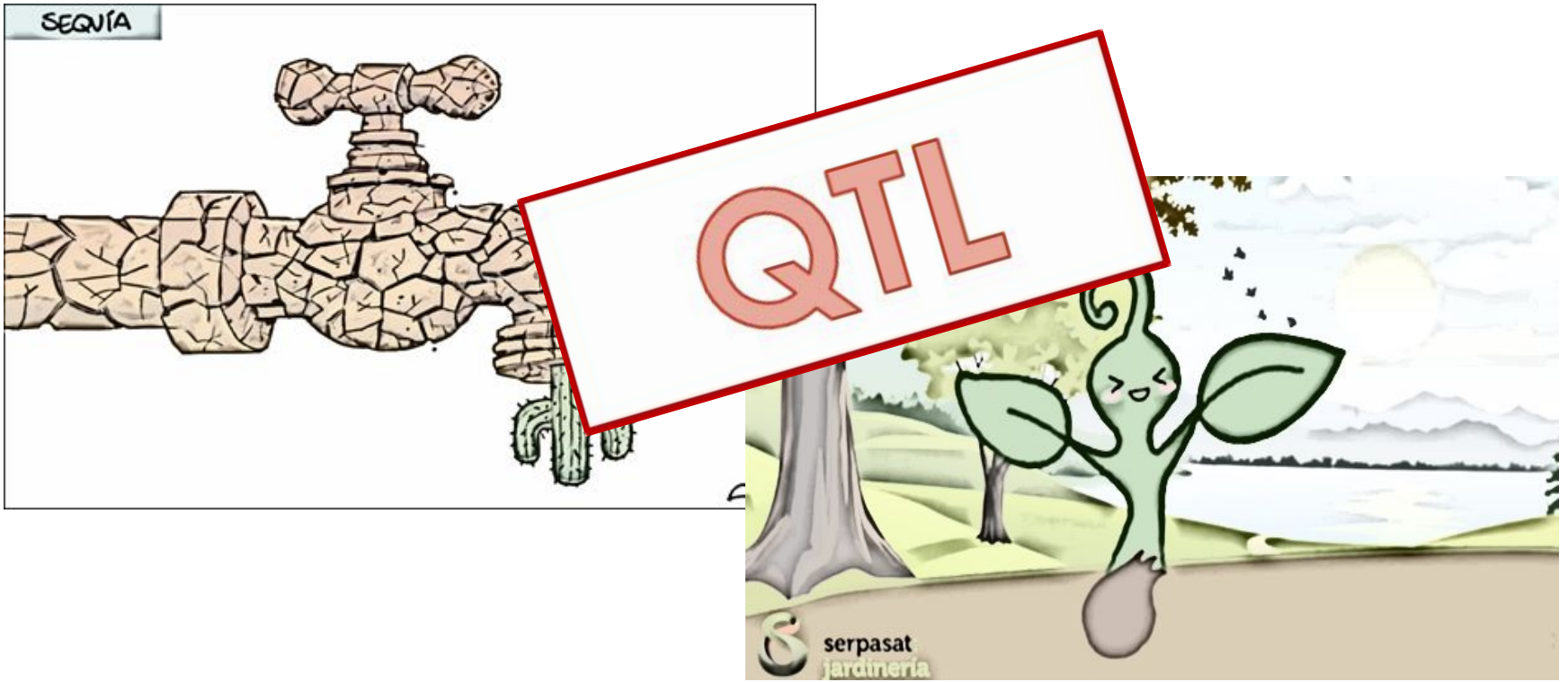
Caracteres de tipo
Cuantitativo



Caracteres de tipo
Cuantitativo

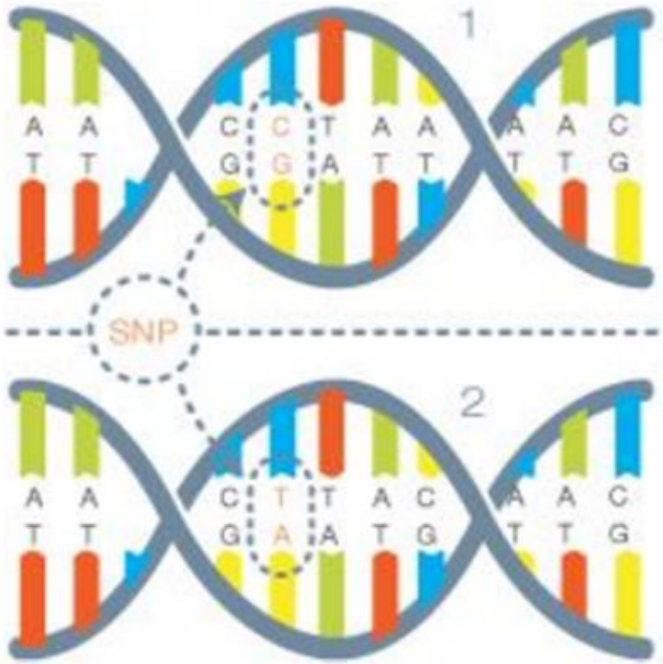


Caracteres de tipo
Cuantitativo

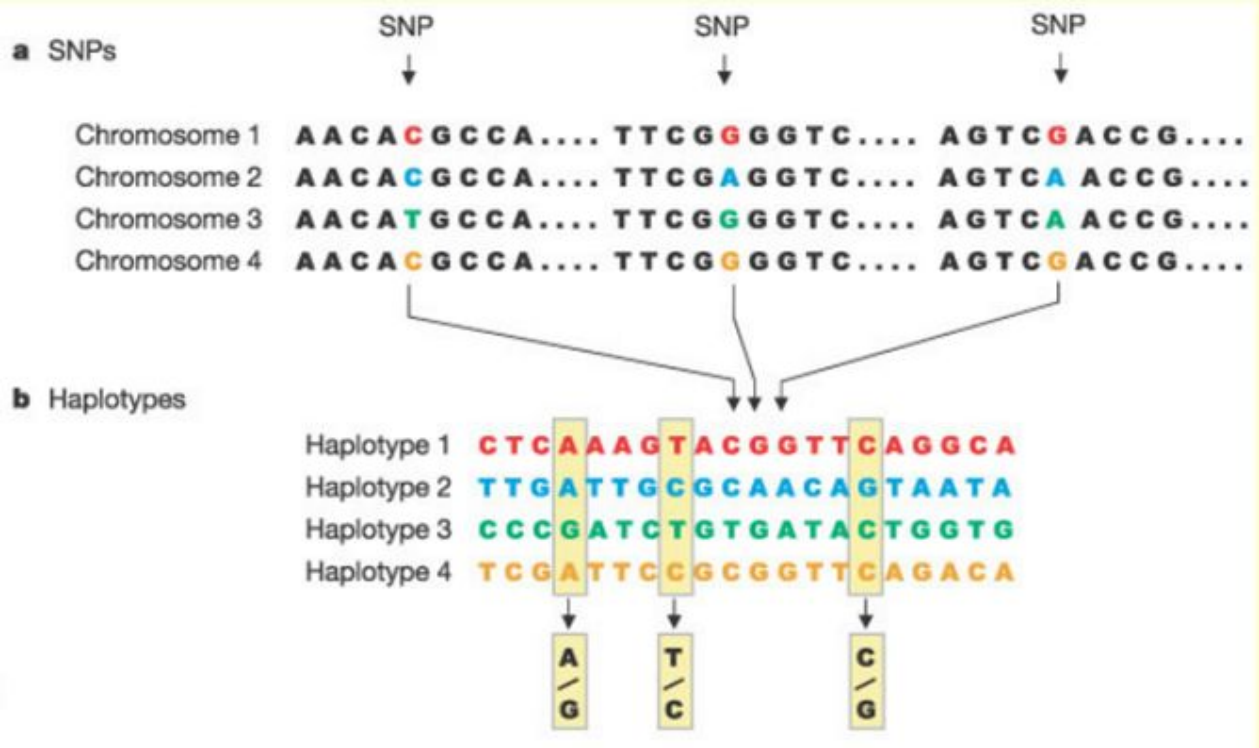


Marcadores

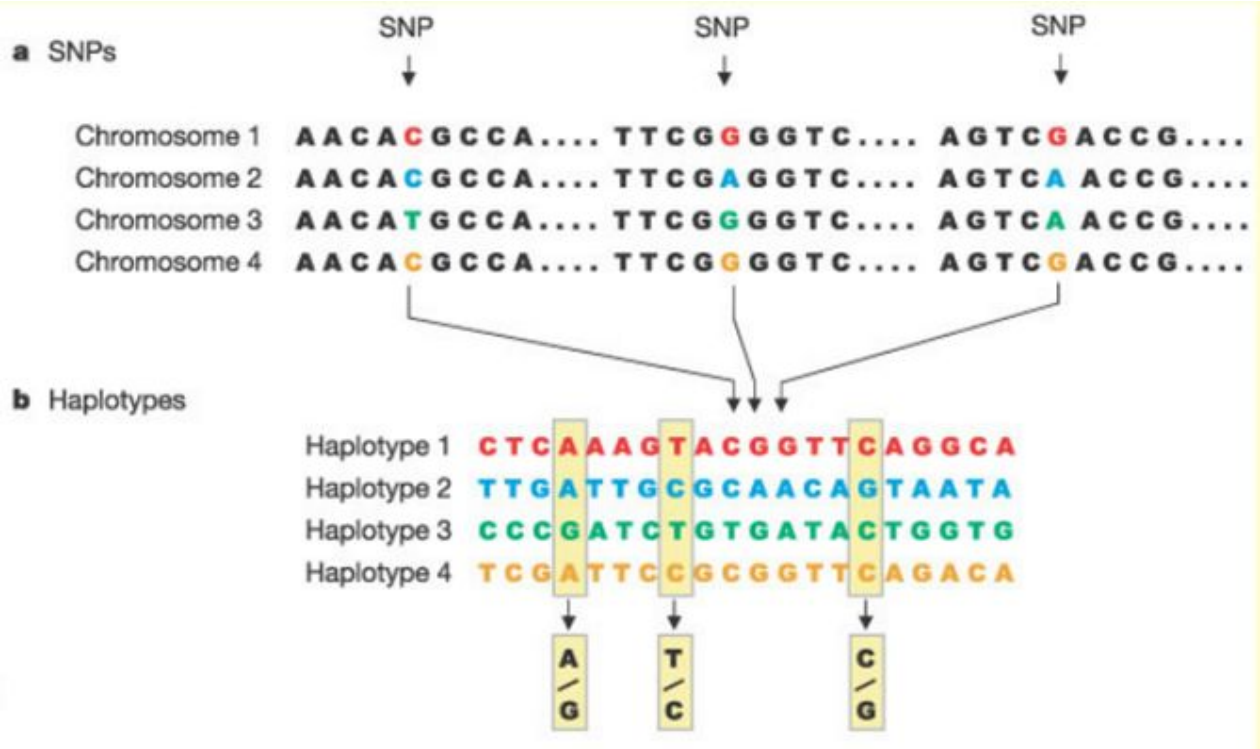
SNPs



Haplotipos


a SNPs
SNP
SNP
SNP
Chromosome 1 AACACGCCA.... TTCGGGGTC.... AGTCGACCG....
Chromosome 2 AACACGCCA.... TTCGAGGTC.... AGTCA ACCG....
Chromosome 3 AACATGCCA.... TTCGGGGTC.... AGTCA ACCG....
Chromosome 4 AACACGCCA.... TTCGGGGTC.... AGTCGACCG....
b Haplotypes
Haplotype 1 CTCAAAGTACGGTTCAGGCA
Haplotype 2 TTGATTGCGCAACAGTAATA
Haplotype 3 CCCGATCTGTGATACTGGTG
Haplotype 4 TCGATTCCGCGGTTCAGACA
A/G
T/C
C/G

Haplotipos



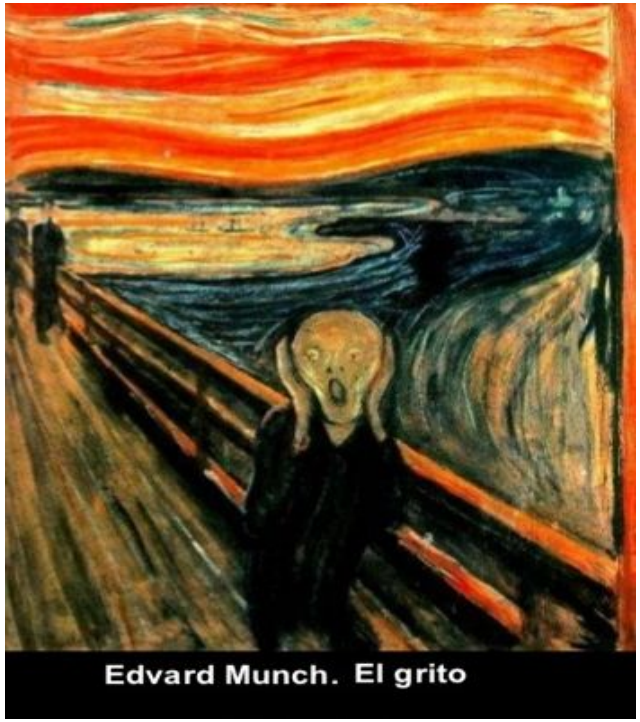
Dada la alta variabilidad alélica, la probabilidad de que dos individuos no relacionados presenten un mismo haplotipo, es prácticamente nula.

