PEC1

cserrano

2025-03-25

En este trabajo se realizará un anásis exploratorio de un estudio metabolómico, escogido desde la base de datos **Metabolomics Workbench**. Por otro lado, usaremos la librería **metabolomicsWorkbenchR** para extraer los datos directamente desde la web y **SummarizedExperiment** para manejar los datos de manera estructurada. Ambas librerías se encuentran en el repositorio **Bioconductor**. Para utilizar estas librerías, primero asegurarse de tener instalado Bioconductor y luego instalar metabolomicsWorkbenchR y SummarizedExperiment como indica en sus respectivos links.

En esta sección, importamos las librerías necesarias que utilizaremos a lo largo de este trabajo

```
library(metabolomicsWorkbenchR)
library(SummarizedExperiment)
library(knitr)
library(dplyr)
library(tidyr)
library(ggplot2)
library(ComplexHeatmap)
library(stringr)
library(tibble)
```

La librería **metabolomicsWorkbenchR** permite realizar consultas a directamente a su página web. Aquí se utiliza la función do_query para solicitar información de estudios, precisamente con la palabra Bacterial.

```
estudios = do_query(
    context = 'study',
    input_item = 'study_title',
    input_value = 'Bacterial',
    output_item = 'summary'
)
```

El resultado fue almacenado en la variable estudios, la cual se puede visualizar como una tabla. Aquí se muestran los primeros resultados de la consulta.

```
kable(head(estudios))
```

study_id	study_title	species	institute	analysis_type	number_of_samples	study_type	department	last_name	first_name	email	phone	submit_date	study_summary	subject_species
ST003710	(p)ppGpp and DksA play crucial role in reducing the efficacy of -lactam antibiotics by modulating bacterial membrane permeability	Vibrio cholerae	Translational Health Science And Technology Institute (THSTI)	LC-MS	72	NA	NA	NA	NA	NA	NA	NA	NA	NA
ST003324	Metabolic Profiling Unveils Enhanced Antibacterial Synergy of Polymyxin B and Teixobactin against Multi-Drug Resistant Acinetobacter baumannii	Acinetobacter bau-mannii	Monash University	LC-MS	58	NA	NA	NA	NA	NA	NA	NA	NA	NA
ST003389	Disruption of Glucose Homeostasis by Bacterial Infection Orchestrates Host Innate Immunity Through NAD+/NADH Balance	Mus mus-culus	Northwest A&F University	LC-MS	24	NA	NA	NA	NA	NA	NA	NA	NA	NA

study_id	study_title	species	institute	analysis_type	number_of_samples	study_type	department	last_name	first_name	email	phone	submit_date	study_summary	subject_species
ST003187	Untargeted plasma metabolomics on bacterial culture supernatants	Escherichia coli	Harvard University	LC-MS	7	NA	NA	NA	NA	NA	NA	NA	NA	NA
ST002973	Examine the through-filter recovery of metabolites extracted from a complex bacterial medium		Duke University	LC-MS	20	NA	NA	NA	NA	NA	NA	NA	NA	NA
ST002932	Metabolomic Characteristics of Nontuberculous Mycobacterial Pulmonary Disease	Homo sapiens	Seoul National University College of Medicine and Hospital	LC-MS	418	NA	NA	NA	NA	NA	NA	NA	NA	NA

Teniendo estas alternativas de interés, se decidió seleccionar el estudio ST003521. La justificación de este estudio es mi relación al estudio de microbiología y genes de resistencia. Además, este estudio pudo ser obtenido por los métodos que aquí se muestran y asociarlo con una publicación para mayor información (Se pobraron otros estudios con la misma metodología y varios fallaban). Para mayor información, se puede visitar directamente el estudio **ST003521** en su página web.

