

# Documentación CIBERSORTx

2025-01-28

## MANUAL DE USO DE CIBERSORTx

Se ha trabajado con el archivo de datos counts\_normalized.tsv. Este archivo contenía valores NaN y filas de genes duplicados, por lo tanto, se procedió a realizar un filtrado de los datos para que el análisis con Cibersortx funcione correctamente.

En primer lugar, se debe cargar el dataset en el environment. Una vez cargado correctamente, se procede a eliminar las filas NaN

```
# Cargar datos
data <- read.table("/Users/clarapodaru/Documents/paraClara/cibersort/input/counts_normalized.tsv", header = TRUE, sep = "\t")

# Eliminar las filas de tipo "gene" que presentan valores NaN
data <- data[!is.na(data$gene), ]

# Comprobar si siguen habiendo valores Na
any(is.na(data$gene)) # Debería devolver FALSE
```

Como bien se puede ver, ya no hay valores NaN pero sí valores duplicados de tipo “gene”.

Se procede a visualizar la estructura del archivo y las 10 primeras filas del dataset.

```
# Verificar la estructura del archivo
str(data)

# Mostrar las 10 primeras filas del dataset
head(data, 10)
```

```
##          gene PED210_21873 PED120_12005 PED120_12001 PED120_12002 PED120_12003
## 1    TSPAN6   1.049219     0.0000   11.38244     0.0000     0.0000
## 2    DPM1    1597.960820   1621.8347  1651.71918   1764.2250  1694.1807
## 3    SCYL3   1562.287368   1653.2108  1503.74740   1488.9472  1173.9547
## 4 Clorf112  2225.393893   2225.0512  3059.98052   2371.0597  2217.3088
## 5    FGR    2896.894172   2423.4719  1264.08371   1413.0929  1985.1299
## 6    FUCA2   771.176102   803.4048   798.03584   760.9903   581.8985
## 7    GCLC   887.639431   958.0757  1047.18490   1085.2064  443.3167
## 8    NFYA   1563.336587   1442.4165  1332.37837   1529.3213  1642.6660
## 9    STPG1   261.255577   219.1907   117.61859   179.8482  127.6984
## 10   NIPAL3  1386.018545   1236.0414   892.88954   1458.3608  1273.3563
##          PED120_12004 PED172_21769 PED172_21768 PED172_21770 PED172_21771
## 1    2.879619   3.465335   4.809901   10.79235   7.279089
## 2    1566.512792  1688.484293  1587.267178  1668.18922  1560.636710
## 3    1262.712976  1207.669115  1334.266410  1328.23014  1406.320021
## 4    2822.026720  2672.639325  3031.199321  3014.92053  3408.069532
## 5    1020.824972  937.373014  818.645072   715.37874  1924.591167
## 6    705.506680   711.259930  760.926266   558.11876  596.885309
## 7    1052.500782  1008.412374  1127.440687   853.36667  1004.514300
## 8    1472.925171  1439.846534  1549.749954   1659.70951  1522.785447
## 9    136.781907   121.286712   181.814240   195.03321  139.758511
## 10   935.876208  1047.397389  1236.144439   1091.56929  957.928130
##          PED125_12504 PED125_12500 PED125_12506 PED27_2701 PED27_2700 PED27_2705
## 1    2.115067   4.353738   0.8938104   4.002597   2.459661  2.856499
## 2    1679.363099  1569.522555  1663.3811210  1362.884430  1301.160516  1449.673302
## 3    1629.659027  1233.196293  1526.6281330  1480.961055  1234.749676  1390.400945
## 4    2492.606312  