



Integración Proteo-genómica para la detección de biomarcadores inflamatorios y sus reguladores genéticos en la Enfermedad Inflamatoria Intestinal mediante Olink proteomics

Francisco José Salamanca-Rivera

Facultad de Ciencias, Departamento de Biología, Universidad Nacional de Colombia, Bogotá, Colombia

Institute of Clinical Molecular Biology, Kiel University, University Hospital Schleswig-Holstein, Kiel, Germany

Resumen

Introducción: La enfermedad inflamatoria intestinal (EII) es una enfermedad crónica y heterogénea, lo que dificulta el pronóstico y las opciones de tratamiento. Aunque se ha identificado más de 200 loci genómicos asociados con la susceptibilidad, no se ha estudiado exhaustivamente la variabilidad en la expresión de las proteínas circulantes.

Objetivo: Caracterizar los perfiles proteómicos de pacientes con EII utilizando Olink Proteomics® e integrar estos datos con información genómica.

Métodos: En un estudio de casos y controles, se analizaron 73 proteínas plasmáticas relacionadas con la inflamación en 154 pacientes con EII y 369 individuos sanos. La combinación de proteómica y mapeo genético permitirá identificar pQTL y comprender mejor la relación entre la genética y la expresión de proteínas en la EII.

Resultados: Se identificaron 21 proteínas asociadas con la inflamación crónica en la enfermedad inflamatoria intestinal (EII), entre ellas TGF- α , MMP-10, CXCL9 e IL-17A, replicadas en estudios previos. En la enfermedad de Crohn (EC), FGF-19, FGF-23, IFN- γ y CCL11 mostraron alta consistencia entre análisis, mientras que en colitis ulcerativa (CU), MMP-10, CXCL11 e IL-17A fueron las más relevantes. Además, se identificaron cuatro pQTLs trans en *RORA*, *STARD5*, *IGSF5* y *RHOBTB3*, regulando indirectamente la expresión de proteínas clave.

Conclusión: Los hallazgos resaltan la complejidad de la regulación genética en la EII y subrayan la importancia de integrar datos proteogenómicos. La identificación de pQTLs trans abre nuevas oportunidades para intervenciones terapéuticas personalizadas.

Palabras clave: Enfermedad Inflamatoria Intestinal; Enfermedad de Crohn; Colitis Ulcerativa; pQTLs; Loci; Locus; Desequilibrio de Ligamiento; SNP.

1. Introducción

La enfermedad inflamatoria intestinal (EII, IBD por sus siglas en inglés), que incluye la enfermedad de Crohn (EC, CD por sus siglas en inglés) y la Colitis ulcerativa (CU, UC por sus siglas en inglés), es un trastorno crónico y recurrente caracterizado por una prevalencia global creciente y una marcada heterogeneidad clínica. La diversidad de manifestaciones de la enfermedad, acompañada por la progresión y la respuesta terapéutica

complica el pronóstico, la selección del tratamiento y el monitoreo a largo plazo, resaltando la necesidad urgente de personalizar los acercamientos en el manejo de los pacientes (Colombel et al., 2017).

Si bien los avances en la investigación genómica han identificado más de 200 loci de riesgo asociado con la susceptibilidad a la EII, estos hallazgos explican solo una fracción de la variabilidad de la enfermedad, lo que deja brechas críticas en la comprensión de los mecanismos funcionales y los objetivos terapéuticos elaborables (de Lange et al., 2017; Liu et al., 2015).

Cada vez hay más pruebas de que las proteínas circulantes son mediadores dinámicos de las vías de la enfermedad, lo que ofrece una perspectiva de la fisiopatología de la EII que la genómica por sí sola no puede captar. A diferencia de las variantes genéticas, las proteínas reflejan procesos biológicos en tiempo real, como interacciones con desencadenantes ambientales, desregulación inmunitaria y daño tisular. A pesar de los amplios perfiles proteómicos en poblaciones sanas, los análisis sistemáticos en la EII siguen siendo escasos, lo que limita la traducción de los descubrimientos proteómicos en herramientas clínicas para el diagnóstico, la monitorización de la enfermedad o la terapia dirigida (Sun et al., 2018; Zhernakova et al., 2018). Además, la integración de datos proteómicos con variantes de riesgo genético, llamados enfoques proteogenómicos, es prometedora para dilucidar los mecanismos subyacentes a los loci conocidos, identificar nuevos biomarcadores y acelerar el desarrollo de fármacos (Bourgonje et al., 2019; Folkersen et al., 2020).

Este estudio aborda esas lagunas a través de un doble enfoque, la definición de proteómica plasmática de la EII utilizando ensayos de extensión de proximidad de alto rendimiento (PEA) con Olink Proteomics® y la identificación de determinantes genéticos de los niveles de expresión de proteínas (loci de rasgo cuantitativo de proteínas, pQTLs) dentro de una cohorte de la EII.

Empleando un diseño de casos y controles (154 pacientes con EII frente a 369 controles sanos emparejados), analizamos 73 proteínas relacionadas con el panel inflamatorio predefinido de Olink, junto con la genotipificación de todo el genoma a través del Global Screening Array (GSA; ~700.000) de Illumina. Para mitigar los factores de confusión, el modelo incorpora componentes genéticos, de sexo y cohortes.

Teniendo en cuenta lo expuesto, este trabajo pretende desentrañar los subtipos moleculares de EII, priorizar dianas terapéuticas y avanzar en estrategias de medicina de precisión. Por último, este marco proteogenómico pretende transformar la gestión de la EII vinculando la predisposición genética a biomarcadores proteicos que sean procesables, ofreciendo así una hoja de ruta para la estratificación de los pacientes y las intervenciones personalizadas.

2. Materiales y métodos

2.1. Método de muestreo:

Se reclutaron participantes provenientes de dos cohortes distintas. La cohorte EMGE compuesta por 193 controles sanos y la cohorte KINDRED con un total de 230 individuos, de los cuales 154 presentaban la enfermedad inflamatoria intestinal (EII), subdividiéndose en 76 casos de Colitis Ulcerativa (CU), 78 de enfermedad de Crohn (EC), junto con 76 controles sanos. En total, se conformó una muestra de 423 individuos, todos ellos provenientes del Hospital Universitario de Schleswig-Holstein ubicado en Kiel, Alemania. Las muestras fueron recolectadas mediante extracción sanguínea, siendo el suero el tipo de muestra utilizado para el análisis. El reclutamiento de los sujetos del estudio fue aprobado por el comité ético y las juntas de revisión institucional de todos los participantes individuales. Se obtuvo el consentimiento informado por escrito de todos los participantes en el estudio.

El estudio empleó un muestreo por conveniencia. El reclutamiento de participantes comenzó en octubre del 2013, y continúa en curso. Hasta abril de 2021, la cohorte incluyó 1497 pacientes diagnosticados con EII, junto con 1813 familiares de primer y segundo grado que inicialmente no presentaban sintomatología asociada, todos residentes de Alemania. Se estableció un criterio de inclusión con edad mínima de 7 años. Los participantes proporcionaron datos mediante cuestionarios estandarizados y muestras biológicas (sangre, heces y cabello). Se implementó un protocolo de seguimiento bienal, durante el cual se recolectaron sistemáticamente nuevos datos clínicos y biomateriales para futuros análisis longitudinales (Rausch et al., 2024).

2.2. Análisis de datos Proteómicos:

El análisis proteómico se realizó a partir del suero sanguíneo de 440 individuos, utilizando el panel Olink Inflammation, el cual incluye la detección de 92 proteínas estrechamente relacionadas con procesos inflamatorios. El análisis fue llevado a cabo por el equipo de Olink Proteomics en el laboratorio de servicio analítico en Uppsala, Suecia. (Olink Target 96 — Olink®, s. f.). Los NPX calculados por el software de Olink® fueron los usados para los análisis posteriores.

2.3. Análisis de datos Genómicos:

El ADN genómico fue extraído a partir de muestras de sangre y procesado utilizando el Illumina Infinium Global Screening Array (GSA) MD 24v1.0 (Illumina, 2017). El proceso de extracción, amplificación y genotipado se realizó siguiendo la metodología descrita por Rausch et al. (2024).

Los datos generados fueron alineados a la referencia GRCh37 (hg19) y convertidos a formato .ped/.map mediante la plataforma GenomeStudio (Illumina).

Para garantizar la calidad de los datos fenotípicos del estudio, se aplicaron filtros de control de calidad, siguiendo el pipeline para controles de calidad en estudios de asociación genómica BIGWAS (Kässens et al., 2021) (<https://github.com/ikmb/gwas-qc/>).

2.3.1. Imputación:

Para incrementar el número de SNPs a evaluar, se realizó un proceso de imputación, el cual infiere variantes genéticas que no fueron directamente genotipadas, a través de la comparación con datos o un panel de referencia. La plataforma Trans-Omics for Precision Medicine (TOPMed) del NHLBI, fue usada para imputar, utilizando los datos genotipados post-control de calidad. El panel de referencia usado fue el ofrecido por defecto el cual proviene de más de 150.000 genomas humanos de diversas poblaciones (Taliun et al., 2021).

2.4. Análisis exploratorio:

A través de un análisis descriptivo y gráfico de la distribución de los datos de expresión proteica se evaluó la forma de la distribución para cada cohorte y cada enfermedad (Anexo1, Gráfica 1A y 2A)

Los valores atípicos fueron identificados como aquellos que caen por debajo del percentil 3 y por encima del percentil 97 de la distribución de los datos, lo que permite una detección robusta sin asumir una distribución puntual.

```
#Análisis Exploratorio

#Carga de paquetes
packages <- function(requirements,quiet=FALSE){
  has <- requirements %in% rownames(installed.packages())
  if(any(!has)){
    message("Installing packages...")
    setRepositories(ind=c(1:7))
    r <- getOption("repos")
    r["CRAN"] <- "https://cran.uni-muenster.de/"
    #options(install.packages.check.source = "no")
    install.packages(requirements[!has],repos=r)
  }
  if(quiet){
    for(r in requirements)
      {suppressMessages(require(r,character.only=TRUE))} }
    else for(r in requirements)
      {message(paste(r,suppressMessages
                     (require(r,character.only=TRUE)),sep=': '))}
  }
}

packages(c("tidyverse", "data.table", "ggrepel", "dplyr", "openxlsx", "fs", "ggplot2",
  "snpStats", "qqman", "CMplot", "purrr", "MASS", "writexl", "viridis",
  "car", "ResourceSelection", "kableExtra", "locuszoomr", "biomaRt", "BiocManager",
  "EnsDb.Hsapiens.v75", "rsnps"))

####LIMPIEZA DEL DATASET####

rawdata <- read.delim2("Data/Olink_Kiel_rawdata.txt",
                      sep = " ", header = TRUE,
                      stringsAsFactors = FALSE) %>% as.tibble

#Generar una nueva columna cohort_status
rawdata <- rawdata%>%
  mutate(cohort=ifelse(str_detect(Assay,'KIN'),'KNIDRED',"EMGE"),
```

```

    cohort_status=(paste0(cohort,"_",status)),.after=Assay)

#Corregir comas y pasar variables de tipo numerico
rawdata[, 8:ncol(rawdata)] <-
  lapply(rawdata[, 8:ncol(rawdata)],
    function(x) {as.numeric(gsub(",", ".", x)) })
rawdata <- as.data.frame(rawdata)

#Pasar a long format
rawdata_long <- rawdata%>%
  pivot_longer(cols = -c(1:7),names_to = "protein",
    values_to = "value")%>%
  mutate(value=as.numeric(str_replace(value,",",".")))
write.table(rawdata_long, file = "Data/Olink_Kiel_Longdata.txt")

# Graficas para observar la distribucion de
#los datos crudos segun la cohorte

# Crear el directorio "plots" si no existe
if (!dir.exists("Plots")) {
  dir.create("Plots", recursive = TRUE)
}

#Generar histograma

histograma_rawdata <- function(){
  rawdata_long<- rawdata_long %>%
    mutate(cohort_status = dplyr::recode(cohort_status,
      "EMGE_0" = "Sano EMGE",
      "KNIDRED_0" = "Sano KINDRED",
      "KNIDRED_1" = "Enfermo KINDRED"
    ))

  histogram <-
    ggplot(rawdata_long,
      aes(x = value, fill = cohort_status)) +
    geom_histogram(color = "black",
      bins = 30, alpha = 0.7,
      position = "identity") +
    facet_wrap(~protein, scales = 'free') +
    scale_fill_grey(start = 0.3, end = 0.8) +
    labs(
      title = "Distribución de valores por proteína según
      su cohorte",
      x = "Valor",
      y = "Frecuencia",
      fill = "Cohorte"
    ) +
    theme_classic(base_size = 14) +
    theme(
      legend.position = "top",
      axis.text.x =
        element_text(angle = 45, hjust = 1),
      plot.title =
        element_text(size=40,hjust = 0.5, face = "bold"),
      strip.background =
        element_rect(color = "black",
          fill = "white",
          linewidth = 1),
      panel.border =