Clusterización de Haplotipos

A menudo, el procedimiento para la identificación del tipo de SNP que tiene cada individuo no se completa, lo que lleva a un **faltante** de valores.

Clusterización de Haplotipos

A menudo, el procedimiento para la identificación del tipo de SNP que tiene cada individuo no se completa, lo que lleva a un **faltante** de valores.



Clusterización de Haplotipos

El objetivo que se tiene es identificar los individuos que tengan todos sus marcadores especificados, a los que llamaremos ***haplotipos***, los cuales serán tomados de referencia para emparentar los marcadores incompletos con un haplotipo semejante.

Esta operación es lo que se conoce como ***clusterización***.

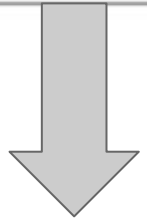
R en acción

- Paquete: **clusterhap***
- Agrupa genotipos en haplotipos semejantes

*Gaston Quero, Sebastian Simondi, with contributions from Victoria Bonnecarrere and Lucia Gutierrez (2016). Clusterhap: Clustering Genotypes in Haplotypes. R package version 0.1. https://CRAN.R-project.org/package=clusterhap

Ejemplo

Ind	*SNP.1*	*SNP.2*	*SNP.3*	*SNP.4*	*SNP.5*	*SNP.6*
ind_1	A	C	G	A	T	C
ind_2	NA	C	NA	A	T	C
ind_3	C	C	T	A	T	C
ind_4	NA	C	NA	A	T	G
ind_5	C	NA	NA	A	T	C



Ejemplo

Ind	*SNP.1*	*SNP.2*	*SNP.3*	*SNP.4*	*SNP.5*	*SNP.6*
ind_1	A	C	G	A	T	C
ind_2	NA	C	NA	A	T	C
ind_3	C	C	T	A	T	C
ind_4	NA	C	NA	A	T	G
ind_5	C	NA	NA	A	T	C

Ind	*SNP.1*	*SNP.2*	*SNP.3*	*SNP.4*	*SNP.5*	*SNP.6*
ind_1	1	2	3	1	4	2
ind_2	0	2	0	1	4	2
ind_3	2	2	4	1	4	2
ind_4	0	2	0	1	4	3
ind_5	2	0	0	1	4	2

Ejemplo:
Haplotipos

Ind	*SNP.1*	*SNP.2*	*SNP.3*	*SNP.4*	*SNP.5*	*SNP.6*
ind_1	1	2	3	1	4	2
ind_2	0	2	0	1	4	2
ind_3	2	2	4	1	4	2
ind_4	0	2	0	1	4	3
ind_5	2	0	0	1	4	2

- Haplo 1
- Haplo 2

Ejemplo:
Individuo 5

	SNP.1	*SNP.2*	*SNP.3*	*SNP.4*	*SNP.5*	*SNP.6*
ind_5	2	0	0	1	4	2
haplo 1	1	2	3	1	4	2
haplo 2	2	2	4	1	4	2