2648.161148  2807.4584110  2783.806523  2317.000389  3004.322946
## 5    1665.615164  1333.332267  1895.7718210  1209.785078  3295.945351  1717.470094
## 6    769.884343   826.121789   870.5713124   685.444813  848.582945  871.946355
## 7    935.917092  1096.053546   918.8370730   864.561048  649.350427  841.953115
## 8    1597.933024  1486.801533  1540.0352890  1605.041575  1468.417444  1388.258571
## 9    194.586152   174.149521   150.1601442   150.097404  182.014893  138.540207
## 10   1230.968921  1116.733801  1029.6695600   982.637673  1168.338837  1044.764552
##          PED27_2706 PED27_2707 PED72_7203 PED72_7204 PED72_7201 PED72_7200
## 1    0.0000   3.535198   3.232243   3.454757   2.398822   1.844126
## 2    1509.8475  1518.956845  1617.737692  1681.314983  1516.055682  1549.065540
## 3    1237.9547  1342.196933  1696.927649  1423.359808  1522.052738  1431.041499
## 4    2641.9324  2489.957962  2578.521966  2289.352183  2009.013661  2690.579312
## 5    3890.7147  1938.467036  1697.735710  1347.355158  2437.203439  1252.161311
## 6    554.6133   668.152468   766.041625   780.775040   662.074950  717.364875
## 7    712.2149   748.283628   1039.974231  1117.038037  961.927735  883.336183
## 8    1252.3915  1478.891265  1595.920051  1583.430207  1288.167566  1659.713078
## 9    167.2261   131.980734   144.642881   125.522831   200.301661  195.477318
## 10   1256.0006  1098.268254  1156.334984  1100.915838  1567.630361  1110.163637
##          PED72_7202 PED127_12709 PED127_12700 PED127_12708 PED65_6504 PED65_6506
## 1    2.520687   4.792164   1.886258   2.819093   2.460477   1.530481
## 2    1635.926108  1586.206283  1508.063272  1753.475870  1648.519259  1678.938125
## 3    1507.371052  1410.094256  1454.304919  1585.269985  1345.880649  1364.679272
## 4    1862.787972  2675.824572  2093.746381  2669.211258  2676.998439  2733.949988
## 5    2156.448052  1368.162821  1510.892659  1503.986136  1077.688710  1223.874981
## 6    732.259683   710.438313   658.304042   833.511841  861.166777  833.602217
## 7    943.997423   951.843574   700.744847   945.805714  1018.637274  1102.966948
## 8    1509.891739  1372.954985  1602.376172  1404.848031  1468.904474  1551.908166
## 9    202.915334   97.041321   226.350960   138.605408  98.419060  132.641724
## 10   1385.117714  1034.508403  1142.129219  1076.423692  1065.386327  1111.639676
```

```
## PED65_6502 PED65_6503 PED65_6500
## 1 0.0000 3.17369 4.829601
## 2 1560.0454 1719.34638 1627.575517
## 3 1504.7614 1340.09046 1751.535274
## 4 2126.7064 2535.77804 2693.307458
## 5 877.6335 1041.76363 1375.631334
## 6 761.8826 801.35664 694.657602
## 7 983.0186 1086.19529 881.402172
## 8 1665.4305 1755.05039 1532.593365
## 9 134.7547 157.09764 182.719902
## 10 1252.5281 982.25695 1066.536874
```

```
# Filtrar filas con valores no válidos en la columna gene
data_clean <- data[!is.na(data$gene) & data$gene != "NaN", ]

# Comprobar si hay datos duplicados de tipo gene
any(duplicated(data$gene)) # Debería devolver FALSE
```