```

```

        element_rect(color = "black",
                      fill = NA,
                      linewidth = 1)
    )
    #Guardar el gráfico en la carpeta "plots"
    ggsave("Plots/Histograms_rawdata.pdf",
           histogram, width = 25, height = 25)
    return(histogram)
}
#Ver
histograma_rawdata()

#Generar boxplots
boxplot_rawdata <- function(){
  rawdata_long<- rawdata_long %>%
    mutate(cohort_status = dplyr::recode(cohort_status,
      "EMGE_0" = "Sano EMGE",
      "KNIDRED_0" = "Sano KINDRED",
      "KNIDRED_1" = "Enfermo KINDRED"
    ))
  boxplot <- ggplot(rawdata_long,
                    aes(x = cohort_status,
                       y = value, fill = cohort_status)) +
    geom_boxplot(outlier.shape = 16, outlier.size = 2,
                 alpha = 0.7, color = "black") +
    facet_wrap(~protein, scales = 'free') +
    scale_fill_grey(start = 0.3, end = 0.8) +
    labs(
      title =
        "Distribución de valores por proteína según su cohorte",
      x = "Cohorte",
      y = "Valor",
      fill = "Cohorte"
    ) +
    theme_classic(base_size = 14) +
    theme(
      legend.position = "top",
      axis.text.x = element_text(angle = 45, hjust = 1),
      plot.title = element_text(size=40,hjust = 0.5,
                                face = "bold"),
      strip.background = element_rect(color = "black",
                                       fill = "white",
                                       linewidth = 1),
      panel.border = element_rect(color = "black",
                                   fill = NA,
                                   linewidth = 1)
    )
  # Guardar el gráfico
  ggsave("Plots/Boxplots_rawdata.pdf", boxplot,
         width = 20, height = 30)
  return(boxplot)
}

boxplot_rawdata()

```

2.5. Proteínas asociadas a la condición de salud del individuo:

Al ser un experimento binomial de casos y controles se escogió un modelo de regresión logística binaria para evaluar la asociación entre la condición de los individuos y la expresión de las proteínas, incluyendo las variables cohorte de procedencia y el género de las personas como variables de confusión (confounder).

Con el fin de identificar las proteínas propias de cada una de las condiciones. Se realizaron cuatro análisis de regresión logística: EII vs Controles, CU vs Controles, EC vs Controles y CU vs EC,

El modelo logístico se implementó en R utilizando el paquete glm (R Core Team, 2024). Del resultado se extrajeron los valores Z, beta, desviación estándar y p de cada una de las proteínas (Fórmula 1). Para evitar falsos positivos, se controló la tasa de descubrimientos falsos (FDR) mediante el método de Benjamini-Hochberg, considerando significativas a todas aquellas proteínas cuyo valor p ajustado por FDR fuera inferior a 0.05.

$$\logit\left(\frac{P(Condicion=1)}{1-P(Condicion=1)}\right) = \beta_0 + \beta_1 \cdot \text{Expresion Proteina} + \beta_2 \cdot \text{Genero} + \beta_3 \cdot \text{Cohorte}$$

Table 1: Fórmula 1

Parámetro	Descripción
β_0	Logaritmo del odds (log-odds) de tener la condición cuando todas las variables predictoras son iguales a 0 (valor de referencia).
β_1	Cambio de log-odds de la condición por cada una unidad de cambio en la expresión de la proteína.
β_2	Efecto del género en los log-odds de la condición.
β_3	Efecto de la cohorte en los log-odds, ajustando por diferencias entre grupos.
Condición	Variable binaria que indica presencia o ausencia de la enfermedad (EII), o subtipos (EC o CU).
Expresión proteica	Media de los valores de NPX de la proteína en distintas placas.
Género	Variable de ajuste con valores Masculino o Femenino.
Cohorte	Procedencia del individuo como factor de ajuste (KINDRED o EMGE)

```
#Modelo: Regresion logistica for IBD vs Controles
get_assocs <- function(){

  # Crear los directorios si no existe
  dir.create("Data/IBDvsControl", recursive = TRUE, showWarnings = FALSE)
  dir.create("Plots/IBDvsControl", recursive = TRUE, showWarnings = FALSE)

  #Calcula los cuantiles
  qci <- rawdata_long%>%group_by(protein,cohort_status)%>%
    summarise(qci=quantile(value,(0+0.03),na.rm=T))
  qcf <- rawdata_long%>%group_by(protein,cohort_status)%>%
    summarise(qcf=quantile(value,(1-0.03),na.rm=T))

  ##Actualiza los valores atipicos encontrados como NA (los valores bajo qci
  #y sobre qcf se reemplazan por NA)
  rawdata_long_qc <- rawdata_long%>%
    left_join(qci,by=c("protein","cohort_status"))%>%
    left_join(qcf,by=c("protein","cohort_status"))%>%
    mutate(value_qc=case_when(value<qci~NA_real_,value>qcf~NA_real_,TRUE~value))

  #genera grafico de barras
  Grafico_de_barras <- function(){
    rawdata_long_qc <- rawdata_long_qc %>%
      mutate(cohort_status = dplyr::recode(cohort_status,
        "EMGE_0" = "Sano EMGE",
        "KNIDRED_0" = "Sano KINDRED",
```

```

    "KNIDRED_1" = "Enfermo KINDRED"
  ))
#grafico de barras
outliers_eliminados <- rawdata_long_qc %>%
  filter(is.na(value_qc)) %>%
  count(protein, cohort_status) %>%
  ggplot(aes(x = reorder(protein, n), y = n,
    fill = cohort_status)) +
  geom_col(color = "black", width = 0.7) +
  geom_text(aes(label = n),
    position = position_stack(vjust = 0.5),
    size = 6, color = "black") +
  coord_flip() +
  scale_fill_grey(start = 0.3, end = 0.8) +
  labs(
    title = "Número de outliers eliminados por proteína IBD+vsControl",
    x = "Proteína",
    y = "Número de outliers",
    fill = "Cohorte"
  ) +
  theme_classic(base_size = 14) +
  theme(
    legend.position = "top",
    axis.text.y = element_text(size = 10),
    plot.title = element_text(size=40,hjust = 0.5, face = "bold")
  )
# Guardar el gráfico
ggsave("Plots/IBDvsControl/Outliers_eliminados.pdf",
  outliers_eliminados, width = 25,
  height = 25)
return(outliers_eliminados)
}

```

Grafico_de_barras()

```

#Boxplot sin outliers
boxplot_sin_outliers <- function(){
  rawdata_long_qc <- rawdata_long_qc %>%
    mutate(cohort_status = dplyr::recode(cohort_status,
      "EMGE_0" = "Sano EMGE",
      "KNIDRED_0" = "Sano KINDRED",
      "KNIDRED_1" = "Enfermo KINDRED"
    ))

# Boxplot sin outliers con las etiquetas corregidas
#en la leyenda
boxplot_no_outliers <- ggplot(rawdata_long_qc %>% drop_na(),
  aes(x = cohort_status,
    y = value,
    fill = cohort_status)) +
  geom_boxplot(outlier.shape = 16, outlier.size = 3,
    alpha = 0.7, color = "black") +
  facet_wrap(~protein, scales = 'free') +
  scale_fill_grey(start = 0.3, end = 0.8) +
  labs(
    title = "Distribución de valores por proteína
    según su cohorte sin outliers IBD+vsControl",
    x = "Cohorte",

```

```

    y = "Valor",
    fill = "Cohorte"
  ) +
  theme_classic(base_size = 14) +
  theme(
    legend.position = "top",
    # legend.title = element_text(size = 30, face = "bold"),
    # legend.text = element_text(size = 30),
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(size = 40,
                              hjust = 0.5, face = "bold"),
    strip.background = element_rect(color = "black",
                                     fill = "white",
                                     linewidth = 1),
    panel.border = element_rect(color = "black",
                                 fill = NA, linewidth = 1)
  )
  # Guardar el gráfico

  ggsave("Plots/IBDvsControl/Boxplots_rawdata_No_Outliers.pdf",
         boxplot_no_outliers, width = 20, height = 30)
  return(boxplot_no_outliers)
}

```

```
boxplot_sin_outliers()
```

```

# Guardar datos filtrados
write.csv(rawdata_long_qc,
         file =
           "Data/IBDvsControl/Olink_Kiel_Longdata_No_outliers.txt",
         row.names = FALSE)

# Inicializar dataframe para almacenar resultados de asociaciones
assocs <- tibble()

# Eliminar filas con NA antes del análisis
rawdata_long_qc <- rawdata_long_qc %>% drop_na()

for (p in unique(rawdata_long_qc$protein)) {

  # Modelo logístico
  assoc <- glm(status ~ value_qc + Gender + cohort,
              data = rawdata_long_qc %>% filter(protein == p),
              family = binomial(link = "logit"))

  # Extraer resultados y formatear tabla
  assoc_sum <- summary(assoc)$coefficients %>%
    as.data.frame() %>%
    rownames_to_column("variable") %>%

```



```

mutate(protein = p, .before = 1) %>%
rename(Pvalue = "Pr(>|z|)") %>%
mutate(variable = str_replace_all(variable, "\\(|\\|)", "")) %>%
as_tibble()

colnames(assoc_sum) <- c("protein",
                        "variable",
                        "Estimate",
                        "StdError",
                        "Zvalue",
                        "Pvalue")

# Acumular resultados
assocs <- bind_rows(assocs, assoc_sum)
}

# Guardar tabla de asociaciones
write_tsv(assocs, file = "Data/IBDvsControl/Olink_Kiel_assocs.txt")

# Aplicar corrección de FDR
assocdf <- assocs %>%
  filter(variable == "value_qc") %>%
  mutate(PvalueFDR = p.adjust(Pvalue, method = "fdr"))

# Gráfico de cascada (dot plot)
sortedplot <- assocdf %>% arrange(Estimate)

p4 <- ggplot(assocdf %>%
  mutate(protein =
    factor(protein, levels = sortedplot$protein)),
  aes(x = Estimate, y = protein, color = PvalueFDR < 0.05)) +
geom_vline(xintercept = 0,
  linetype = "dashed",
  color = "black",
  linewidth = 0.8) + # Línea punteada en x = 0
geom_point(size = 4) +
scale_color_manual(values = c("gray50", "black")) + # Invertir colores
labs(
  title = "Asociaciones por proteína IBD+vsControl",
  x = "Estimación del efecto",
  y = "Proteína",
  color = "FDR < 0.05"
) +
theme_classic(base_size = 14) +
theme(
  legend.position = "top",
  legend.title = element_text(size = 14, face = "bold"),
  legend.text = element_text(size = 12),
  axis.text.x = element_text(angle = 45, hjust = 1),
  axis.text.y = element_text(size = 13),
  plot.title = element_text(size = 20, hjust = 0.5, face = "bold"),
  strip.background = element_rect(color = "black", fill = "white", linewidth = 1),
  panel.border = element_rect(color = "black", fill = NA, linewidth = 1)
)

ggsave("Plots/IBDvsControl/Dotplot_associations.pdf", p4, width = 8, height = 15)

# Gráfico de volcán
p5 <- ggplot(assocdf, aes(x = Estimate, y = -log10(Pvalue),
  label = ifelse(Pvalue < 0.05, protein, "")),