Ejemplo:
Individuo 5

	SNP.1	*SNP.2*	*SNP.3*	*SNP.4*	*SNP.5*	*SNP.6*
ind_5	2			1	4	2
haplo 1	1			1	4	2
haplo 2	2			1	4	2

Ejemplo:
Individuo 5

	SNP.1	SNP.2	SNP.3	SNP.4	SNP.5	SNP.6
ind_5	2			1	4	2
haplo 1	1			1	4	2
haplo 2	2			1	4	2

Haplotipo 1

Ind 1

Haplotipo 2

Ind 3

Ind 5

Ejemplo:
Individuo 2

	SNP.1	SNP.2	SNP.3	SNP.4	SNP.5	SNP.6
ind_2		2		1	4	2
haplo 1		2		1	4	2
haplo 2		2		1	4	2

Ejemplo:
Individuo 2

	SNP.1	SNP.2	SNP.3	SNP.4	SNP.5	SNP.6
ind_2		2		1	4	2
haplo 1		2		1	4	2
haplo 2		2		1	4	2

Haplotipo 1

Ind 1

Ind 2

Haplotipo 2

Ind 3

Ind 5

Ind 2

Ejemplo:
Individuo 4

	SNP.1	SNP.2	SNP.3	SNP.4	SNP.5	SNP.6
ind_4		2		1	4	3
haplo 1		2		1	4	2
haplo 2		2		1	4	2

Ejemplo:
Individuo 4

	SNP.1	SNP.2	SNP.3	SNP.4	SNP.5	SNP.6
ind_4		2		1	4	3
haplo 1		2		1	4	2
haplo 2		2		1	4	2

Haplotipo 1

Ind 1

Ind 2

Haplotipo 2

Ind 3

Ind 5

Ind 2

Indeterminados

Ind 4

Algoritmo función clusterhap

```
> clusterhap::clusterhap
function (x, Print = FALSE)
{
    H_data_o <- x
    H_data <- x
    dir.create("clusterhap_reports", showWarnings = F)
    hplq.result <- NULL
    hplq.final <- NULL
    id <- H_data[, 1]
    Q <- H_data[, -1]
    hplq <- NULL
    hp <- NULL
    hp.1 <- NULL
    cq <- NULL
    w <- NULL
    cr <- NULL
    c.Q <- rowSums(Q == 0)
    y <- which(c.Q == 0)
    b <- Q[y, ]
    Y <- b[!duplicated(b), ]
    for (i in 1:nrow(Q)) {
        for (j in 1:nrow(Y)) {
            cq <- rowSums(Q[i, ] == 0)
            w <- Q[i, ] - Y[j, ]
            cr <- rowSums(w == 0)
            zeros <- cq + cr
            if (zeros == ncol(Q)) {
                hp <- cbind(hp, i)
                hp.1 <- cbind(hp.1, j)
                hpl <- cbind(Q[i, ], j)
                hpl1 <- cbind(id[i], hpl)
                hplq <- rbind(hplq, hpl1)
            }
        }
    }
    u <- H_data[-c(hp), ]
    if (nrow(u) == 0) {
```

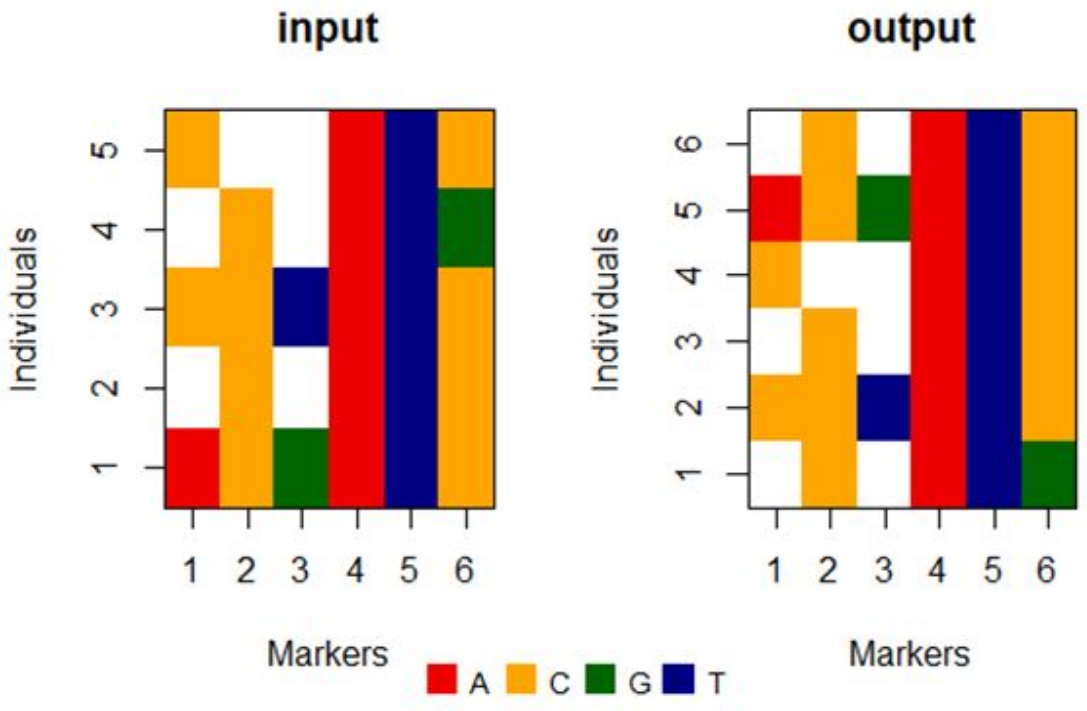
Algoritmo función clusterhap

```
    u <- u
}
else {
    undetermined <- cbind(u, "undetermined")
}
fa <- NULL
for (i in 1:nrow(Y)) {
    fa <- (cbind(fa, rowSums(hp.1 == i)))
}
hap.id <- paste("haplo", 1:nrow(Y))
Y <- cbind(hap.id, Y)
colnames(hplq)[colnames(hplq) == "id[i]"] <- "id.geno"
colnames(hplq)[colnames(hplq) == "j"] <- "haplo.qtl"
if (nrow(u) > 0) {
    colnames(undetermined) <- colnames(hplq)
    hplq.final <- rbind(hplq, undetermined)
}
else {
    hplq.final <- hplq
}
if (nrow(u) > 0) {
    fa <- cbind(fa, nrow(u))
    fr <- (fa/nrow(hplq.final)) * 100
    fh <- round(rbind(fa, fr), 1)
    colnames(fh) <- c(paste("haplo", 1:nrow(Y)), "undetermined")
    freq <- c("freq.abs", "freq.rel")
    fh <- data.frame(freq, fh)
}
else {
    fr <- (fa/nrow(hplq.final)) * 100
    fh <- round(rbind(fa, fr), 1)
    colnames(fh) <- c(paste("haplo", 1:nrow(Y)))
    freq <- c("freq.abs", "freq.rel")
    fh <- data.frame(freq, fh)
}
d <- duplicated(hplq.final[, 1]) | duplicated(hplq.final[,
```

