Facultad de Ciencias Experimentales

GRADO EN BIOLOGÍA

Universidad de Jaén
Facultad de ciencias experimentales

Trabajo Fin de Grado

ANÁLISIS COMPUTACIONAL DE RIESGO POLIGÉNICO: APLICACIÓN EN LA ENFERMEDAD DE LAS ARTERIAS CORONARIAS (EAC)

Alumno/a: Emma Zaragoza López

Tutor: Francisco J. Esteban

Departamento de Biología Experimental

Jaén. Junio, 2022

Facultad de
Ciencias Experimentales

Trabajo Fin de Grado

ANÁLISIS COMPUTACIONAL DE RIESGO POLIGÉNICO: APLICACIÓN EN LA ENFERMEDAD DE LAS ARTERIAS CORONARIAS (EAC)

Alumno/a: Emma Zaragoza López

Fdo. Francisco J. Esteban
Jaén, 6 de junio de 2023

ÍNDICE

ABSTRACT

Numerous studies are currently being carried out to improve health and well-being. One of them is the possibility of predicting the state of a disease, or how susceptible an individual is to suffering from it, by means of the calculation of polygenic risk. There are different tools that can be used for its calculation, in this case, we will make use of an R package, bigsnpr, which, by means of the LDpred tool and its various functions, applied to a clinical case of coronary artery disease or EAC, will help to understand and visualise the procedure of this calculation.

Key words: Bigsnpr, LDpred, Poligenic Risk, Rstudio.

RESUMEN

Actualmente se están llevando a cabo numerosos estudios encaminados en la mejora de la salud y bienestar. Uno de ellos, es la posibilidad de predecir el estado de una enfermedad, o cómo de susceptible es el individuo de padecerla, por medio del cálculo de riesgo poligénico. Hay diferentes herramientas que pueden emplearse para su cálculo, en este caso, se hará uso de un paquete de R, bigsnpr, que, por medio de la herramienta LDpred y sus diversas funciones, aplicadas a un caso clínico de enfermedades de las arterias coronarias o EAC, ayudarán a comprender y visualizar el procedimiento de este cálculo.

Palabras clave: Bigsnpr, LDpred, Riesgo Poligénico, Rstudio.

1. INTRODUCCIÓN

1.1. Definición e importancia del riesgo poligénico

La mayor parte de los trastornos no trasmisibles a la descendencia presentan un gran polimorfismo genético, cuya base genética comprende un pequeño efecto sobre el riesgo de padecer una enfermedad. Estas variaciones genéticas son de gran importancia para el estudio de la vía biológica de un trastorno o para predecir el riesgo de presentar esa enfermedad. (Lewis and Vassos, 2020) Por ello, en los últimos años se ha considerado una necesidad para la salud pública identificar aquellas personas con mayor riesgo de padecer una enfermedad, con el fin de conseguir una detección más rápida y eficaz. (Khera *et al*., 2018)

La puntuación del riesgo poligénico (PRS), se considera hoy en día una posible vía para predecir el estado de una enfermedad y evaluar el riesgo de padecerla. El término de riesgo poligénico fue descrito por primera vez a principios del siglo XX (Fischer, 1919). Se define como una aproximación o estimación de la predisposición genética de un individuo a un carácter o enfermedad. Para ello, es necesaria una carga genética, adquirida por una suma de variantes de riesgo, cuyo conjunto presente suficiente información para poder detectar aquellos genes que presenten un riesgo mayor. (Lewis and Vassos, 2020). Se basa por tanto en el perfil genotípico y los datos de asociación del genoma completo (GWAS) relevantes.

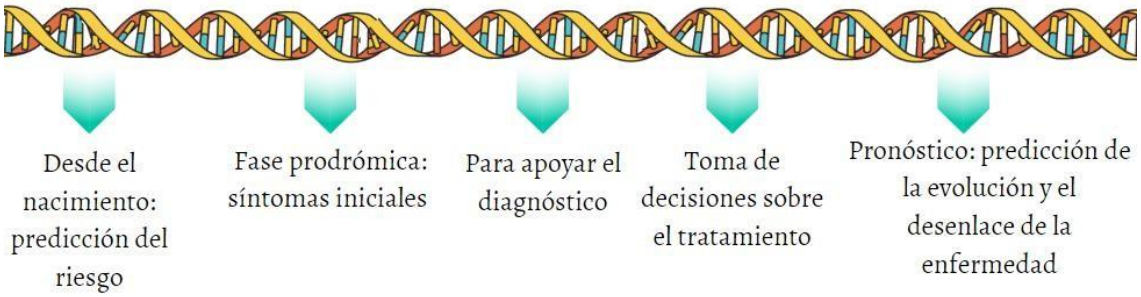


Ilustración 1. Línea de vida de la importancia de las puntuaciones poligénicas en puntos críticos del desarrollo de una enfermedad

1.1.1. Riesgo poligénico y el factor ambiental

Un aspecto importante a tener en cuenta con respecto a las puntuaciones de riesgo poligénico es la implicación del factor ambiente. Es decir, la interacción gen-ambiente (GxEs) conlleva a que una persona herede la capacidad de captar las variaciones genéticas, en función del ambiente que rodea al individuo (Mullins, *et al*., 2016). Esta evidencia se ha visto sobre todo en familias cuyas implicaciones de la PRS calculadas a partir de los alelos paternos se manifiestan en los fenotipos de la descendencia. (Lewis and Vassos, 2020)

1.1.2. Aplicaciones

Hay un amplio abanico de utilidades de puntuajes de riesgo poligénico (PRS) que a día de hoy suponen un continuo hábito de investigación en el ámbito biomédico. Entre estas aplicaciones encontramos: estudios de aleatorización mendeliana para inferir relaciones causales, la identificación de etiologías, con el propósito de obtener información del genoma siendo de gran utilidad clínica para el estudio de enfermedades y para estudios experimentales, evaluaciones de interacción gen-ambiente y gen-gen, la estratificación y ordenación de pacientes y subfenotipado. (Choi *et al*., 2020).

Ilustración 2. Aplicación clínica de la genómica en enfermedades humanas

1.2. Cálculo del riesgo poligénico

1.2.1. Control de calidad

Para que nuestro análisis de puntuación de riesgo poligénico tenga un valor y poder significativo, y sea relevante, se debe de realizar un control de calidad (QC) de los datos base y objetivo. De esta manera nos aseguramos que no habrá errores ni a la hora del cálculo ni en los resultados. (Choi *et al*., 2020). Podemos eliminar aquellos sesgos que se producen sobre todo en estudios genéticos de asociación de muestra control y prueba, evitando los posibles falsos positivos o negativos, al reducir el número de asociaciones. (Anderson *et al*., 2010).

Para el control de calidad se recomienda el uso de herramientas como PLINK, para el análisis de datos de SNP y evaluaciones de tasa de fracaso por individuo y SNP, pues esta herramienta puede leer archivos binarios, lo que agiliza el estudio (Marees *et al*., 2018). También hay otras opciones para el análisis de SNP como por ejemplo SNPTEST (Marchini *et al*., 2007) y Genabel (Aulchenko *et al*., 2007). Para la identificación de valores inusuales ancestrales se puede hacer uso de SMARTPCA (Anderson *et al*., 2010). En el caso de estudios de genotipos en familias hay otras herramientas que permiten visualizar la asociación en GWAS (Ott, Kamatani and Lathrop, 2011)

Además para poner en práctica todos los apartados que conlleva el control de calidad, se puede consultar el siguiente tutorial: https://choishingwan.github.io/PRS-Tutorial/

1.2.2. Cálculo de las puntuaciones de riesgo poligénico

La puntuación del riesgo poligénico generalmente se calcula mediante la suma de una ponderación de los alelos de riesgo, puntuando estos con valores de 0, 1 o 2, según su tamaño de efecto, es decir, relación impar logarítmica o coeficiente beta, viéndose implicados un gran número de SNP asociados. (Nguyen *et al*., 2022)

$$\mathrm{PR\_S}_i = \sum_{j=1}^{metro} X_{yo} \hat{\beta}_{jj}$$

Ilustración 3. Fórmula usada para el cálculo del riesgo poligénico donde X_{yoi} y X_{jj} es el genotipo para el i $^{-ésimo}$ individuo y el j $^{-ésimo}$ SNP

La divergencia genómica puede darse por multitud de aspectos como variaciones estructurales, mutaciones, efectos de los propios SNP o incluso por diferencias en la recombinación. Al haber multitud de posibilidades, suele ser difícil y poco frecuentes de caracterizar. Aun así, se ha observado cómo estas variaciones dejan una pequeña marca o señal en los SNP a lo largo del genoma permitiendo su estudio en amplias muestras de población (Balagué-Dobón *et al*. 2022).

Por tanto, el siguiente paso para el estudio del genoma es la construcción de una matriz de SNP para la obtención de datos genéticos, donde el resultado final es una puntuación única, reflejando la carga genética que presenta un individuo, la cual será proporcional al riesgo de adquirir una determinada enfermedad.

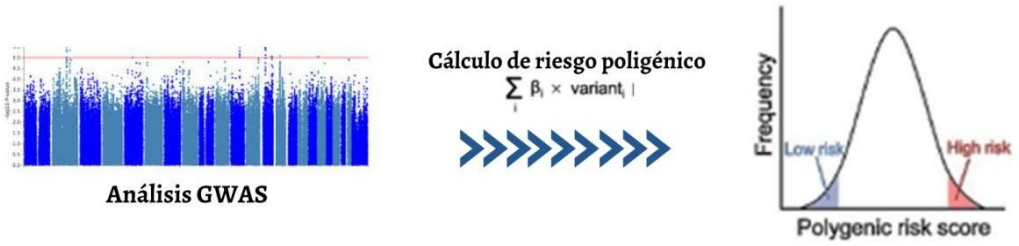


Ilustración 4. Descripción general de la obtención de puntajes de riesgo poligénico.

1.2.3. Disminución de las estimaciones del tamaño del efecto GWAS y control del desequilibrio del ligamento (LD)

Las matrices SNP se construyen por la formación del llamado “etiqueta SNP”, es decir, por la selección de un conjunto de SNP que reflejan el máximo de variantes de ADN sin genotipo, por la asociación entre los alelos en la población, a esto se le conoce como desequilibrio del Ligamento o LD (Nguyen *et al*., 2022). Hay mayor número de correlaciones entre SNP en regiones donde se da el desequilibrio del ligamento, lo que conlleva a una sobreexpresión en estas secciones. Para la eliminación y/o selección de concretos SNP se hace

uso de unos modelos más complejos como son LDpred y Lassosum realizando la poda de LD, reteniendo los SNP más significativos (Lambert *et. al*., 2019) (Mak *et al*., 2017) (Vilhjálmsson *et al*., 2015).

Aun así, dado que la mayor parte de las variantes genéticas son concretas y raras, para el análisis de PRS a una escala mayor requiere que esa personalización de las matrices SNP se generalice, es decir, se resuma el tamaño de los efectos de los SNP individuales para posteriormente poder aplicarlas a superpoblaciones. (Nguyen *et al*., 2022) (Lambert *et al*., 2019) (Chen *et. al.*, 2020).

1.2.4. Limitaciones presentes a la hora de calcular el riesgo poligénico

A pesar de la simplicidad y rapidez del método de puntuación del riesgo poligénico, hay una serie de limitaciones que suponen nuevos desafíos para la ciencia, como por ejemplo, el reducido tamaño de GWAS usados en estudios anteriores, afectando a la precisión del impacto estimado en el riesgo de manifestar la enfermedad. Otro inconveniente es la limitación de los métodos computacionales para la elaboración de una predicción fielmente representativa, así como la carencia de amplios conjuntos de datos requeridos para ratificar y ensayar la RPS. (Khera *et al*., 2018).

Además la mayoría de estudios de riesgo poligénico se han realizado utilizando individuos europeos por lo que la gran parte de los GWAS usados para la imputación están sesgados, inclinados hacia las ascendencias europeas, esto conlleva a una menor precisión en datos de ascendencia no europea. Otro aspecto a tener en cuenta es la influencia del sexo, este parámetro está poco investigado aunque se han observado diferencias predictivas de PRS. Actualmente la mayoría de GWAS no se han distinguido entre sexos, e incluso a menudo, excluyen los cromosomas sexuales, en concreto el X a la hora de realizar el análisis. (Khramtsova *et al*., 2019) (Lambert *et al*., 2019).

1.3. Herramientas para el cálculo del riesgo poligénico

A la hora de identificar las variaciones nucleotídicas en genes, así como su efecto sobre la proteína involucrada, el uso de técnicas convencionales supone una gran dificultad, sumándole su lento y laborioso proceso a la hora de estudiarlo. Para amenizar estos inconvenientes es de gran ayuda la ejecución de programas de biología computacional. (Al Mehdi *et al*., 2019)(Takahashi *et al.,* 2012).

En los últimos años, la cantidad de datos de todo el genoma utilizados para estudios de asociación ha aumentado considerablemente, introduciendo millones de variantes nuevas que posteriormente serán medidas para cada individuo. Por tanto, la actualización y optimización del software debe ser primordial para evitar la obsolescencia. (Al Mehdi *et al*., 2019)

Para el cálculo de riesgo poligénico se puede hacer uso de diversas herramientas. La elección de una u otra será en función de nuestro objetivo y compatibilidad con los datos obtenidos. Entre las más utilizadas, por su rapidez y eficiencia, encontramos:

- PRSice y PLINK: utilizan métodos de agrupamiento, o poda, y umbralización, donde se seleccionan un conjunto de variables genéticas realizando una "poda" en el desequilibrio del ligamento (LD), teniendo en consideración la asociación con el rasgo que se estudia (Lewis and Vassos, 2020).
- LDpred: utiliza un enfoque bayesiano basado en la correlación entre las variables, sin centrarse en la identificación de un subconjunto de SNP, es decir, realiza la predicción del total del genoma. (Lewis, and Vassos, 2020).
- PRS-CS (Poligenic risk score-continuous shrinkage): es un método de predicción poligénica bayesiana caracterizado por la realización de una reducción continua (CS) previa en los tamaños de efecto SNP. Utiliza estadísticas de resumen del genoma total junto, con un panel externo de desequilibrio del ligamento (LD). Esta herramienta aporta grandes ventajas computacionales, además de permitir un estudio multivariante de los patrones focalizados de LD (Ge *et al*., 2019).
- SBayesR: Es una herramienta muy novedosa, pudiendo ser una de las más potentes que existen hoy en día para el cálculo de riesgo poligénico, aunque se encuentra aún en pleno desarrollo. Utiliza una regresión múltiple bayesiana empleando estadísticas resumidas del genoma completo (GWAS), lo que aporta mejor predicción a su método. (Lloyd-Jones *et al*., 2019)

Por lo tanto, una vez descrito el riesgo poligénico y los procedimientos generales para la obtención de su valor, podemos visualizar dichos pasos en el siguiente esquema:

Datos control

Datos problema

CONTROL DE CALIDAD

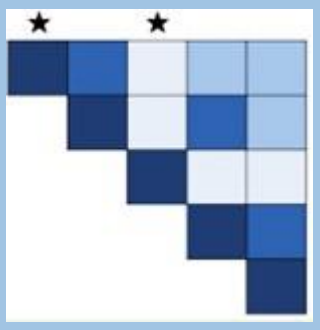
Control de calidad de ambos conjuntos de datos de GWAS

Algunos controles de calidad requieren un cuidado especial en PRS

Mantener el conjunto de SNP que se superponen entre los datos base y objetivo.