```

```

                                color = PvalueFDR < 0.05)) +
geom_hline(yintercept = -log10(0.015),
           linetype = "dashed",
           color = "black",
           linewidth = 0.8) + # Umbral de p-valor 0.05
geom_vline(xintercept =
           c(-0.5, 0.5),
           linetype = "dashed",
           color = "black",
           linewidth = 0.8) + # Umbrales de efecto
geom_point(size = 4) +
geom_text_repel(size = 5, box.padding = 0.5, max.overlaps = 15) +
# Etiquetas de proteínas significativas
scale_color_manual(values = c("gray50", "black")) +
# Invertir colores: negro para significativos
labs(
  title = "Volcán de asociaciones IBD+vsControl",
  x = "Estimación del efecto",
  y = "-log10(P-valor)",
  color = "FDR < 0.05"
) +
theme_classic(base_size = 14) +
theme(
  legend.position = "top",
  legend.title = element_text(size = 14, face = "bold"),
  legend.text = element_text(size = 12),
  axis.text.x = element_text(angle = 45, hjust = 1),
  axis.text.y = element_text(size = 13),
  plot.title = element_text(size = 20, hjust = 0.5, face = "bold"),
  strip.background = element_rect(color = "black", fill = "white", linewidth = 1),
  panel.border = element_rect(color = "black", fill = NA, linewidth = 1)
)

ggsave("Plots/IBDvsControl/Volcanoplot_associations.pdf", p5, width = 10, height = 10)

# Filtrar proteínas significativas
result <- subset(assocdf, PvalueFDR < 0.05)
write.csv(result, file = "Data/IBDvsControl/Olink_Kiel_filtrado_FDR.txt",
          row.names = FALSE)
}

get_assocs()

```

2.6. Identificación de pQTLs asociados a las proteínas previamente identificadas:

Con base en los niveles de expresión de las proteínas significativas, los individuos se distribuyeron en dos grupos; casos y controles. Aquellos cuya expresión proteica supera el percentil 75 fueron clasificados como casos, mientras que aquellos con niveles por debajo del percentil 50 se consideraron controles (Figure 1).

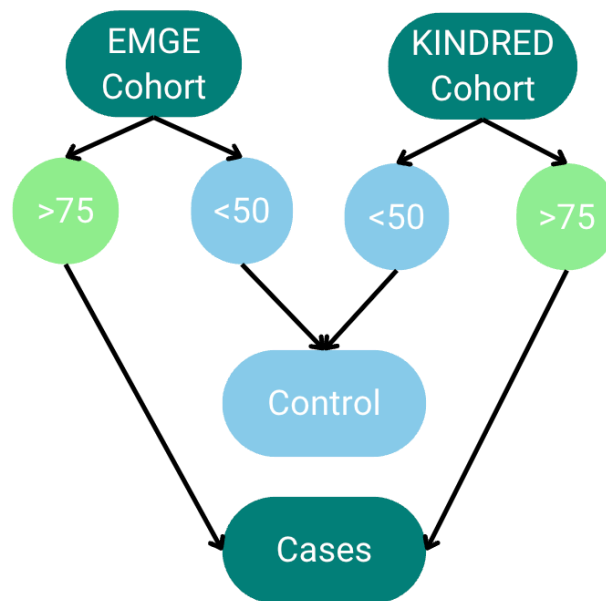


Figure 1: **Distribución realizada en grupos por cada proteína.** Los individuos cuya expresión proteica supera el percentil 75 fueron clasificados como casos, mientras que aquellos con niveles por debajo del percentil 50 se consideraron controles.

```

#distribucion de individuos

#NORMALIZE PROTEINS####
raw_significative <- read.delim("Data/Olink_Kiel_Significative_Proteins_df.txt")

#SEPARAR EMGES Y KIN####

#KIN
df_kin <- subset(raw_significative, grepl("^KIN", Assay))
df_kin_long <- df_kin%>%pivot_longer(cols = -c(1:9),
                                     names_to = "protein",
                                     values_to = "value")%>%
  mutate(value=as.numeric(str_replace(value,",",".")))

#EMGE
df_emge <- subset(raw_significative, grepl("^EMGE", Assay))
df_emge_long <- df_emge%>%
  pivot_longer(cols = -c(1:9),
               names_to = "protein",
               values_to = "value")%>%
  mutate(value=as.numeric(str_replace(value,",",".")))

#RECODE.TXT PARA CADA PROTEINA####
# Definimos la función
separar_grupos_proteina <- function(data_emge, data_kin, proteina) {
  # Crear la ruta del directorio dentro de Data/Proteinas
  dir_name <- file.path("Data", "Proteinas", proteina)

  # Verificar si el directorio no existe y crearlo
  if (!dir.exists(dir_name)) {
    dir.create(dir_name, recursive = TRUE)
    # 'recursive = TRUE' asegura que se creen las carpetas intermedias si no existen
  }
  # Filtramos las filas donde 'protein' coincide con la proteína especificada para EMGE
  protein_data_emge <- subset(data_emge, grepl(proteina, protein))

```

```
# Filtramos las filas donde 'protein' coincide con la proteína especificada para KIN
protein_data_kin <- subset(data_kin, grepl(proteina, protein))

# Combinamos los datasets EMGE y KIN para la proteína especificada
combined_data_50 <- bind_rows(
  protein_data_emge %>%
    filter(value > quantile(value, 0.01, na.rm = TRUE) &
           value <= quantile(value, 0.50, na.rm = TRUE)),
  protein_data_kin %>%
    filter(value > quantile(value, 0.01, na.rm = TRUE) &
           value <= quantile(value, 0.50, na.rm = TRUE))
)

combined_data_75 <- bind_rows(
  protein_data_emge %>%
    filter(value >= quantile(value, 0.75, na.rm = TRUE) &
           value <= quantile(value, 0.99, na.rm = TRUE)),
  protein_data_kin %>%
    filter(value >= quantile(value, 0.75, na.rm = TRUE) &
           value <= quantile(value, 0.99, na.rm = TRUE))
)

combined_all <- bind_rows(
  combined_data_50 %>% mutate(group = 'control'),
  combined_data_75 %>% mutate(group = 'case')
)

# Renombramos las columnas según el formato esperado
combined_all <- combined_all %>%
  rename(FID = personal_id, IID = ID, SEX = Gender, COHORT= cohort, phenoPlink = group)

# Crear y guardar el plot usando phenoPlink original
plot <- ggplot(combined_all, aes(x = phenoPlink, y = value, fill = phenoPlink)) +
  geom_boxplot() +
  theme_minimal() +
  labs(title = paste("Distribución de valores para", proteina),
       x = "Grupo",
       y = "Valor") +
  scale_fill_manual(values = c("control" = "skyblue", "case" = "lightgreen"))

ggsave(filename =
  file.path(dir_name, paste0(proteina, "_distribution_plot.png")),
  plot = plot, width = 10, height = 6)

# Convertimos los valores de la columna phenoPlink a numéricos (1 y 2)
combined_all$phenoPlink[combined_all$phenoPlink == "control"] <- 1
combined_all$phenoPlink[combined_all$phenoPlink == "case"] <- 2

# Convertimos los valores de la columna COHORT a numéricos (1 y 2)
combined_all$COHORT[combined_all$COHORT == "EMGE"] <- 0
combined_all$COHORT[combined_all$COHORT == "KNIDRED"] <- 1

# Filtramos las columnas necesarias
combined_all <- combined_all[, c("FID", "IID", "SEX", "COHORT", "phenoPlink")]

# Aseguramos que las columnas FID e IID sean de tipo carácter sin comillas
combined_all$FID <- as.character(combined_all$FID)
combined_all$IID <- as.character(combined_all$IID)

# Guardamos el DataFrame combinado en un archivo de texto dentro del directorio
```

```
write.table(combined_all, file.path(dir_name, paste0(proteina, "_recode.txt")),
            sep = "\t", row.names = FALSE, col.names = FALSE, quote = FALSE)
}

proteinas_significativas <- c("IL7", "IL.17A", "CXCL11", "CXCL9", "CXCL1", "CD6",
                             "TGF.alpha", "CCL11", "FGF.23", "PD.L1", "HGF",
                             "IL10", "TNF", "CCL23", "DNER", "IFN.gamma",
                             "FGF.19", "CSF.1", "VEGFA", "MMP.10", "CXCL10")

# Aplicamos la función para cada proteína
for (proteina in proteinas_significativas) {
  separar_grupos_proteina(df_emge_long, df_kin_long, proteina)
}
```

Esta clasificación permitió una separación contrastante entre los individuos con diferencias marcadas en la expresión proteica, lo que facilitó la detección de asociaciones genéticas.

Se usó PLINK 2.0 (C. C. Chang, et al., 2015) para evaluar las frecuencias alélicas entre casos y controles, con el objetivo de evaluar la asociación entre cada SNP y los niveles de expresión de las proteínas mediante regresiones logísticas. Para este análisis, se utilizaron los datos genómicos imputados anteriormente.