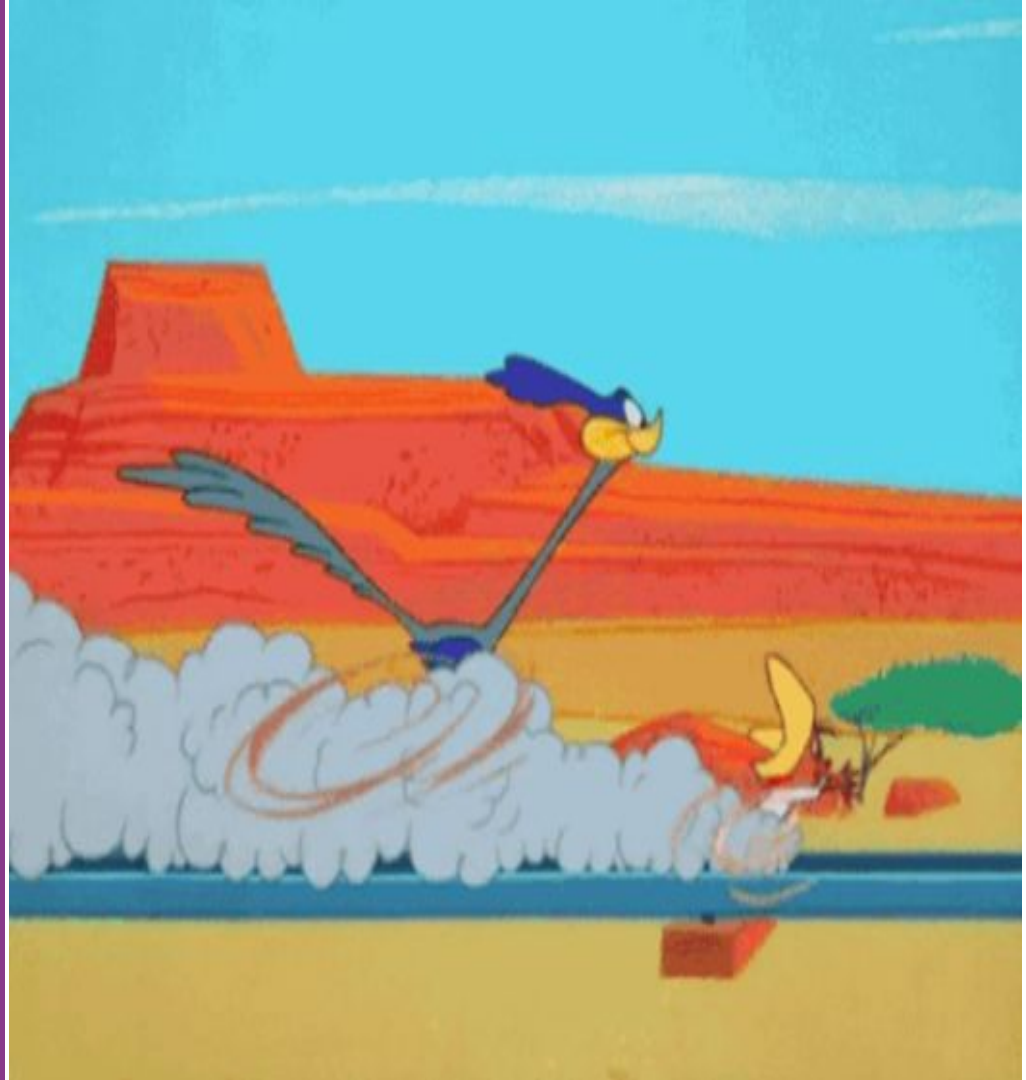
Ejemplo
clusterhap

```
> clusterhap::clusterhap(sim, Print = TRUE)
$h.result
   id.geno SNP.1 SNP.2 SNP.3 SNP.4 SNP.5 SNP.6     haplo.qtl
1    ind.1     1     2     3     1     4     2             1
2    ind.2     0     2     0     1     4     2             1
21   ind.2     0     2     0     1     4     2             2
3    ind.3     2     2     4     1     4     2             2
5    ind.5     2     0     0     1     4     2             2
4    ind.4     0     2     0     1     4     3 undetermined

$haplotypes
   hap.id SNP.1 SNP.2 SNP.3 SNP.4 SNP.5 SNP.6
1 haplo 1     1     2     3     1     4     2
3 haplo 2     2     2     4     1     4     2

$duplicates
   id.geno SNP.1 SNP.2 SNP.3 SNP.4 SNP.5 SNP.6 haplo.qtl
2    ind.2     0     2     0     1     4     2         1
21   ind.2     0     2     0     1     4     2         2

$freq
      freq haplo.1 haplo.2 undetermined
1 freq.abs     2.0       3          1.0
2 freq.rel    33.3      50         16.7

$und
    ind SNP.1 SNP.2 SNP.3 SNP.4 SNP.5 SNP.6
4 ind.4     0     2     0     1     4     3
```