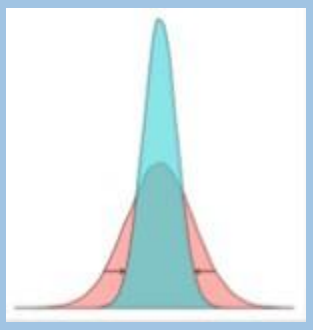
CÁLCULO DE PUNTUACIÓN DE RIESGO POLIGÉNICO

LDadjustment



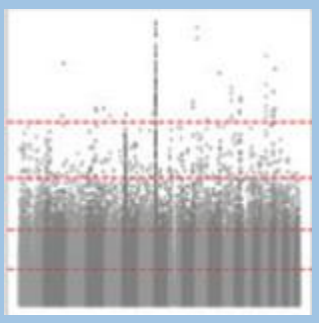
Ejemplo: Clumping

Beta Shrinkage



Ejemplo: Lasso/ridge

p-value thresholding

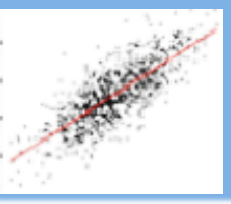


Ejemplo: PRS como multiple p

ANÁLISIS

Obtención PRS y pruebas de asociación

ID	BMI	PRS
101	24.1	0.43
102	28.3	1.61
103	31.2	0.83
104	19.4	3.54



PRUEBAS DE VALIDACIÓN: PRUEBAS OUT-OF-SAMPLE PRS

K-fold cross-validation

Pruebas complementarias

2. OBJETIVO

El objetivo principal de este Trabajo de Fin de Grado es desarrollar Script computacionales de aplicaciones de riesgo poligénico.

De este objetivo general derivan los siguientes objetivos específicos:

-Conocer qué es el “Riesgo Poligénico”.

-Conocer cómo se calcula el riesgo poligénico.

-Detectar herramientas de cálculo de riesgo poligénico.

-Aplicar el uso de una de las herramientas en un caso concreto: la enfermedad de arterias coronarias (EAC)

-Integrar diferentes conocimientos adquiridos a lo largo del grado.

3. MATERIAL Y MÉTODOS

3.1. Marco general

Para la realización de este trabajo se ha utilizado el programa “Rstudio”. Este realiza análisis de datos usando un lenguaje de programación que posibilita la asociación de funciones nuevas o presentes, con la finalidad de realizar análisis de grandes dimensiones, interactivos y reproducibles (R Core Team, 2013).

Rstudio consta de 15303 paquetes, cada uno con una aplicación específica, lo que confiere a este programa una gran versatilidad, pudiéndose aplicar en numerosos campos como cálculos bayesianos, financieros, wavelets, graficación de mapas etc. (Jiménez, 2019)

Para el cálculo y estudio de riesgo poligénico se ha utilizado el paquete “bigsnpr”, el cual depende de “bigstatsr”. Este último paquete provee de diversas funciones con el fin de realizar un rápido análisis estadístico de datos a gran escala encriptado como matrices. Además, puede emplear matrices de un tamaño mayor al de la memoria RAM por atribuir esa memoria a archivos binarios del disco, usando FBM (un tipo especial de matriz grande respaldada por archivos) (Privefl, 2023).

Como se ha comentado anteriormente, el paquete bigsnpr depende de bigstatsr, utilizando un tipo particular de FBM, llamado 'FBM.code256', con la finalidad de almacenar genotipos. Bigsnpr ejecuta algoritmos concretos para el análisis de matrices de polimorfismo de un solo nucleótido (SNP), como llamamientos a software externo para aspectos de procesamiento, ejecuciones de entrada/salida (E/S) de archivos PLINK binarios y operaciones de análisis de datos en datos SNP (adelgazamiento, prueba, predicción y trazado). (Privefl, 2023).

3.1.1. Funciones Bigsnpr

A continuación se pueden observar las diversas funciones del paquete bigsnpr obtenidas de un estudio realizado por Privé *et al*. en 2018. Las funciones requeridas para el desarrollo de este trabajo serán indicadas en el apartado de

resultados. Para más información acerca de este paquete se puede consultar: https://privefl.github.io/bigsnpr/index.html

Tipos bigSNP	
bigSNP-class/bigSNP	Representan llamadas de genotipo y datos de alelos imputados.
snp_subset()/subset(<bigSNP>)	Subconjuntos de bigSNP.

Funciones de entrada/salida	
snp_readBed() snp_readBed2()	Lee archivos PLINK almacenados en un “bigSNP” objeto y crea otro archivo que almacena los valores de la matriz FBM.
snp_writeBed()	Usado para escribir archivos bed/bim/fam desde un bigSNP.
snp_readBGEN()	Leer archivos BGEN en un "bigSNP"
snp_readBGI()	Leer información de variantes de un archivo BGI
snp_attach()	Cargar un "bigSNP" desde los archivos de respaldo encontrados en R.
snp_attachExtdata()	Adjunte un "bigSNP" para ejemplos y pruebas.

Control de calidad/parentesco/imputación	
download_plink() **download_plink2()**	Descargar PLINK
download_1000G()	Descargar 1000G
snp_plinkQC()	Control de calidad
snp_plinkKINGQC()	Poda basada en relaciones
snp_plinkIBDQC()	Identidad por descendencia
snp_plinkRmSamples()	Eliminar muestras
download_beagle()	Descargar Beagle 4.1
snp_beagleImpute()	imputación
snp_fastImpute()	Imputación rápida
snp_fastImputeSimple()	Imputación rápida

Pruebas múltiples - parcelas	
snp_gc()	Control Genómico
snp_qq()	Gráfico QQ
snp_manhattan()	parcela manhattan
snp_pcadapt() bed_pcadapt()	detección de valores atípicos
snp_MAX3()	estadística MAX3
snp_fst()	Índice de fijación

Utilidades	
snp_prodBGEN()	Producto matriz BGEN
snp_match()	Coincidencia de alelos
snp_asGeneticPos()	Interpolar a posiciones genéticas
snp_ancestry_summary()	Estimación de proporciones de ascendencia
snp_simuPheno()	Simular fenotipos
snp_modifyBuild()	Modificar la construcción del genoma
snp_split()	Split-parApply-Combinar
snp_save()	Guardar modificaciones
snp_getSampleInfos()	Obtener información de la muestra
snp_MAF()	MAF
snp_scaleAlpha() **snp_scaleBinom()**	Escala binomial (n, p)
same_ref()	Determinar la divergencia de referencia
sub_bed()	Reemplace la extensión '.bed'
seq_log()	Secuencia, espaciada uniformemente en una escala logarítmica
coef_to_liab()	Escala de responsabilidad

Archivadores de cama PLINK	
bed()	cama de referencia
bed_MAF()	Frecuencias alélicas
snp_autoSVD() bed_autoSVD()	SVD truncado mientras limita LD
bed_clumping() snp_clumping() **snp_pruning() snp_indLRLDR()**	Aglomeración de LD
bed_counts()	cuenta
bed_cprodVec()	Producto cruzado con un vector
snp_pcadapt() bed_pcadapt()	Detección de valores atípicos
bed_prodVec()	Producto con un vector
bed_projectPCA()	Proyección de PCA
bed_projectSelfPCA()	Proyección de PCA
bed_randomSVD()	SVD parcial aleatorizado
bed_scaleBinom()	Escala binomial (2, p)
bed_tcrossprodSelf()	tcrossprod / GRM

3.2. Archivos mapeados en memoria

Para el acceso a los datos, el paquete bigsnpr utiliza el llamado: mapeo de memoria a través del paquete BH dentro de R. De esta manera, se puede acceder a los datos situados en el disco como si estuvieran en la memoria (Privefl, 2023).

3.3. Gestión de datos, preprocesamiento e imputación

Para la gestión de datos se utiliza un único formato, un objeto FBM particular, llamado 'FBM.code256' para almacenar tanto llamadas de genotipo como datos. Puede incorporar 256 valores distintos arbitrarios ocupando un pequeño espacio (Privefl, 2023).

En cuanto al procesamiento de los datos, se utiliza PLINK mediante llamadas del sistema. Gracias a este paquete se puede realizar los controles de calidad, uso de diversos formatos de entrada/sallida (E/S) como por ejemplo vcf, bed/bim/fam, ped/map como archivos entrada y bed/bim/fam, como archivos de salida.
Por último, el paquete bigsnpr facilita la conversión veloz entre los archivos PLINK bed/bim/fam y el objeto 'bigSNP'. Este último contiene el genotipo FBM (FBM.code256), un conjunto de datos que incluye detalles acerca de las muestras, así como otro conjunto de datos que brinda información sobre los SNP (Privefl, 2023).

3.4. Pruebas de asociación y puntaje de riesgo poligénico

Los paquetes de R también contienen funciones para el cálculo de PRS mediante dos métodos distintos, ambos utilizados en este estudio. El primero de ellos es el ampliamente utilizado método “Clumping + Thresholding” o “Agrupación + Umbral” (C + T), también conocido como “Pruning + Thresholding” o "Poda + Umbral". Este método se basa en estadísticas de resumen de GWAS univariadas. Bajo el modelo C + T, se aprende un coeficiente de regresión de forma independiente para cada SNP, junto con un valor *p* correspondiente (parte de la GWAS). En primer lugar, los SNP se agrupan (C) de manera que solo se mantienen aquellos que presentan una correlación débil entre sí. A continuación, se realiza la umbralización (T), que consiste en eliminar aquellos SNP que presentan un nivel de significación por debajo de un determinado umbral de valor *p*. La PRS se define como la suma de los recuentos de alelos de los SNP restantes, ponderados por los coeficientes de regresión correspondientes (Privefl, 2023).

Por otro lado, el segundo método que lleva a cabo tanto bigstatsr como bigsnpr es la ejecución de diversos modelos. Son principalmente regresiones lineales dispersas y logísticas muy rápidas donde se disminuye el número de SNP incorporados en los modelos predictivos (Privefl, 2023).
En concreto se ha utilizado el modelo LDpred2-grid para el cálculo de puntuaciones poligénicas y estimación de la incertidumbre de las mismas.
LDpred2-grid junto con LDpred2-auto son dos modelos nuevos de LDpred2 siendo más eficaces y rápidos para estudios de riesgo poligénico. El modelo

principal de LDpred es LDpred2-grid, el cual ajusta una cuadrícula de valores para los hiperparámetros *p* (la proporción de variantes causales), h2 (la heredabilidad del SNP) y, posiblemente, la opción de escasez (como un tercer hiperparámetro) mediante un conjunto de validación (Privé *et al*., 2021).

3.5. Procedencia de datos

Los datos utilizados en este trabajo proceden de dos fuentes: Una del proyecto de los mil genomas, realizado por Auton *et al*., en 2015. Este proyecto consistía en la obtención de una descripción exhaustiva de la variación genética humana común. Para ello, se emplearía la secuenciación del genoma completo en un conjunto de diversos individuos pertenecientes a múltiples poblaciones de origen africano, asiático, europeo, asiático oriental y asiático meridional. Estos datos fueron utilizados para poner de manifiesto las diferentes funciones de bigsnpr y para el cálculo de puntuaciones poligénicas mediante agrupamiento y umbralización apilados (SCT). Se pueden obtener los datos a través del siguiente enlace:
www.dropbox.com/s/k9ptc4kep9hmvz5/1kg_phase1_all.tar.gz?raw=1

Por otro lado, se trabajó con los datos tomados de un estudio realizado por Reed *et al*., en 2015, acerca de la enfermedad de las arterias coronarias (EAC) y los factores de riesgo cardiovascular. Los datos que seleccionó este proyecto proceden de un estudio anterior realizado por Devaney *et al*., en 2011. En el análisis, se incorporó a la muestra un total de n = 3850 participantes reclutados durante el período comprendido entre julio de 1998 y marzo de 2003. Se llevó a cabo un estudio de casos y controles anidado, donde se incluyeron individuos de ascendencia europea con diagnóstico de EAC angiográfico grave como casos, así como controles angiográficos normales, para la genotipificación de todo el genoma. Estos datos fueron utilizados para el cálculo de riesgo poligénico usando LDpred2-grid.

4. RESULTADOS

Este apartado podemos dividirlo en cuatro secciones en función de lo que se quiere analizar:

4.1. Características principales de las funciones de cálculo poligénico utilizando el paquete bigsnpr.

Como hemos visto anteriormente bigsnpr presenta una gran variedad de funciones. En este caso, se pueden poner de manifiesto algunas de ellas para comprender sus funciones y características:

Primero de todo se extraen los datos y se agrupan. Esto se consigue con la función “str” de la siguiente manera:

```
str(obj.bigSNP, max.level = 2, strict.width = "cut")
```

De esta manera tenemos todos los datos compactos del conjunto de individuos. Datos de genotipo, de procedencia (europea, africana y euroasiática), sexo (parental y del individuo) y el mapa genético.

Ahora se crean los siguientes archivos con los que se trabajarán en el estudio:

```
G   <- obj.bigSNP$genotypes
CHR <- obj.bigSNP$map$chromosome
POS <- obj.bigSNP$map$physical.pos
y   <- obj.bigSNP$fam$affection - 1
sex <- obj.bigSNP$fam$sex
pop <- obj.bigSNP$fam$family.ID
NCORES <- nb_cores()
```

Ilustración 5. G: se refiere a la matriz FBM que utiliza bigsnpr; CHR: vector que especifica el cromosoma de cada SNPs; POS: vector que especifica la la posición física en un cromosoma de cada SNP; y:vector de fenotipos (uso de código binario 1/0) ; sex: sexo de cada individuo (1 para hombres, 2 para mujeres); pop: procedencia de los individuos (africana; euroasiática; europea) (*Analysis of Massive SNP Arrays,* s. f.-a., 2023)

A partir de este punto ya podemos trabajar con las funciones que nos ofrece bigsnpr para nuestro estudio de riesgo poligénico.

4.1.1. Análisis de los componentes principales (PC) con el fin de estudiar la estructura de la población.

Usamos la función "big_randomSVD" para calcular los principales componentes, o PC, de una matriz de genotipo de gran escala. Utiliza proyecciones aleatorias como método para la descomposición de valores singulares parciales (SVD), usando un algoritmo PCA, llamado Arnoldi, que hace uso de multiplicaciones de matriz vectorial en función del formato FBM.code256 (Privé, 2022a)

Para la representación de los valores del índice de los componentes principales ejecutamos:

```
svd <- big_randomSVD(G, big_scale(), ncores = NCORES)
```

"svd" contiene:

- d: valores de estudio→ 2153.2889 / 2039.0605 / 647.4233 / 561.8824 / 5 08.8336 / 498.7515 / 485.2461 / 480.1129 / 463.1290 / 460.2539
- u: los vectores eje izquierda
- v: los vectores singulares eje derecha
- niter: el número de la iteración del algoritmo→ 6
- nops: número de multiplicaciones Matriz-Vector utilizadas→ 114
- center: el vector de centrado.
- scale: el vector de escala.

