

---

Proyecto Final:

# Distancias genéticas a partir de prueba de 3 puntos

---







Miguel Angel Rodriguez Feregrino  
Ivan Oseas Jacobo Cruz  
Santiago Figueroa D'gremy

# Introducción

- Debido a ligamiento de genes se rompe la segunda ley de Mendel. Por lo tanto ya no hay segregación independiente entre genes.
- Este ligamento es debido a que dos genes se encuentran en el mismo cromosoma y a distancias cortas, por lo que el número de quiasmas que se pueden formar durante la recombinación homóloga disminuye.
- La distancia se expresa en centiMorgans.
- Una distancia mayor a 50 cM evidencia que no hay ligamiento entre genes.
- La prueba de 3 puntos compara tres genes para identificar las dos distancias a partir de la generación de dobles recombinantes y recombinantes simples en la progenie.

# Principales funciones y necesidades del proyecto:

Son necesarios los paquetes qtl y ape

```
1 library("qtl")
2 library(ape)
3
4 ▶ prueba3puntos.no <- function(gen1,gen2,gen3, dominancia = FALSE){}
191
192 ▶ prueba3puntos <- function(data, frecuencias, wt = FALSE, fr.wt = FALSE){}
358
359 prueba3puntos.inv <- function(Coeficiente_consistencia, Distancia_zona1, Distancia_zona2,
360 ▶ gen_parental = c("A","B","C"), total = FALSE, inf.observados = FALSE){}
442
443 ▶ as.mapa.genetico.matrix <- function(x, etiquetas = FALSE) {}
474
475 ▶ as.mapa.genetico <- function(data, Genes = c("A","B","C")) {}
492
493 ▶ linkmap <- function(object, chr, chr.space = 2, m.cex = 0.6, interval = FALSE, ruler = FALSE, ...){
```

# **1.- Calcular distancias genéticas, Coeficiente de coincidencia y de interferencia.**

Hay dos funciones para el cálculo de distancias a partir de frecuencia de observados luego de cruza de prueba:

**1.-prueba3puntos.no**

**2.-prueba3puntos**

# 1.- prueba3puntos.no

Esta función no necesita una base de datos preestablecida que contenga las frecuencias ya que funciona con readlines.

Argumentos de la función:

**gen1** = El nombre o la característica fenotípica que estamos evaluando para el gen1. Ej. gen "v" que únicamente se expresa en drosophila cuando es recesivo ya que es mutante, también se podría ingresar como "vermillion".

**gen2** = El nombre o la característica fenotípica que estamos evaluando para el gen2.

**gen3** = El nombre o la característica fenotípica que estamos evaluando para el gen3.

**dominancia** = Argumento lógico (TRUE o FALSE) que indica si los genes estudiados tienen expresión fenotípica dominante o recesiva.

La lógica de la función es, por medio de readlines, obtener como es la expresión en los 2 fenotipos parentales, y a partir de estas 2 combinaciones compara con las 8 posibles combinaciones existentes al considerar al parental 1 como aquel con mayor frecuencia de observados. Por tanto se puede crear una estructura de fenotipos para cada tipo de fenotipo de la descendencia a partir de la combinación con la que coincide el parental 1.

**Posibles combinaciones:** (si, si, si), (si, si, no), (si, no, si), (si, no, no), (no, si, si), (no, si, no), (no, no, si), (no, no, no).

Una vez se obtiene que combinación es se elige por medio de if la misma para preguntar por cada frecuencia de descendientes, esta información la acomoda en una base de datos llamada “frecuencias\_fenotipicas”, ya separadas según si son:

- **parentales, recombinantes simples zona 1, recombinantes simples zona 2, dobles recombinantes.**

```

> Ejemplo1_3puntos_no_base_datos <- prueba3puntos.no("k","e","cd") # debemos ingresar los genes entre comillas.
Hay expresion fenotipica de k en el progenitor 1 (Aquel fenotipo con mayor frecuencia en la progene)? s/n
s
Hay expresion fenotipica de e en el progenitor 1(Aquel fenotipo con mayor frecuencia en la progene)? s/n
n
Hay expresion fenotipica de cd en el progenitor 1(Aquel fenotipo con mayor frecuencia en la progene)? s/n
s
Hay expresion fenotipica de k en el progenitor 2(Aquel fenotipo con la segunda mayor frecuencia en la progene)? s/n
n
Hay expresion fenotipica de e en el progenitor 2(Aquel fenotipo con la segunda mayor frecuencia en la progene)? s/n
s
Hay expresion fenotipica de cd en el progenitor 2(Aquel fenotipo con la segunda mayor frecuencia en la progene)? s/n
n
Cual es la frecuencia de individuos luego de cruza de prueba que presentan expresion fenotipica unicamente de k y cd ?:
1773
Cual es la frecuencia de individuos luego de cruza de prueba que presentan expresion fenotipica unicamente de e ?:
1761
Cual es la frecuencia de individuos luego de cruza de prueba que presentan expresion fenotipica unicamente de k y e ?:
138
Cual es la frecuencia de individuos luego de cruza de prueba que presentan expresion fenotipica unicamente de cd ?:
128
Cual es la frecuencia de individuos luego de cruza de prueba que presentan expresion fenotipica unicamente de k ?:
89
Cual es la frecuencia de individuos luego de cruza de prueba que presentan expresion fenotipica unicamente de e y cd ?:
97
Cual es la frecuencia de individuos luego de cruza de prueba que presentan expresion fenotipica de k e cd ?:
8
Cual es la frecuencia de individuos luego de cruza de prueba que no presentan expresion fenotipica de alguno de los genes (wt)??:
6

```

# RESULTADO

```
> Ejemplo1_3puntos_no_base_datos
```

```
$Frecuencias
```

	parentales	recombinantes_simples_1	recombinantes_simples_2	recombinantes_dobles
1	1761	128	97	6
2	1773	138	89	8

```
$Distancias_Zona_1
```

```
[1] "La distancia en zona 1 es: 7"
```

```
$Distancias_Zona_2
```

```
[1] "La distancia en zona 2 es: 5"
```

```
$Coeficiente_coincidencia
```

```
[1] "El coeficiente de coincidencia es: 1"
```

```
$Coeficiente_interferencia
```

```
[1] "El coeficiente de interferencia es: 1.11022302462516e-16"
```



## 2.- prueba3puntos

Esta función necesita de una base que tenga una columna con las frecuencias de observación para los 8 posibles fenotipos.

Argumentos de la función:

**data** = Una base de datos que contiene las frecuencias.

**frecuencias** = El nombre de la columna de la base de datos que contiene las frecuencias

**wt** = Un vector que contenga el nombre de los genes como te gustaría que fueran representados en el fenotipo wt. Ej. Si estamos trabajando los genes vt, v, ct, entonces el wt tendría ("vt", "v", "ct"), mientras que el mutante tendría ("vt-", "v-", "ct-").

**fr.wt** = La frecuencia de observación del fenotipo silvestre dentro de la base de datos.

# RESULTADOS

\$Frecuencias

	genotipo	frecuencias
Parental_1	k - e cd -	1773
Parental_2	k e - cd	1761
Recombinante_simple_zona_1_1	k - e - cd	138
Recombinante_simple_zona_1_2	k e cd -	128
Recombinante_simple_zona_2_1	k - e cd	97
Recombinante_simple_zona_2_2	k e - cd -	89
Recombinante_doble_1	k - e - cd -	8
Recombinante_doble_2	k e cd	6

\$Distancias\_Zona\_1

[1] 7

\$Distancias\_Zona\_2

[1] 5

\$Coeficiente\_coincidencia

[1] 1

\$Coeficiente\_interferencia

[1] 1.110223e-16

## **2.- Calcular frecuencias a partir de distancia genética y coeficiente de coincidencia**

La función para calcular frecuencias, osea un cálculo de prueba de 3 puntos inverso es:

**prueba3puntos.inv**

# prueba3puntos.inv

Esta función necesita los datos que normalmente se obtendrán a partir de las funciones anteriores:  
Distancia zona 1, Distancia zona 2, coeficiente de coincidencia.

Argumentos de la función:

**Coeficiente\_coincidencia** = El coeficiente CC.

**Distancia\_zona1** = La distancia entre gen 1 y gen 2.

**Distancia\_zona2** = La distancia entre gen 2 y gen 3.

**Gen\_parental1** = Argumento opcional. Un vector que contenga los genes o los fenotipos como queremos representarlos para el fenotipo parental1.

**Total** = Argumento opcional. El número total de progenie, si se ingresa se calculan los esperados.

**inf.observados** = Argumento opcional. Lógico (TRUE o FALSE) si se ingresa TRUE entonces se calculan las frecuencias y los esperados para cada uno de los 8 posibles fenotipos de la progenie.