Ejemplo
clusterhap



Optimización



Optimización

Utilizando funciones vectorizadas

Familia **apply**

Optimización

```
> clust2(sim, Print = TRUE)
$h.result
    ind SNP.1 SNP.2 SNP.3 SNP.4 SNP.5 SNP.6        hap.id
1 ind.1     1     2     3     1     4     2      haplo 1
2 ind.2     0     2     0     1     4     2      haplo 1
3 ind.3     2     2     4     1     4     2      haplo 2
4 ind.2     0     2     0     1     4     2      haplo 2
5 ind.5     2     0     0     1     4     2      haplo 2
6 ind.4     0     2     0     1     4     3 undetermined

$haplotypes
   hap.id SNP.1 SNP.2 SNP.3 SNP.4 SNP.5 SNP.6
1 haplo 1     1     2     3     1     4     2
3 haplo 2     2     2     4     1     4     2

$duplicates
    ind SNP.1 SNP.2 SNP.3 SNP.4 SNP.5 SNP.6  hap.id
2 ind.2     0     2     0     1     4     2 haplo 1
4 ind.2     0     2     0     1     4     2 haplo 2

$freq
       freq haplo.1 haplo.2 undetermined
fa freq.abs     2.0       3          1.0
fr freq.rel    33.3      50         16.7

$und
    ind SNP.1 SNP.2 SNP.3 SNP.4 SNP.5 SNP.6        hap.id
6 ind.4     0     2     0     1     4     3 undetermined
```

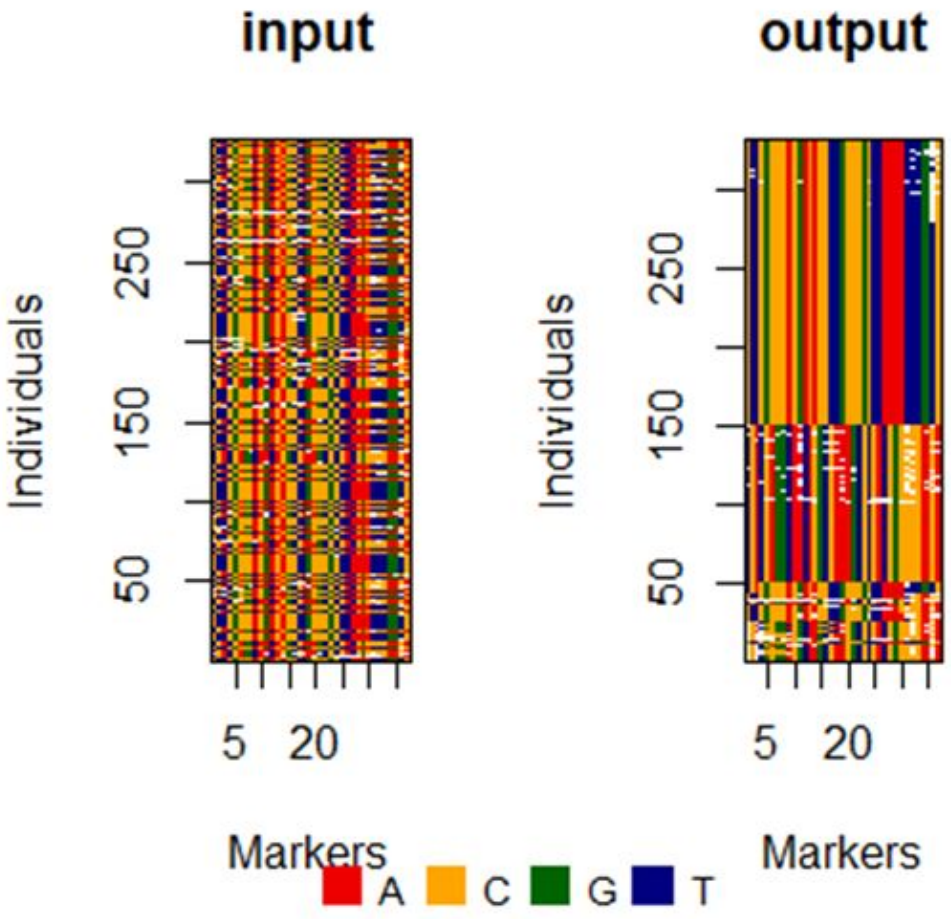
Algunas bases de datos:

