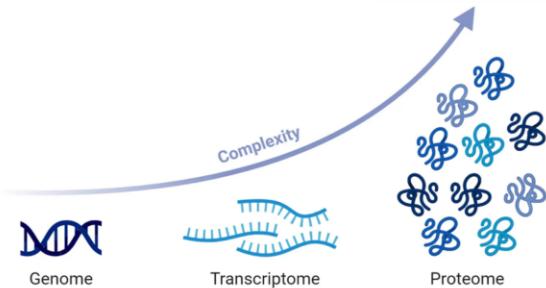


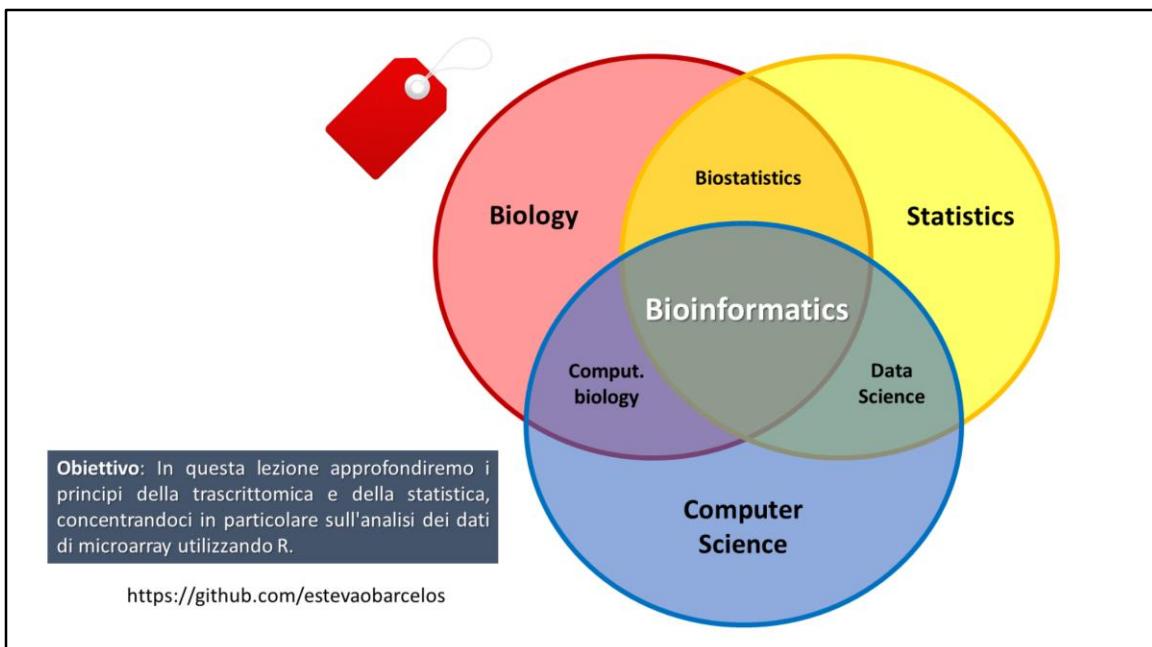
Analisi del trascrittoma con R

PhD. Estevao Barcelos
Università degli Studi di Perugia - UNIPG



2024 Training on Microarray/RNA-seq data analysis using R Studio/Geo2R NCBI pipeline

Obiettivo: In questa lezione approfondiremo i principi della trascrittomica, concentrandoci in particolare sull'analisi dei dati di microarray utilizzando R.



La bioinformatica si esprime trasversalmente nel corpus di discipline e tecniche multidisciplinari capaci di supportare la ricerca biologica nei suoi vari ambiti applicativi. Come possiamo vedere, si trova all'intersezione di altre discipline ed è per questo che dobbiamo sviluppare competenze in questi tre ambiti, fondamentalmente. Per questo ho diviso le diapositive in colori che rappresentano queste discipline.

Innanzitutto...

• INTRODUCTION TO R AND RSTUDIO

- **R** è un linguaggio di programmazione specifico per la statistica e la grafica computazionali. È uno dei linguaggi più ampiamente utilizzati da statistici, analisti di dati e ricercatori per gestire, manipolare, analizzare e visualizzare.
- **Rstudio** è un ambiente di sviluppo integrato per R che consente agli utenti di interagire più facilmente con R integrando diversi aspetti della scripting.

In order to use RStudio, R needs to be installed first.

R è un software e un vero e proprio linguaggio di programmazione. Potremmo pensare a R come a un ambiente statistico. La sua diffusione è sempre più rapida e in crescita perché è un software disponibile gratuitamente e quindi chiunque può scaricarlo sul proprio PC e notebook e lavorarci in totale autonomia. È assolutamente personalizzabile e inoltre con R Studio ha una struttura facilmente accessibile. Quindi, R è un linguaggio che nasce come evoluzione del linguaggio S nel 1996. Prima di spiegare come funziona l'Rstudio dobbiamo appunto installare R e Rstudio.

R and Rstudio Installation

- **Step 1:**

- Vai al sito Comprehensive R Archive Network (CRAN) e clicca su “Download R for Windows” (or “Download R for MacOS”). <https://cran.r-project.org>

- **Step 2:**

- Clicca su collegamento sottodirectory “base”.