A continuación se procede a obtener los datos del estudio. Primero se obtendrá el summary del estudio. En la función do_query, se especifíca el estudio que al que se quiere acceder, mientras que en output_item se indica "summary"

Summary de estudio seleccionado

```
summary <- do_query(
    context = "study",
    input_item = "study_id",
    input_value = "ST003521",
    output_item = "summary")
```

Dentro de los campos disponibles en summary se encuentran: "study_id", "study_title", "species", "institute", "analysis_type", "number_of_samples", "study_type" "department", "last_name", "first_name", "email", "phone", "submit_date", "study_summary", "subject_species". En este caso en particular, solo están disponible los siguientes campos.

```
cat("ID de estudio: ", summary$study_id,"\n")
```

```
## ID de estudio:  ST003521
```

```
cat("Nombre de estudio: ", summary$study_title,"\n")
```

```
## Nombre de estudio:  Metabolic Profiling Unveils Enhanced Antibacterial Synergy of Polymyxin B and Te
```

```
cat("Especies de estudio: ", summary$species,"\n")
```

```
## Especies de estudio:  Acinetobacter baumannii
```

```
cat("Instituto del estudio: ", summary$institute,"\n")
```

```
## Instituto del estudio:  Monash University
```

```
cat("Tipo de análisis: ", summary$analysis_type,"\n")
```

```
## Tipo de análisis:  LC-MS
```

```
cat("Número de muestras: ", summary$number_of_samples,"\n")
```

```
## Número de muestras:  58
```

A continuación se obtienen los datos del estudio. Para ello se indica su ID, y en output_item se indicará salida de tipo "SummarizedExperiment". De esta manera, los datos se obtendrán en formato del objeto **SummarizedExperiment**, el cuál permite contener mucha información del estudio de manera estructurada dentro del mismo objeto.

En este caso, al obtener los datos, se obtiene una lista con 2 elementos SummarizedExperiment.Estos se almacenarán en variable llamada se_list

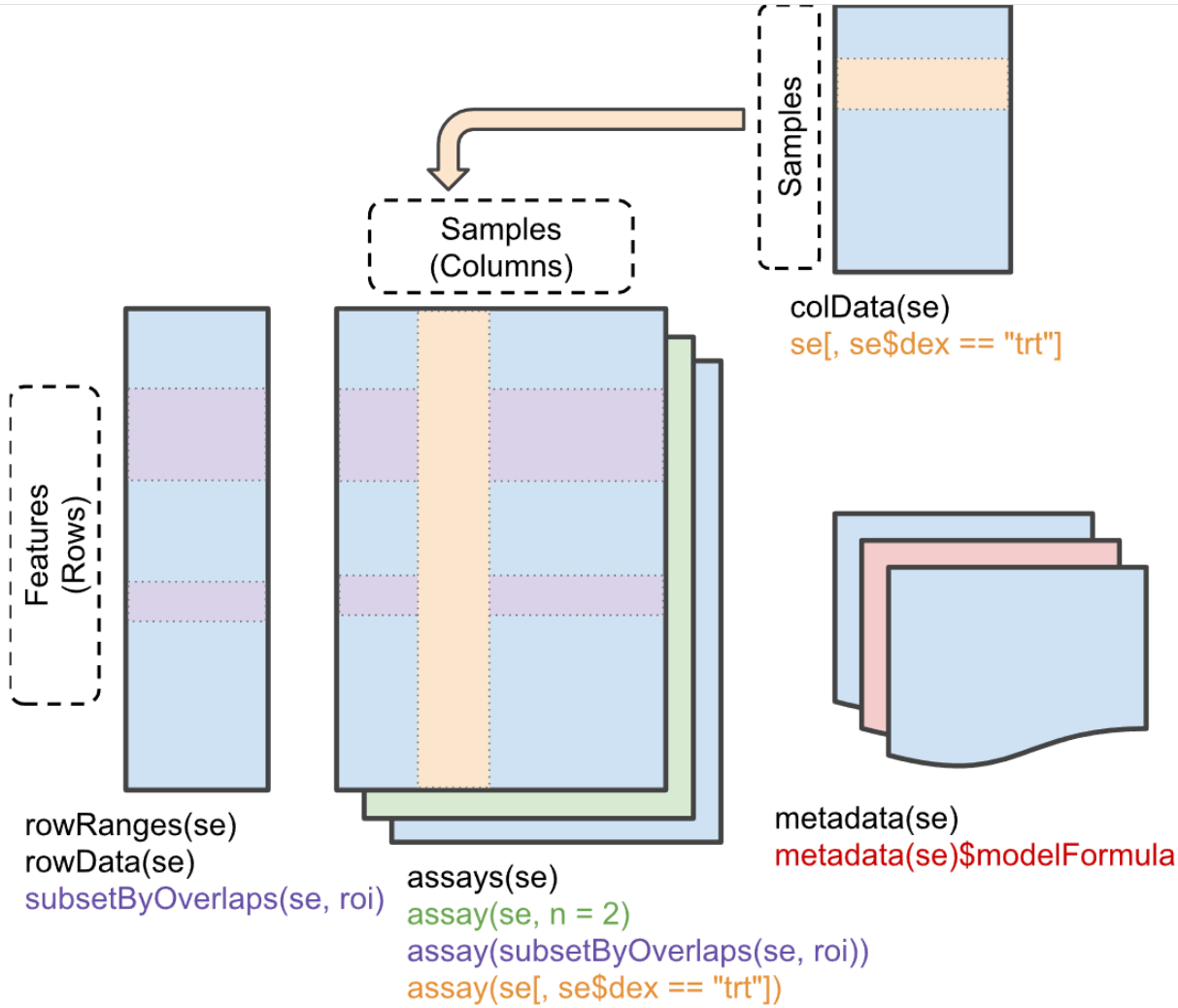
Carga de Datos en clase SummarizedExperiment

```
se_list = do_query(
    context = 'study',
    input_item = 'study_id',
    input_value = 'ST003521',
    output_item = 'SummarizedExperiment'
)
```

```
n=1
for (se in se_list)
{cat('Clase de elemento', n ,' : ',class(se), '\n')
n = n+1}
```

```
## Clase de elemento 1  :  SummarizedExperiment
## Clase de elemento 2  :  SummarizedExperiment
```