### **3.- Construir mapas genéticos a partir de los datos obtenidos en la prueba de 3 puntos**

Las funciones para construir mapas genéticos a partir de los datos obtenidos son:

1.- `as.mapa.genetico`

2.- `as.mapa.genetico.matrix`

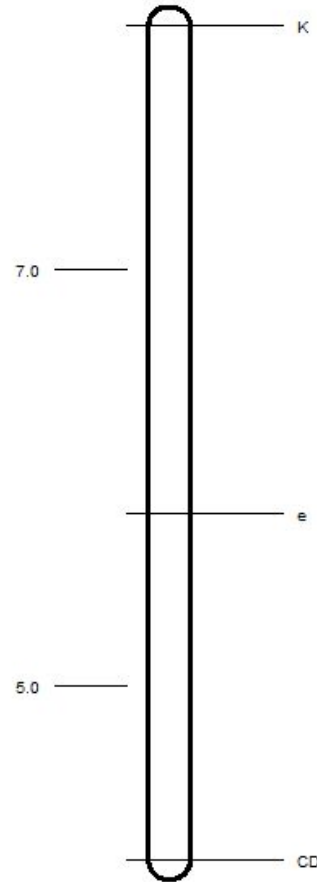
# 1.- as.mapa.genetico

Para generar este mapa se requieren 3 cosas:

- Un objeto creado por cualquiera de las 3 funciones anteriores.
- El paquete qtl
- La función linkmap (creada por Junli Zhang)

> Como resultado arroja un mapa genético realizado a partir de una prueba de 3 puntos, de manera que nos indica la distancia que hay entre cada gen para saber si están ligados o no.

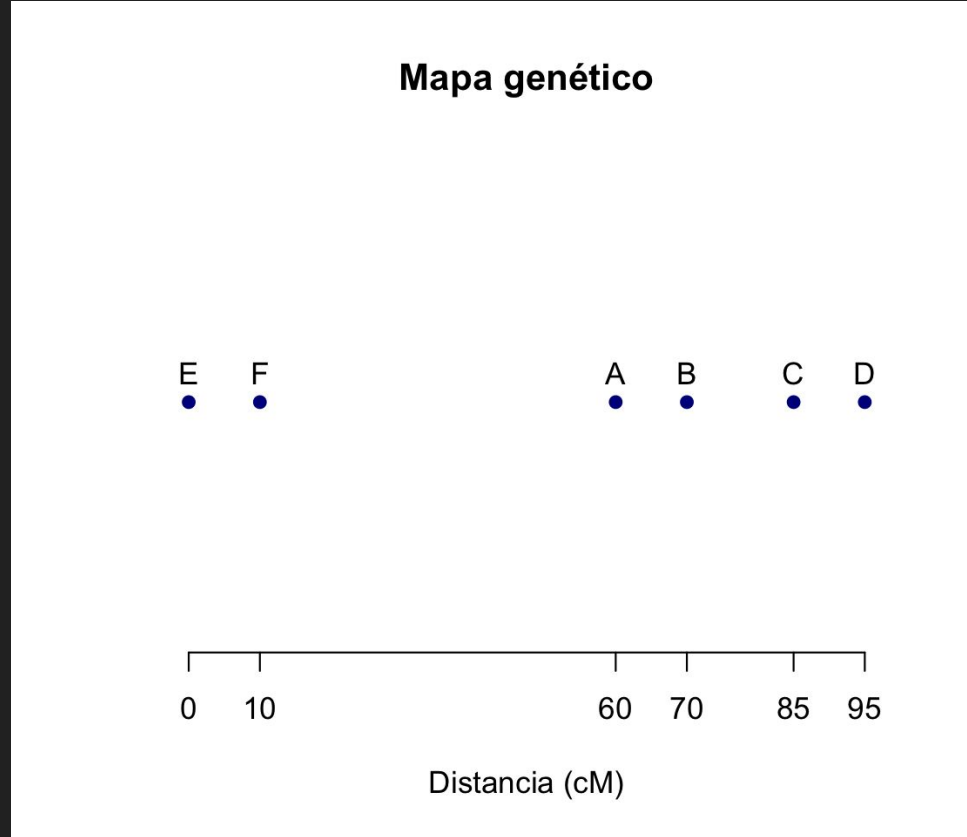
Prueba de 3 puntos



## 2.- as.mapa.genetico.matrix

input: matriz cuadrada

opcional: vector con nombres



# Función adicional necesaria para construcción de mapas:

**Linkmap:** Es una función que genera un juego de cromosomas con marcadores moleculares a partir de un objeto tipo cross.

**Esta función fue creada por Junli Zhang quien la compartió en GitHub**