- **Step 3:**

- Clicca su “Download R-4.4.2 for Windows”. Il collegamento consente di scaricare un'estensione di installazione (.exe file).

- **Step 4:**

- Eseguire il file .exe e seguire la procedura guidata di installazione accettando le impostazioni predefinite.
Una volta installato R, puoi procedere all'installazione di RStudio.

- **Step 5:**

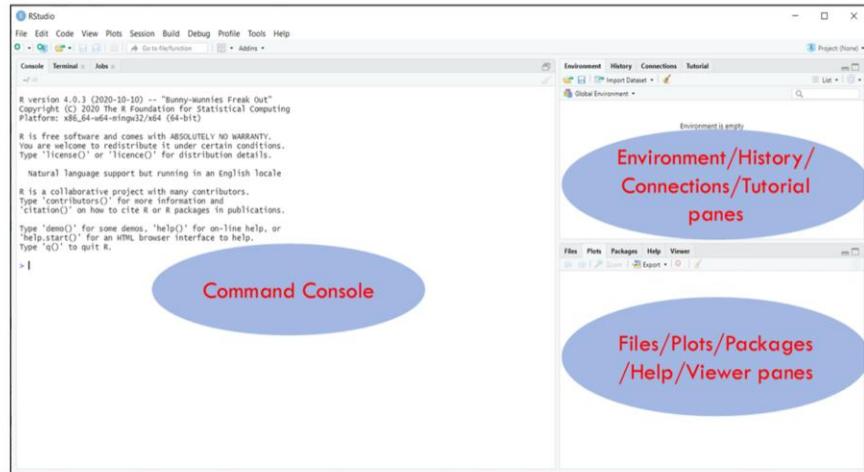
- Vai al sito Web di download di RStudio e fai clic su tasto “Download RStudio for Windows” (or the link for the MacOS version).

- **Step 6:**

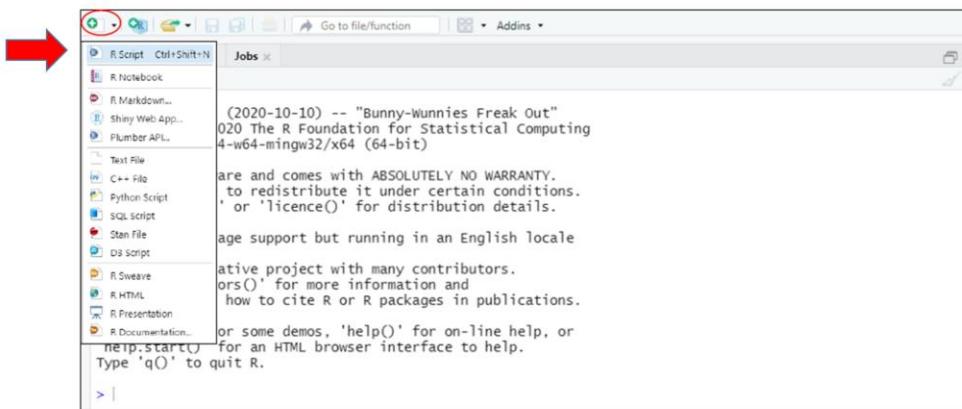
- Eseguire il file .exe e seguire le istruzioni di installazione.

Per installare R dobbiamo andare sul sito web CRAN e scaricare l'ultima versione. CRAN (Comprehensive R Archive Network) è un repository globale dedicato all'archiviazione e alla distribuzione di pacchetti, documentazione e risorse relative al linguaggio di programmazione R. Svolge un ruolo essenziale per la comunità R fornendo un accesso semplice e centralizzato a strumenti ed estensioni. Dopodiché, dobbiamo installare anche il software Rstudio. Andiamo direttamente sul sito web Rstudio e procediamo con le istruzioni.

Rstudio Interface



Rstudio: command console



The screenshot shows the RStudio interface. A red arrow points to the 'File' menu icon in the top-left corner of the toolbar. The main window displays the R command console output:

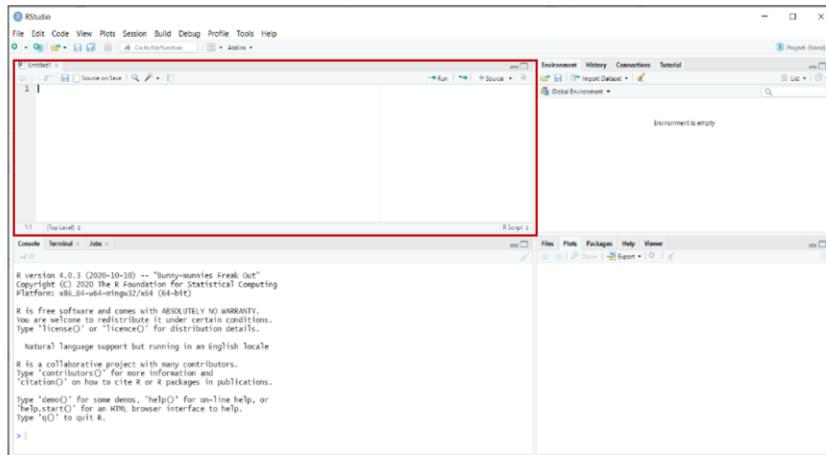
```
(2020-10-10) -- "Bunny-Wunnies Freak Out"
020 The R Foundation for Statistical Computing
4-w64-mingw32/x64 (64-bit)

[REDACTED]
```