Los niveles de significancia de los SNP evaluados se visualizaron a través de Manhattan plots por cada proteína utilizando el paquete qqman de R (Turner, 2018).

```
#Edicion del FAM file, anadiendo las covariables
OLINKiel <-
  read.table("Data/PLINK_Files/Common_inputed_PLINK_file/OLINKiel.fam",
            quote="\"", comment.char="") %>% rename(
  FID=V1,
  IID=V2,
  PID=V3,
  MID=V4,
  SEX=V5,
  PHENO=V6)

#Eliminar patrones reros del FAM
OLINKiel$IID <- sub("^0_", "", OLINKiel$IID)
OLINKiel$IID <- sub("^P[0-9]+_", "", OLINKiel$IID)

OLINKiel <- OLINKiel %>%
  left_join(datos %>% dplyr::select(old_id, personal_id), by = c("IID" = "old_id")) %>%
  # Reemplazar la columna FID con personal_id
  mutate(FID = personal_id) %>%
  # Mover la columna FID al principio y eliminar personal_id
  dplyr::select(FID, everything()) %>%
  dplyr::select(-personal_id) # Eliminar la columna personal_id

OLINKiel <- OLINKiel %>%
  left_join(datos %>% dplyr::select(old_id, Gender), by = c("IID" = "old_id")) %>%
  mutate(SEX = Gender) %>%
  dplyr::select(-Gender)

OLINKiel$SEX[is.na(OLINKiel$SEX)] <- -9
OLINKiel$FID[is.na(OLINKiel$FID)] <- 0

OLINKiel$SEX <- ifelse(OLINKiel$SEX == -9, 0, OLINKiel$SEX + 1)

write.table(OLINKiel,
            "Data/PLINK_Files/OLINKiel.fam/Common_inputed_PLINK_file/Edited/OLINKiel.fam",
```

```
row.names = FALSE, col.names = FALSE, quote = FALSE)
```

```
#evaluar frecuencias alelicas
process_experiment_folders <- function(parent_folder, plink_raw_file) {

  # Obtener la lista de carpetas (nombres de experimentos)
  experiment_names <- list.dirs(parent_folder, full.names = FALSE, recursive = FALSE)

  # Función interna para procesar cada experimento
  process_single_experiment <- function(experiment_name) {
    # Definir rutas para el experimento actual
    experiment_path <- file.path(parent_folder, experiment_name)
    recode_file <- file.path(experiment_path, paste0(experiment_name, "_recode.txt"))
    # Archivo recode dentro de la carpeta del experimento
    outpath <- experiment_path

    # Comando PLINK para MAF usando plink_raw_file
    maf_command <- paste0("plink --bfile ", plink_raw_file,
                          " --keep ", recode_file,
                          " --chr 1-22,X,Y --maf 0.01 --make-bed --out ",
                          file.path(outpath, experiment_name))

    system(maf_command)

    # Comando PLINK para Pruning
    pruning_command <- paste0("plink --bfile ", file.path(outpath, experiment_name),
                              " --indep-pairwise 200 50 0.25 --out ",
                              file.path(outpath, experiment_name))

    system(pruning_command)

    # Comando PLINK para PCA
    pca_command <- paste0("plink --bfile ", file.path(outpath, experiment_name),
                          " --extract ",
                          file.path(outpath, paste0(experiment_name, ".prune.in")),
                          " --pca 5 --out ",
                          file.path(outpath, experiment_name))

    system(pca_command)

    # Leer eigenvec y preparar phenox
    eigenvec_file <- file.path(outpath, paste0(experiment_name, ".eigenvec"))
    eigenvec <- read.table(eigenvec_file, quote="\"",
                          comment.char="",
                          header = FALSE,
                          as.is = TRUE) %>%

    as_tibble() %>%
    rename(FID = V1, IID = V2, PC1 = V3, PC2 = V4, PC3 = V5, PC4 = V6, PC5 = V7)

    phenox_file <- file.path(experiment_path, paste0(experiment_name, "_recode.txt"))
    phenox <- read.delim(phenox_file, header = FALSE) %>%
    as_tibble() %>%
    rename(FID = V1, IID = V2, SEX = V3, COHORT = V4, phenoPlink = V5)

    write.table(phenox, file.path(outpath, paste0(experiment_name, ".txt")),
                row.names = FALSE, col.names = FALSE, sep = "\t", quote = FALSE)

    # Combinar eigenvec con phenox
    covars <- phenox %>%
    left_join(eigenvec %>% dplyr::select(-FID), by = "IID") %>%
    dplyr::select(FID, IID, SEX) %>%
    ### select(FID, IID,
    #SEX, COHORT, PC1, PC2, PC3, PC4, PC5)
```

```
#Quitar los pca, prueba con sex y cohort#
distinct(IID, .keep_all = TRUE) %>%
drop_na()

# Guardar covariables y .pheno
write.table(covars, file.path(outpath, paste0(experiment_name, "_covars.txt")),
            row.names = FALSE, col.names = TRUE, sep = "\t", quote = FALSE)
write.table(phenox %>% dplyr::select(FID, IID, phenoPlink),
            file.path(outpath, paste0(experiment_name, "_plink.pheno")),
            row.names = FALSE, col.names = FALSE, sep = "\t", quote = FALSE)

# Actualizar fenotipo con PLINK
update_pheno_command <- paste0("plink --bfile ",
                                file.path(outpath, experiment_name),
                                " --pheno ",
                                file.path(outpath,
                                              paste0(experiment_name,
                                                         "_plink.pheno")),
                                " --make-bed --out ",
                                file.path(outpath, experiment_name))

system(update_pheno_command)

# Asociación con PLINK
assoc_command <- paste0("plink --bfile ",
                        file.path(outpath, experiment_name),
                        " --pheno ",
                        file.path(outpath, paste0(experiment_name,
                                                    "_plink.pheno")),
                        " --assoc --covar ",
                        file.path(outpath,
                                  paste0(experiment_name,
                                           "_covars.txt")),
                        " --out ",
                        file.path(outpath, experiment_name))

system(assoc_command)

# Regresión logística y resultados
logistic_command <- paste0("plink --bfile ",
                            file.path(outpath, experiment_name),
                            " --pheno ",
                            file.path(outpath,
                                        paste0(experiment_name,
                                                "_plink.pheno")),
                            " --logistic --covar ",
                            file.path(outpath,
                                        paste0(experiment_name,
                                                "_covars.txt")),
                            " --out ",
                            file.path(outpath, experiment_name))

system(logistic_command)

print("Reading and filtering assoc results, this could take a while")
assoc_logistic_results <-
  fread(file.path(outpath,
                  paste0(experiment_name, ".assoc.logistic"))) %>%
  filter(TEST == "ADD") %>%
  drop_na()

write.table(assoc_logistic_results,
```

```

        file.path(outpath,
                  paste0(experiment_name,
                        "_assoc_logistic_results.txt")),
        row.names = FALSE, col.names = TRUE, sep = "\t", quote = FALSE)

lower_threshold <- 5e-8
upper_threshold <- 1e-5

# Filtrar los SNPs significativos
filtered_snps <- assoc_logistic_results %>%
  filter(P > lower_threshold & P <= upper_threshold) %>%
  dplyr::select(CHR, SNP, BP, A1, TEST, NMISS, OR, STAT, P)

# Crear el nombre del archivo de salida
txt_file_path <- file.path(outpath, paste0(experiment_name,
                                           "_significatives.txt"))

# Guardar el dataframe completo en un archivo de texto (formato tabulado)
write.table(filtered_snps, file = txt_file_path, sep = "\t", row.names = FALSE,
            quote = FALSE)

# Nombre del archivo de la gráfica Manhattan
manhattan_plot_path <- file.path(outpath, paste0(experiment_name, "_manhattan.png"))

# Graficar Manhattan
png(manhattan_plot_path, width = 1000, height = 600)
# Tamaño ajustado para mejor visibilidad

manhattan(assoc_logistic_results, chr = "CHR", bp = "BP", p = "P", snp = "SNP",
          main = paste("Manhattan Plot for", experiment_name), ylim = c(0, 10))

# Agregar líneas horizontales para los umbrales
abline(h = -log10(lower_threshold), col = "red")
abline(h = -log10(upper_threshold), col = "blue")

if (nrow(filtered_snps) > 0) {
  message(paste("Theres", nrow(filtered_snps),
                "significant SNPs within the specified threshold range.))
} else {
  message("No significant SNPs found within the specified threshold range.")
}

dev.off()

}

# Aplicar la función a cada nombre de experimento
walk(experiment_names, process_single_experiment)

}

# Llamar a la función con la carpeta madre y el archivo PLINK común
process_experiment_folders("Data/Proteinas",
                          "Data/PLINK_Files/Common_inputed_PLINK_file/Edited/OLINKiel")

```

Estos gráficos son ampliamente usados en estudios de asociación del genoma completo (GWAS), para representar la significancia estadística ($-\log_{10}$ del valor p) de múltiples SNPs según su posición cromosómica. En ellos, cada punto representa un SNP individual, con cromosomas distinguidos mediante etiquetas y colores en el eje X,

mientras que en el eje Z muestra la magnitud de la asociación estadística. Los SNPs que superan el umbral de significancia establecido aparecen en la parte superior del gráfico.

El Manhattan Plot permitió identificar las regiones genómicas con variantes significativamente asociadas a la expresión de cada proteína estudiada, facilitando así la detección de loci de rasgos cuantitativos (pQTLs) que modulan los niveles proteicos de interés. Además se utilizó la información de la imputación para evaluar visualmente la significancia de los pQTLs identificados. La visualización se realizó a través del paquete LocusZoom (Heinz et al., 2017).

```
#VER GENES ASOCIADOS AL SNP####

#LOCUS PLOT con ensembl####

library(EnsDb.Hsapiens.v75)# basado en GRCh37 (hg19).

# Crear la carpeta "Plots" si no existe
dir.create("Plots/Locus", showWarnings = FALSE, recursive = TRUE)

#Lista de proteínas con snps significativos
proteinas_significativas <- c("CD6","CXCL10","CXCL11","FGF.23","HGF","VEGFA")

#funcion
locusplot <- function(proteina){

  # Cargar datos GWAS específicos de la proteína
  proteina_gwas <-
    read.delim(paste0("Data/Proteinas/",
                      proteina, "/",
                      proteina,
                      "_assoc_logistic_results.txt"))

  # Cargar lista de snps significativos
  snps_significativos <-
    read.delim(paste0("Data/Proteinas/",
                      proteina, "/",
                      proteina,
                      "_significatives.txt")) %>%
    dplyr::pull(SNP)

  for (snp in snps_significativos) {

    token_ld="fad50e02d366"
    genome_version="hg38"

    loc <- locus(
      data = proteina_gwas,
      chrom = "CHR",      # Columna del cromosoma
      pos = "BP",         # Columna de la posición base
      p = "P",            # Columna de los valores P
      labs = "SNP",       # Columna de los SNPs
      index_snp = snp,    # El SNP de interés (rsID)
      flank = 10^5,       # Flanqueo alrededor del SNP
      ens_db = "EnsDb.Hsapiens.v75" # Usar la base de datos de ensembl
    )

    # 3. Graficar el locus inicial
    locus_plot(loc, labels = c("index", snp))

    # 4. Enlazar los datos de LD usando el token proporcionado
    loc <- link_LD(loc, token = token_ld)

    # 5. Enlazar los datos de recombinación para el genoma deseado
```

```
loc <- link_recomb(loc, genome = genome_version)

# 6. Graficar el locus después de enlazar LD y recombinación
locus_plot(loc, labels = c("index", snp))

# Nombre del archivo de salida
output_file <- paste0("Plots/Locus/", proteina, "-", snp, "_locusplot.png")

# Guardar el plot en un archivo PNG
png(output_file, width = 1200, height = 800, res = 150)
# Ajusta el tamaño y la resolución

# Graficar nuevamente el locus plot al guardar
locus_plot(loc, labels = c("index", snp))

# Agregar el título
mtext(paste("Locus Plot for", proteina, "-", snp), side = 3,
      line = 5,
      cex = 0.8,
      font = 2)

# Cerrar la imagen PNG para guardar el archivo
dev.off()
}

}

for (proteina in proteinas_significativas) {
  locusplot(proteina)
}
```

Un Locus Plot es un gráfico que examina una región genómica específica asociada a un rasgo de interés. A diferencia del Manhattan Plot, el Locus Plot se enfoca en una región cromosómica donde hay una señal significativa. Este tipo de visualización proporciona información sobre el grado de desequilibrio de ligamiento (LD por sus siglas en inglés) entre el SNP principal y los SNPs circundantes, así como a los genes a los que están asociados. El eje X muestra la posición física en el cromosoma, mientras que el eje Y representa la significancia estadística ($-\log_{10}$ del valor p).

Los Locus Plots generados facilitaron la caracterización de la arquitectura genética de las regiones con pQTLs significativos y el análisis de la relación con genes candidatos que podrían estar influyendo en los niveles de expresión de la proteína asociada con IBD.

Además, se ajustaron los Locus Plots considerando el LD, utilizando el paquete LDlinkR (Machiela & Chanock, 2015) en R, el cual emplea datos del 1000 Genomes Project (Auton et al., 2015b) para calcular los valores de LD en el conjunto de datos analizado.