Mediante la función ggplot podemos representar en una gráfica los valores (d) en función de los componentes principales, 10 por defecto, mediante una proyección aleatoria.

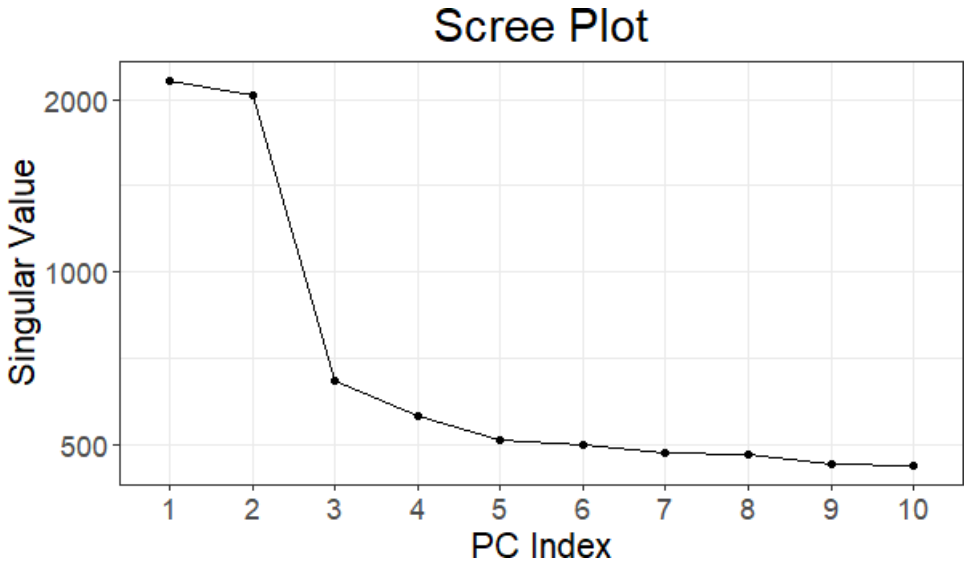


Ilustración 5. Representación del índice de los componentes principales (PC) en función de los valores (d) de la carpeta svd.

Para representar las primeras gráficas de los componentes principales, usamos:

```
plot(svd, type = "scores") + aes(color = pop)
```

La función “aes” se utiliza para realizar mapeos estéticos, son propiedades visuales que se pueden asignar a variables (x e y) para revelar información acerca de nuestros datos (Grolemund, 2023).

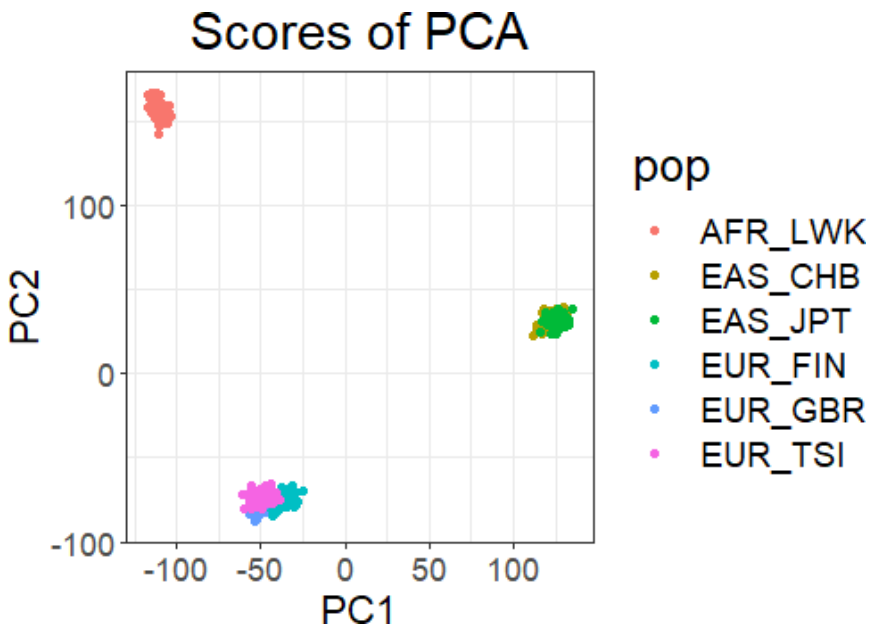


Ilustración 6. Representación de las dos primeras PC en función de sus lugares de procedencia.

Podemos representar las demás PC de la siguiente manera:

```
plot(svd, type = "scores", scores = 3:4) + aes(color = pop)
```

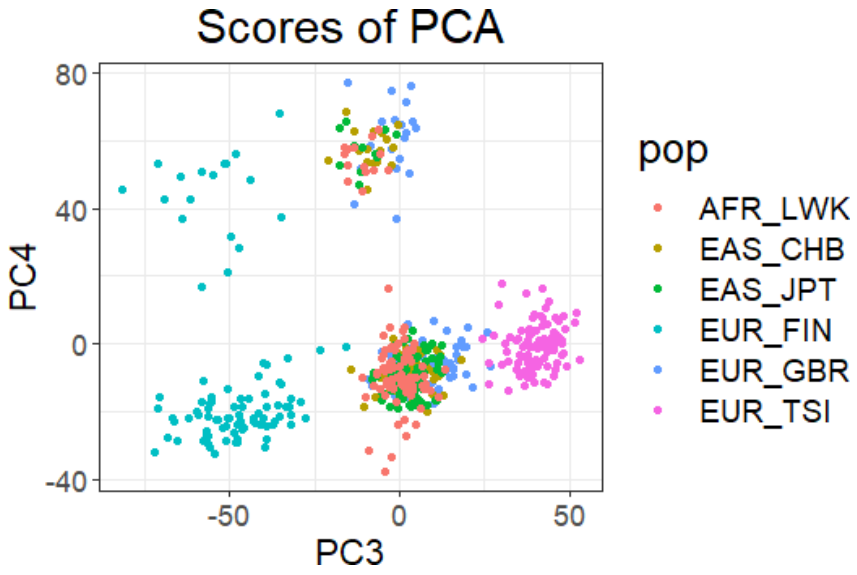
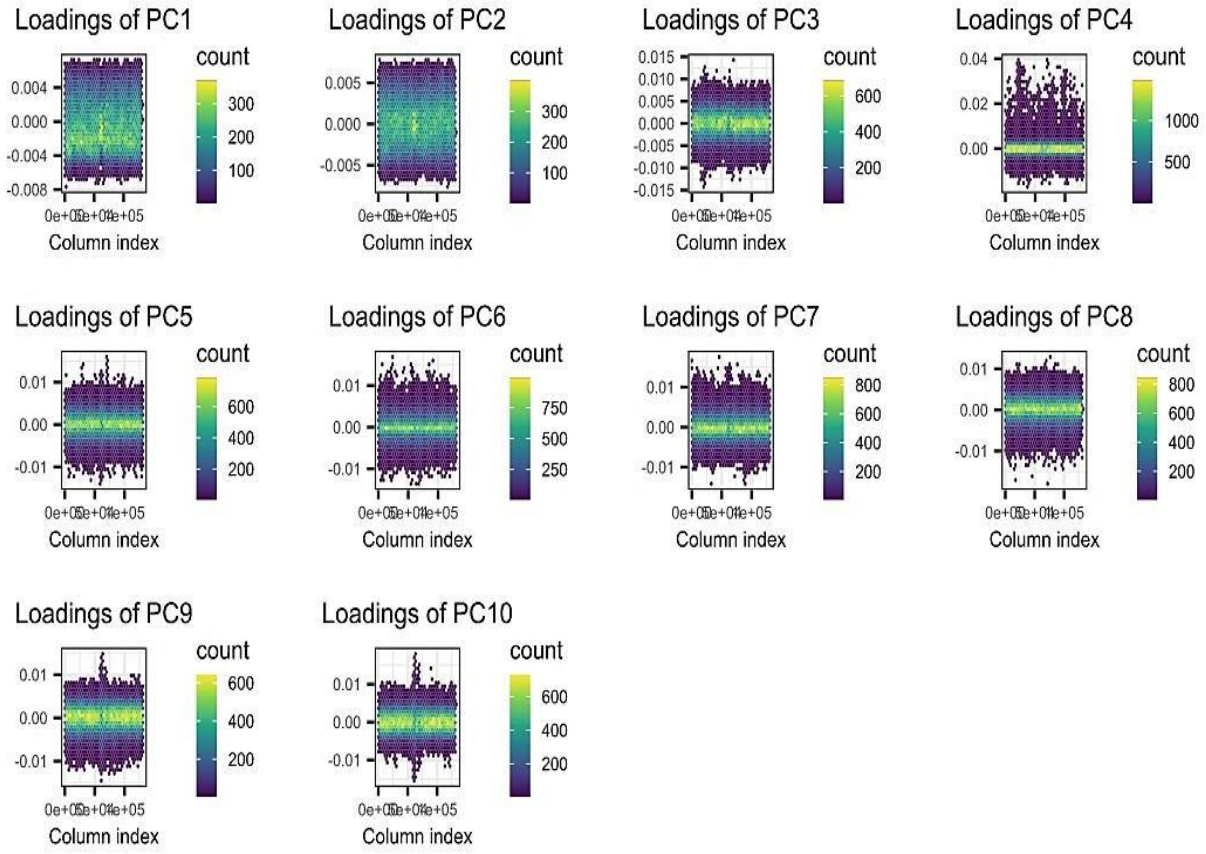


Ilustración 7. Representación de los PC 3 y 4 según sus lugares de procedencia.

En esta gráfica, sin embargo, vemos cómo hay un solapamiento de puntos de las poblaciones africanas y euroasiáticas (AFR y EAS). Esto podría indicar que hay una cierta correlación entre los datos.

Para visualizar las cargas de cada PC:

```
plot(svd, type = "loadings", loadings = 1:10, coeff = 0.4)
```



Al representar las cargas de las PC comprobamos que efectivamente hay ciertas correlaciones entre nuestros datos. Para subsanar este problema podemos realizar una poda o un agrupamiento de los datos con el fin de mantener un subconjunto de SNP que casi no están correlacionados entre sí.

4.1.2. Estudio de asociación del genoma completo o GWAS

Para el estudio y representación de GWAS utilizamos la siguiente función:

```
gwas <- big_univLogReg(G, y, covar.train = svd$u, ncores = NCORES)
```

La función “big_univlogreg” se refiere a pendientes de regresiones logísticas por cada columna de una matriz grande de archivos.
Para la representación de GWAS, “big_univlogreg” utiliza los datos de genotipo, “y”, refiriéndose a un vector de respuestas dependiente de “ind.train”, otro vector que emplea aquellas filas que se le indique (si no se especifica utiliza todas las filas por defecto). En este caso, se utiliza los datos de “u” de “svd”. (Privé, 2022a)

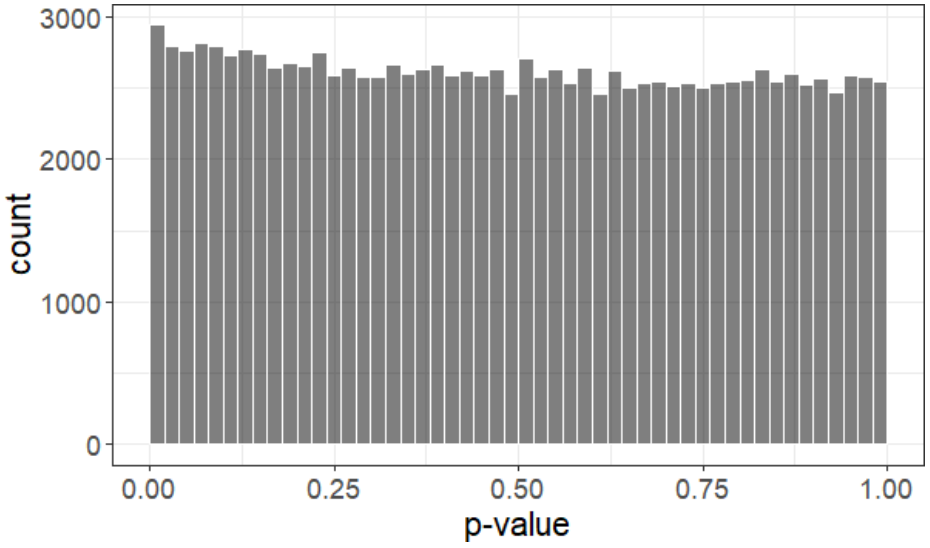


Ilustración 8. Histograma de GWAS

Una vez obtenido el histograma de la asociación completo del genoma podemos evaluar si nuestros datos siguen o no una distribución normal (Gómez-Herazo, 2018). Para ello se genera el gráfico "Q-Q" o "cuantiles-cuantiles" de la siguiente forma:

```
plot(gwas, type = "Q-Q") + xlim(1, NA)
```

"xlim" es una función que aporta un valor numérico en el eje "x", en este caso, 1. NA se utiliza para calcular el límite referido al rango de datos utilizados. (Privé, 2022a)

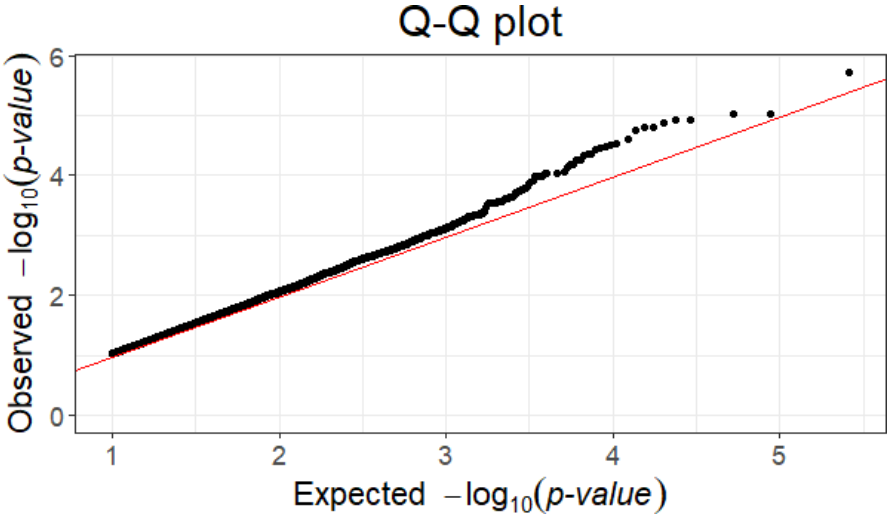


Ilustración 9. Gráfico "cuantiles-cuantiles"

Tras la obtención de la gráfica, podemos observar que, efectivamente, los datos siguen una distribución normal pues los puntos del gráfico se aproximan a la línea diagonal recta.