The console also shows the standard R startup message and a prompt at the bottom: > |

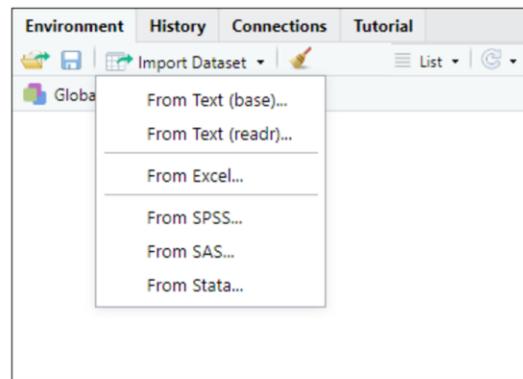
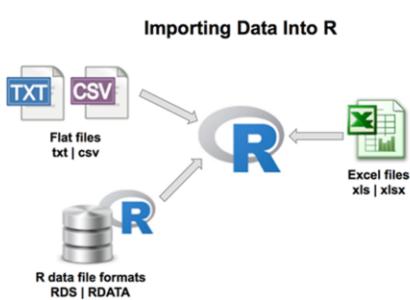
Novità: "quarto document" si riferisce a documenti creati usando il framework **Quarto**, che è uno strumento per la creazione di report dinamici, articoli scientifici, presentazioni e siti web, basati su **R Markdown** e supporta molteplici linguaggi di programmazione (come R, Python e Julia). Publish reproducible, production quality articles, presentations, dashboards, websites, blogs, and books in HTML, PDF, MS Word, ePub, and more.

Rstudio interface with editor window open



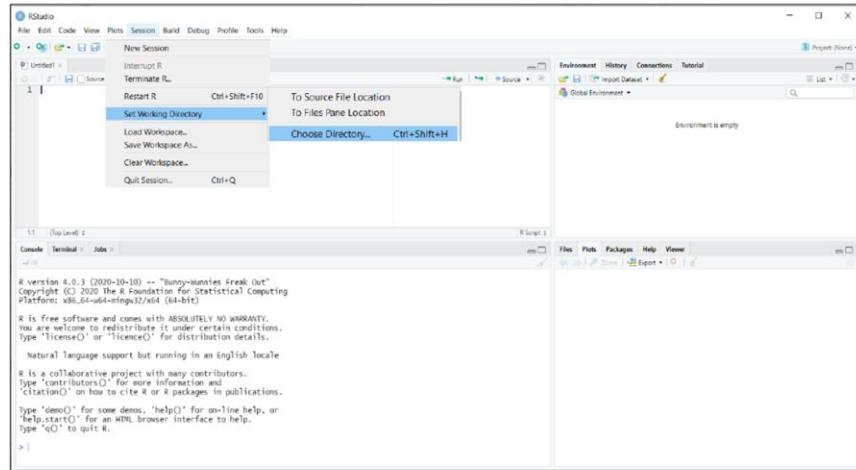
Editore, console di comando e altri 2 quadri (Environment and Files).

Import dataset tab - scheda importazione dataset



Possiamo importare il set di dati da altri formati oltre a R.

Setting the Working Directory



Set Working Directory: opzioni.

Get and Set the Working Directory

Getwd function

```
# Find the path of your working directory  
getwd()
```

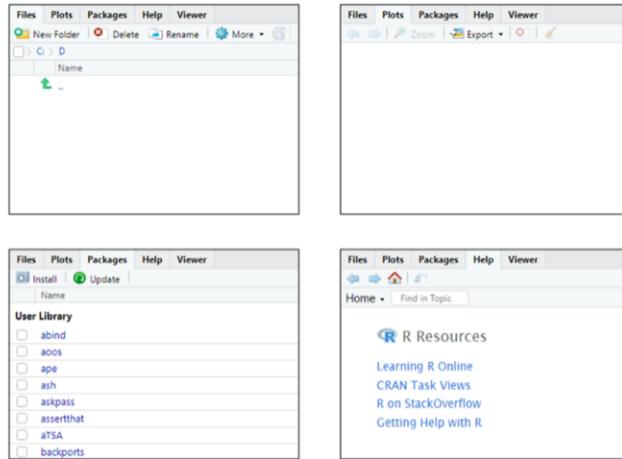
Setwd function

```
# Set the path of your working directory  
setwd("My\\Path")  
setwd("My/Path") # Equivalent
```

```
dir()  
list.files() # Equivalent
```

Use getwd() and setwd().

Rstudio: Files/Plots/Packages/Help/Viewer panels



The options of the two panels.

Packages

- **Packages:** estensioni che contengono codice, dati e documentazione in un formato standardizzato che può essere installato e utilizzato dagli utenti di R per risolvere specifici problemi analitici.

- La versione base di R include già molti pacchetti utili che consentono di eseguire attività elementari come calcoli semplici, esplorazione dei dati e caricamento di file di dati di testo.