Exploraremos cada una de las matrices que contiene el objeto SummarizedExperiment. A continuación, una imagen representativa obtenida directamente desde la **documentación de la librería**.



Observaremos la matriz 'assay'. Lo haremos con el primer elemento de los datos, pero se pueden realizar las mismas acciones con ambos. Para ello utilizamos la funcion assay() y tomamos un subset para visualizar los primeros resultados tanto de filas como de columnas. Observamos también las dimensiones del dataset, donde obtenemos que este objeto consta de 513 filas y 58 columnas.

```
se1 <- se_list$AN005782
se2 <- se_list$AN005783
cat('Dimensiones:\n Filas: ',dim(assay(se1))[1], '\n', 'Columnas: ', dim(assay(se1))[2])
```

```
## Dimensiones:
##  Filas:  513
##  Columnas:  58
```

```
kable(assay(se1)[1:10,5:10])
```

Blank_5	COM_1h_1	COM_1h_2	COM_1h_3	COM_1h_4	COM_3h_1
NA	280668	179652	152334	186482	452100
NA	49312	52133	54610	16889	86925
NA	22367	1054362	127405	7622910	1115624
NA	12447	NA	9947	NA	15240
NA	92754	151013	108064	177670	138774
NA	159694	222957	246475	178822	208951
NA	157407	118697	98355	57821	64563
48040	88726	179481	283192	357103	151205

Blank_5	COM_1h_1	COM_1h_2	COM_1h_3	COM_1h_4	COM_3h_1
NA	85908	60717	65437	37010	105943
NA	95089	85478	70942	41815	126359

Ahora observaremos específicamente la matriz de Features (o Rows). Accedemos a ellas mediante la función rowData(). Al igual que antes, solo se mostrarán los primeros resultados para evitar extender demasiado el documento.

Aquó observamos que las dimensiones de esta matriz es de 513 filas y 3 columnas. Aquí se muestran los valores de metabolitos denotados con ids pero también con su nombre químico. Las columnas que se encuentran son: "metabolite_name", "metabolite_id" y "refmet_name"

```
cat('Dimensiones:\n Filas: ',dim(rowData(se1))[1], '\n', 'Columnas: ', dim(rowData(se1))[2])
```

```
## Dimensiones:
##  Filas:  513
##  Columnas:  3
```

```
kable(rowData(se1)[1:10,])
```

