Cambio_identificadores_LM22

Clara Podaru

2025-02-11

CAMBIO DE IDENTIFICADORES EN LA MATRIZ DE REFERENCIA LM22 DE GENE SYMBOL A ENSEMBL ID

Se ha querido procesar la matriz de referencia con los 22 tipos de células inmunes mediante la sustitución de sus Gene symbols con su correspondiente Ensembl Gene IDs. Este cambio se ha llevado a cabo con las siguientes librerias: bioMart, HGNChelper, dplyr.

El flujo de trabajo consiste en tres pasos generales:

-Extraer los Ensembl Gene IDs de los genes encontrados en LM22.txt empleando bioMart.

-Manejar los genes sin un Ensembl ID (debido a que tienen gene symbols desactualizados).

-Sustituir los gene symbols de LM22 con sus Ensembl IDs, asegurándose de que hay 547 genes y que eso no varía.

El primer paso consiste en cargar los paquetes necesarios y configurar el directorio de trabajo.

Antes de comenzar a trabajar, se deben cargar los paquetes de R necesarios y configurar el directorio en el que queremos trabajar.

```
# Definir el directorio de trabajo
workingD <- rstudioapi::getActiveDocumentContext()$path
setwd(dirname(workingD))
```

```
# Cargar paquetes necesarios
library(HGNChelper) #permite detectar los gene symbols desactualizados
```

```
## Please cite our software :)
##
##  Sehyun Oh et al. HGNChelper: identification and correction of invalid gene symbols for human and mouse. F1000 Research 2020, 9:1493. DOI: https://doi.org/10.12688/f1000research.28033.1
##
##  Type 'citation("HGNChelper")' for a BibTeX entry.
```

```
library(biomaRt) #busca los Ensembl IDs de los genes en la base de datos
library(dplyr) #ayuda a la manipulación de los datos
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:biomaRt':
##
##     select
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

1.Cargar datos y conectarse a Ensembl

A la hora de cargar los datos, en la fichero_path, es necesario introducir la ruta personal donde esté el archivo de cada uno.

```
# Cargar datos
fichero_path <- "/Users/clarapodaru/Documents/cibersortx/data/input/original/LM22.txt"
data <- read.table(fichero_path, header = TRUE, sep = "\t", stringsAsFactors = FALSE)
```

Nos conectamos a bioMArt para buscar los Ensembl Gene IDs en la base de datos de bioMart. Una vez nos hemos conectado a bioMart, debemos extraer los Ensembl IDs. Extraemos de la base de de datos de bioMArt los Ensembl IDs de los genes que coincidan con el gene symbol.

```
# Conectarse a la base de datos de Ensembl
ensembl <- useMart("ensembl", dataset = "hsapiens_gene_ensembl")
```

2.Obtener los Ensembl IDs

```
# Extraer los Ensembl IDs para cada Gene Symbol
gene_symbols <- data$Gene.symbol

tabla_de_conversiones <- getBM(
  attributes = c("hgnc_symbol", "ensembl_gene_id", "chromosome_name"),
  filters = "hgnc_symbol",
  values = gene_symbols,
  mart = ensembl)
```

Hay algunos genes que se encuetran en un scaffold, y eso no nos interesa. Nos queremos quedar únicamente con los genes localizados en los cromosomas reales. Por lo tanto, con este paso nos aseguramos de eso mismo.

```
# Filtrar los cromosomas reales (1-22, X)
tabla_de_conversiones_2 <- tabla_de_conversiones[tabla_de_conversiones$chromosome_name %in% c(as.character(1:22),
"X"), ]
```

Al buscar en la base de datos de bioMart, hay algunos genes que tienen asignado más de un Esembl ID. Por lo tanto, tenemos que manejar esos duplicados y quedarnos con una sola entrada por gen.

Cuando se consulta Ensembl usando la biomaRt, se extraen a veces más de un Ensembl ID por Gene symbol. Esto puede ocurrir porque un gene puede tener múltiples variantes de trascripción, Ensembl puede guardar diferentes mapeos para otras referencias o algunos genes existen en diferentes contextos genómicos, lo cual lleva a la creación de múltiples entradas.

```
# Detectar duplicados
dups <- tabla_de_conversiones_2[duplicated(tabla_de_conversiones_2$hgnc_symbol), ]
dups
```

```
##     hgnc_symbol ensembl_gene_id chromosome_name
## 249     GUSBP11 ENSG00000228315              22
```