Se ha comprobado que hay genes duplicados dentro del dataset ya que la comprobación anterior nos ha devuelto el valor TRUE. Por lo tanto, hay que identificar qué genes están duplicados.

```
# Identificar genes duplicados
duplicated_genes <- data_clean$gene[duplicated(data_clean$gene)]
print(unique(duplicated_genes)) # Mostrar los genes duplicados
```

Podemos observar que ese grupo de genes se encuentran duplicados. Para realizar el filtrado de los datos se procederá a calcular el promedio de los valores de los genes duplicados. Para ello se empleará el paquete dplyr (si no se encuentra instalado, habría que realizar un paso anterior poniendo el siguiente comando `install.packages("dplyr")`).

```
library(dplyr)

# Calcular el promedio de los valores genes duplicados
data_clean_avg <- data_clean %>%
  group_by(gene) %>%
  summarise(across(everything(), mean))
```

Una vez ejecutado lo anterior, volvemos a comprobar si siguen habiendo genes duplicados.

```
# Comprobar si los duplicados se resolvieron
any(duplicated(data_clean_avg$gene)) # debería devolver FALSE
```

```
## [1] FALSE
```

Como bien se puede comprobar, los genes duplicados han sido sustituidos por los valores promedio.

Por último, se debe guardar el dataset filtrado en la carpeta de trabajo empleando los siguientes comandos.

```
# Almacenar el archivo filtrado
write.table(data_clean_avg, "/Users/clarapodaru/Documents/paraClara/cibersort/input/counts_normalized_cleaned_final.tsv",
            sep = "\t", row.names = FALSE, quote = FALSE)
```

## **¿COMO EJECUTAR CIBERSORTx?**

CIBERSORTx es una herramienta para llevar a cabo citometrías *in silico* la cual ha sido diseñada para obtener la proporción de tipos celulares concretos en conjuntos de datos de RNAseq masivo. Para ejecutarla, primero se debe registrar en la página web: <https://cibersortx.stanford.edu/> (<https://cibersortx.stanford.edu/>)

### **1. Instalar Docker Desktop**

Una vez realizado el registro en la página web, se debe solicitar el token de acceso y descargar las instrucciones de cómo instalar Docker.

Se puede instalar Docker a través del siguiente link: <https://www.docker.com/products/docker-desktop> (<https://www.docker.com/products/docker-desktop>)

Una vez descargado sigue las instrucciones de configuración en función de tu sistema operativo (MacOS, Windows, Linux...). En este caso, CIBERSORTx se ha instalado en un dispositivo Macbook Apple Silicon MacOS.

Después de la instalación, abre Docker Desktop en tu dispositivo. Seguramente debas iniciar sesión o crear una cuenta de Docker en caso de que no lo hayas realizado.

### **2. Descargar imágenes de Docker**

Para utilizar Docker con la herramienta CIBERSORTx, es necesario extraer las imágenes que contienen el software. Para ello se deben usar los siguientes comandos.

```
docker pull cibersortx/fractions
docker pull cibersortx/hires
docker pull cibersortx/gep
```

Estos comandos sirven para descargar las imágenes de los tres módulos principales de CIBERSORTs de Docker Hub.

- docker pull cibersortx/fractions : esta imagen se usa la estimar las fracciones de diferentes tipos celulares a partir de datos de RNA-seq masivo.
- docker pull cibersortx/hires : esta imagen se usa la imputar la variación de expresión génica a nivel de muestra para tipos específicos.
- docker pull cibersortx/gep : esta imagen sirve para generar los perfiles de expresión génica (GEPs) para los distintos tipos celulares.

### **3. Ejecución de CIBERSORTx empleando Docker**

A continuación se muestran los comandos empleados para trabajar con cibersortx desde el terminal.

```
docker run --platform linux/amd64 -v /Users/clarapodaru/Documents/cibersortx/da
ta/input:/src/data -v /Users/clarapodaru/Documents/cibersortx/data/output:/src/
outdir cibersortx/fractions --username dclara.podaru@universidadcatolicadeavil
a.es --token 10b592755a1ac3b5a81b2af4497c5cf --mixture /src/data/counts_nor
malized_cleaned_final.tsv --sigmatrix /src/data/LM22.txt --sourceGEPs /src/data/LM
22_source_GEPs.txt --outdir /src/outdir --rmbatchBmode TRUE --perm 100 --verbos
e TRUE
```

Ouput:

```
>Running CIBERSORTxFractions...
>[Options] username: dclara.podaru@universidadcatolicadeavila.es
>[Options] token: 10b592755a1ac3b5a81b2af4497c5cf
>[Options] mixture: /src/data/counts_normalized_cleaned_final.tsv
>[Options] sigmatrix: /src/data/LM22.txt
>[Options] sourceGEPs: /src/data/LM22_source_GEPs.txt
>[Options] outdir: /src/outdir
>[Options] rmbatchBmode: TRUE
>[Options] perm: 100
>[Options] verbose: TRUE
>=====CIBERSORTx Settings=====
>Mixture file: /src/data/counts_normalized_cleaned_final.tsv
>Signature matrix file: /src/data/LM22.txt
>Number of permutations set to: 100
>Enable verbose output
>Do B-mode batch correction
>=====CIBERSORTx=====
>Batch correction:.
Read 9541 items
>Run CIBERSORTx on B-mode batch corrected mixtures.
>=====CIBERSORTx Settings=====
>Mixture file: /src/outdir//CIBERSORTx_Mixtures_Adjusted.txt
>Signature matrix file: /src/data/LM22.txt
>Number of permutations set to: 100
>Enable verbose output
>=====CIBERSORTx=====
P 1 %
P 2 %
P 3 %
P 4 %
P 5 %
P 6 %
P 7 %
P 8 %
P 9 %
P 10 %
P 11 %
P 12 %
P 13 %
P 14 %
P 15 %
P 16 %
P 17 %
P 18 %
P 19 %
P 20 %
P 21 %
P 22 %
P 23 %
P 24 %
P 25 %
P 26 %
P 27 %
P 28 %
P 29 %
```