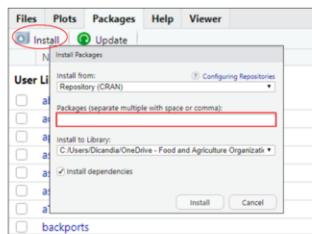


Packages (2)

- A) Per **installare un pacchetto** da CRAN (il repository ufficiale per i pacchetti R forniti dagli utenti) e quindi caricarlo, utilizzare i seguenti comandi:

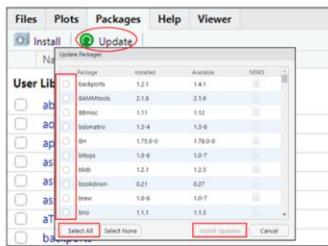
```
install.packages("name of the package")
library("name of the package")
```

- B) Un altro modo per installare i pacchetti è utilizzare la scheda **Installa pacchetti**:



Packages (3)

- Talvolta i pacchetti R vengono aggiornati per migliorarne o modificarne la funzionalità. Si consiglia di aggiornare occasionalmente i pacchetti installati sul computer.
- Puoi aggiornare i pacchetti R installati in RStudio cliccando sul pulsante Aggiorna nella barra degli strumenti nel pannello Pacchetti.



BiocManager



- Il **BiocManager** è un pacchetto R progettato per facilitare l'installazione, l'aggiornamento e la gestione dei pacchetti **Bioconductor**, una piattaforma open-source dedicata all'analisi di dati biologici come genomica e trascrittomico.
- Offre strumenti per garantire compatibilità tra i pacchetti e la versione di R in uso, semplificando il lavoro dei ricercatori nel campo della bioinformatica.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
```

Il **Bioconductor** è una piattaforma open-source basata sul linguaggio di programmazione R, progettata per fornire strumenti software e risorse per l'analisi e la comprensione di dati biologici complessi, come quelli provenienti da genomica, trascrittomico, proteomica e altre discipline omiche. Offre una vasta collezione di pacchetti R per analisi statistiche, visualizzazione e gestione di dati biologici ad alta dimensione, con particolare attenzione a dati provenienti da tecnologie come sequenziamento RNA-seq o microarray. Bioconductor promuove la riproducibilità e l'integrità scientifica, fornendo strumenti aggiornati e standardizzati che permettono ai ricercatori di analizzare dati biologici in modo efficiente e robusto.

La funzione requireNamespace() controlla se il pacchetto specificato (in questo caso, "BiocManager") è disponibile nel sistema. L'argomento quietly = TRUE evita di stampare messaggi inutili durante la verifica. Il simbolo ! nega il risultato della verifica: se il pacchetto **non è disponibile** (cioè requireNamespace restituisce FALSE), allora il codice all'interno del blocco if verrà eseguito.

List of packages used in the training



<https://www.ncbi.nlm.nih.gov/geo/>

- **GEOquery:** è uno strumento R progettato per interagire con il database Gene Expression Omnibus (GEO) e consente di scaricare, importare e gestire i dati di esperimenti di espressione genica.

```
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("GEOquery")
```

getGEO()

getGSEMatrix()

Meta()

GSMList()

getGEOSuppFiles()

GEOquery::getGPL()

GPLList()

List of packages used in the training

- **limma**: (Linear Models for Microarray Data) è un pacchetto per l'analisi dei dati di espressione genica derivanti da tecnologie microarray o RNA-seq.