	metabolite_name	metabolite_id	refmet_name
ME917515	10,11-dihydro-20-trihydroxy-leukotriene B4	ME917515	
ME917423	1-(14-methyl-pentadecanoyl)-2-(8-[3]-ladderane-octanyl)-sn-glycerol	ME917423	
ME917123	1,2-dihexadecanoyl-sn-glycero-3-phosphosulfocholine	ME917123	
ME917224	1,3,5-trimethoxybenzene	ME917224	1,3,5-Trimethoxybenzene
ME917120	1-4-beta-D-Glucan	ME917120	1,4-beta-D-Glucan
ME917031	1,6-anhydro-N-acetylmuramate	ME917031	
ME917246	1-(beta-D-Ribofuranosyl)-1,4-dihydronicotinamide	ME917246	1-(beta-D-Ribofuranosyl)-1,4-dihydronicotinamide
ME917307	1-carboxymethylpyridinium	ME917307	
ME917341	1-deoxyxylonojirimycin	ME917341	
ME917309	1-Hexadecanoyl-2-(9Z-octadecenoyl)-sn-glycero-3-cytidine-5'-diphosphate	ME917309	

Ahora revisamos la matriz Samples. Para ello accedemos con la función colData(). En este caso, las dimensiones son 58 filas y 7 columnas. Notar que ahora las filas corresponden a las columnas de la matriz assay. Las columnas de esta matriz corresponden a: "local_sample_id", "study_id", "sample_source", "mb_sample_id", "raw_data", "Sample_source", "raw_file_name".

Los nombres que se observan en assay corresponden a local_sample_id. Se asume que el último sufijo "_N" corresponde al número de replicado de la muestra. En sample_source se muestra el nombre más descriptivo de la muestra. En mb_sample_id se tienen códigos de especies bacterianas. Todas corresponden a Acinetobacter baumannii ATCC 19606.

```
cat('Dimensiones:\n Filas: ',dim(colData(se1))[1], '\n', 'Columnas: ', dim(colData(se1))[2])
```

```
## Dimensiones:
##  Filas:  58
##  Columnas:  7
```

```
kable(head(colData(se1)))
```

	local_sample_id	study_id	sample_source	mb_sample_id	raw_data	Sample_source	raw_file_name
Blank_1	Blank_1	ST003521	*BLANK*	SA386750		Extraction blank	Blank_1
Blank_2	Blank_2	ST003521	*BLANK*	SA386751		Extraction blank	Blank_2
Blank_3	Blank_3	ST003521	*BLANK*	SA386752		Extraction blank	Blank_3
Blank_4	Blank_4	ST003521	*BLANK*	SA386753		Extraction blank	Blank_4
Blank_5	Blank_5	ST003521	*BLANK*	SA386754		Extraction blank	Blank_5
COM_1h_1	COM_1h_1	ST003521	Combination_1h	SA386726		Combination_1h	COM_1h_1

Por último, exploramos la metadata del estudio. Para ello usamos la función metadata(). Esto nos entrega una lista con información. Aquí observamos la base de datos de donde se obtuvieron los datos, el id del

estudio, el id del análsis (Este estudio tiene 2 análisis), el tipo de análsis (HILIC POSITIVE ION MODE corresponde a un modo en HPLC. Segundo análisis corresponde a HILIC NEGATIVE ION MODE), la unidad de medida y la descripción del estudio.

```
metadata(se1)
```

```
## $data_source
## [1] "Metabolomics Workbench"
##
## $study_id
## [1] "ST003521"
##
## $analysis_id
## [1] "AN005782"
##
## $analysis_summary
## [1] "HILIC POSITIVE ION MODE"
##
## $units
## [1] "peak height"
##
## $name
## [1] "ST003521:AN005782"
##
## $description
## [1] "Metabolic Profiling Unveils Enhanced Antibacterial Synergy of Polymyxin B and Teixobactin agains
##
## $subject_type
## [1] NA
```

Ahora, teniendo mayor información de cómo se estructuran los datos y cual es el contexto del estudio, procederemos a explorar como darle un sentido a ellos.