Los estudios de asociación del genoma completo (GWAS) también utilizan los diagramas de Manhattan como herramienta gráfica. Estos diagramas permiten visualizar gran cantidad de puntos de datos, algunos de ellos no significativos y con bajos valores variables, mientras que otros puntos de datos relevantes se agrupan en torres en el gráfico. Por lo general, los valores de "*p"* se utilizan para generar los gráficos, pero se transforman mediante la función -log10 (pval), de manera que los valores más pequeños de "*p*" se corresponden con valores transformados más altos. (Gel and Serra, 2017). Se puede obtener dicho diagrama de la siguiente manera:

```
snp_manhattan(gwas, CHR, POS, npoints = 20e3) +
  geom_hline(yintercept = -log10(5e-8), color = "red")
```

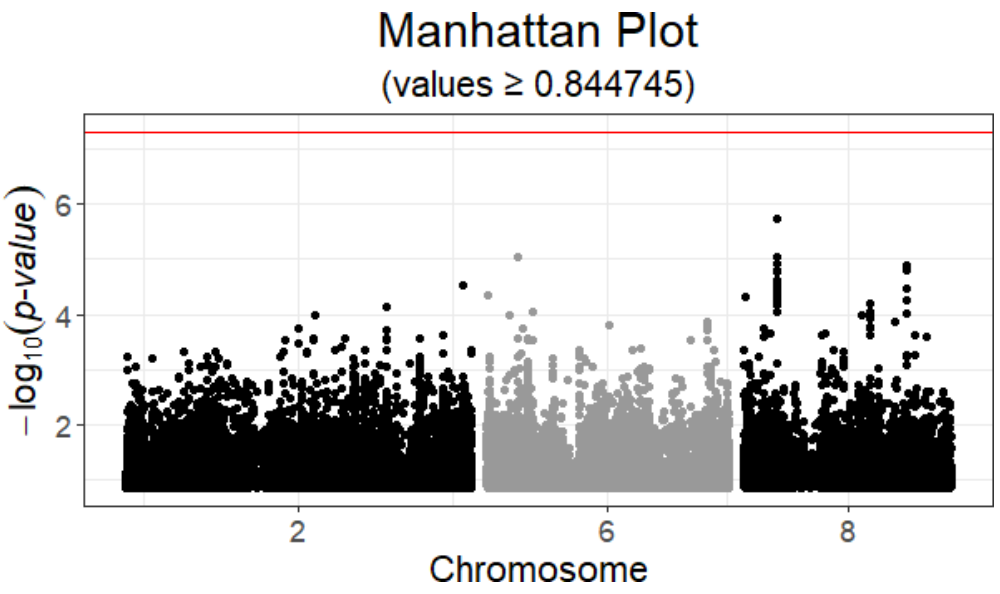


Ilustración 10. Gráfico de Manhattan. No muestra ningún dato significativo por la ausencia vidual de torres de puntos en él.

4.2. Puntuación de riesgo poligénico (PRS)

4.2.1. Clumping and Thersholding (C+T)

Método basado en la aglutinación de LD. Su cálculo se realiza de la siguiente forma:

```
sumstats <- bigreadr::fread2("public-data-sumstats.txt")
lpval <- -log10(sumstats$p)
ind.keep <- snp_clumping(G, CHR, ind.row = ind.train, S = lpval, infos.pos
THR <- seq_log(1, 8, length.out = 20)
prs <- snp_PRS(G, sumstats$beta[ind.keep], ind.keep = ind.keep,
               lps.keep = lpval[ind.keep], thr.list = THR)
```

Mediante la función de “fread2”, se leen los archivos guardados en el archivo “public-data-sumstats.txt” y se crea un vector nuevo, “sumstats”. Sobre los datos de este, se realiza el logaritmo en base 10 y el resultado se guarda como “lpval”. Gracias a la función “snp_clumpling” obtenemos la aglutinación de LD. Para ello se necesita la matriz FBM.code256 (G); el vector de posición de cada cromosoma (CHR); un vector que especifique qué filas y columnas se deben utilizar (ind.row=ind.train); un vector de estadística que refleja la relevancia de cada SNP (S) y, por último, un vector que refleje la posición exacta del SNP en el cromosoma (info.pos).
Ahora se genera “THR” con los valores obtenidos a partir de la función seq_long. Esta indica los valores iniciales y finales que se usaran en la secuencia, en este caso, del valor 1 al 8, y, con una longitud (length.out) de 20.

Una vez realizado el agrupamiento de los valores, se debe establecer el umbral óptimo para esta prueba. Para ello:

```
aucs <- apply(prs[ind.train, ], 2,
              AUC, target = y[ind.train])
plot(THR, aucs, xlab = "-log10(p-value)",
     ylab = "AUC", pch = 20)
```

El área bajo la curva ROC o AUC, es un buen método para evaluar el desempeño de las PRS en un conjunto de datos GWAS (Song *et al*., 2019). Para ello, se hace uso de la función “apply”, que, como su nombre indica, aplica una función a una matriz obteniendo un vector, matriz o lista de valores.

Los valores obtenidos son los siguientes:

EJE X	EJE Y
1	0.6397955
1.11565791776154	0.6370835
1.24469258946402	0.6397955
1.38865114261466	0.6413230
1.38865114261466	0.6384239
1.72844378656321	0.6386421
1.92835199588499	0.6309424
2.15138117244036	0.6262976
2.40020543915621	0.6151065
2.67780820244894	0.6148259
2.98751792330897	0.6281680
3.33304802559418	0.6037283
3.71854142003362	0.5586365
4.14862017778477	0.5329811

4.62844094913088	0.5588391
5.16375679178962	0.5000000
5.76098615015504	0.5000000
6.42728981253506	0.5000000
7.17065676910285	0.5000000
8	0.5000000

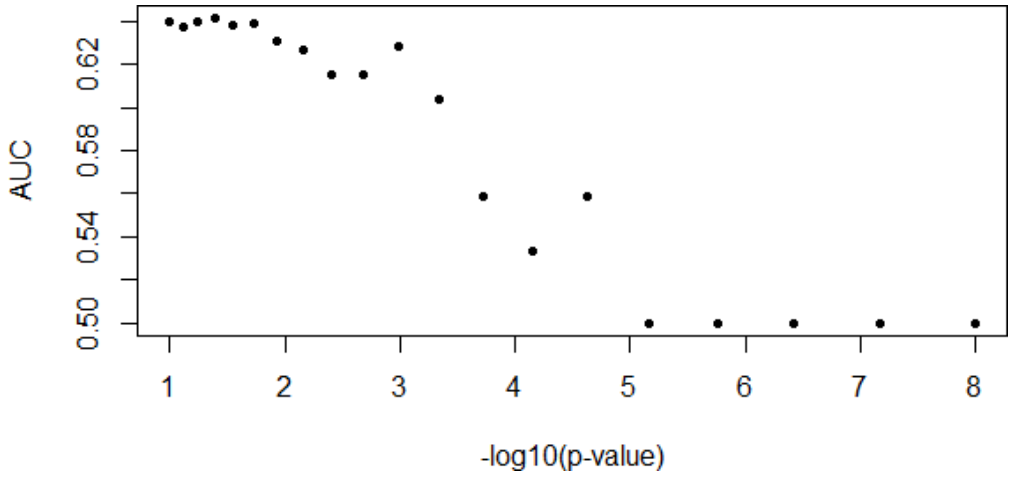


Ilustración 11. Gráfico obtenido tras el análisis de AUC

Una vez obtenidos todos los datos para representar el área bajo la curva ROC, se procede a evaluar la calidad de análisis del test:

```
AUC(prs[ind.test, which.max(aucs)], y[ind.test])
```

En este proceso, se utiliza una nueva función, "which.max", especificando el valor máximo del vector numérico, aucs.

Finalmente, el valor obtenido es: 0.6689111. Cuanto más próximo a 1 sea el valor obtenido, mejor será su análisis. Por tanto, obteniendo un valor de 0,669, el análisis realizado puede mejorarse notablemente.

4.2.2. Regresión Logística Penalizada (PRL)

La regresión logística penalizada (PRL) es un método de clasificación que se utiliza habitualmente en la práctica y que generaliza la regresión logística estándar al incorporar un término de penalización en los coeficientes (Park and Liu, 2011). Para realizar este análisis se necesita:

```
mod <- big_spLogReg(G, y[ind.train],
       ind.train, covar.train = svd$u[ind.train, ],
       K = 5, alphas = 10^(-(0:4)), ncores = NCORES)
```

“big_spLogReg” realiza un ajuste de la regresión logística penalizada mediante un lazo (o la red elástica) sobre una matriz de gran tamaño, en este caso, “FBM.code256”. (Privé, 2022a)

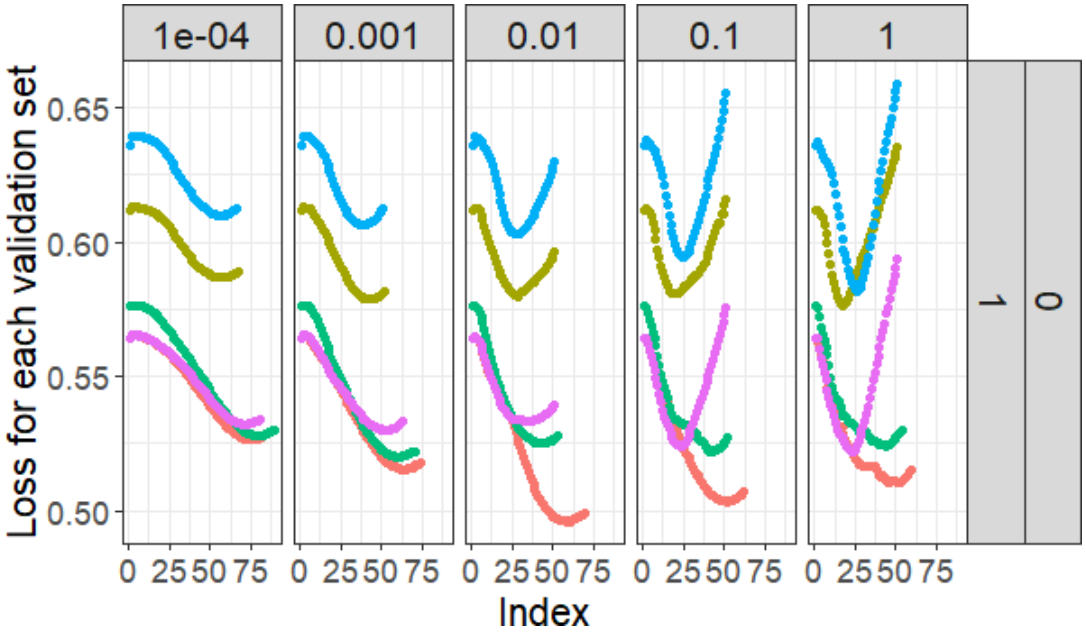


Ilustración 12. Gráfica PRL. Representa los puntos por pérdida de validación donde Alpha=0.0001, 0,001; 0,01; 0,1; 1/k=5 (número de gráficas representadas)

El resumen de los datos es el siguiente:

Alpha	Power_adaptive	Power_scale	Validation_loss	Intercept beta	nb_var
0.0001	0	1	0.557	-1.41	67876
0.001	0	1	0.550	-1.94	16008
0.01	0	1	0.547	-2.16	2928
0.1	0	1	0.545	-2.30	722
1	0	1	0.543	-2.40	398

De nuevo, una vez plasmado nuestros datos en la gráfica, realizamos un análisis para evaluar nuestra predicción de PRL:

```
pred <- predict(mod, G,
ind.test, covar.row = svd$u[ind.test, ])
AUC(pred, y[ind.test])
```

El valor de predicción es 0.7148903, aún sigue siendo mejorable puesto que lo ideal sería un número bastante más cercano a 1, pero en relación al anterior análisis se ha obtenido un valor ligeramente superior.

4.3. Cálculo del riesgo poligénico usando el modelo LDpred2-grid.

4.3.1. Obtención de datos

En este apartado, a diferencia de los demás, se emplean datos procedentes del estudio de Reed *et al*., realizado en 2015, basado en la enfermedad de las arterias coronarias (EAC)

Primero descargamos los datos:

```
zip <- runonce::download_file(
  "https://figshare.com/ndownloader/files/38019072",
  dir = "tmp-data", fname = "GWAS_data.zip")
unzip(zip, exdir = "tmp-data", overwrite = FALSE)
```

Ordenamos los datos en función de su posición en el cromosoma, puesto que al descarnárnoslos no se encontraban organizados. Podemos conseguirlo con la función PLINK y "glue":

```
library(bigsnpr)
plink <- download_plink("tmp-data")
system(glue::glue(
  "{plink} --bfile tmp-data/GWAS_data",
  " --make-bed --out tmp-data/GWAS_data_sorted"
))
```

PLINK emplea tanto métodos unidimensionales como multidimensionales y llevar a cabo enlaces en cadena de ensayos para diversos grupos (Weeks, 2022). "glue" presenta todas las capacidades necesarias para la integración de los datos, procesarlos y analizarlos (Bryan and Maintainer, 2022). El resultado es de 1401 individuos con 500.000 variantes (Privé, 2022b).

Se realiza un control de calidad para "aclarar" nuestros datos y eliminar datos redundantes:

```
bedfile2 <- snp_plinkQC(plink, "tmp-data/GWAS_data_sorted")
```

De esta forma nos aseguramos que nuestros resultados de los análisis posteriores son correctos. Una vez realizado, contamos con 404.663 variantes (Privé, 2022b).

Creamos una matriz, "obj.bigsnp", donde se almacenan datos respecto al genotipo, el mapa genético y datos familiares para la identificación del sujeto.

```
(rds <- snp_readBed2(bedfile2, ncores = nb_cores()))
obj.bigsnp <- snp_attach(rds)
str(obj.bigsnp, max.level = 2)
```

Si quisiéramos añadir más información sobre los sujetos de estudio, como por ejemplo aspectos clínicos o fenotipo, se puede realizar de la siguiente manera:

```
clinical <- bigreadr::fread2("tmp-data/GWAS_clinical.csv")
# Get the same order as for the genotypes
# (to match over multiple columns, use `vctrs::vec_match()`)
ord <- match(obj.bigsnp$fam$family.ID, clinical$FamID)
pheno <- clinical[ord, ]
```

La siguiente tabla muestra datos adicionales características de cada individuo que se han guardados en "pheno":

FamID	CAD	sex	age	tg	hdl	ldl
10002	1	1	60	*NA*	*NA*	*NA*
10004	1	2	50	55	23	75
10005	1	1	55	105	37	69
10007	1	1	52	314	54	108

Ilustración 13. CAD: enfermedad de las arterias coronarias; sex: sexo; age: edad; HDL: "high density lipoprotein"; LDL: "low density protein".