2.7. Enriquecimiento Funcional:

Para identificar los genes asociados con los SNPs seleccionados, se realizó un análisis con la librería `rsnps` en R (Rüeger et al., 2022). Se estableció una lista de SNPs previamente identificados, agrupados por la proteína con la que se encuentran potencialmente relacionados. Luego, cada SNP fue consultado en la base de datos de NCBI mediante la función `ncbi_snp_query()` de `rsnps`, con el objetivo de recuperar información sobre los genes a los que están asignados.

```
#Mapear SNPs a Genes

install.packages("rsnps")
library(rsnps)

# Definir la lista de SNPs con su proteína asociada
```

```
snp_list <- list(
  "CD6" = c("rs77069221", "rs11670764"),
  "CXCL10" = c("rs8090960"),
  "CXCL11" = c("rs1121619", "rs6495562", "rs148236522"),
  "FGF.23" = c("rs9314154", "rs3756706", "rs6871090", "rs1735141"),
  "HGF" = c("rs17237297"),
  "VEGFA" = c("rs2946834")
)

# Crear un dataframe vacío
snp_gene_table <- data.frame(Proteina = character(),
                             SNP = character(),
                             Gene = character(),
                             stringsAsFactors = FALSE)

# Iterar sobre cada proteína y sus SNPs
for (proteina in names(snp_list)) {
  for (snp in snp_list[[proteina]]) {
    snp_info <- ncbi_snp_query(snp) # Consultar el SNP en NCBI

    # Verificar si se obtuvo información
    if (!is.null(snp_info) && nrow(snp_info) > 0) {
      gene_name <- snp_info$gene # Extraer el nombre del gen

      # Si el SNP no tiene un gen asociado, poner "No encontrado"
      if (is.na(gene_name) || gene_name == "") {
        gene_name <- "No encontrado"
      }

      # Agregar a la tabla
      snp_gene_table <- rbind(snp_gene_table,
                             data.frame(Proteina = proteina,
                                         SNP = snp,
                                         Gene = gene_name,
                                         stringsAsFactors = FALSE))
    } else {
      snp_gene_table <- rbind(snp_gene_table,
                             data.frame(Proteina = proteina,
                                         SNP = snp,
                                         Gene = "No encontrado",
                                         stringsAsFactors = FALSE))
    }
  }
}

# Mostrar la tabla final
view(snp_gene_table)

# Crear el directorio si no existe
dir.create("Data/Análisis_funcional", recursive = TRUE, showWarnings = FALSE)
# Guardar la tabla en un archivo CSV
write.csv(snp_gene_table, "Data/Análisis_funcional/SNPstoGenes.txt", row.names = FALSE)
```

La lista de genes identificados se analizó mediante la herramienta FUMA (Functional Mapping and Annotation of GWAS) (Watanabe et al., 2017), utilizando el módulo Gen2Func. Esto permitió determinar las funciones biológicas y los procesos moleculares en los que participan los genes asociados a los SNPs, así como su expresión en distintos tejidos y su posible papel en la patogénesis de la EII.

Finalmente, los genes identificados se analizaron en la base de datos STRING (Search Tool for the Retrieval of Interacting Genes/Proteins) (Szklarczyk et al., 2021) para evaluar las interacciones funcionales entre las

proteínas codificadas.

3. Resultados:

3.1. Estructura de la muestra en estudio:

Se observó que 73 de las 92 proteínas fueron detectadas en más del 75% de las muestras, lo que representa el 78.49% del total de proteínas analizadas en el panel de *Olink Inflammation*. Estos resultados coinciden con lo reportado por Olink, donde la detectabilidad esperada en plasma con ácido etilendiaminotetraacético (EDTA), basada en muestras de donantes sanos, supera el 75%. Sin embargo este valor puede variar según las condiciones del individuo, la muestra y los métodos de preparación (Figure 2).

De las 440 muestras, 432 cumplieron los criterios de calidad, lo que representa al el 98% del total analizado. En promedio, el coeficiente de variación intraensayo fue de 5% mientras que el de interensayo fue de 10%, ambos dentro de los rangos de referencia recomendados, <15% y <25%, respectivamente.

La distribución del coeficiente de variación intraensayo mostró que 77 proteínas presentaron una variación inferior al 5%, mientras que 15 tuvieron un coeficiente entre el 5% y 10%, evidenciando alta precisión y repetibilidad. Por otro lado, la variación interensayo reveló que 64 proteínas tuvieron un coeficiente menor al 10% y 28 se ubicaron entre el 10% y 20%, reflejando una buena reproducibilidad.

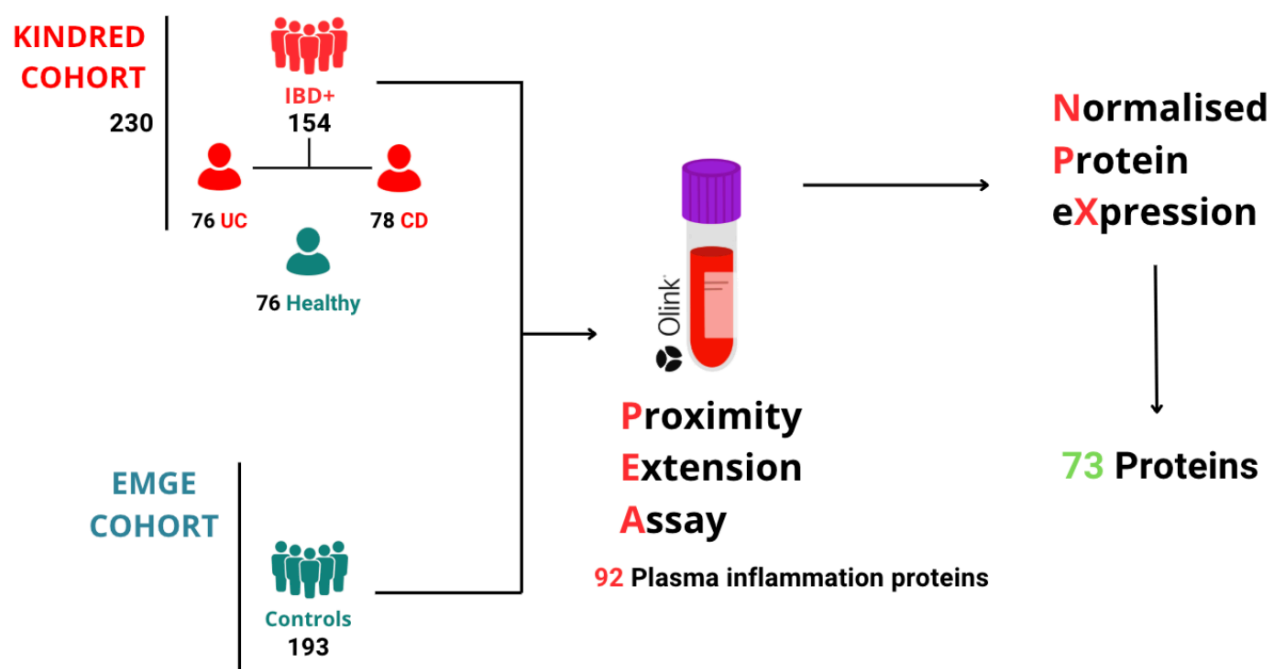


Figure 2: **Composición de la muestra analizada.** Se evaluaron muestras de plasma de las cohortes KINDRED (n = 320) y EMGE (n = 193 controles) mediante la tecnología Proximity Extension Assay (PEA) de Olink. La cohorte KINDRED incluyó 154 individuos con enfermedad inflamatoria intestinal (EII), subdivididos en 76 con colitis ulcerativa (CU) y 78 con enfermedad de Crohn (EC), además de 76 individuos sanos como controles internos. Tras la normalización y el control de calidad, 73 de las 92 proteínas medidas fueron retenidas para el análisis final.

Al aplicar el control de calidad (Anexo 2) para las variantes en las secuencias, se excluyeron un total de 208,713 variantes (28.6% del total inicial), quedando 521,584 variantes antes de la imputación. La eliminación se debió principalmente a una alta tasa de datos faltantes (missingness > 2%) y a la exclusión de variantes monomórficas o con anomalías en la anotación. En cuanto a las muestras, 83 individuos fueron eliminados debido a duplicación, parentesco cercano ($\hat{r} > 0.1875$) o anomalías en las métricas de heterocigosidad y missingness. Como resultado, se retuvieron 349 individuos, de los cuales 301 eran casos y 48 controles.

De la imputación realizada en TOPMed, bajo el panel de referencia (Trans-Omics for Precision Medicine), se imputaron 10,905,855 variantes para los 349 individuos.

3.2. La expresión de 21 proteínas resultó estar asociada con IBD:

Se identificaron 21 proteínas cuya expresión se asoció significativamente con la EII, diferenciadas por comparaciones de grupos: EII vs. Controles, CU vs. EC, EC vs. Controles y CU vs. Controles. La lista de proteínas significativamente asociadas en cada comparación, junto con sus respectivos p-valores, se encuentra en la Tabla 2.

Las diferencias en la expresión proteica se ilustran en la Figura 3, donde los Volcano Plots muestran las proteínas con cambios relevantes entre los grupos comparados, resaltando aquellas con asociaciones significativas y, por ende, con un mayor impacto en la enfermedad. Además, estos gráficos permiten visualizar la dirección del efecto, indicando qué proteínas presentan una mayor o menor expresión en cada condición analizada.

Table 2: P-valores de las proteínas en las comparaciones del estudio: Se identificaron proteínas significativas en cada una de las comparaciones realizadas, junto con sus respectivos P-valores. En total, se distinguieron 21 proteínas significativas en las cuatro comparaciones realizadas. En la comparación **EII vs. Controles**, se detectaron 16 proteínas asociadas, destacándose **CCL11** ($p = 0.000031$). En **EC vs. Controles**, se identificaron 18 proteínas, siendo **FGF-19** la más significativa ($p = 0.000002$). Para la comparación **CU vs. EC**, únicamente **FGF-19** mostró una asociación significativa ($p = 0.001120$). Finalmente, en **CU vs. Controles**, se encontraron 7 proteínas con asociaciones relevantes, resaltando nuevamente **CCL11** con un p-valor significativo ($p = 0.000291$).

Proteína	Pvalue.ECvsC	Pvalue.EIIvsC	Pvalue.CUvsEC	Pvalue.CUvsC
IL7	0.023681	-	-	-
IL.17A	0.000404	0.000075	-	0.001958
CXCL11	0.023334	0.002613	-	0.004638
CXCL9	0.023334	0.004363	-	0.008500
CXCL1	0.011110	0.018026	-	-
CD6	0.038323	-	-	-
TGF.alpha	0.024597	0.04182	-	-
FGF.23	0.01110	0.026278	-	-
PD.L1	0.000404	0.002613	-	-
IL10	0.007886	0.002613	-	-
TNF	0.007490	0.006898	-	-
CCL23	0.023334	-	-	-
DNER	0.003760	0.018928	-	-
FGF.19	0.000002	0.000065	0.001120	-
CSF.1	0.007490	-	-	-
CCL11	0.001225	0.000031	-	0.000291
HGF	0.003227	0.005208	-	-
IFN.gamma	0.000298	0.002613	-	-
VEGFA	-	0.030690	-	-
MMP.10	-	0.005936	-	0.001336
CXCL10	-	-	-	0.009817