```
# Eliminar duplicados con menos transcritos (gen GUSBP11)
discard_dups <- c("ENSG00000272578")
tabla_data_ensembl_cleaned <- tabla_de_conversiones_2[!(tabla_de_conversiones_2$ensembl_gene_id %in% discard_dup
s), ]
```

El gen GUSBP11 es el que tenía el dos Ensembl ID ya que, posee dos entradas en Esembl. Al buscar este gen en la base de datos de Ensembl, nos datos cuenta de que hay dos entradas con los siguientes Esembl IDs:

- ENSG00000228315: esta entrada del gen tiene 289 transcritos. https://www.ensembl.org/Homo_sapiens/Gene/Summary?db=core;g=ENSG00000228315;r=22:23630618-23717423
- ENSG00000272578: esta entrada del gen tiene 1 trascrito. https://www.ensembl.org/Homo_sapiens/Gene/Summary?db=core;g=ENSG00000272578;r=22:23659869-23717295;t=ENST00000435868

Por lo tanto, se ha escogido ENSG00000228315 porque tiene más trascritos.

Y se ha vuelto a actualizar la tabla de conversiones que contiene los Esembl IDs.

3. Fusionar LM22 con los Ensembl IDs obtenidos

A continuación, se sustituyen los Ensembl IDs detectados hasta ahora en el archivo original 'LM22.txt' que está almacenado en la variable 'data'. Esta sustitución se hace fusionando la variable 'data' con 'tabla_data_ensembl_cleaned' (que contiene los Ensembl IDs detectados). La variable 'tabla_data_ensembl_cleaned' tiene 3 columnas (hgnc_symbol, ensembl_gene_id, chromosome_name) y los genes de la columna 'hgnc_symbol' que coincidan con la columna 'Gene symbol' de 'data', se sustituyen por el Esembl ID. Y despues, se descarta la columna extra 'ensembl_gene_id'.

Los cambios se van a almacenar en la variable 'lm22_actualizado' y en un archivo de texto denominado 'LM22_actualizado.txt' .

```
# Fusionar 'LM22.txt' con 'tabla_data_ensembl_cleaned'
lm22_actualizado <- data %>%
  left_join(tabla_data_ensembl_cleaned %>% select(hgnc_symbol, ensembl_gene_id),
            by = c("Gene.symbol" = "hgnc_symbol")) %>%
  mutate(Gene.symbol = coalesce(ensembl_gene_id, Gene.symbol)) %>%  #si hay Ensembl ID, reemplasarlo
  select(-ensembl_gene_id)  #Descartar la columna extra

# Guardar 'LM22_actualizado.txt' con los primeros Ensembl IDs detectados (opcional)
write.table(lm22_actualizado, "LM22_actualizado.txt", sep = "\t", row.names = FALSE, quote = FALSE)
```

4. Detectar genes sin Ensembl ID

Una vez que se ha actualizado el archivo 'LM22' a 'LM22_actualizado', comprobamos qué Gene symbols faltan por sustituir por su Ensembl ID. Comparamos la columna 'Gene symbol' de 'data' con la columna 'hgnc_symbol' de 'tabla_de_conversiones_2' y los genes que no de esta columna que no coincidan con la de 'Gene symbol' de otro archivo se van a ir guardando dentro de la variable 'genes_sin_ensembl' y exportados a un archivo denominado 'genes_sin_ensembl_id.txt'.

```
# Buscar genes sin Ensembl ID
genes_sin_ensembl <- data$Gene.symbol[!data$Gene.symbol %in% tabla_de_conversiones_2$hgnc_symbol]
genes_sin_ensembl #imprimir genes
```

```
##  [1] "ATHL1"       "C1lorf80"    "CXorf57"     "EMR1"        "EMR2"
##  [6] "EMR3"        "FAIM3"       "FAM159B"     "FAM212B"     "FAM659"
## [11] "FLJ13197"    "GPR1"        "GPR97"       "GSTT1"       "HIST1H2AE"
## [16] "HIST1H2BG"   "KIAA0226L"   "KIAA0754"    "KIR3EL"      "LILRA3"
## [21] "LINC00597"   "LOC100130100" "LOC126987"   "LRMP"        "NARC8J"
## [26] "SEPT5"       "SEPT8"       "VSNL1"
```

```
# Guardar los genes sin esembl en un archivo denominado "genes_sin_ensembl" (opcional)
write.table(genes_sin_ensembl, "genes_sin_ensembl_id.txt", sep = "\t", row.names = FALSE, quote = FALSE)
```

5. Corregir nombres desactualizados con HGNC

Se emplea el paquete HGNChelper para buscar los gene symbols que se han quedado obsoletos o están incorrectos. Se extrae los nombre alternativos/actualizados y sus Ensembl IDs. Y se guardan los genes problemáticos con nombres de HGNC que todavía no son oficiales.

```
# Revisar gene symbols desactualizados
corrected_genes <- checkGeneSymbols(genes_sin_ensembl, unmapped.as.na = FALSE)
```

```
## Maps last updated on: Sat Nov 16 10:35:32 2024
```

```
## Warning in checkGeneSymbols(genes_sin_ensembl, unmapped.as.na = FALSE): x
## contains non-approved gene symbols
```

```
# Genes sin sustituto en HGNC
non_approved_symbols <- corrected_genes[!corrected_genes$Approved, ]
write.table(non_approved_symbols, "non_approved_symbols.txt", sep = "\t", row.names = FALSE, quote = FALSE)

# Genes aprobados pero sin Ensembl ID
problematic_genes <- corrected_genes[corrected_genes$Approved, ]
colnames(problematic_genes) <- c('gene_symbol', 'status', 'suggestion')