Observamos anteriormente que las columnas de las muestras presentaban un sufijo '_N', el cual indica el número de réplica, por lo que una opción será agrupar las columnas con nombre equivalente y luego promediar sus valores para cada metabolito (filas)

Primero creamos un dataframe con la matriz del experimento 1 y seleccionamos los nombres del experimento 1. Para ello seleccionaremos los nombres descriptivos de las muestras en matriz colData. Generamos una lista unica de muestras, las que usaremos para identificar qué columnas se usarán para obtener promedio

```
data_df1 <- as.data.frame(assay(se1))
col_names1 <- se1@colData$sample_source
row_names1 <- se1@elementMetadata$metabolite_name

colnames(data_df1) <- col_names1 # Se asigan nombres de columnas a dataframe
rownames(data_df1) <- row_names1 # Se asigan nombres de filas a dataframe

unicos1 <- unique(col_names1[duplicated(col_names1)]) # Generamos una lista unica de nombres de columna

data_avg1 <- data.frame(row.names = rownames(data_df1)) # Crea dataframe vacío con largo de filas

for (name in unicos1) {
  data_avg1[[name]] <- rowMeans(data_df1[, colnames(data_df1) == name], na.rm = TRUE)
} # Itera sobre lista de nombres unicos para determinar promedio
```

Ahora realizamos el mismo ejercicio con el segundo objeto.

```
data_df2 <- as.data.frame(assay(se2))
col_names2 <- se2@colData$sample_source
row_names2 <- se2@elementMetadata$metabolite_name

colnames(data_df2) <- col_names2 # Se asigan nombres de columnas a dataframe
rownames(data_df2) <- row_names2 # Se asigan nombres de filas a dataframe

unicos2 <- unique(col_names2[duplicated(col_names2)]) # Generamos una lista unica de nombres de columna

data_avg2 <- data.frame(row.names = rownames(data_df2)) # Crea dataframe vacío con largo de filas

for (name in unicos2) {
  data_avg2[[name]] <- rowMeans(data_df2[, colnames(data_df2) == name], na.rm = TRUE)
} # Itera sobre lista de nombres unicos para determinar promedio
```

Ahora, teniendo las matrices con promdeios de ambos objetos provenientes de summarizedexperiment, concatenamos los dataframes antes creados en uno solo

```
data_avg <- rbind(data_avg1, data_avg2)
```

```
kable(data_avg[1:10,1:6])
```

	BLANK	Combination_1h	Combination_3h	Combination_6h	Control_1h	Control_3h
10,11-dihydro-20-trihydroxy-leukotriene B4	NaN	199784.0	514042.00	452823.75	1596473.75	1255597.5
1-(14-methyl-pentadecanoyl)-2-(8-[3]-ladderane-octanyl)-sn-glycerol	NaN	43236.0	72269.50	67371.25	108278.50	105704.0
1,2-dihexadecanoyl-sn-glycero-3-phosphosulfocholine	12395.0	2206761.0	2414995.75	2950121.75	2920088.25	1991086.8
1,3,5-trimethoxybenzene	NaN	11197.0	30593.50	24576.50	84796.75	137355.0
1-4-beta-D-Glucan	NaN	132375.2	169030.25	131530.25	423642.25	236291.0
1,6-anhydro-N-acetylmuramate	17519.5	201987.0	221758.50	279904.25	434804.50	288969.8
1-(beta-D-Ribofuranosyl)-1,4-dihydronicotinamide	NaN	108070.0	80567.33	89926.00	85004.75	109576.5
1-carboxymethylpyridinium	55594.0	227125.5	186415.50	215638.50	236746.00	233803.8
1-deoxyxylonojirimycin	NaN	62268.0	118603.75	106085.50	196470.75	248249.5
1-Hexadecanoyl-2-(9Z-octadecenoyl)-sn-glycero-3-cytidine-5'-diphosphate	NaN	73331.0	132536.25	142474.75	225906.50	327031.8

A continuación, trabajaremos los cambios en las mediciones de los distintos metabolitos. Para ello, deberemos calcular el cambio de cada metabolito respecto a una condición, en este caso, compararemos las mediciones de las distintas condiciones respecto a la condición control. Extraeremos de nuestros datos las columnas '*BLANK*' y '*QC*'

```
drops <- c("_BLANK_" , "_QC_")
data_avg <- data_avg[ , !(names(data_avg) %in% drops)]
```