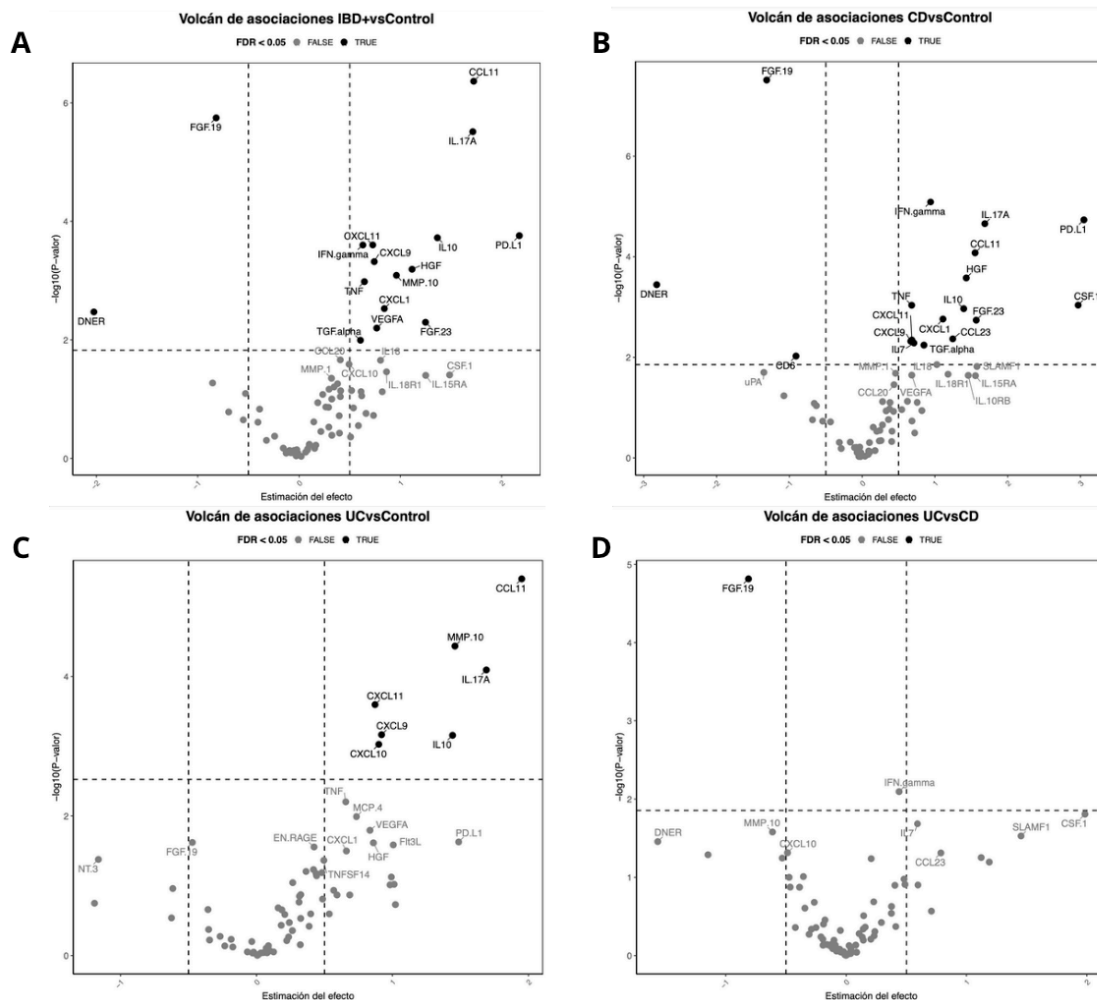


Figure 3: **Los gráficos de volcán muestran la distribución de las proteínas diferencialmente expresadas en cada comparación:** (A) Enfermedad inflamatoria intestinal (IBD) vs. Controles, (B) Enfermedad de Crohn (CD) vs. Controles, (C) Colitis ulcerosa (UC) vs. Controles y (D) Colitis ulcerosa vs. Enfermedad de Crohn. Los puntos resaltados corresponden a proteínas con una diferencia significativa según $FDR < 0.05$. El eje X representa la estimación del efecto, mientras que el eje Y muestra el valor $-\log_{10}(p\text{-valor})$.

Las diferencias en la expresión de las proteínas se evidencian de la misma manera en la Figura 4, donde estos gráficos representan la estimación del efecto de cada proteína en las comparaciones realizadas, permitiendo visualizar la magnitud y dirección del efecto. En la comparación EII vs. Controles, se identificaron 2 proteínas subexpresadas y 14 sobreexpresadas. En EC vs. Controles, hubo 3 proteínas subexpresadas y 15 sobreexpresadas. En UC vs. Controles, todas las 7 proteínas identificadas fueron sobreexpresadas. Finalmente, en EC vs. UC, únicamente se identificó una proteína subregulada.

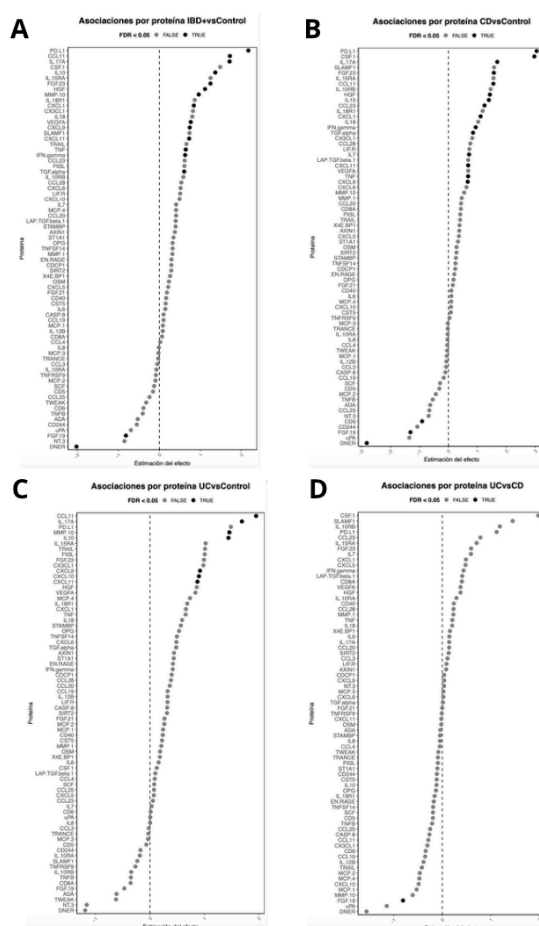


Figure 4: **Distribución de proteínas significativas en cada comparación mediante Dot Plots.** Gráficos tipo forest plot que representan la estimación del efecto de la expresión de proteínas en cada comparación: (A) Enfermedad inflamatoria intestinal (IBD) vs. controles, (B) Enfermedad de Crohn (CD) vs. controles, (C) Colitis ulcerativa (UC) vs. controles, y (D) Colitis ulcerativa vs. Enfermedad de Crohn. Cada punto representa una proteína, con su estimación del efecto en el eje X y los nombres de las proteínas en el eje Y. Las proteínas significativamente asociadas ($FDR < 0.05$) están resaltadas en negro.

3.3. Proteínas asociadas a IBD resultaron tener pQTLs:

Entre las 23 proteínas identificadas como significativamente asociadas a variaciones genéticas tras el análisis de regresión logística, que evaluó la relación entre 15,775,290 SNPs y la expresión proteica (Anexo1 Figura 2B), seis presentaron al menos un SNP con una asociación estadísticamente significativa: HGF, CXCL10, CXCL11, CD6, FGF23 y VEGFA (Tabla 3, Anexo1 Figura 2C).

Table 3: **Proteínas con asociaciones significativas a la expresión proteica.** Se identificaron seis proteínas con SNPs significativamente asociados a su expresión, detallando su cromosoma, posición genómica y valor de p. Destacan CD6 y CXCL11, cada una con dos SNPs asociados, y FGF23, que presentó tres SNPs con asociación significativa.

Protein	CHR	SNP	BP	A1	TEST	NMISS	OR	STAT	P
CD6	4	rs77069221	186643337	T	ADD	107	10.95	4.573	4.798e-06
CD6	19	rs11670764	58481529	A	ADD	107	0.1485	-4.422	9.791e-06
CXCL10	18	rs8090960	7261088	G	ADD	105	10.79	4.701	2.595e-06
CXCL11	8	rs1121619	3612178	A	ADD	103	8.302	4.431	9.378e-06
CXCL11	15	rs6495562	81319776	C	ADD	103	10.43	4.504	6.671e-06
FGF23	5	rs9314154	95782453	T	ADD	111	0.1198	-4.726	2.285e-06
FGF23	5	rs6871090	95797882	A	ADD	111	0.142	-4.558	5.153e-06
FGF23	21	rs1735141	39739173	A	ADD	111	5.589	4.504	6.655e-06
HGF	15	rs17237297	60610443	T	ADD	110	6.674	4.434	9.234e-06
VEGFA	12	rs2946834	102394036	A	ADD	109	0.1493	-4.495	6.956e-06

Se identificaron cuatro pQTLs significativos localizados en regiones con una alta densidad de variantes en LD alto. Estos pQTLs corresponden a rs17237297 en el cromosoma 15 para la proteína HGF, rs6495562 en el cromosoma 15 para CXCL11, rs1735141 en el cromosoma 21 para FGF23 y rs6871090 en el cromosoma 5 para FGF23. La concentración de SNPs en LD alto en estas regiones indica una influencia en la regulación de la expresión de estas proteínas (Figure 5).

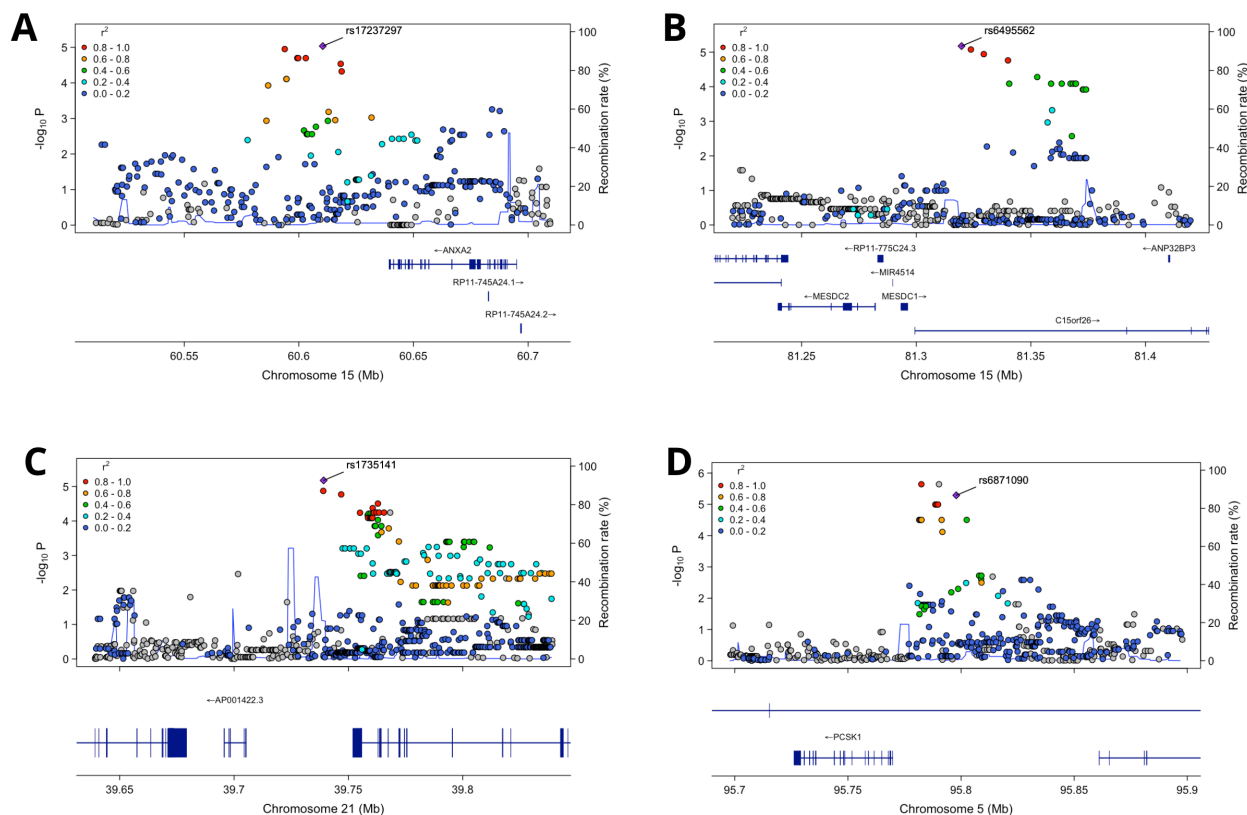


Figure 5: **Los Locus Plots muestran los SNPs significativos encontrados junto con el disequilibrio de ligamiento.** (A) rs17237297 para la proteína HGF, ubicado en el cromosoma 15. (B) rs6495562 para la proteína CXCL11, ubicado en el cromosoma 15. (C) rs1735141 para la proteína FGF23, ubicado en el cromosoma 21. (D) rs6871090 para la proteína FGF23, ubicado en el cromosoma 5. Todos los SNPs se encuentran en una zona de alto LD.