# Obtener Ensembl IDs para genes corregidos
tabla_de_conversiones_3 <- getBM(
  attributes = c("hgnc_symbol", "ensembl_gene_id", "chromosome_name"),
  filters = "hgnc_symbol",
  values = non_approved_symbols$Suggested.Symbol,
  mart = ensembl)

# Filtrar cromosomas reales
tabla_de_conversiones_4 <- tabla_de_conversiones_3[tabla_de_conversiones_3$chromosome_name %in% c(as.character(1:
22), "X"), ]

# Fusionar non_approved_symbols con Ensembl IDs corregidos
non_approved_symbols_merged <- merge(corrected_genes, tabla_de_conversiones_4, by.x = "Suggested.Symbol", by.y =
"hgnc_symbol", all.x = TRUE)
```

6. Fusionar los 28 genes problemáticos con LM22

Hay que realizar algunas anotaciones manuales de los siguientes genes ya que no se han detectado automáticamente. Por lo tanto, creamos un data frame con dos columnas:

- Suggested.Symbol: donde las entradas son los genes en formato Gene symbol.
- ensembl_gene_id: donde las entradas son los genes en formato Ensembl ID

A LOC100120100 no le asignamos ningún Ensembl ID ya que tras realizar una ardua búsqueda no se ha localizado dicho gen en niguna base de datos.

```
# Anotaciones manuales
new_df <- data.frame(Suggested.Symbol = c("LOC126987", "LOC100130100", "LINC00597", "LILRA3", "GSTT1"),
                     ensembl_gene_id = c("ENSG00000237480", "LOC100130100", "ENSG00000173517", "ENSG00000104974",
"ENSG00000277656"))

# Fusionar datos corregidos con anotaciones manuales
data_merged <- non_approved_symbols_merged %>%
  left_join(new_df %>% select(Suggested.Symbol, ensembl_gene_id), by = "Suggested.Symbol") %>%
  mutate(ensembl_gene_id = coalesce(ensembl_gene_id.x, ensembl_gene_id.y)) %>%
  select(-ensembl_gene_id.x, -ensembl_gene_id.y)
```

7. Aplicar las correcciones finales a LM22

Una vez se han actualizado los datos nos aseguramos de que los nombres de las dos variables coincidan para que no haya errores. Después, se reemplazan los genes de la columna 'Gene.symbol' de 'lm22_actualizado' por su Ensembl ID, en caso de que se encuentre en 'data_merged'.

Se crea una variable índice que se encarga de buscar coincidencias entre la columna 'Gene.symbol' de lm22_actualizado y la columna 'x' de 'data_merged'. Cuando se encuentren coincidencias, se reemplazará el Gene symbol por su Ensembl ID.

```
# Asegurarse de que los nombres coinciden antes de actualizar los datos
lm22_actualizado$Gene.symbol <- trimws(toupper(lm22_actualizado$Gene.symbol))
data_merged$Suggested.Symbol <- trimws(toupper(data_merged$x))

# Reemplazar los gene.symbol en LM22_actualizado con su ensembl id si está en data_merged
indice <- match(lm22_actualizado$Gene.symbol, data_merged$x)  #buscar coincidencias en 'x'
lm22_actualizado$Gene.symbol[!is.na(indice)] <- data_merged$ensembl_gene_id[indice[!is.na(indice)]]  # Reemplazar
solo donde hay coincidencia
```

8. Almacenar el archivo final con los 547 genes

Antes de generar el archivo de texto final, abrimos 'lm22_actualizado' para asegurarnos de que todos los genes se encuentran en formato Ensembl ID y, en caso de que falte alguno por actualizar, se realizará de forma manualmente directamente una vez creemos el archivo de texto. Debemos buscar dichos genes en el UCSC Genome Browser y buscar su Ensembl ID.

Se han detectado los siguientes genes sin convertir:

- CXORF57: al buscarlo el gen aparece con su nombre actualizado, RADX = ENSG00000147231. https://genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&lastVirtModeType=default&lastVirtModeExtraState=&virtModeType=default&virtMode=0&nonVirtPosition=&position=chrX%3A106611978%2D106679438&hgsid=2451272961_XTFFpSejqmQZNph84u5CiCrv6F9V
- C11ORF80: ENSG00000173715. https://genome.ucsc.edu/cgi-bin/hgTracks?db=hg38&lastVirtModeType=default&lastVirtModeExtraState=&virtModeType=default&virtMode=0&nonVirtPosition=&position=chr11%3A66744736%2D66843515&hgsid=2451273455_7mQjJpIlHy0Radqkw5h2jqK7eVVDD

Una vez tenemos los datos finales nos aseguramos de tener 547 genes en total y almacenamos los datos en un archivo llamado LM22_definitivo.txt.

Abrimos el archivo creado y actualizamos manualmente los dos genes identificados anteriormente.

```
# Verificar resultado final
print(length(unique(lm22_actualizado$Gene.symbol)))  # 547 genes
```

```
## [1] 547
```

```
# Guardar 'LM22_final.txt' con todos los Ensembl IDs corregidos
write.table(lm22_actualizado, "LM22_definitivo.txt", sep = "\t", row.names = FALSE, quote = FALSE)
```