4.3.2. Ldpred-grid

Una vez obtenidos y ordenados todos los datos se realiza el análisis de los mismos, con el fin de calcular la puntuación de riesgo poligénico. Se pueden utilizar diferentes modelos, en este caso, se utiliza LDpred-grid.

- *Disposición de datos*

  En primer lugar se ejecutan los datos obtenidos en el anterior apartado

```
library(bigsnpr)
obj.bigsnp <- snp_attach("tmp-data/GWAS_data_sorted_QC.rds")
G <- obj.bigsnp$genotypes
NCORES <- nb_cores()
map <- dplyr::transmute(obj.bigsnp$map,
                        chr = chromosome, pos = physical.pos,
                        a0 = allele2, a1 = allele1)
```

Gracias a los datos genéticos y fenotípicos, de alrededor de 500.000 personas, recopilados por Biobanco del Reino Unido (Bycroft *et al.*, 2018), podemos descargar algunas estadísticas resumidas de GWAS en referencia a las enfermedades coronarias. (Privé, 2022b).

```
gz <- runonce::download_file(
  "https://figshare.com/ndownloader/files/38077323",
  dir = "tmp-data", fname = "sumstats_CAD_tuto.csv.gz")
readLines(gz, n = 3)
```

Una vez obtenidos, se leen los datos guardados como "gz", seleccionándose solo algunas columnas y guardándose en una estadística resumen, sumstats:

```
sumstats <- bigreadr::fread2(
  gz,
  select    = c("chr", "pos", "allele2", "allele1",
                "beta", "se", "freq", "info"),
  col.names = c("chr", "pos", "a0", "a1",
                "beta", "beta_se", "freq", "info"))
```

chr	pos	a0	a1	beta	beta_se	freq	info	n_eff
1	721290	G	C	0.0361758959	0.02908659	0.03588903	0.9419181	78136.41
1	752566	G	A	-0.0340838523	0.01445730	0.84079991	0.9975869	78136.41
1	777122	A	T	-0.0187253871	0.01550889	0.87106963	0.9973021	78136.41

El siguiente paso es hacer coincidir los datos de las estadísticas resumen GWAS con los datos problema:

```
library(dplyr)
info_snp <- snp_match(sumstats, map,
                      return_flip_and_rev = TRUE) %>%
  mutate(freq = ifelse(`_REV_`, 1 - freq, freq),
         `_REV_` = NULL, `_FLIP_`= NULL) %>%
  print()
```

Gracias a la función snp_match podemos hacer coincidir los alelos de GWAS con la información de SNP (Privé, 2022a). Como resultado obtenemos 62.168 SNP ambiguos se han eliminado, han sido emparejadas 340.210 variantes, 170.170 se han volteado y 101.606, invertido (Privé, 2022b).

chr	pos	a0	a1	beta	beta_se	freq	info	n_eff	_NUM_ID_.ss
1	752566	T	C	0.0341	0.0145	0.1592	0.997	78136	2
1	785989	G	A	0.0183	0.0155	0.13007	0.991	78136	4

Tabla 1. "chr": cromosoma / "pos": posición/ "a0": alelo 0/ "a1": alelo 1/ "beta": coeficiente beta / "beta_se": / "freq": frecuencia alélica/ "info": información /"n_eff": tamaño de la muestra /"_NUM_ID.ss": número identificación.

Después, se puede llevar a cabo un control de calidad de las estadísticas resumidas de GWAS, corroborando si estas coinciden con los datos internos que poseemos (Privé, 2022b):

```
af_ref <- big_colstats(G, ind.col = info_snp$`_NUM_ID_`,
                       ncores = NCORES)$sum / (2 * nrow(G))
sd_ref <- sqrt(2 * af_ref * (1 - af_ref))
sd_ss <- with(info_snp, 2 /
                sqrt(n_eff * beta_se^2 + beta^2))
is_bad <-
  sd_ss < (0.5 * sd_ref) | sd_ss > (sd_ref + 0.1) |
  sd_ss < 0.05 | sd_ref < 0.05
```

“big_colstats” realiza estadísticas en cada columna de las variantes que se le indica y se guarda en un vector, en este caso, af_ref. Mediante “sqrt” obtenemos la raíz cuadrada de “af_ref”, y, “with”, se utiliza para una expresión de r a un entorno introducido por parámetros, es decir, se aplica “sqrt(n_eff * beta_se^2 + beta^2)” a cada una de las filas de “info_snp”, cuyo resultado se guarda en un vector al que llamaremos sd_ss (Privé, 2019).

A continuación representaremos los datos obtenidos haciendo coincidir ambas desviaciones estándar:

```
library(ggplot2)
ggplot(slice_sample(data.frame(sd_ref, sd_ss, is_bad), n = 50e3)) +
  geom_point(aes(sd_ref, sd_ss, color = is_bad), alpha = 0.5) +
  theme_bigstatsr(0.9) +
  scale_color_viridis_d(direction = -1) +
  geom_abline(linetype = 2) +
  labs(x = "Standard deviations in the reference set",
       y = "Standard deviations derived from the summary statistics",
       color = "To remove?")
```

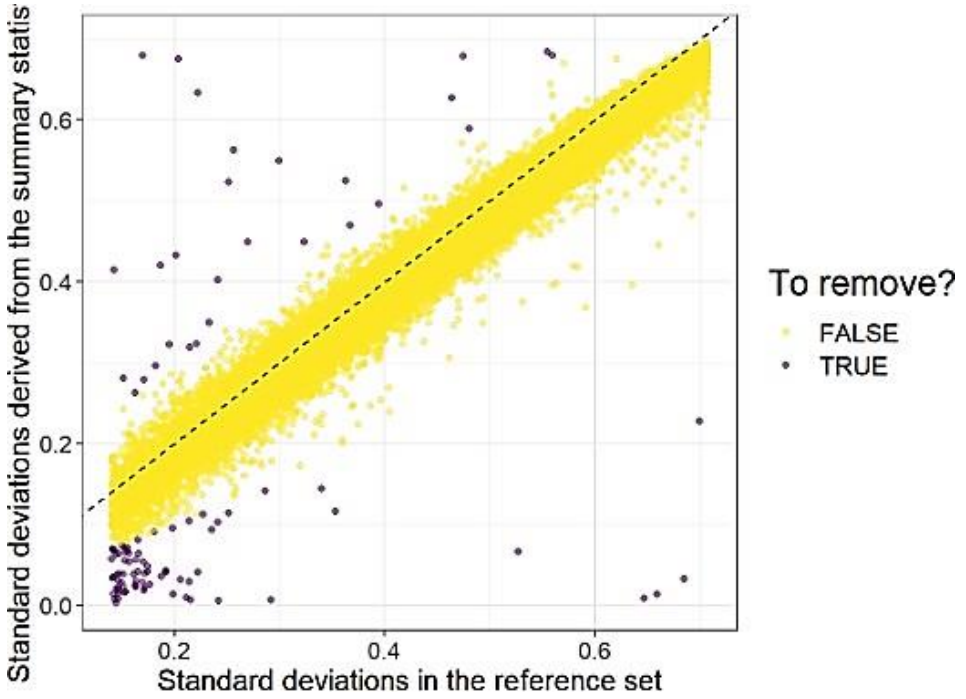


Ilustración 14. Control de calidad de las estadísticas de resumen comparando las desviaciones estándar.

El gráfico muestra los datos que son relevantes para el estudio, representados de color amarillo (FALSE), y también, aquellos que no lo son, siendo de color morado (TRUE).

Refinamos los datos comparando ahora las desviaciones estándar de las frecuencias alélicas

```
sd_af <- with(info_snp, sqrt(2 * freq * (1 - freq) * info))

ggplot(slice_sample(data.frame(sd_af, sd_ss), n = 50e3)) +
  geom_point(aes(sd_af, sd_ss), alpha = 0.5) +
  theme_bigstatsr(0.9) +
  geom_abline(linetype = 2, color = "red") +
  labs(x = "Standard deviations derived from allele frequencies",
   y = "Standard deviations derived from the summary statistics")
```

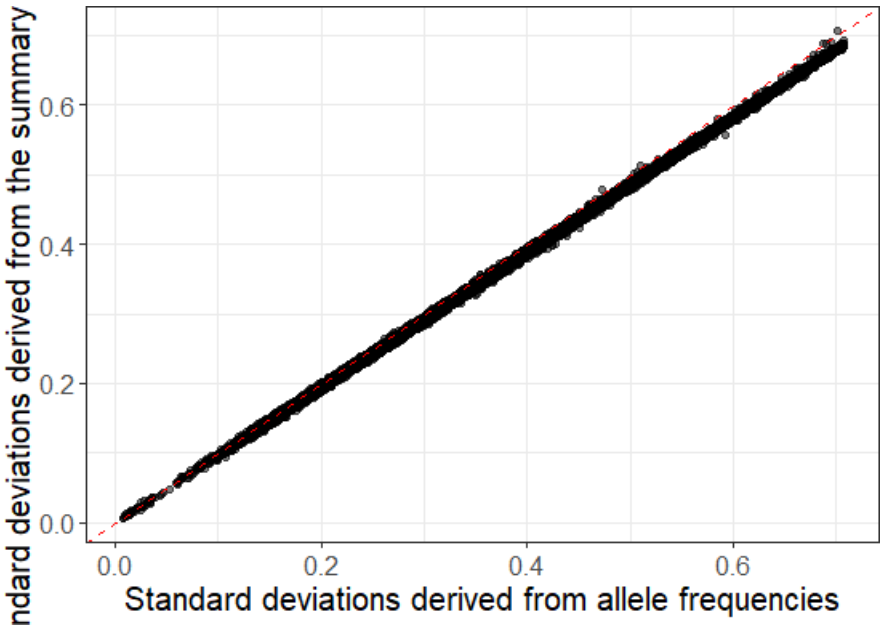


Ilustración 15. Control de calidad de las estadísticas de resumen comparando las desviaciones estándar de las frecuencias alélicas.

Observamos cómo "el ruido" que había en el gráfico anterior (ilustración 14), se ha eliminado, quedado solo aquellos datos más representativos, ceñidos a la función de regresión lineal.

Ahora podemos analizar la correlación que hay entre los cromosomas y creamos la matriz de todo el genoma. Descargamos las posiciones genéticas ya calculadas:

```
gen_pos <- readRDS(runonce::download_file(
  "https://figshare.com/ndownloader/files/38247288",
  dir = "tmp-data", fname = "gen_pos_tuto.rds"))
```

Realizamos la matriz de correlación para 22 cromosomas:

```
for (chr in 1:22)

  print(chr)

  corr0 <- runonce::save_run(
```

Obtenemos los índices para "sumstats" y para el genotipo, cuyos valores son guardados en vectores a los que llamamos "ind.chr" y "ind.chr2", respectivamente:

```
ind.chr <- which(df_beta$chr == chr)
ind.chr2 <- df_beta$`_NUM_ID_`[ind.chr]
```

Para obtener posiciones genéticas (en cM), a partir de posiciones físicas (en pb) usamos la función “gen_pos”:

```
POS2 <- gen_pos[ind.chr2]
```

Posteriormente, “snp_cor” calcula correlaciones significativas de las variables, basándose en el coeficiente de correlación Pearson, tomando como valores aquellos SNP próximos del mismo cromosoma, generando de este modo una matriz. El coeficiente de correlación Pearson o prueba estadística Pearson R, calcula la relación entre diferentes variables. (Privé, 2022a):

```
snp_cor(G, ind.col = ind.chr2, size = 3 /
        1000, infos.pos = POS2,
        ncores = NCORES)

 file = paste0("tmp-data/corr_chr", chr, ".rds"))
```

Generamos una matriz donde almacenamos todos los datos, matriz SFBM:

```
if (chr == 1) {
  ld <- Matrix::colSums(corr0^2)
  corr <- as_SFBM(corr0, "tmp-data/corr", compact = TRUE)
} else {
  ld <- c(ld, Matrix::colSums(corr0^2))
  corr$add_columns(corr0, nrow(corr))
}
```

En este caso la matriz SFBM cuenta con 343856 filas y 343856 columnas.

A continuación, extraemos todos los datos gracias a la función “file.size”:

```
file.size(corr$sbk) / 1024^3
```

Generando un único valor que representa la correlación que existe en todos los cromosomas de la matriz: 1.838061. Este valor nos indica que hay una correlación positiva entre las variables, de tal forma que si el valor de una aumenta o disminuye, causa el mismo efecto en la otra.

4.3.3. Análisis con LDpred2-grid

Calculamos la regresión lineal de la puntuación de LD, dicho valor es guardado en un vector, “ldsc”:

```
(ldsc <- with(df_beta, snp_ldsc
              (ld, length(ld), chi2 = (beta / beta_se)^2,
               sample_size = n_eff, blocks = NULL)))
```

A partir de “ldsc” podemos calcular la heredabilidad “h^2”. Obtenemos un valor de 0.1545607, guardado como “ldsc_h2_est”:

```
ldsc_h2_est <- ldsc[["h2"]]
```

Ahora empleamos el modelo LDpred2-grid para la obtención de la puntuación de riesgo poligénico de la enfermedad de las arterias coronarias (CAD). El array generado, “lsdc_h2_est”, se combina con valores de 0,3; 0,7; 1; 1,4, cuyo valor será guardado en como un vector, “h2_seq”.

```
(h2_seq <- round(ldsc_h2_est * c(0.3, 0.7, 1, 1.4), 4))
```

Los valores de “h2_seq” son los siguientes: 0.0464, 0.1082, 0.1546, 0.2164.

Al igual que hemos obtenido los valores de heredabilidad, calculamos los de *p*-valor:

```
(p_seq <- signif(seq_log(1e-5, 1, length.out = 21), 2))
```

“seq_log” aplica una función logarítmica entre un rango de valores, con una extensión de 21 cromosomas. El mínimo establecido es $1e^{-5}$, y el máximo, 1. El resultado es guardado como “p_seq”

El siguiente paso es relacionar ambos parámetros (“h^2” y “*p*”), para ello se hace uso de la función “expand.grid”. Genera todas las posibles combinaciones entre las variables establecidas:

```
params <- expand.grid(p = p_seq,
          h2 = h2_seq, sparse = c(FALSE, TRUE))
dim(params)
```

“dim” indica las dimensiones de nuestro marco de datos guardado como “params”, que consta de 168 datos con 3 columnas (*p*, h^2, true/false).