3.4. Inflamación y asociaciones:

El análisis de los SNPs mediante la consulta con rsnp permitió identificar los genes asociados a cada variante. (Table 4).

Table 4: **Relación de Snp con Genes descritos en la proteína con posible asociación.**

Proteína	SNP	Gen
CD6	rs77069221	<i>FAT1</i>
CD6	rs11670764	<i>ZNF446/LOC112268251</i>
CXCL10	rs8090960	<i>LOC105371973</i>
CXCL11	rs1121619	<i>CSMD1</i>
CXCL11	rs6495562	<i>STARD5</i>
CXCL11	rs148236522	<i>STARD5/TMV3-AS1</i>
FGF23	rs9314154	<i>RHOBTB3</i>
FGF23	rs3756706	<i>RHOBTB3</i>
FGF23	rs6871090	<i>RHOBTB3</i>
FGF23	rs1735141	<i>IGSF5</i>
HGF	rs17237297	<i>RORA/ RORA-AS1</i>
VEGFA	rs2946834	<i>LINC02456</i>

El análisis de expresión diferencial de los genes exportados de rsSNPs (DEG) en distintos tejidos revela una regulación significativa en múltiples órganos, con una mayor representación en el sistema digestivo (intestino delgado, esófago, tracto gastrointestinal) y el sistema nervioso (corteza cerebral, cerebelo, tálamo). La significancia estadística, medida como $-\log_{10}(\text{p-value})$, indica una expresión diferencial más marcada en estos tejidos, destacando algunos con valores particularmente elevados. En la comparación global, se identificaron genes con expresión diferencial tanto en sentido positivo como negativo, lo que sugiere una posible implicación en procesos fisiológicos específicos o en la patogénesis de enfermedades relacionadas con estos órganos (Figure 6).

En el análisis de expresión genética, se observan patrones de expresión diferencial en distintos tejidos, con algunos genes mostrando niveles elevados en tejidos específicos, mientras que en otros la expresión es mínima o nula.

Entre los genes con mayor asociación, RORA y STARD5 presentan una expresión alta en *skin sun-exposed* y *skin not-exposed*. FAT1 muestra una mayor expresión en *cell cultured fibroblasts*, mientras que GLRX se expresa principalmente en *cells transformed lymphocytes*. Por otro lado, RHOBTB3 tiene una expresión destacada en *nerve tibial* y *ovary*, y SLC27A5 se expresa predominantemente en el *hígado*.

Por el contrario, CSMD1, RP11-219B17.1 y SLC27A5 son genes que no presentan casi expresión en el análisis (Figure 7).

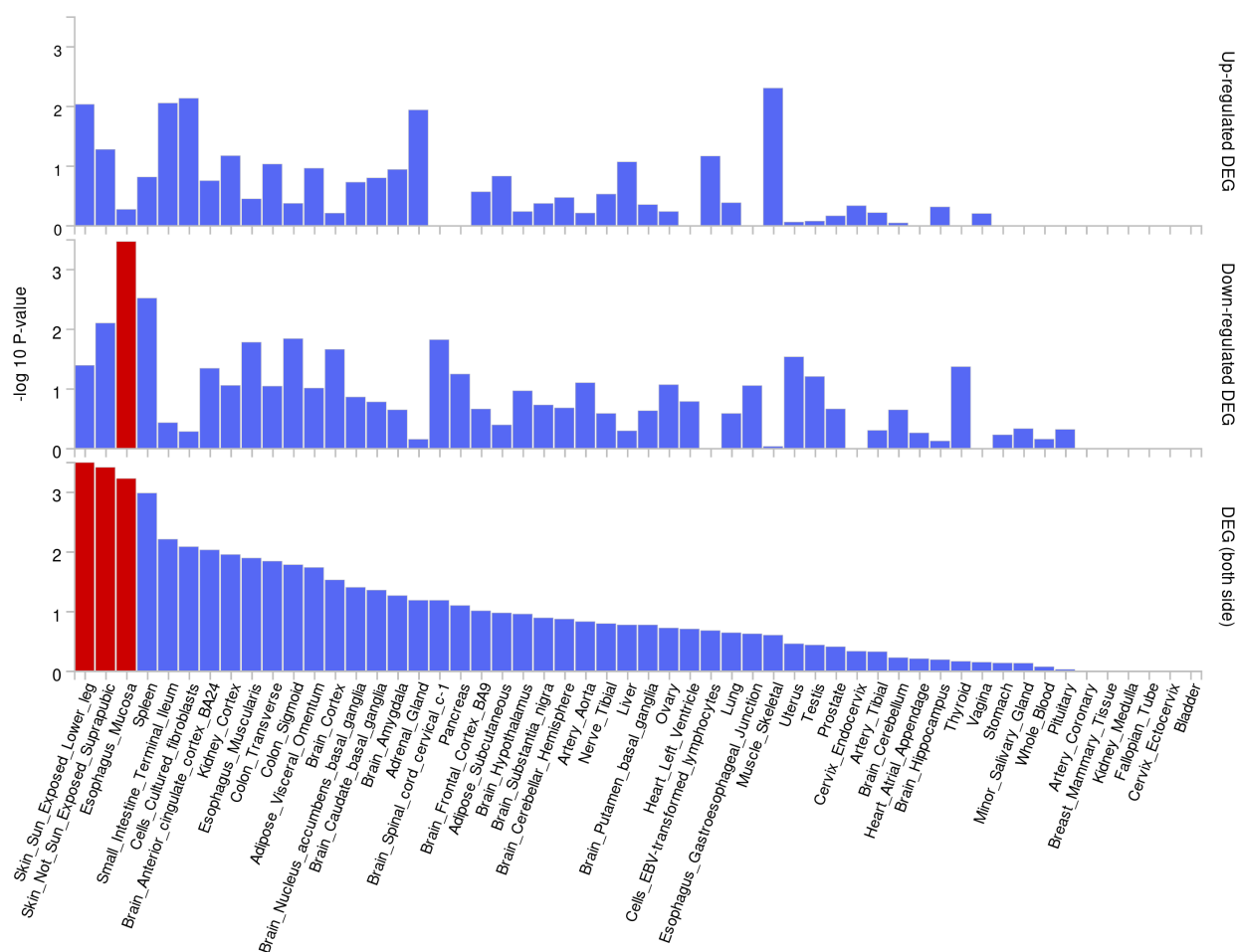


Figure 6: Asociación de los Genes diferencialmente expresados según la especificidad de 54 tejidos. Con los genes dados de rsSNPs. El análisis de expresión diferencial de genes asociados a rsSNPs muestra una regulación significativa en múltiples tejidos, con mayor asociación en el sistema digestivo y nervioso. La significancia estadística indica una expresión diferencial marcada en estos órganos, sugiriendo su posible rol en procesos fisiológicos o patológicos.

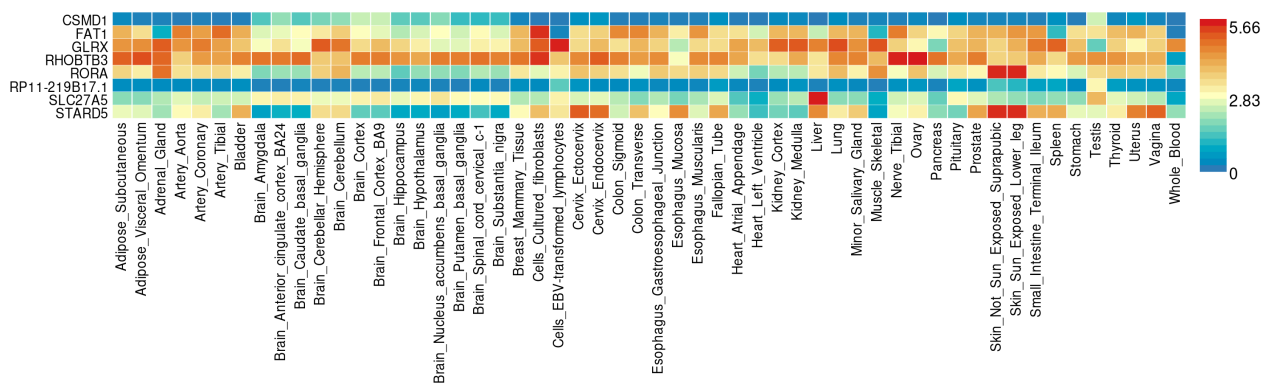


Figure 7: **Mapa de calor de expresión génica en 54 tejidos.** El gráfico representa la expresión diferencial de genes en diversos tejidos, donde el eje Y muestra los genes analizados y el eje X los tejidos. La escala de colores indica los niveles de expresión, con tonos azules representando menor expresión y tonos rojos mayor expresión.

La red de interacción de proteínas (Figure 8) muestra aquellas asociadas con inflamación (CXCL10, CXCL11, HGF y FGF23) y sus conexiones directas. Aun cuando CSMD1, FAT1, ZNF446, RHOBTB3, IGSF5, STARD5 y CD6 no aparecen directamente vinculadas en la red, los análisis genéticos realizados revelan que variantes específicas (SNPs) en sus genes podrían regular indirectamente estas proteínas inflamatorias, sugiriendo un posible efecto en los procesos de inflamación.

Las líneas verde claro indican asociaciones detectadas por minería de textos, las negras representan coexpresión, las violetas señalan homología proteica, las fucsias reflejan evidencia experimental y las celestes corresponden a asociaciones documentadas en bases de datos curadas.

Se destacan especialmente las interacciones de CXCL10 con HGF y CXCL11, ya que muestra coexpresión con ambas. Además, la homología entre CXCL10 y CXCL11 respalda la fiabilidad de su interacción.

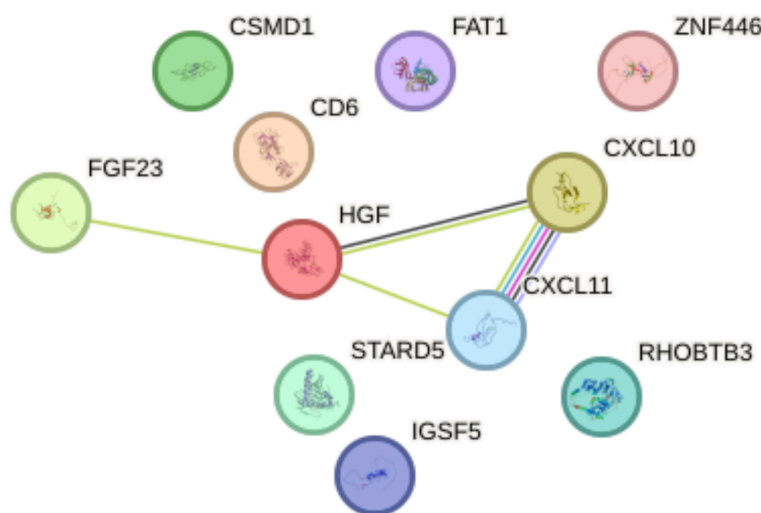


Figure 8: **Red de interacción de proteínas** aquellas asociadas con inflamación CXCL10, CXCL11, HGF y FGF23 y sus conexiones directas. Y las detectadas en las posiciones de los snps CSMD1, FAT1, ZNF446, RHOBTB3, IGSF5, STARD5 y CD6.

4. Discusión:

Se identificaron varias proteínas asociadas con la inflamación crónica en enfermedades inflamatorias intestinales (EII), lo que sugiere su participación en mecanismos patogénicos comunes. Entre ellas, TGF- α , MMP-10, CXCL9 e IL-17A fueron replicadas en ambos estudios analizados, reforzando su papel clave en la inflamación intestinal (Bourgonje et al., 2021; Andersson et al., 2017).