```
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("limma")
```



normalizeBetweenArrays()

lmFit()

eBayes()

topTable()

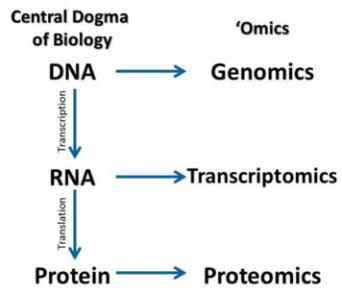
plotMA()

vennDiagram()

L'analisi del trascrittoma

- La conoscenza del genoma di un organismo multicellulare è molto importante per comprendere il suo potenziale funzionale...
- Per comprendere come funzionano le diverse tipologie di cellule, organi o tessuti, o come avviene il differenziamento cellulare, è necessario conoscere quale porzione di informazione genetica viene utilizzata, ovvero quali geni vengono espressi.
- Assumiamo che il livello di mRNA trascritti da un dato gene sia un buon indicatore del livello di espressione delle proteine corrispondenti.

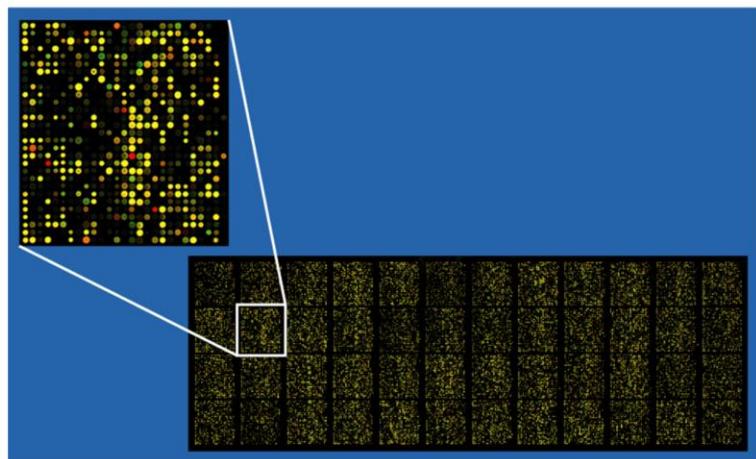
"Mentre la genomica analizza le sequenze di DNA, la trascrittomica si concentra sull'attività dei geni, rivelando quali vengono 'accesi' e 'spenti' e in quale quantità."



Perché la trascrittomica è importante?



Microarray

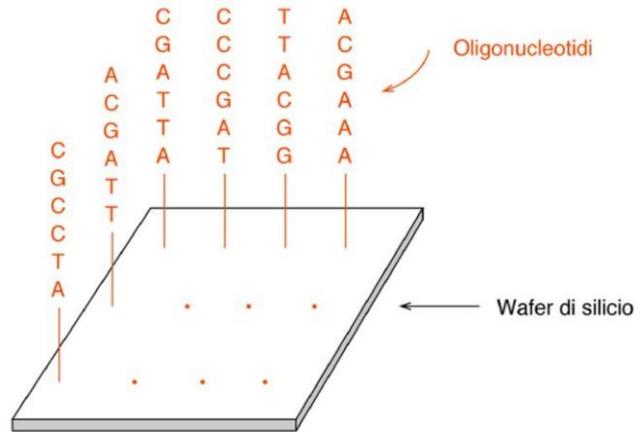


Analisi di una libreria di cDNA attraverso microarray

- **Microarray:** superficie (una piastra/chip) che contiene migliaia di sonde di DNA immobilizzate in posizioni ben definite.
 - Ogni **sonda** corrisponde a un **gene specifico**.
- **Campione:** estrazione di RNA cellulare, conversione a cDNA con marcatura con fluorescenza di tutti i cDNA (diversa fluorescenza per i diversi campioni di ibridazione ed esame microarray).

• Chip a DNA

- Microarray ad alta densità con oligo sintetizzati in superficie (1 milione/cm²),
- Uno o più oligo corrispondono a un DNA espresso (gene).



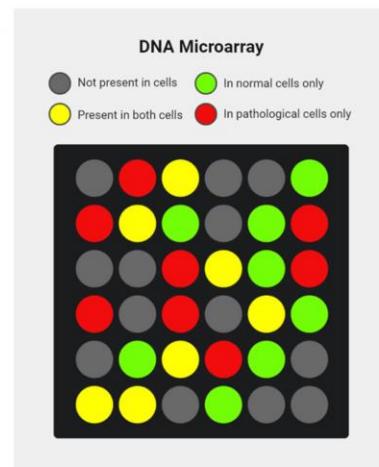
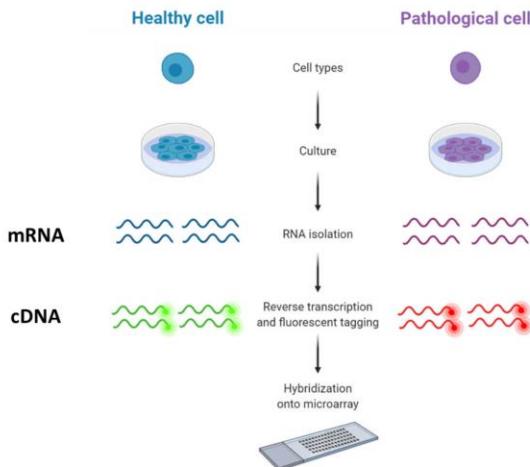
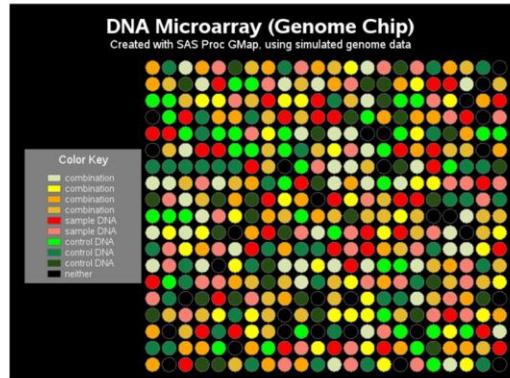
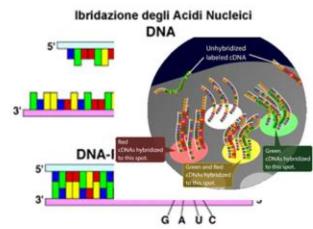
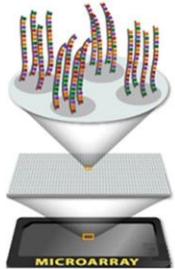


Image By Sagar Aryal, created using biorender.com

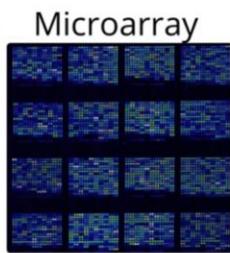
Microarray per l'analisi dell'espressione genica



Biological Sciences



Surescan Dx



Microarray

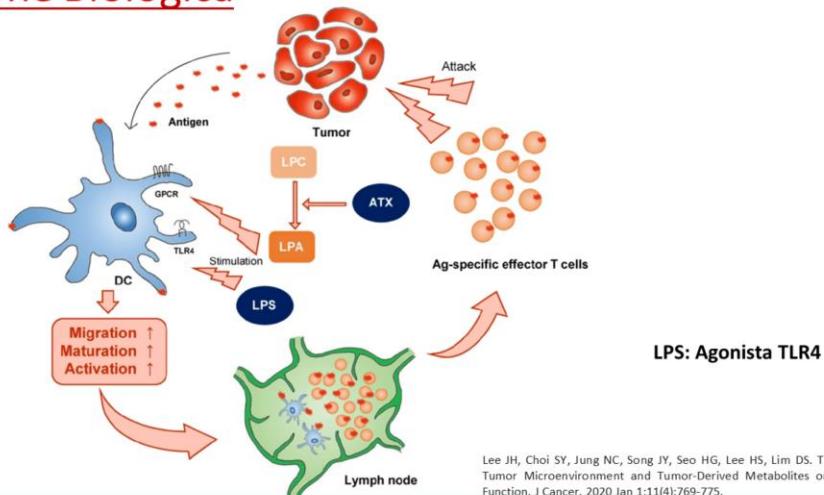


Gene chip 3000 7g

<https://www.youtube.com/watch?v=wZN070rl7VA&t=139s>

Questione Biologica

Classical stimulators induce DC activation. Classical stimulators (e.g. LPA and LPS) influence DC function.

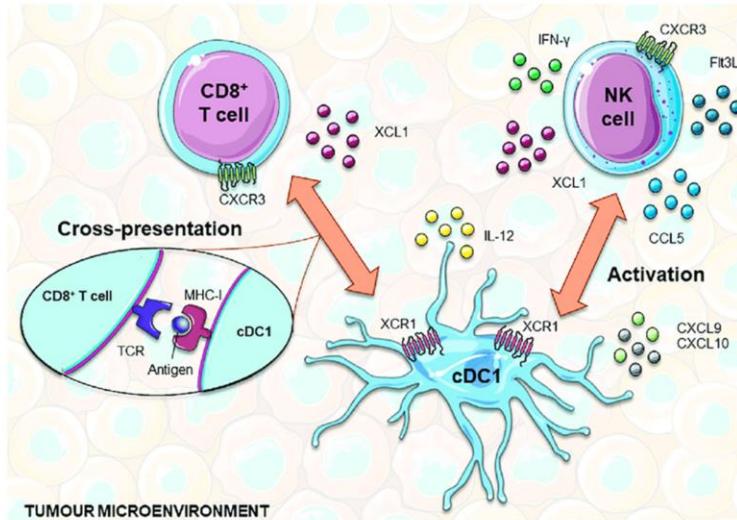


Lee JH, Choi SY, Jung NC, Song JY, Seo HG, Lee HS, Lim DS. The Effect of the Tumor Microenvironment and Tumor-Derived Metabolites on Dendritic Cell Function. *J Cancer*. 2020 Jan 1;11(4):769-775.

Classical stimulators induce DC activation. Classical stimulators (e.g. LPA and LPS) influence DC function. ATX (autotaxin), LPC (lyso-phosphatidylcholine), LPA (lysophosphatidic acid).

Sebbene il lipopolisaccaride (LPS) sia un fattore di maturazione comunemente utilizzato negli studi sui topi, l'interleuchina (IL)-1 β , l'IL-6, il fattore di necrosi tumorale (TNF)- α e la prostaglandina E2 (PGE2) sono comunemente utilizzati nella maturazione delle DC derivate dai monociti (mo-DC) per gli studi clinici. Lo stimolo di maturazione è un fattore importante che determina il successo o il fallimento della terapia con DC.