P 30 %  
P 31 %  
P 32 %  
P 33 %  
P 34 %  
P 35 %  
P 36 %  
P 37 %  
P 38 %  
P 39 %  
P 40 %  
P 41 %  
P 42 %  
P 43 %  
P 44 %  
P 45 %  
P 46 %  
P 47 %  
P 48 %  
P 49 %  
P 50 %  
P 51 %  
P 52 %  
P 53 %  
P 54 %  
P 55 %  
P 56 %  
P 57 %  
P 58 %  
P 59 %  
P 60 %  
P 61 %  
P 62 %  
P 63 %  
P 64 %  
P 65 %  
P 66 %  
P 67 %  
P 68 %  
P 69 %  
P 70 %  
P 71 %  
P 72 %  
P 73 %  
P 74 %  
P 75 %  
P 76 %  
P 77 %  
P 78 %  
P 79 %  
P 80 %  
P 81 %  
P 82 %  
P 83 %  
P 84 %  
P 85 %

P 86 %  
P 87 %  
P 88 %  
P 89 %  
P 90 %  
P 91 %  
P 92 %  
P 93 %  
P 94 %  
P 95 %  
P 96 %  
P 97 %  
P 98 %  
P 99 %  
P 100 %

column B cells naive B cells memory Plasma cells T cells CD8 T cells CD4 naive T cell s CD4 memory resting T cells CD4 memory activated T cells follicular helper T cells r egulatory (Tregs) T cells gamma delta NK cells resting NK cells activated Monocytes M acrophages M0 Macrophages M1 Macrophages M2 Dendritic cells resting Dendritic cells a ctivated Mast cells resting Mast cells activated Eosinophils Neutrophils P-value Corr elation RMSE %Completed

	B cells	naive B cells	memory Plasma cells	T cells	CD8 T cells	CD4 naive T cell s	CD4 memory resting T cells	CD4 memory activated T cells	follicular helper T cells	r egulatory (Tregs)	T cells	gamma delta	NK cells	resting NK cells	activated Monocytes	M acrophages	M0 Macrophages	M1 Macrophages	M2 Dendritic cells	resting Dendritic cells	a ctivated Mast cells	resting Mast cells	activated Eosinophils	Neutrophils	P-value	Corr elation	RMSE	%Completed			
1	0.130128	0.286625	0.120973	0	0	0	0.0810261	3.22581																							
2	0.202025	0.214481	0.17104	0	0	0	0.0740237	6.45161																							
3	0.189292	0.319804	0.0912852	0	0	0	0.0144083	0.0273692	9.67742																						
4	0.191997	0.259598	0.130481	0	0	0	0.0318421	12.9032																							
5	0.119322	0.46657	0.0880767	0	0	0	0.00418843	16.129																							
6	0.220347	0.284978	0.172019	0	0	0	0.0299058	0.0242541	19.3548																						
7	0.227029	0.231697	0.161793	0	0	0	0.00347558	22.5806																							
8	0.163649	0.298784	0.168029	0	0	0	0.00774293	0.0222803	25.8065																						
9	0.125705	0.360598	0.116885	0	0	0	0.030502	29.0323																							
10	0.105016	0.392574	0.0867303	0	0	0	0.0122908	32.2581																							
11	0.187188	0.232913	0.163794	0	0	0	0.046945	35.4839																							
12	0.196442	0.283842	0.123751	0	0	0	0.0165296	0.0377147	38.7097																						
13	0.216261	0.297399	0.123834	0	0	0	0.0187738	0.00288721	41.9355																						
14	0.234551	0.184559	0.163589	0	0	0	0.0186291	0.0508061	45.1613																						
15	0.161584	0.251496	0.23744	0	0	0	0.0690434	48.3871																							
16	0.188843	0.247586	0.201426	0	0	0	0.00574835	0.0554381	51.6129																						
17	0.103395	0.341008	0.0965923	0	0	0	0.0418062	54.8387																							
18	0.143731	0.264144	0.174779	0	0	0	0.0126616	0.0926953	58.0645																						
19	0.248131	0.202912	0.155225	0	0	0	0.0223711	0.0155109	61.2903																						
20	0.176361	0.308124	0.0769077	0	0	0	0.0620382	64.5161																							
21	0.127096	0.190498	0.1105	0	0	0	0.169667	67.7419																							
22	0.378066	0.094459	0.191264	0	0	0	0.023585	0.0158076	0.0299642	70.9677																					
23	0.18596	0.223455	0.128889	0	0	0	0.000445161	0.0703741	74.1935																						
24	0.152584	0.349818	0.159384	0	0	0	0.00403861	0.0215413	77.4194																						
25	0.175789	0.239824	0.115096	0	0	0	0.00561096	80.6452																							
26	0.207833	0.245458	0.199944	0	0	0	0.0471591	83.871																							
27	0.157238	0.306525	0.190919	0	0	0	0.00521706	0.0151258	87.0968																						
28	0.151718	0.309338	0.166114	0	0	0	0.0423919	90.3226																							
29	0.192025	0.276538	0.136384	0	0	0	0.00928469	0	0.0163943	93.5484																					
30	0.147961	0.273313	0.226542	0	0	0	0.0158407	0.00859038	96.7742																						
31	0.152787	0.319757	0.155326	0	0	0	0.0213691	100																							