Acotamos los datos, eliminando aquellos valores no significativos, quedándonos con únicamente 16 de los 168 obtenidos al principio. El valor mínimo, en este caso, es de $1e^{-4}$ y el máximo es 0.5:

```
(params <- expand.grid(p = signif(seq_log(1e-4, 0.5,
                        length.out = 16), 2),
                        h2 = round(ldsc_h2_est, 4), sparse = TRUE))
```

“params” ahora contiene los siguientes datos:

Datos	*p*	h2
1	0.00010	0.1546
2	0.00018	0.1546
3	0.00031	0.1546
4	0.00055	0.1546
5	0.00097	0.1546
6	0.00170	0.1546
7	0.00300	0.1546
8	0.00530	0.1546
9	0.00940	0.1546
10	0.01700	0.1546
11	0.02900	0.1546
12	0.05200	0.1546
13	0.09100	0.1546
14	0.16000	0.1546
15	0.28000	0.1546
16	0.50000	0.1546

Una vez hayamos acotado los datos de “*p*” a un valor de heredabilidad podemos aplicar la función de nuestro modelo, “snp_ldpred2_grid”:

```
beta_grid <- snp_ldpred2_grid(corr, df_beta,
             params, ncores = NCORES)
```

Emplea los datos de correlación (“corr”), el tamaño de la muestra (“df_beta”) y los parámetros calculados de *p* y h^2 (“params”).

```
params$sparsity <- colMeans(beta_grid == 0)
```

Mediante la función “colMeans” se puede realizar la media de las columnas de “beta_grid”.

```
pred_grid <- big_prodMat(G, beta_grid,
                         ind.col = df_beta[["_NUM_ID_"]],
                         ncores = NCORES)
```

“big_prodMat” genera un producto a partir de diferentes archivos. En este caso se implica el genotipo, los datos de “beta_grid” anteriormente calculados y el tamaño de muestra.
Finalmente calculamos las puntuaciones de riesgo poligénico que serán guardadas como “params$score”:

```
params$score <- apply(pred_grid, 2, function(x) {
  if (all(is.na(x))) return(NA)
  summary(glm(
    CAD ~ x + sex + age,
    data = obj.bigsnp$fam, family = "binomial"
  ))$coef["x", 3]
})
```

Los datos son representados mediante la función “ggplot”:

```
ggplot(params, aes(x = p, y = score, color = as.factor(h2))) +
  theme_bigstatsr() +
  geom_point() +
  geom_line() +
  scale_x_log10(breaks = 10^(-5:0), minor_breaks = params$p) +
  facet_wrap(~ sparse, labeller = label_both) +
  labs(y = "GLM Z-Score", color = "h2") +
  theme(legend.position = "top", panel.spacing = unit(1, "lines"))
```

Eje x= *p*	Eje y= Params_score
0.00010	7.403187
0.00018	7.643910
0.00031	7.723712
0.00055	7.958732
0.00097	8.224475
0.00170	8.750395
0.00300	9.123094
0.00530	9.280899
0.00940	9.497044
0.01700	9.527174
0.02900	9.303520
0.05200	9.014294
0.09100	8.478751
0.16000	7.946352
0.28000	7.369360
0.50000	7.958732

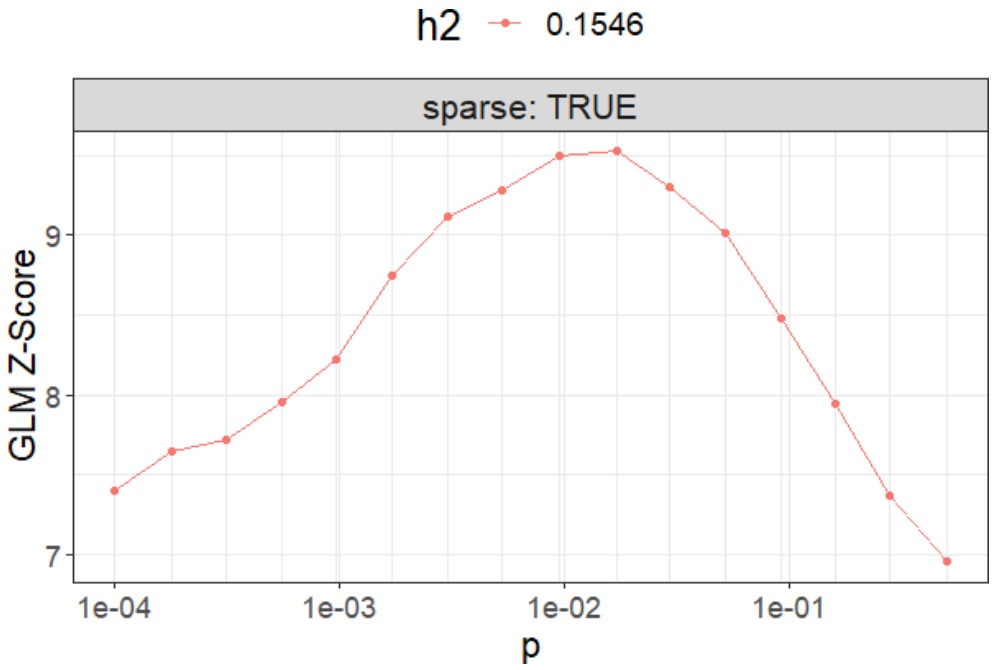


Ilustración 16. Riesgo poligénico de la enfermedad de las arterias coronarias (CAD)

Finalmente obtenemos el valor de riesgo poligénico de la enfermedad de las arterias coronarias (EAC) gracias a la función “big_prodvec”, que realiza el producto de las matrices que se le indica. En este caso de “G”, “best_beta_grid”, “ind.test”, “df_beta[[“_NUM_ID_”]]”. El resultado es guardado cmo un array, llamado “pred”:

```
pred <- big_prodvec(G, best_beta_grid,
                    ind.row = ind.test,
                    ind.col = df_beta[["_NUM_ID_"]])
pcor(pred, y[ind.test], NULL)
```

Por último, “pcor” establece una función de correlación para hallar el valor riesgo poligénico final, siendo este un resultado de 0.68290814.

Esta ponderación refleja la forma en la que diversas variantes genéticas acumuladas contribuyen al desarrollo de la EAC. En concreto, el 68,29% de las personas de este estudio presentan una predisposición genética moderada de padecer dicha enfermedad, considerándose población de riesgo en comparación al resto. Aunque dicho porcentaje no sea determinante de que el paciente pueda desarrollar la enfermedad, se debería de tener en cuenta, puesto que este es superior al 50%, y, sobre todo, habría que hacer hincapié en aquellos factores externos modificables que contribuyen a que la persona reduzca las posibilidades de presentar EAC. Estos factores son por ejemplo el estilo de vida, la dieta o factores ambientales.

4.4. Estimación de la incertidumbre del riesgo poligénico.

Las estimaciones de la incertidumbre suelen estudiarse a nivel de cohorte pero aún hay pocas investigaciones a nivel individual (Ding *et al*., 2022). Debido a esto, junto con la posibilidad de no transmitir adecuadamente la incertidumbre en la estimación, y el no recibir asesoramiento oportuno sobre los enfoques para predecir el riesgo global (no solo asociado a la PRS), conlleva a relevantes estimaciones incorrectas para el individuo, como por ejemplo, casos de un "falso positivo", asignando al individuo como "alto riesgo" según la puntuación de la PRS (Adeyemo *et al*., 2021).

A continuación demostramos algunos métodos de PRS bayesianos capaces de estimar la variación de la PRS de un individuo, produciendo intervalos de confianza precisos para el valor genético de un solo individuo.

El primer paso es la obtención de la distribución posterior completa del valor genético, denotada por GVi=xTib, donde, GVi hace referencia al valor genético. Su expresión en un individuo es el resultado de multiplicar vector genotipo x vector efecto, y, para ello, se necesitan obtener muestras MCMC o muestras del Método de cadenas de Markov Monte Carlo. Es un método orientado a la obtención de la distribución posterior de los parámetros del modelo, así como, la estimación de cantidades de interés (González, Melgar and Vega, 2015)

Para obtener muestras MCMC se realiza lo siguientes:

```
best_param <- data.frame(p = 0.01, h2 = 0.2, sparse = FALSE)
```

"best_param" es un hiperparámetro para obtener los valores de MCMC. Por otro lado, "data.frame", forma, a partir de una serie de variales (p, h2, sparse), un marco de datos que serán imprescindibles para la realización de los pasos posteriores.

```
posterior_beta_samples <- snp_ldpred2_grid(
  corr, df_beta, best_param,
  return_sampling_betas = TRUE, num_iter = 500)
dim(posterior_beta_samples)
```

Aplicamos la función snp_ldpred2_grid, puesto que en nuestro método utilizado ha sido el método LDpred2-grid.
"num_iter" establece el número de muestras que se utilizarán para obtener el valor genético del individuo, en este caso, son 500 muestras.

Al ejecutar la función snp_ldpred2_grid se obtiene un valor de 45.337 MCMC (Privé, 2022a).

Ahora, para obtener el valor genético del individuo se multiplica su genotipo (g) por las muestras posteriores β que hemos generado en el apartado anterior:

```
posterior_gv_samples <- big_prodMat
(G, posterior_beta_samples, ind.col = df_beta[["_NUM_ID_"]])
dim(posterior_gv_samples)
```

Finalmente obtenemos un valor de 503, a partir de 500 muestras. Esto indica que hay poca autocorrelación entre las variables. Para confirmar este hecho calculamos una serie de estimaciones de covarianza y autocorrelación mediante la función acf o “Auto and Cross Covariance and Correlation Function Estimation”.

```
acf(posterior_gv_samples[1, ], lag.max = 10, plot = TRUE)$acf
```

Obtenemos los siguientes resultados:

Eje (x,y)	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)	(5,0)
Puntaje	1	0,0481	0,0091	0,0214	-0.0083	0.0339

Eje (x,y)	(6,0)	(7,0)	(8,0)	(9,0)	(10,0)
Puntaje	0.0846	0.0115	0.0096	-0.0079	0.0960

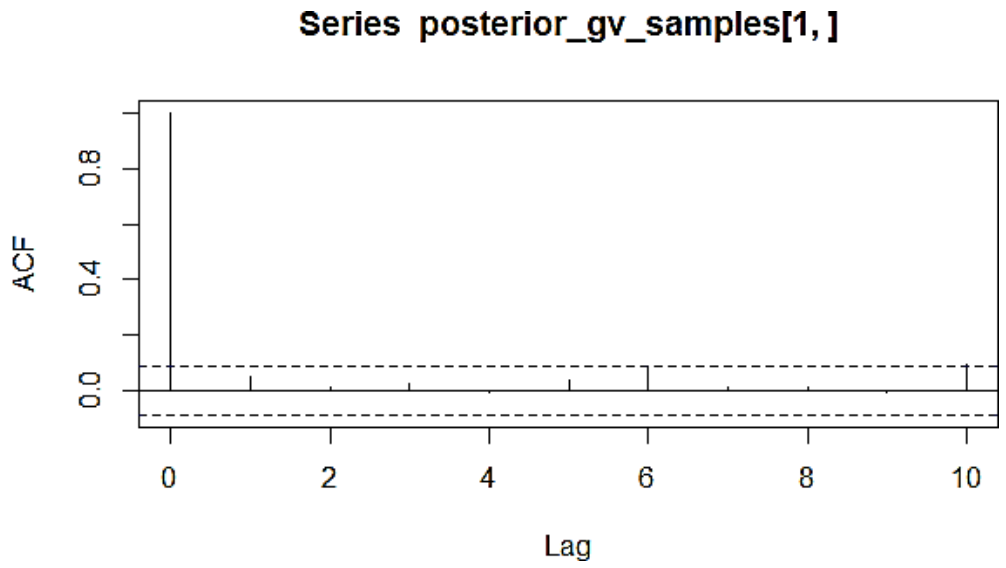


Ilustración 17. Estimación de la función de correlación y covarianza cruzada y automática.

Finalmente realizamos la suma de filas y columnas de los datos obtenidos gracias a la función “rowMeans” a partir de los resultados de “acf” y los ejemplos de la muestra “gv”:

```
rowMeans(apply(posterior_gv_samples, 1, function(x) {
  acf(x, lag.max = 10, plot = FALSE)$acf
}))
```

Obteniendo los datos finales de autocorrelación:

Resultados	1,0000	0.020861623	0.0317	0.0176	0.0024	0.0213
	0.0514	-0.0154	-0.0058	-0.0176	0.0514	0.0671

Una vez realizado este paso, podemos calcular la media posterior, denotada por PRSˆi≡E(GVi|Data), junto con la varianza posterior, var(GVi/Datos). Donde GVi hace referencia al valor genético del individuo i en un GWAS concreto, y, ρ, es el intervalo donde se sitúa el valor genético del individuo con probabilidad ρ. Por tanto, al calcular la media posterior, calculamos la puntuación de riesgo poligénico (Privé, 2022b).

```
samples <- posterior_gv_samples[1, ]
posterior_gv_mean <- mean(samples)
posterior_gv_var <- var(samples)
```

“Mean” es la función de la media aritmética y “va”r de la varianza. Obtenemos unos valores de 0.209086 y 0.3866804, respectivamente, creando unos vectores llamados: “posterior_gv_mean” y “posterior_gv_var”

Para finalizar con el estudio de la incertidumbre del riesgo poligénico, otra opción para poder calcularlo es mediante la construcción de intervalos de valor genético.

```
rho <- 0.95
bound <- (1 - rho) / 2
samples <- posterior_gv_samples[1, ]
mean <- mean(samples)
lower_ci <- quantile(samples, bound)
upper_ci <- quantile(samples, 1 - bound)
hist(samples,
     main = "Posterior distribution of genetic value", xlab = NULL)
abline(v = c(lower_ci, mean, upper_ci),
       col = c("blue", "red", "blue"), lty = c(2,1,2))
legend("topright", legend = c("Credible Interval", "PRS"),
       col = c("blue", "red"), lty = c(2,1), cex = 0.8)
```

El rango de valores que contiene el verdadero valor genético del individuo con probabilidad ρ=0,95, se conoce como “rho”
“bound” es el límite del rango, con valor de 0,0025. La media (mean) obtenida a partir de los valores de las muestras es de 0,2091.
A la hora de establecer los límites de confianza, inferior (-1,07) y superior (1,44), se utilizan “lower_ci” y “upper_ci” respectivamente.

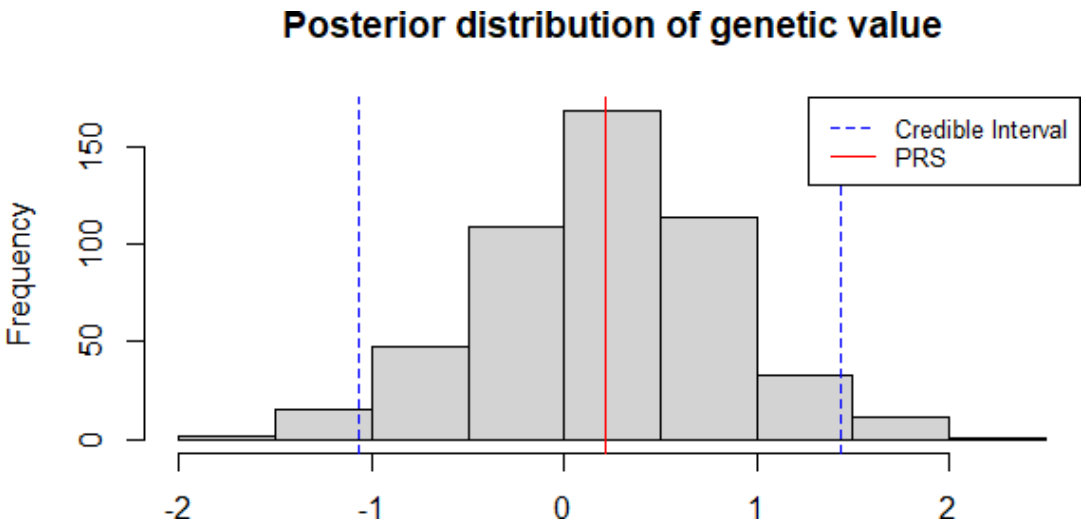


Ilustración 18. Distribución posterior del valor genético con un intervalo de confianza del 95%.