Gli agonisti TLR [il lipopolisaccaride agonista TLR-4 (LPS), il poli (I:C) agonista TLR-3 e il resiquimod agonista TLR-7/8], le citochine [TNF- α , IL-1 β , IL-6, interferone (IFN)- α e IFN- γ], i ligandi del recettore costimolatori (CD40L) e PGE2 sono stati tutti utilizzati da soli o in vari cocktail per maturare e programmare le DC. Il risultato di maturazione desiderato è indurre un'elevata espressione di molecole MHC, molecole costimolatorie come CD80, CD86 e CD40 e chemiochine come CCR7, nonché la secrezione di citochine Th1 come IFN- γ , in modo da polarizzare le DC verso l'attivazione Th1 18.



cDC1 interagisce con le cellule T CD8+ e NK per sviluppare risposte antitumorali. Le cellule NK e T CD8+ esprimono XCL1, che attrae XCR1+ cDC1 nel microambiente tumorale. Inoltre, le cellule NK possono anche produrre CCL5, aiutando a reclutare questo sottoinsieme di DC. A loro volta, le cDC1 sono la principale fonte delle chemiochine CXCL9 e CXCL10, chemioattrattivi per le cellule T e NK. Funzionalmente, le cDC1 sono altamente capaci di presentare in modo incrociato antigeni tumorali tramite MHC-I alle cellule T CD8+ e di produrre IL-12, che promuove la citotossicità delle cellule T e la produzione di INF-γ da parte delle cellule NK. Inoltre, le cellule NK producono Flt3L che mantiene la vitalità e le capacità funzionali delle cDC1 all'interno del microambiente tumorale e può anche promuovere la loro differenziazione locale dai precursori reclutati. cDC1, cellula dendritica classica 1; Flt3L, ligando della tirosina chinasi 3 tipo FMS; IFN-γ, interferone gamma; MHC-I, complesso maggiore di istocompatibilità I; NK, anticorpo naturale killer; TCR, recettore delle cellule T; XCL1, ligando della chemiochima motivo X-C 1.

GEO – Gene Expression Omnibus

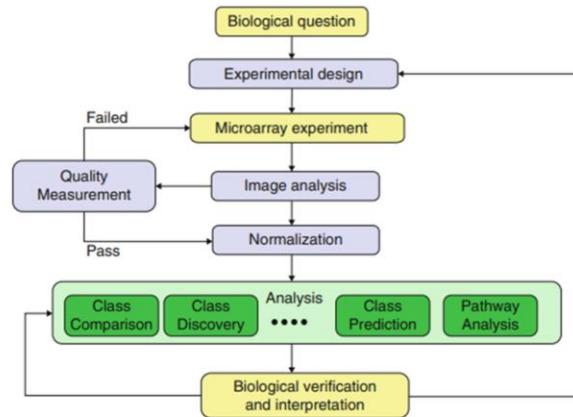
The screenshot shows the NCBI GEO Accession Display page for series GSE203450. At the top, there are links for HOME, SEARCH, SITE MAP, GEO Publications, FAQ, MIAME, Email GEO, and a login status. The main content area displays experimental details, including the title "Expression data from mouse dendritic cells (cDCs)", organism "Mus musculus", experiment type "Expression profiling by array", and a summary about LPS-induced tolerogenic enzymes in DC subsets. It also includes an overall design section and contributor/citation information.

GEO accession:
GSE203450

Series GSE203450 Query DataSets for GSE203450

Status: Public on Jun 14, 2022
Title: Expression data from mouse dendritic cells (cDCs)
Organism: *Mus musculus*
Experiment type: Expression profiling by array
Summary: We used microarrays to understand the role of LPS in inducing tolerogenic enzymes in DC subsets
Overall design: WT bone marrow derived DCs (BMDCs) were sort-purified and treated with LPS for RNA extraction and hybridization on Affymetrix microarrays. We sought to obtain gene expression signature that are controlled by LPS in DC subtypes
Contributor(s): Gargaro M, Murphy KM, Fallarino F
Citation(s): Gargaro M, Scallisi G, Manni G, Briseño CG et al. Indoleamine 2,3-dioxygenase 1 activation in mature cDC1 promotes tolerogenic education of inflammatory cDC2 via metabolic communication. *Immunity* 2022 Jun 14;55(6):1032-1050.e14. PMID: 35704993

Pipeline Microarray Analysis



Part 1. Load required libraries