rice_qtl

```
> str(rice_qtl)
'data.frame':   326 obs. of  38 variables:
 $ idMG      : Factor w/ 326 levels "El_Paso_144",..: 2 206 288 68 36 211 304 92 200 64 ...
 $ S1_1001162: int  2 4 0 0 4 4 2 2 4 4 ...
 $ S1_1001511: int  4 2 2 2 2 2 4 4 2 2 ...
 $ S1_1009316: int  4 1 0 1 1 1 4 4 1 1 ...
 $ S1_1019523: int  2 4 4 4 4 4 2 2 4 4 ...
 $ S1_1019634: int  3 2 3 2 2 2 3 3 2 2 ...
 $ S1_1019648: int  2 1 2 1 1 1 2 2 1 1 ...
 $ S1_1020373: int  2 3 3 3 3 3 2 2 3 3 ...
 $ S1_1029542: int  2 3 3 3 3 3 2 2 3 3 ...
 $ S1_1030434: int  1 4 1 4 4 4 1 1 4 4 ...
 $ S1_1031758: int  2 1 2 1 1 1 2 2 1 1 ...
 $ S1_1049575: int  3 1 3 0 1 1 3 3 1 1 ...
 $ S1_1053530: int  1 4 4 4 4 4 1 1 4 4 ...
 $ S1_1053535: int  2 1 1 1 1 1 2 2 1 1 ...
 $ S1_1053703: int  1 0 1 2 2 2 1 1 2 2 ...
 $ S1_1059696: int  2 3 3 3 3 3 2 2 3 3 ...
 $ S1_1062835: int  2 4 4 0 4 4 2 2 4 4 ...
 $ S1_1062848: int  4 2 2 0 2 2 4 4 2 2 ...
 $ S1_1062853: int  4 1 1 0 1 1 4 4 1 1 ...
 $ S1_1062957: int  3 1 1 1 1 1 3 3 1 1 ...
 $ S1_1063050: int  2 0 2 1 1 1 2 2 1 1 ...
 $ S1_1066894: int  2 3 3 0 3 3 2 2 3 3 ...
 $ S1_1067113: int  2 4 4 4 4 4 2 2 4 4 ...
 $ S1_1067116: int  3 2 2 2 2 2 3 3 2 2 ...
 $ S1_1067668: int  2 0 4 4 4 4 2 2 4 4 ...
 $ S1_1069360: int  4 0 2 2 2 2 4 4 2 2 ...
 $ S1_1069362: int  4 0 1 1 1 1 4 4 1 1 ...
 $ S1_1069363: int  1 0 4 4 4 4 1 1 4 4 ...
 $ S1_1069365: int  1 0 2 2 2 2 1 1 2 2 ...
 $ S1_1069784: int  1 3 3 3 3 3 1 1 3 3 ...
 $ S1_1081413: int  1 0 2 2 2 2 1 1 2 2 ...
 $ S1_1089727: int  4 0 2 0 2 2 4 4 2 2 ...
 $ S1_1188838: int  4 0 2 2 2 2 4 4 2 2 ...
 $ S1_1196763: int  4 2 2 2 2 2 4 4 2 2 ...
 $ S1_1197550: int  3 1 1 1 1 1 3 3 1 1 ...
 $ S1_1199213: int  3 1 1 1 1 1 3 3 1 1 ...
 $ S1_1204373: int  4 2 2 0 2 2 4 0 2 2 ...
 $ S1_1206067: int  2 1 1 0 1 1 2 2 1 1 ...
```

Algunas bases de datos:

rice_qtl



Algunas bases de datos:

rice_qtl

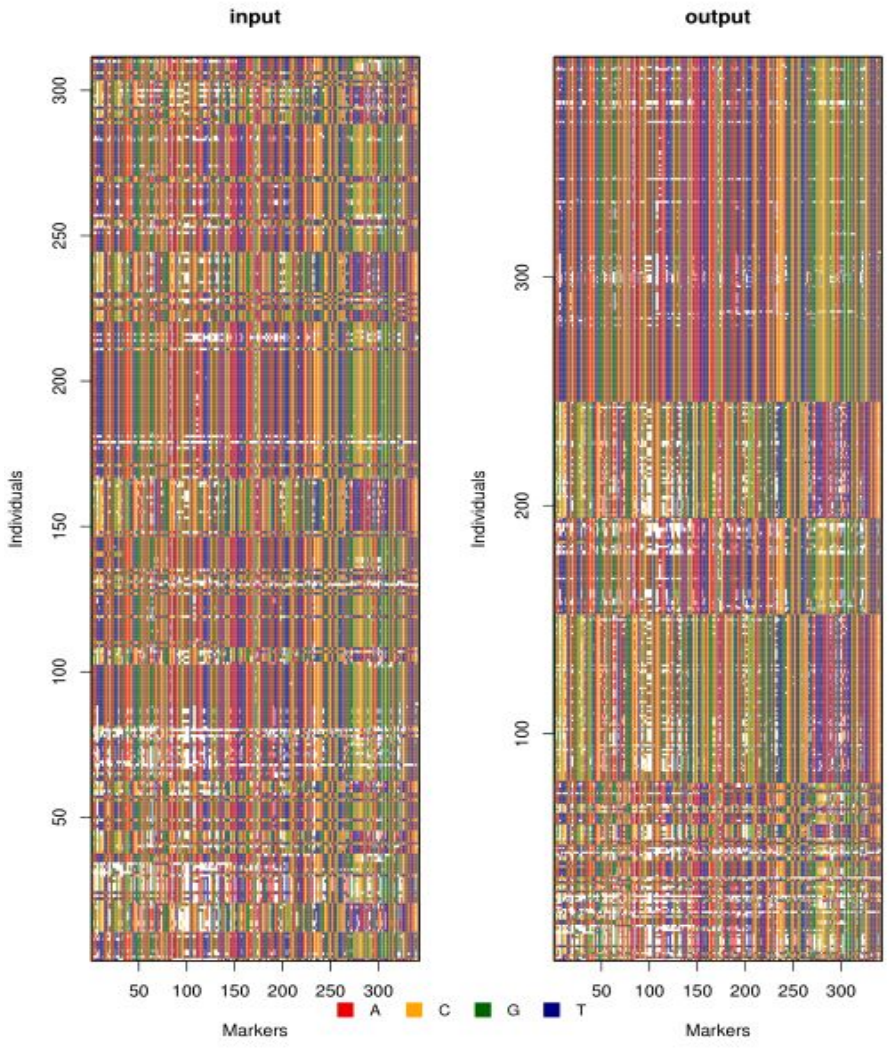
```
> system.time(clusterhap(rice_qtl))
   user  system elapsed
  22.06    0.15   22.33
> system.time(clust2(rice_qtl))
   user  system elapsed
   1.64    0.12    1.78
```