4.5. Cálculo de puntuaciones poligénicas mediante agrupamiento y umbralización apilados (SCT) utilizando el modelo LDpred.

El primer paso para cualquier cálculo, como se ha explicado en el primer apartado de resultados, es descargar los datos de genotipo con los que vamos a trabajar y generar un archivo donde, además de guardarlos, se van a compactar y ordenar.
Una vez que tengamos todos los datos requeridos procedemos al cálculo.

Primero debemos hacer coincidir las variantes entre los datos del genotipo y las estadísticas resumen:

```
names(sumstats) <- c("chr", "rsid", "pos",
                     "a0", "a1", "beta", "p")
map <- obj.bigSNP$map[,-(2:3)]
names(map) <- c("chr", "pos", "a0", "a1")
info_snp <- snp_match(sumstats, map)
```

Mediante la función “c” podemos combinar diferentes argumentos, con el fin de originar un vector. En este caso utiliza los datos existentes en

"chr", "rsid", "pos", "a0", "a1", "beta", "p" que se guardarán en sumstats y "chr", "rsid", "pos", "a0", "a1" que se guardarán en “map”

Significado de las variables	
“chr”	Número de cromosomas
“pos”	Posición genética
“a0”	Alelo de referencia
“a1”	Alelo derivado

La tarea de función “snp_match” es hacer coincidir los alelos de las estadísticas resumen y la información de SNP. Esta coincidencia se realiza mediante el emparejamiento de los valores de sumstats ("pos" o "rsid") y map ("chr", "a0", "a1"), teniendo en cuenta la posibilidad de cambios de cadena y alelos de referencia inversos que peden tener efectos opuestos (Privé, 2022a). El resultado se almacena en “info_sn”, mostrándose de la siguiente manera:

chr	pos	a0	a1	rsid	beta	p	_NUM_ID_.ss	_NUM_ID_
2	16506	C	T	rs13400442	-0.072950062	0.792531845	1	1
2	22398	T	C	rs7597758	-0.332453022	0.084632598	3	3
2	28228	A	G	rs13383216	-0.544474054	0.028018845	4	4
2	32003	GTA	G	rs148885999	-0.468086130	0.043919800	5	5
2	32005	A	G	rs73138586	-0.050955787	0.805599345	6	6

Ilustración 19. Tabla info_snp: presenta 111.860 variantes emparejadas 18.932 SNP ambiguos.

4.5.1. Aglomeración (cumpling)

Realizamos las siguientes operaciones:

```
all_keep <- snp_grid_clumping(G, CHR,
            POS, ind.row = ind.train,
            lps = lpval, exclude = which(is.na(lpval)),
            ncores = NCORES)
attr(all_keep, "grid")
```

“snp_grid_clumping” obtiene un conjunto de variables mediante el apilamiento o agrupamiento repetido de diversos valores de hiperparámetros. En este caso se utilizan 28 valores de los cuales 7 son umbrales de correlación y 4 “window sizes”, generando así un conjunto de variantes. Por otro lado, “attr” asocia atributos concretos de un objeto.

Gracias a estas funciones generamos la siguiente tabla:

```
    size thr.r2 grp.num thr.imp
1   5000   0.01       1       1
2  10000   0.01       1       1
3  20000   0.01       1       1
4  50000   0.01       1       1
5   1000   0.05       1       1
6   2000   0.05       1       1
7   4000   0.05       1       1
8  10000   0.05       1       1
9    500   0.10       1       1
10  1000   0.10       1       1
11  2000   0.10       1       1
12  5000   0.10       1       1
13   250   0.20       1       1
14   500   0.20       1       1
15  1000   0.20       1       1
16  2500   0.20       1       1
17   100   0.50       1       1
```

Ilustración 20. “thr.r2” indica el umbral de correlacion entre dos SNPs y “thr.imp” es un vector de umbrales (por defecto es 1)

4.5.2. Umbralización (Thresholding)

```
multi_PRS <- snp_grid_PRS(G, all_keep,
             beta, lpval, ind.row = ind.train,
             backingfile = "public-data-scores",
             n_thr_lps = 50, ncores = NCORES)
dim(multi_PRS)
```

Por un lado, la función “snp_grid_PRS” crea una matriz en disco FBM que almacena los valores C+T para cada parámetro y para cada cromosoma de manera individual. Por otro lado, “dim”, fija la dimensión de un objeto, en este caso, de “multi_PRS” (Privé, 2022a)

Tras la ejecución de estas funciones obtenemos un resultado de 4200 puntajes de C+T (Clumping+Thresholding) para 400 individuos.

4.5.3. Apilamiento de las predicciones C+T

A continuación se realiza el agrupamiento de las predicciones realizadas por el método “Clumping+Thresholding”.

```
final_mod <- snp_grid_stacking(multi_PRS,
             y[ind.train], ncores = NCORES, K = 4)
summary(final_mod$mod)
```

“snp_grid_stacking” es similar a la función “snp_grid_PRS”, explidada en el primer apartado de los resultados. “snp_grid_stacking” se encarga del

apilamiento de muchos valores de riesgo poligénico obtenidos a través de puntajes “C+T” (Privé, 2022a).

El resumen del paquete generado, “final_mod”, es el siguiente:

Alpha	Power_adaptive	Power_scale	Validation_loss	Intercept beta	nb_var
0.0001	0	1	0.569	-0.975	3492
0.01	0	1	0.571	-0.942	687
1	0	1	0.575	-0.899	151

Podemos visualizar mediante una gráfica el efecto de la asociación del genoma completo (GWAS) como resultado del apilamiento.

```
new_beta <- final_mod$beta.G
ind <- which(new_beta != 0)
```

Se crea un nuevo vector, “new_beta”, con el que se trabajará para el cálculo de los siguientes apartados.

```
library(ggplot2)
ggplot(data.frame(y = new_beta, x = beta)[ind, ]) +
  geom_abline(slope = 1, intercept = 0, color = "red") +
  geom_abline(slope = 0, intercept = 0, color = "blue") +
  geom_point(aes(x, y), size = 0.6) +
  theme_bigstatsr() +
  labs(x = "Effect sizes from GWAS",
       y = "Non-zero effect sizes from SCT")
```

Las funciones geom- incluyen líneas de referencia en una cuadricula con orientación horizontal, vertical o diagonal definida por su pendiente e intersección. En este caso se hace uso de “geom_abline” para trazar una línea horizontal azul en el punto 0 (pendiente 0), y otra línea diagonal, roja, cuya pendiente es 1 y el punto de intersección de ambas es 0. Por otro lado, se utiliza también “geom_point” para generar diagramas de dispersión mostrando una relación entre dos variables (Privé, 2022a).

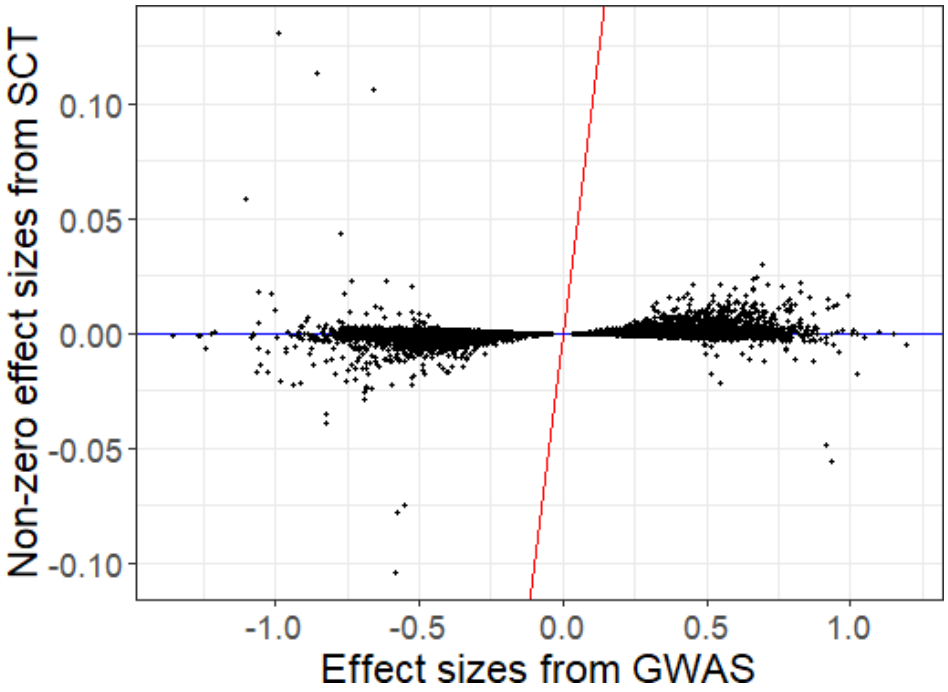


Ilustración 21. Efecto resultante del apilamiento en función del tamaño de efecto GWAS y tamaños de efecto distintos de 0 de SCT.

A partir del vector creado "new_beta" podemos calcular el puntuaje de riesgo, por un lado, como vemos a continuación:

```
pred <- final_mod$intercept +
  big_prodvec(G, new_beta[ind],
              ind.row = ind.test, ind.col = ind)
AUCBoot(pred, y[ind.test])
```

"AUCBoot" calcula la predicción del área bajo la curva ROC, obteniendo el valor de riesgo poligénico, hallado en este caso mediante el método de agrupamiento y umbralización.

El resultado es el siguiente:

Media	2.5%	97.5%	Sd
0.68965743	0.58961496	0.78379429	0.04921019

Ilustración 22. 2.5% y 97.5% hace referencia a los cuantiles y Sd es la desviación típica.

El puntaje de riesgo poligénico es de 0.68965743, es decir, un 68,96% de la población presenta una predisposición genética hacia el desarrollo de la enfermedad, haciéndoles más susceptibles. Como se ha explicado anteriormente en el apartado 4.3.3, este porcentaje no asegura que la persona vaya a presentar la enfermedad pero, al ser población de riesgo se deberían atender aquellos factores modificables para el individuo que se encuentran al alcance de la población y que contribuyen a una reducción de este valor.

Por otro lado, podemos valorar la gráfica del Área bajo la curva (AUC):

```
ggplot(data.frame(
  Phenotype = factor(y[ind.test], levels = 0:1,
                     labels = c("Control", "Case")),
  Probability = 1 / (1 + exp(-pred)))) +
  theme_bigstatsr() +
  geom_density(aes(Probability,
                   fill = Phenotype), alpha = 0.3)
```

“theme_bigstatsr” se utiliza para añadir el color adecuado a la gráfica en función de los datos obtenidos. “geom_density” crea un gráfico de densidad mediante ggplot2 (Privé, 2022a).

En la siguiente gráfica podemos observar el área de distribución de la población control y los individuos problema. Así podríamos comprobar visualmente la gravedad o rango de afección del factor que se pretende estudiar:

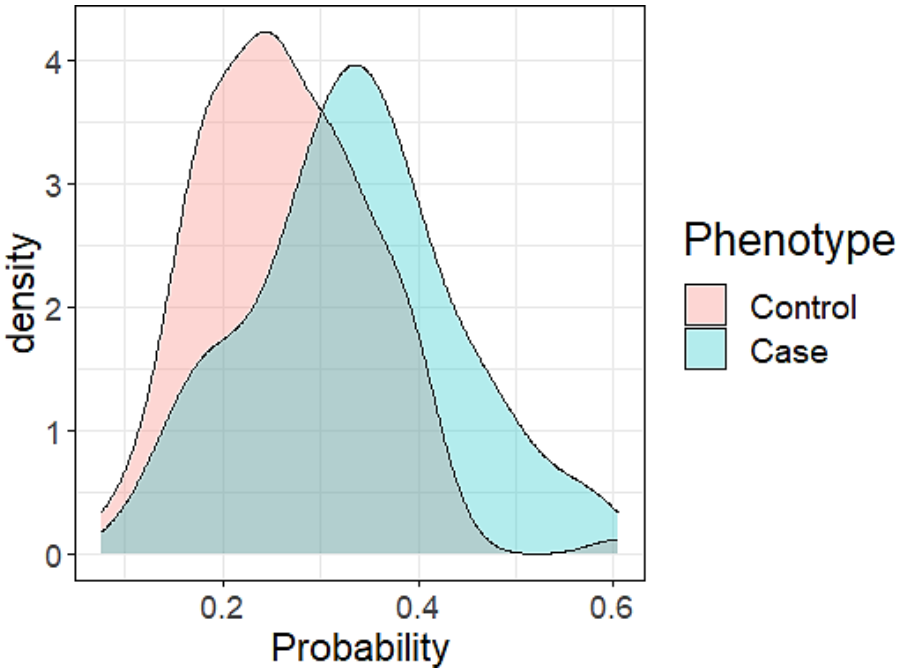


Ilustración 23. Gráfica AUC exponiendo los datos control y datos problema del estudio.

5. DISCUSIÓN Y CONCLUSIÓN

Las puntuaciones de riesgo poligénico son actualmente uno de los principales objetos de estudio como posible vía para la predicción de enfermedades. Se han desarrollado diferentes herramientas que facilitan su cálculo, aunque todavía falta avanzar en el desarrollo de este campo pues aún hay diversas limitaciones, aún no resueltas, para hallar un perfil de riesgo potencial.

En este trabajo se ha estudiado qué es "Rstudio" y cómo se utiliza, con el fin de conocer herramientas para la PRS. "Rstudio" es un programa que permite e análisis de datos de grandes dimensiones utilizando un lenguaje especifico de programación. Cuenta con multitud de paquetes, cada uno con una función y aplicación concreta (R Core Team, 2013). Para este estudio se ha utilizado el paquete "bigsnpr", el cual, se han de manifiesto algunas de sus funciones útiles para nuestro trabajo, aunque para el propio cálculo de riesgo poligénico, se ha empleado el modelo LDpred2-grid, como futura herramienta de conocimiento para la predicción de enfermedades.

Un aspecto de gran importancia para un buen rendimiento predictivo del modelo PRS es la realización de controles de calidad, por medio de la eliminación de regiones donde se da el desequilibrio del ligamento (LD), tanto antes de iniciar el análisis, como durante su realización, de esta manera eliminas los sesgos de confusión y la poligenidad. Esto se consigue a través de la regresión logística penalizada (PRL)

A pesar de haber realizado unos buenos controles de calidad, a la hora del cálculo de PRS se ha observado una gran correlación entre las variables. Esto puede ser debido a las limitaciones que presenta el cálculo de riesgo poligénico, como por ejemplo, el haber empleado un pequeño tamaño poblacional. Podría solucionarse aumentando los datos del conjunto de GWAS, incorporando información adicional biológica, como el efecto de un gen sobre el fenotipo del individuo. (Song, *et al.* 2019).