```
-----#
# Pre-processing Microarray Data - QC, Normalization, DEGs and Visualization
#-----#
# Analysis of datasets from: GSE203450 (Microarray Affymetric Data from DCs)

# Set Working Directory.-----
setwd("") # Set the Working Directory on your computer.

# Part 1. Load required libraries.=====
print("loading libraries...")

# Install Bioconductor Manager (needed for some packages).-----
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

# Install the required libraries.-----
BiocManager::install("GEOquery")
library(GEOquery)

# BiocManager::install("limma")
library(limma)|
```

Dopo aver installato i pacchetti, prova ad aprirli con la funzione library. Se tutto è ok la funzione non dovrebbe restituire nessun warning.

Altri modi per aprire le biblioteche

➤ **require("Nome del pacchetto")**

```
> if (require(GEOquery)) {  
+   print("Pacchetto caricato con successo!")  
+ }
```

➤ Per i pacchetti non disponibili su CRAN, caricare da altre fonti come **GitHub**. Utilizza devtools o remotes per installare e caricare:

```
> if (!require(devtools)) install.packages("devtools")  
devtools::install_github("username/repository")  
library(package_name)|  
  
> devtools::install_github("tidyverse/ggplot2")  
library(ggplot2)|
```

Part 2. Import and explore data from GEO

```
# Part 2. Import and explore data from GEO.=====
# print("Importing data...")

id <- "GSE203450"
gse <- getGEO(id, GSEMatrix =TRUE, AnnotGPL=TRUE)

# Get some information about the file.-----
list(gse)
names(pData(gse[[1]]))      # print the sample info.
length(gse)                 # check how many platforms was used.

# Select the dataset.-----
gse <- gse[[1]]

# if more than one dataset is present, you can analyse the other dataset by
# changing the number inside the [...]
# e.g. gse <- gse[[2]]