En la enfermedad de Crohn (EC), varias proteínas mostraron una alta consistencia entre estudios. FGF-19 y FGF-23, factores de crecimiento fibroblástico implicados en la regeneración epitelial, fueron detectados de manera recurrente. IFN-gamma, una citocina fundamental en la respuesta inmune adaptativa, se encontró consistentemente elevada. Asimismo, CCL11, un reclutador de eosinófilos, fue replicado en los tres estudios (Bourgonje et al., 2021; Andersson et al., 2017).

En el caso de la colitis ulcerativa (CU), se identificaron menos proteínas replicadas. MMP-10, además de su presencia en la EII en general, se encontró elevada de manera consistente en pacientes con CU en los dos estudios. CXCL11, una quimiocina clave en la migración de células inmunes, e IL-17A, previamente vinculada a la EII en general, también fueron replicadas en el contexto de la CU (Bourgonje et al., 2021; Andersson et al., 2017).

Adicionalmente, el análisis de expresión diferencial de los genes asociados a rsSNPs (DEG) en distintos tejidos reveló una regulación significativa en múltiples órganos, con mayor representación en el sistema digestivo (intestino delgado, esófago, tracto gastrointestinal) y el sistema nervioso (corteza cerebral, cerebelo, tálamo). Esto permite inferir que los genes identificados podrían estar implicados en la patogénesis de la enfermedad, interactuando también en otros tejidos.

La red de interacción de proteínas (Figure 8) aporta información adicional sobre los mecanismos reguladores. Aunque las proteínas inflamatorias como CXCL10, CXCL11, HGF y FGF23 muestran conexiones directas, los SNPs asociados a la regulación de estas proteínas se localizan en genes distintos, específicamente en *CSMD1*, *FAT1*, *ZNF446*, *RHOBTB3*, *IGSF5*, *STARD5* y *CD6*, lo que indica que actúan como pQTLs de tipo trans. Esto implica que, en lugar de estar codificados en los genes de las propias proteínas, dichos SNPs regulan indirectamente su expresión mediante mecanismos que pueden incluir la modulación de factores de transcripción, la estabilidad del ARNm o la participación en redes de señalización.

El hallazgo de pQTLs de tipo trans resalta la complejidad de la regulación genética en la EII y enfatiza la importancia de la integración de datos proteogenómicos para revelar los mecanismos patológicos subyacentes. Esto abre nuevas vías para el desarrollo de intervenciones terapéuticas personalizadas basadas en la modulación de este tipo de pQTLs.

No obstante, el estudio presenta algunas limitaciones, entre ellas el tamaño muestral (349 individuos), que podría afectar la generalización de los resultados, un posible sesgo poblacional y la necesidad de ajustar por variables clínicas adicionales. Asimismo, a pesar del uso de metodologías avanzadas como el *Proximity Extension Assay* (PEA) y la imputación con el panel TOPMed, es necesaria la validación de estos hallazgos en estudios futuros con cohortes más amplias y métodos complementarios.

5. Agradecimiento:

Agradezco profundamente a los doctores Guillermo Torres y Janina Dose, del Instituto Clínico de Biología Molecular de Kiel, por su invaluable orientación y apoyo a lo largo de este estudio. Asimismo, extendiendo mi gratitud a la profesora Maryam Chaib de Mares, de la Universidad Nacional de Colombia, por su acompañamiento y sus valiosos aportes durante mi formación.

De igual manera, agradezco a mi familia—mi madre, mis tías, mis hermanos—y a mis amigos, quienes estuvieron presentes a lo largo de mi pregrado y, en innumerables ocasiones, fueron el motor que me impulsó a seguir adelante.

6. Reproducibilidad y Documentación:

Todo el flujo de análisis se realizó mediante scripts en R versión 4.4.2. Los conjuntos de datos procesados y scripts están disponibles en Github (<https://github.com/fsalamancar>).

7. Referencias:

Olink Target 96 — Olink®. (s. f.). Olink®. <https://olink.com/products/olink-target-96>

Colombel JF, Narula N, Peyrin-Biroulet L. Management strategies to improve outcomes of patients with inflammatory bowel diseases. *Gastroenterology* 2017;152:351–61.e5.

Rausch, P. et al. (2024). First Insights into Microbial Changes within an Inflammatory Bowel Disease Family Cohort Study. *medRxiv*. <https://doi.org/10.1101/2024.07.23.24310327>

de Lange KM, Moutsianas L, Lee JC, et al. Genome-wide association study implicates immune activation of multiple integrin genes in inflammatory bowel disease. *Nat Genet* 2017;49:256-61

Liu JZ, van Sommeren S, Huang H, et al.; International Multiple Sclerosis Genetics Consortium; International IBD Genetics Consortium. Association analyses identify 38 susceptibility loci for inflammatory bowel disease and highlight shared genetic risk across populations. *Nat Genet* 2015;47:979-86

Sun BB, Maranville JC, Peters JE, et al. Genomic atlas of the human plasma proteome. *Nature* 2018;558:73-9.

Zhernakova DV, Le TH, Kurilshikov A, et al.; LifeLines cohort study; BIOS consortium. Individual variations in cardiovascular-disease-related protein levels are driven by genetics and gut microbiome. *Nat Genet* 2018;50:1524-32.

Bourgonje AR, von Martels JZH, Gabriëls RY, et al. A combined set of four serum inflammatory biomarkers reliably predicts endoscopic disease activity in inflammatory bowel disease. *Front Med (Lausanne)* 2019;6:251.

Folkersen L, Gustafsson S, Wang Q, et al. Genomic and drug target evaluation of 90 cardiovascular proteins in 30,931 individuals. *Nat Metab* 2020;2:1135-48.

Taliun, D., Harris, D. N., Kessler, M. D., Carlson, J., Szpiech, Z. A., Torres, R., Taliun, S. A. G., Corvelo, A., Gogarten, S. M., Kang, H. M., Pitsillides, A. N., LeFaive, J., Lee, S., Tian, X., Browning, B. L., Das, S., Emde, A., Clarke, W. E., Loesch, D. P., . . . Abecasis, G. R. (2021). Sequencing of 53,831 diverse genomes from the NHLBI TOPMed Program. *Nature*, 590(7845), 290-299. <https://doi.org/10.1038/s41586-021-03205-y>

Das, S., Forer, L., Schönherr, S., Sidore, C., Locke, A. E., Kwong, A., Vrieze, S. I., Chew, E. Y., Levy, S., McGue, M., Schlessinger, D., Stambolian, D., Loh, P., Iacono, W. G., Swaroop, A., Scott, L. J., Cucca, F., Kronenberg, F., Boehnke, M., . . . Fuchsberger, C. (2016). Next-generation genotype imputation service and methods. *Nature Genetics*, 48(10), 1284-1287. <https://doi.org/10.1038/ng.3656>

Kässens, J. C., Wienbrandt, L., & Ellinghaus, D. (2021). BIGwas: Single-command quality control and association testing for multi-cohort and biobank-scale GWAS/PheWAS data. *GigaScience*, 10(6). <https://doi.org/10.1093/gigascience/giab047>

Auton, A., Abecasis, G. R., Altshuler, D. M., Durbin, R. M., Abecasis, G. R., Bentley, D. R., Chakravarti, A., Clark, A. G., Donnelly, P., Eichler, E. E., Flicek, P., Gabriel, S. B., Gibbs, R. A., Green, E. D., Hurles, M. E., Knoppers, B. M., Korbel, J. O., Lander, E. S., Lee, C., . . . Abecasis, G. R. (2015). A global reference for human genetic variation. *Nature*, 526(7571), 68-74. <https://doi.org/10.1038/nature15393>

Sherry, S. T. (2001). dbSNP: the NCBI database of genetic variation. *Nucleic Acids Research*, 29(1), 308-311. <https://doi.org/10.1093/nar/29.1.308>

“Illumina, Inc.”. Infinium Global Screening Array v1.0 (GSA v1.0) User Guide. Illumina, 2017. <https://emea.support.illumina.com/global-screening-array-v1-0-support-files.html>

Turner, S. D. (2018). qqman: An R package for visualizing GWAS results using Q-Q and Manhattan plots. *Journal of Open Source Software*, 3(25), 731. <https://doi.org/10.21105/joss.00731>

R Core Team. (2024). R: A language and environment for statistical computing. R Foundation for Statistical Computing. <https://www.R-project.org/>

Heinz, S., Benner, C., Spann, N., Bertolino, E., Lin, Y. C., Laslo, P., Cheng, J. X., Murre, C., Singh, H., & Glass, C. K. (2017). Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. *Molecular Cell*, 38(4), 576-589. <https://doi.org/10.1016/j.molcel.2017.08.011>

Rausch, P., Ratjen, I., Tittmann, L., Enderle, J., Wacker, E. M., Jaeger, K., Ruehlemann, M. C., Ellul, P., Kruse, R., Halfvarsson, J., Roggenbuck, D., Ellinghaus, D., Jacobs, G., Krawczak, M., Schreiber, S., Bang, C., Lieb,

- W., & Franke, A. (2024). First Insights into microbial changes within an Inflammatory Bowel Disease Family Cohort study. medRxiv (Cold Spring Harbor Laboratory). <https://doi.org/10.1101/2024.07.23.24310327>
- Rüeger, S., & Gustavsen, J. (2022). rsnp 0.5.0: New ncbi_snp_query() Features. En Front Matter. <https://doi.org/10.59350/b9s6j-hmt03>
- Watanabe, K., Taskesen, E., van Bochoven, A., & Posthuma, D. (2017). FUMA: Functional mapping and annotation of genetic associations. Nature Communications, 8, 1826. <https://doi.org/10.1038/s41467-017-01261-5>
- Szklarczyk, D., Gable, A. L., Nastou, K. C., Lyon, D., Kirsch, R., Pyysalo, S., ... & Jensen, L. J. (2021). The STRING database in 2021: Customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. Nucleic Acids Research, 49(D1), D605–D612. <https://doi.org/10.1093/nar/gkaa1074>
- Chang, C. C., Chow, C. C., Tellier, L. C. A. M., Vattikuti, S., Purcell, S. M., & Lee, J. J. (2015). Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience, 4(1), 7. <https://doi.org/10.1186/s13742-015-0047-8>
- Auton, A., Abecasis, G. R., Altshuler, D. M., Durbin, R. M., Abecasis, G. R., Bentley, D. R., Chakravarti, A., Clark, A. G., Donnelly, P., Eichler, E. E., Flicek, P., Gabriel, S. B., Gibbs, R. A., Green, E. D., Hurles, M. E., Knoppers, B. M., Korbel, J. O., Lander, E. S., Lee, C., . . . Abecasis, G. R. (2015b). A global reference for human genetic variation. Nature, 526(7571), 68-74. <https://doi.org/10.1038/nature15393>
- Machiela, M. J., & Chanock, S. J. (2015). LDlink: a web-based application for exploring population-specific haplotype structure and linking correlated alleles of possible functional variants. Bioinformatics, 31(21), 3555-3557. <https://doi.org/10.1093/bioinformatics/btv402>
- Andersson, E., Bergemalm, D., Kruse, R., Neumann, G., D'Amato, M., Repsilber, D., & Halfvarson, J. (2017). Subphenotypes of inflammatory bowel disease are characterized by specific serum protein profiles. PLoS ONE, 12(10), e0186142. <https://doi.org/10.1371/journal.pone.0186142>
- Bourgonje, A. R., Hu, S., Spekhorst, L. M., Zhernakova, D. V., Vila, A. V., Li, Y., Voskuil, M. D., Van Berkel, L. A., Folly, B. B., Charroux, M., Mahfouz, A., Reinders, M. J. T., Van Heck, J. I. P., Joosten, L. A. B., Visschedijk, M. C., Van Dullemen, H. M., Faber, K. N., Samsom, J. N., Festen, E. A. M., . . . Weersma, R. K. (2021). The Effect of Phenotype and Genotype on the Plasma Proteome in Patients with Inflammatory Bowel Disease. Journal Of Crohn S And Colitis, 16(3), 414-429. <https://doi.org/10.1093/ecco-jcc/jjab157>