Otra de las limitaciones que encontramos en este estudio, es el haber empleado solamente datos de poblaciones europeas, limitando su predicción. Este aspecto supone un reto para los futuros estudios, pudiendo afianzarse a poblaciones africanas occidentales o asiáticas orientales (Bulik-Sullivan *et al*., 2015)

En conclusión, se ha desarrollado la aplicación de una herramienta para el cálculo de riesgo poligénico, LDpred2-grid, utilizando diferentes métodos, como la "umbralización y aglomeración" o el cálculo del área bajo la curva (AUC). Además de haber afianzado nuevos conocimientos y puesto en práctica los implementados. En este caso, hemos podido comprobar su utilidad poniendo como ejemplo una enfermedad compleja, como lo es, la enfermedad de las arterias coronarias (EAC). Enfermedad donde se produce el estrechamiento de las arterias coronarias induciendo numerosos y diversos problemas de salud,

desde una angina de pecho hasta infartos de miocardio (Khera and Kathiresan, 2017). Gracias a LDpred2 se puede evaluar el riesgo poligénico de la enfermedad, además de seleccionar aquellas personas que presentan mayor riesgo de desarrollarla. Permite una identificación temprana, lo que posibilita al individuo tomar medidas preventivas y modificar sus hábitos de vida para disminuir el riesgo de desarrollar la enfermedad.

Por tanto, el conocimiento y abordaje de este trabajo de fin de grado ha tenido como finalidad conocer y aplicar una herramienta computacional de predicción de riesgo poligénico en una enfermedad compleja y, además, afianzar los conocimientos adquiridos en diferentes asignaturas del grado para una mejor comprensión de los procesos biológicos, a nivel molecular y su utilidad clínica.

6. BIBLIOGRAFÍA

1. 1000 Genomes Project Consortium, Auton, A., Brooks, L. D., Durbin, R. M., Garrison, E. P., Kang, H. M., Korbel, J. O., Marchini, J. L., McCarthy, S., McVean, G. A., & Abecasis, G. R. (2015). A global reference for human genetic variation. Nature, 526(7571), 68-74. https://doi.org/10.1038/nature15393
2. Adeyemo, A., Balaconis, M. K., Darnes, D. R., Fatumo, S., Moreno, P. G., Hodonsky, C. J., Inouye, M., Kanai, M., Kato, K., Knoppers, B. M., Lewis, A. C. F., Martin, A. R., McCarthy, M. I., Meyer, M. L., Okada, Y., Richards, J. B., Richter, L., Ripatti, S., Rotimi, C. N., Zhou, A. Y. (2021). Responsible use of polygenic risk scores in the clinic: potential benefits, risks and gaps. Nature Medicine, 27(11), 1876-1884. https://doi.org/10.1038/s41591-021-01549-6
3. Al Mehdi, K., Fouad, B., Zouhair, E., Boutaina, B., Yassine, N., Chaimaa, A. E. C., Najat, S., Hassan, R., Rachida, R., Abdelhamid, B., & Halima, N. (2019). Molecular Modelling and Dynamics Study of nsSNP in STXBP1 Gene in Early Infantile Epileptic Encephalopathy Disease. BioMed research international, 2019, 4872101. https://doi.org/10.1155/2019/4872101
4. Anderson, C. A., Pettersson, F. H., Clarke, G. M., Cardon, L. R., Morris, A. P., & Zondervan, K. T. (2010). Data quality control in genetic case-control association studies. Nature protocols, 5(9), 1564-1573. https://doi.org/10.1038/nprot.2010.116
5. Aulchenko, Y. S., Ripke, S., Isaacs, A., & van Duijn, C. M. (2007). GenABEL: an R library for genome-wide association analysis. Bioinformatics (Oxford, England), 23(10), 1294-1296. https://doi.org/10.1093/bioinformatics/btm108
6. Balagué-Dobón, L., Cáceres, A., & González, J. R. (2022). Fully exploiting SNP arrays: a systematic review on the tools to extract underlying genomic structure. *Briefings in bioinformatics*, *23*(2), bbac043. https://doi.org/10.1093/bib/bbac043
7. Bryan, J., & Maintainer, J. (2022). Package “glue”. R-project.org. https://cran.r-project.org/web/packages/glue/glue.pdf
8. Bulik-Sullivan, B. K., Loh, P. R., Finucane, H. K., Ripke, S., Yang, J., Schizophrenia Working Group of the Psychiatric Genomics Consortium, Patterson, N., Daly, M. J., Price, A. L., & Neale, B. M. (2015). LD Score regression distinguishes confounding from polygenicity in genome-wide association studies. Nature genetics, 47(3), 291-295. https://doi.org/10.1038/ng.3211
9. Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L. T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., Cortes, A., Welsh, S., Young, A., Effingham, M., McVean, G., Leslie, S., Allen, N., Donnelly, P., & Marchini, J. (2018). The UK Biobank resource with deep

phenotyping and genomic data. Nature, 562(7726), 203-209. https://doi.org/10.1038/s41586-018-0579-z
10. Chen, S. F., Dias, R., Evans, D., Salfati, E. L., Liu, S., Wineinger, N. E., & Torkamani, A. (2020). Genotype imputation and variability in polygenic risk score estimation. Genome medicine, 12(1), 100. https://doi.org/10.1186/s13073-020-00801-x
11. Choi, S. W., Mak, T. S., & O'Reilly, P. F. (2020). Tutorial: a guide to performing polygenic risk score analyses. Nature protocols, 15(9), 2759-2772. https://doi.org/10.1038/s41596-020-0353-1
12. Devaney, J. M., Knouff, C. W., Thompson, J. R., Horne, B. D., Stewart, A. F., Assimes, T. L., Wild, P. S., Allayee, H., Nitschke, P. L., Patel, R. S., Myocardial Infarction Genetics Consortium, Wellcome Trust Case Control Consortium, Martinelli, N., ... Rader, D. J. (2011). Identification of ADAMTS7 as a novel locus for coronary atherosclerosis and association of ABO with myocardial infarction in the presence of coronary atherosclerosis: two genome-wide association studies. Lancet (London, England), 377(9763), 383-392. https://doi.org/10.1016/S0140-6736(10)61996-4
13. Ding, Y., Hou, K., Burch, K. S., Lapinska, S., Privé, F., Vilhjálmsson, B., Sankararaman, S., & Pasaniuc, B. (2022). Large uncertainty in individual polygenic risk score estimation impacts PRS-based risk stratification. Nature genetics, 54(1), 30-39. https://doi.org/10.1038/s41588-021-00961-5
14. Fischer, R. (1919). XV.—La correlación entre parientes sobre el supuesto de herencia mendeliana. Transacciones de Ciencias Ambientales y de la Tierra de la Sociedad Real de Edimburgo, 52 (2), 399-433. doi:10.1017/S0080456800012163
15. Ge, T., Chen, C. Y., Ni, Y., Feng, Y. A., & Smoller, J. W. (2019). Polygenic prediction via Bayesian regression and continuous shrinkage priors. Nature communications, 10(1), 1776. https://doi.org/10.1038/s41467-019-09718-5
16. Gel, B., & Serra, E. (2017). karyoploteR: an R/Bioconductor package to plot customizable genomes displaying arbitrary data. Bioinformatics (Oxford, England), 33(19), 3088-3090. https://doi.org/10.1093/bioinformatics/btx346
17. Gómez-Herazo, C. J. (2018). Un estudio de diferentes pruebas para el problema general de dos muestras [Thesis]. Retrieved from https://hdl.handle.net/20 500.11801/2112
18. González, L. D., Melgar, D. C., & Vega, V. S. (2015). Enfoque bayesiano del modelo de regresión logística usando cadenas de Markov Monte Carlo. Investigación Operacional, 36(2), 178-186.
19. Grolemund, H. W. A. G. (2023). 3 Data visualisation | R for Data Science. https://r4ds.had.co.nz/data-visualisation.html
20. Jiménez, JU (2019). Introducción a R y RStudio.

21. Khera, A. V., & Kathiresan, S. (2017). Genetics of coronary artery disease: discovery, biology and clinical translation. Nature reviews. Genetics, 18(6), 331-344. https://doi.org/10.1038/nrg.2016.160
22. Khera, A. V., Chaffin, M., Aragam, K. G., Haas, M. E., Roselli, C., Choi, S. H., Natarajan, P., Lander, E. S., Lubitz, S. A., Ellinor, P. T., & Kathiresan, S. (2018). Genome-wide polygenic scores for common diseases identify individuals with risk equivalent to monogenic mutations. Nature genetics, 50(9), 1219-1224. https://doi.org/10.1038/s41588-018-0183-z
23. Khramtsova, E. A., Davis, L. K., & Stranger, B. E. (2019). The role of sex in the genomics of human complex traits. Nature reviews. Genetics, 20(3), 173-190. https://doi.org/10.1038/s41576-018-0083-1
24. Lambert, S. A., Abraham, G., & Inouye, M. (2019). Towards clinical utility of polygenic risk scores. Human molecular genetics, 28(R2), R133-R142. https://doi.org/10.1093/hmg/ddz187
25. Lewis, C. M., & Vassos, E. (2020). Polygenic risk scores: from research tools to clinical instruments. Genome medicine, 12(1), 44. https://doi.org/10.1186/s13073-020-00742-5
26. Lloyd-Jones, L. R., Zeng, J., Sidorenko, J., Yengo, L., Moser, G., Kemper, K. E., Wang, H., Zheng, Z., Magi, R., Esko, T., Metspalu, A., Wray, N. R., Goddard, M. E., Yang, J., & Visscher, P. M. (2019). Improved polygenic prediction by Bayesian multiple regression on summary statistics. Nature communications, 10(1), 5086. https://doi.org/10.1038/s41467-019-12653-0
27. Mak, T. S. H., Porsch, R. M., Choi, S. W., Zhou, X., & Sham, P. C. (2017). Polygenic scores via penalized regression on summary statistics. Genetic epidemiology, 41(6), 469-480. https://doi.org/10.1002/gepi.22050
28. Marchini, J., Howie, B., Myers, S., McVean, G., & Donnelly, P. (2007). A new multipoint method for genome-wide association studies by imputation of genotypes. Nature genetics, 39(7), 906-913. https://doi.org/10.1038/ng2088
29. Marees, A. T., de Kluiver, H., Stringer, S., Vorspan, F., Curis, E., Marie-Claire, C., & Derks, E. M. (2018). A tutorial on conducting genome-wide association studies: Quality control and statistical analysis. International journal of methods in psychiatric research, 27(2), e1608. https://doi.org/10.1002/mpr.1608
30. Mullins, N., Power, R. A., Fisher, H. L., Hanscombe, K. B., Euesden, J., Iniesta, R., Levinson, D. F., Weissman, M. M., Potash, J. B., Shi, J., Uher, R., Cohen-Woods, S., Rivera, M., Jones, L., Jones, I., Craddock, N., Owen, M. J., Korszun, A., Craig, I. W., Farmer, A. E., ... Lewis, C. M. (2016). Polygenic interactions with environmental adversity in the aetiology of major depressive disorder. Psychological medicine, 46(4), 759-770. https://doi.org/10.1017/S0033291715002172

31. Nguyen, D. T., Tran, T. T. H., Tran, M. H., Tran, K., Pham, D., Duong, N. T., Nguyen, Q., & Vo, N. S. (2022). A comprehensive evaluation of polygenic score and genotype imputation performances of human SNP arrays in diverse populations. *Scientific reports*, *12*(1), 17556. https://doi.org/10.1038/s41598-022-22215-y
32. Ott, J., Kamatani, Y., & Lathrop, M. (2011). Family-based designs for genome-wide association studies. Nature reviews. Genetics, 12(7), 465-474. https://doi.org/10.1038/nrg2989
33. Park, S. Y., & Liu, Y. (2011). Robust penalized logistic regression with truncated loss functions. The Canadian journal of statistics = Revue canadienne de statistique, 39(2), 300-323. https://doi.org/10.1002/cjs.10105
34. Privé F., Blum M., Aschard H. Statistical Tools for Filebacked Big Matrices. (2019). https://privefl.github.io/bigstatsr/
35. Privé, F. (2022a). Polygenic scores and inference using LDpred2. Github.Io. https://privefl.github.io/bigsnpr/articles/LDpred2.html
36. Privé, F. (2022b). bigsnpr & bigstatsr. Github.io. https://privefl.github.io/bigsnpr-extdoc/index.html
37. Privé, F., Arbel, J., & Vilhjálmsson, B. J. (2021). LDpred2: better, faster, stronger. Bioinformatics (Oxford, England), 36(22-23), 5424-5431. https://doi.org/10.1093/bioinformatics/btaa1029
38. Privé, F., Aschard, H., Ziyatdinov, A., & Blum, M. G. B. (2018). Efficient analysis of large-scale genome-wide data with two R packages: bigstatsr and bigsnpr. Bioinformatics (Oxford, England), 34(16), 2781-2787. https://doi.org/10.1093/bioinformatics/bty185
39. Privefl. (2023). GitHub - privefl/bigsnpr: R package for the analysis of massive SNP arrays. GitHub. https://github.com/privefl/bigsnpr
40. R Core Team, R. (2013). R: A language and environment for statistical computing.
41. Reed, E., Nunez, S., Kulp, D., Qian, J., Reilly, M. P., & Foulkes, A. S. (2015). A guide to genome-wide association analysis and post-analytic interrogation. Statistics in medicine, 34(28), 3769-3792. https://doi.org/10.1002/sim.6605
42. Song, L., Liu, A., Shi, J., & Molecular Genetics of Schizophrenia Consortium (2019). SummaryAUC: a tool for evaluating the performance of polygenic risk prediction models in validation datasets with only summary level statistics. Bioinformatics (Oxford, England), 35(20), 4038-4044. https://doi.org/10.1093/bioinformatics/btz176
43. Takahashi, Y., Mimori, K., & Mori, M. (2012). Nihon Geka Gakkai zasshi, 113(2), 210-214.
44. Vilhjálmsson, B. J., Yang, J., Finucane, H. K., Gusev, A., Lindström, S., Ripke, S., Genovese, G., Loh, P. R., Bhatia, G., Do, R., Hayeck, T., Won, H. H., Schizophrenia Working Group of the Psychiatric Genomics Consortium, Discovery, Biology, and Risk of Inherited Variants in Breast

Cancer (DRIVE) study, Kathiresan, S., Pato, M., Pato, C., Tamimi, R., Stahl, E., Zaitlen, N., Pasaniuc, B., ... Price, A. L. (2015). Modeling Linkage Disequilibrium Increases Accuracy of Polygenic Risk Scores. American journal of human genetics, 97(4), 576–592. https://doi.org/10.1016/j.ajhg.2015.09.001

45. Weeks, JP. (2022). Paquete "plink". R-proyecto.org. https://cran.r-project.org/web/packages/plink/plink.pdf