Algunas bases de datos:

qYAM

```
> str(datos.prueba)
Classes 'tbl_df', 'tbl' and 'data.frame':       311 obs. of  346 variables:
 $ idMG        : chr  "5287" "8405" "8410" "8424" ...
 $ YAM         : num  67.7 70.5 70.1 69.3 69 ...
 $ PHR         : num  59.7 64.2 64.3 61.1 61.1 ...
 $ GC          : num  15.16 11.46 9.84 8.27 8.28 ...
 $ S6_26894298: num  2 0 2 2 2 2 2 0 2 2 ...
 $ S6_26894513: num  1 1 1 1 1 1 1 1 1 1 ...
 $ S6_26898904: num  3 0 3 3 3 3 3 0 3 3 ...
 $ S6_26909453: num  1 0 1 0 0 1 1 1 1 0 ...
 $ S6_26911368: num  1 0 1 1 1 1 1 1 0 1 ...
 $ S6_26940622: num  4 0 0 0 4 4 4 0 0 0 ...
 $ S6_26940711: num  4 0 0 4 4 4 4 0 4 0 ...
 $ S6_26942437: num  1 1 1 1 1 1 1 1 1 1 ...
 $ S6_26958005: num  4 0 0 0 4 4 4 0 0 4 ...
 $ S6_26973106: num  4 4 4 4 4 4 4 4 4 4 ...
 $ S6_26973138: num  4 4 4 4 4 4 4 4 4 4 ...
 $ S6_26973422: num  3 0 3 3 3 3 3 0 3 3 ...
 $ S6_27015435: num  4 4 4 4 4 4 4 4 4 4 ...
 $ S6_27022159: num  1 1 1 1 1 1 1 1 1 1 ...
 $ S6_27022161: num  3 3 3 3 3 3 3 3 3 3 ...
 $ S6_27022312: num  1 1 1 1 1 1 1 1 1 1 ...
 $ S6_27024677: num  2 2 2 2 2 2 2 2 2 2 ...
 $ S6_27037310: num  4 0 0 4 4 4 4 0 4 0 ...
 $ S6_27037621: num  1 0 1 0 1 1 1 0 0 0 ...
 $ S6_27037652: num  2 0 2 0 2 2 2 0 0 0 ...
 $ S6_27041425: num  3 3 3 3 3 3 3 3 3 3 ...
 $ S6_27042784: num  1 1 1 1 1 1 1 1 1 1 ...
 $ S6_27043406: num  3 3 3 3 3 3 3 3 3 3 ...
 $ S6_27043422: num  1 0 1 1 1 1 1 1 1 1 ...
 $ S6_27043424: num  4 4 4 4 4 4 4 4 4 4 ...
 $ S6_27044221: num  4 0 0 0 4 4 4 0 0 0 ...
 $ S6_27087128: num  4 0 4 4 4 4 4 0 4 4 ...
 $ S6_27087150: num  3 0 3 3 3 3 3 0 3 3 ...
 $ S6_27090546: num  0 0 1 1 1 1 1 0 1 1 ...
 $ S6_27096003: num  0 2 2 2 2 2 2 0 2 2 ...
 $ S6 27096533: num  0 4 4 4 4 4 4 4 4 4 ...
 $ S6_27096548: num  0 4 4 4 4 4 4 4 4 4 ...
 $ S6_27096886: num  0 0 3 0 3 3 3 0 0 3 ...
 $ S6_27099594: num  3 3 3 3 3 3 3 3 3 3 ...
 $ S6_27120079: num  1 1 1 1 1 1 1 1 1 1 ...
 [list output truncated]
```

Algunas bases de datos:
qYAM



3.
Conclusiones

Conclusiones

- Dado un QTL pudimos asociarle un haplotipo a cada individuo, a través de un algoritmo de una menor cantidad de código.
- Comparamos la velocidad de cómputo del algoritmo original vs. el algoritmo nuevo, y obtuvimos una respuesta favorable hacia este último.
- Logramos el análisis de bases de datos numerosas, que el algoritmo original no podía resolver, y además acortamos los tiempos, ya que obtuvimos respuestas más rápidas.
- Por lo tanto, observamos una mejora sustancial al algoritmo original, que se traduce en una mayor capacidad de cómputo.

¡Muchas gracias!