Ahora determinaros el cambio entre condicion normal y condicion experimetal. Esto se puede calcular como $FoldChange$ (FC) o en $log_2 FoldChange$ $log_2(FC)$, el cual se calcula como:

$$FC = \frac{CondicionExperimental}{CondicionControl}$$

$$log_2(FC) = log_2(\frac{CondicionExperimental}{CondicionControl})$$

Calcularemos $log_2(FC)$ puesto que en el estudio de este trabajo se realizó así. Esto permitirá comparar con los resultados del artículo.

```r
names(data_avg) <- gsub(" ", "", names(data_avg))
names(data_avg) <- gsub("-", "", names(data_avg))
drops <- c("Control_1h","Control_3h","Control_6h")
log2FC_data <- data_avg[ , !(names(data_avg) %in% drops)]

# Calcular log2 Fold Change para cada condición y tiempo
log2FC_data$Combination_1h <- log2(data_avg$Combination_1h / data_avg$Control_1h)
log2FC_data$Combination_3h <- log2(data_avg$Combination_3h / data_avg$Control_3h)
log2FC_data$Combination_6h <- log2(data_avg$Combination_6h / data_avg$Control_6h)

log2FC_data$PolymyxinB_1h <- log2(data_avg$PolymyxinB_1h / data_avg$Control_1h)
log2FC_data$PolymyxinB_3h <- log2(data_avg$PolymyxinB_3h / data_avg$Control_3h)
log2FC_data$PolymyxinB_6h <- log2(data_avg$PolymyxinB_6h / data_avg$Control_6h)

log2FC_data$Leu10teixobactin_1h <- log2(data_avg$'Leu10teixobactin_1h' / data_avg$Control_1h)
log2FC_data$Leu10teixobactin_3h <- log2(data_avg$'Leu10teixobactin_3h' / data_avg$Control_3h)
log2FC_data$Leu10teixobactin_6h <- log2(data_avg$'Leu10teixobactin_6h' / data_avg$Control_6h)

# Reemplazar valores infinitos o NaN con NA
log2FC_data <- log2FC_data %>% mutate(across(everything(), ~ ifelse(is.infinite(.) | is.nan(.), NA, .)))

log2FC_data <- log2FC_data %>% replace(is.na(.), 0)

new_order = c("Combination_1h","PolymyxinB_1h","Leu10teixobactin_1h","Combination_3h","PolymyxinB_3h","
log2FC_data <- log2FC_data[, new_order]
```

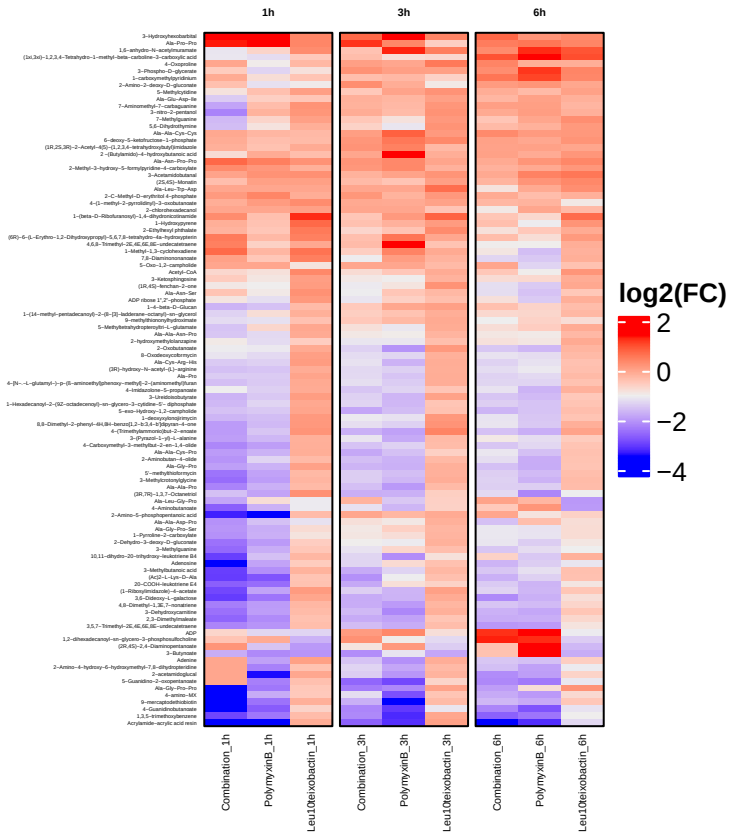
```r
library(ComplexHeatmap)
library(dplyr)

# Ordenamos las columnas agrupando por tiempo
log2FC_data <- log2FC_data %>%
  select(ends_with("_1h"), ends_with("_3h"), ends_with("_6h"))

small_set <- log2FC_data[1:100,]

# Convertir a matriz
matrix_FC <- as.matrix(small_set)

# Crear el heatmap con ajustes para visualización
Heatmap(matrix_FC,
        name = 'log2(FC)',
        cluster_rows = TRUE,
        show_row_dend = FALSE ,
        cluster_columns = FALSE,    # No agrupar columnas (se mantienen en orden por tiempo)
        column_split = rep(c("1h", "3h", "6h"), each = ncol(matrix_FC) / 3),
        border = TRUE,
        row_names_side = "left",   # Mueve etiquetas de filas a la izq
        row_names_gp = gpar(fontsize = 2),
        column_names_gp = gpar(fontsize = 4),
        column_title_gp = gpar(fontsize = 4, fontface = "bold"),# Ajusta tamaño de texto de las etiquet
        heatmap_height = unit(11, "cm"),  # Hace el heatmap más largo
        heatmap_width = unit(8, "cm")    # Ajusta el ancho
)
```