### Explicación de los parámetros:

- docker run: este comando sirve para ejecutar el contenedor de Docker basándose en la imagen especificada.
- --platform linux/amd64: este parámetro especifica la arquitectura empleada a la hora de correr la imagen. Asegura la compatibilidad con la plataforma linux/amd64, arquitectura estándar empleada para la mayoría de contenedores de Docker.
- -v /Users/clarapodaru/Documents/paraClara/cibersort/input:/src/data: la '-v' se emplea asignar un directorio local, en este caso el de entrada de datos, a un directorio dentro del contenedor.
  - /Users/clarapodaru/Documents/paraClara/cibersort/input: -> estp debería ser el directorio local donde se encuentran los ficheros de datos con los que se desea trabajar.
  - /src/data -> esta es la localización dentro del contenedor desde donde se va a poder acceder a los datos.
- -v /Users/clarapodaru/Documents/paraClara/cibersort/output:/src/outdir : la '-v' se emplea asignar un directorio local, en este caso el de salida de datos, a un directorio dentro del contenedor.
  - /Users/clarapodaru/Documents/paraClara/cibersort/output: -> el directorio local donde se van a guardar los datos
  - /src/outdir -> esta es la localización dentro del contenedor donde se van a almacenar los datos
- --username y --token: en estos campos se debe introducir las credenciales para poder iniciar sesión en CIBERSORTx.
- --sigmatrix: se introduce la ruta de la matriz de firmas (LM22.txt)
- --mixture: se introduce la ruta del archivo de mezcla (counts\_normalized\_cleaned\_final.tsv).
- --sourceGEPs: se introduce la ruta del archivo del perfil de expresiones génicas (LM22\_source\_GEPs.txt).
- --outdir: indicar el directorio donde se van a almacenar los datos de salida.
- --rmbatchBmode TRUE: esta opción especifica si se debe aplicar la corrección por lotes empleando el modo B. Si se establece en TRUE se activa la corrección por lotes. Esta corrección es importante para poder mitigar los sesgos que se han introducido por diferentes variables técnicas.
- --perm 100: número de permutaciones el cálculo del p valor (fijado en 100).
- --verbose TRUE: establecer esta opción en TRUE permite generar una documentación detallada del proceso llevado a cabo en el terminal.

Se debe ejecutar el comando anterior en el terminal asegurándose de que los ficheros de entrada y los directorios están escritos correctamente. Por último, los resultados se van a almacenar en el directorio de salida que se haya indicado en el comando.

### **Interpretación de datos:**

#### **Cibersortx\_adjusted.txt (proporciones estimadas de cada tipo celular en cada muestra)**

- Valores entre 0 y 1 -> representan la proporción de cada tipo celular en la muestra -> porque los datos están tipificados.
- P-value <0.05 : tenemos unos resultados significativos.
- Correlation > 0.6 : el ajuste del modelo es confiable
- RMSE (error cuadrático medio) -> valores bajos indican una mejor precisión de la estimación

#### **cibersortx\_mixtures\_adjusted.txt**

cada fila representa un gen y las columnas una muestra.

Para comprobar si los resultados son fiables:

- Se deben filtrar las muestras con P-value > 0.05 (significa que la deconcolución no es significativa).
- Clustering jerárquico para comprobar los patrones de las muestras.

### **Validación y análisis de resultados.**

- Descartar las muestras que presentan un p-value > 0.05 (no significativas).
- Realizar un clustering para identificar patrones celulares.