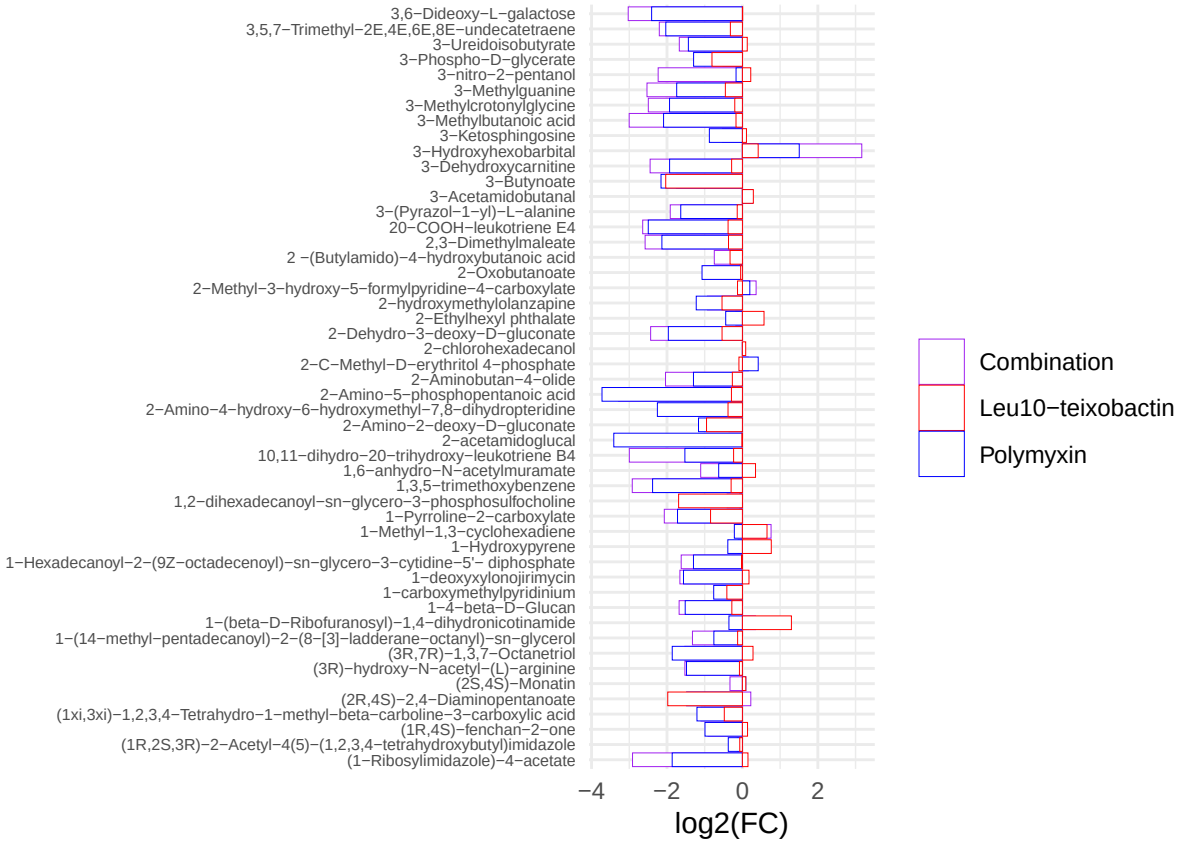

```
df_long <- log2FC_data %>%
  rownames_to_column("Metabolite") %>%
  pivot_longer(cols = -Metabolite, names_to = "Condition", values_to = "log2FC")

condition_1h <- df_long %>%
  filter(str_detect(Condition, "_1h"))

df_short <- condition_1h[1:150,]

df_short$Group <- case_when(
  grepl("Polymyxin", df_short$Condition) ~ "Polymyxin",
  grepl("Leu10teixobactin", df_short$Condition) ~ "Leu10-teixobactin",
  grepl("Combination", df_short$Condition) ~ "Combination")


ggplot(df_short, aes(x = log2FC, y = Metabolite, color = Group)) +
  geom_bar(stat = "identity", position = "identity", fill = "white", linewidth = 0.1) +
  scale_color_manual(values = c("Polymyxin" = "blue", "Leu10-teixobactin" = "red", "Combination" = "purp
  theme_minimal() +
  labs(x = "log2(FC)", y = "", color = "") +
  theme(axis.text.y = element_text(size = 6))
```


3,6−Dideoxy−L−galactose
3,5,7−Trimethyl−2E,4E,6E,8E−undecatetraene
3−Ureidoisobutyrate
3−Phospho−D−glycerate
3−nitro−2−pentanol
3−Methylguanine
3−Methylcrotonylglycine
3−Methylbutanoic acid
3−Ketosphingosine
3−Hydroxyhexobarbital
3−Dehydroxycarnitine
3−Butynoate
3−Acetamidobutanal
3−(Pyrazol−1−yl)−L−alanine
20−COOH−leukotriene E4
2,3−Dimethylmaleate
2 −(Butylamido)−4−hydroxybutanoic acid
2−Oxobutanoate
2−Methyl−3−hydroxy−5−formylpyridine−4−carboxylate
2−hydroxymethylolanzapine
2−Ethylhexyl phthalate
2−Dehydro−3−deoxy−D−gluconate
2−chlorohexadecanol
2−C−Methyl−D−erythritol 4−phosphate
2−Aminobutan−4−olide
2−Amino−5−phosphopentanoic acid
2−Amino−4−hydroxy−6−hydroxymethyl−7,8−dihydropteridine
2−Amino−2−deoxy−D−gluconate
2−acetamidoglucal
10,11−dihydro−20−trihydroxy−leukotriene B4
1,6−anhydro−N−acetylmuramate
1,3,5−trimethoxybenzene
1,2−dihexadecanoyl−sn−glycero−3−phosphosulfocholine
1−Pyrroline−2−carboxylate
1−Methyl−1,3−cyclohexadiene
1−Hydroxypyrene
1−Hexadecanoyl−2−(9Z−octadecenoyl)−sn−glycero−3−cytidine−5'− diphosphate
1−deoxyxylonojirimycin
1−carboxymethylpyridinium
1−4−beta−D−Glucan
1−(beta−D−Ribofuranosyl)−1,4−dihydronicotinamide
1−(14−methyl−pentadecanoyl)−2−(8−[3]−ladderane−octanyl)−sn−glycerol
(3R,7R)−1,3,7−Octanetriol
(3R)−hydroxy−N−acetyl−(L)−arginine
(2S,4S)−Monatin
(2R,4S)−2,4−Diaminopentanoate
(1xi,3xi)−1,2,3,4−Tetrahydro−1−methyl−beta−carboline−3−carboxylic acid
(1R,4S)−fenchan−2−one
(1R,2S,3R)−2−Acetyl−4(5)−(1,2,3,4−tetrahydroxybutyl)imidazole
(1−Ribosylimidazole)−4−acetate
−4
−2
0
2
log2(FC)
Combination
Leu10−teixobactin
Polymyxin