# We can also use that kind of line.
# if (length(gse) > 1) idx <- grep("GPL6246", attr(gse, "names")) else idx <- 1

# Explore the expression dataframe.
pData(gse)                  # print the sample info.
fData(gse)                  # print the gene annotation.
exprs(gse)                  # print the expression data.
```

Assegna l'ID dello studio GEO (Gene Expression Omnibus) "GSE203450" alla variabile id. Questo ID identifica un dataset specifico che sarà scaricato dal database GEO. Usa la funzione getGEO() dal pacchetto **GEOquery** per scaricare i dati associati all'ID specificato. GSEMatrix = TRUE: indica che si vuole ottenere i dati come un oggetto di tipo **ExpressionSet**, che è ottimizzato per analisi di espressione genica in R. AnnotGPL = TRUE: richiede che i dati siano annotati utilizzando la piattaforma GPL (Gene Platform List) per aggiungere informazioni sulle sonde geniche. Il risultato, memorizzato in gse, è una lista che può contenere più oggetti ExpressionSet, uno per ogni piattaforma tecnologica utilizzata nello studio.

list(gse): Stampa la struttura di gse, che è una lista contenente uno o più oggetti ExpressionSet. Questo permette di esaminare il contenuto generale della variabile gse, come una panoramica.

names(pData(gse[[1]])): Estraie i metadati dei campioni associati al primo dataset (selezionato con gse[[1]]). La funzione pData() restituisce i **metadati del campione**, come informazioni sulle condizioni sperimentali, gruppi, codici dei campioni, ecc.; names() elenca le colonne disponibili in questi metadati, permettendo di vedere quali informazioni sono disponibili per i campioni.

length(gse): Determina quante piattaforme (tecniche sperimentali) sono state utilizzate nello studio. Ogni piattaforma è rappresentata da un elemento nella lista

gse. Se il valore restituito è 1, significa che c'è una sola piattaforma, e il dataset rilevante è gse[[1]]. Se il valore è maggiore di 1, lo studio ha utilizzato più piattaforme, e ciascuna deve essere analizzata separatamente.

```

> list(gse)
[[1]]
ExpressionSet (storageMode: lockedEnvironment)
assayData: 28853 features, 18 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: GSM6172191 GSM6172192 ... GSM6172208 (18 total)
  varLabels: title geo_accession ... tissue:ch1 (38 total)
  varMetadata: labelDescription
featureData
  featureNames: 10344614 10344616 ... 10608630 (28853 total)
  fvarLabels: ID Gene title ... GO:Component ID (21 total)
  fvarMetadata: Column Description labelDescription
experimentData: use 'experimentData(object)'
  pubMedIds: 35704993
Annotation: GPL6246

> names(pData(gse[[1]]))      # print the sample info.
[1] "title"                  "geo_accession"          "status"           "submission_date"
[5] "last_update_date"        "type"                 "channel_count"    "source_name_ch1"
[9] "organism_ch1"            "characteristics_ch1" "characteristics_ch1.1" "characteristics_ch1.2"
[13] "treatment_protocol_ch1" "growth_protocol_ch1"   "molecule_ch1"     "extract_protocol_ch1"
[17] "label_ch1"               "label_protocol_ch1"  "taxid_ch1"        "hyb_protocol"
[21] "scan_protocol"          "description"         "data_processing"  "platform_id"
[25] "contact_name"            "contact_phone"       "contact_department" "contact_institute"
[29] "contact_address"         "contact_city"        "contact_state"    "contact_zip/postal_code"
[33] "contact_country"        "supplementary_file"  "data_row_count"   "genotype:ch1"
[37] "strain:ch1"              "tissue:ch1"

> length(gse)                # check how many platforms was used.
[1] 1

```

Result

```

> pData(gse)                                # print the sample info.
   title geo_accession      status submission_date last_update_date type
GSM6172191    WT pDC_untreated_rep1  GSM6172191 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172192    WT pDC_untreated_rep1  GSM6172192 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172193    WT pDC_untreated_rep3  GSM6172193 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172194    WT pDC_activated_(LPS)_rep1  GSM6172194 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172195    WT pDC_activated_(LPS)_rep2  GSM6172195 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172196    WT pDC_activated_(LPS)_rep3  GSM6172196 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172197    WT cDC1_untreated_rep1  GSM6172197 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172198    WT cDC1_untreated_rep2  GSM6172198 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172199    WT cDC1_untreated_rep3  GSM6172199 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172200    WT cDC1_activated_(LPS)_rep1  GSM6172200 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172201    WT cDC1_activated_(LPS)_rep2  GSM6172201 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172202    WT cDC1_activated_(LPS)_rep3  GSM6172202 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172203    WT cDC2_untreated_rep1  GSM6172203 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172204    WT cDC2_untreated_rep2  GSM6172204 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172205    WT cDC2_untreated_rep3  GSM6172205 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172206    WT cDC2_activated_(LPS)_rep1  GSM6172206 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172207    WT cDC2_activated_(LPS)_rep2  GSM6172207 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
GSM6172208    WT cDC2_activated_(LPS)_rep3  GSM6172208 Public on Jun 14 2022  May 20 2022  Jun 14 2022 RNA
channel_count          source_name_ch1 organism_ch1           characteristics_ch1
GSM6172191          1 pDC isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172192          1 pDC isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172193          1 pDC isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172194          1 pDC isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172195          1 pDC isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172196          1 pDC isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172197          1 cDC1 isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172198          1 cDC1 isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172199          1 cDC1 isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172200          1 cDC1 isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172201          1 cDC1 isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172202          1 cDC1 isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172203          1 cDC2 isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs
GSM6172204          1 rmc2 isolated from WT Flt3-BMDCs Mus musculus tissue: Bone marrow derived DCs

```

Result

> fData(gse) # print the gene annotation.

ID	Gene title
10344614 10344614	predicted gene 2889
10344616 10344616	
10344618 10344618	
10344620 10344620	
10344622 10344622	predicted gene 10568
10344624 10344624	
10344633 10344633	lysophospholipase 1
10344637 10344637	transcription elongation factor A (SII) 1
10344653 10344653	ATPase, H ⁺ transporting, lysosomal V _H subunit H
10344658 10344658	opioid receptor, kappa 1
10344674 10344674	RBL-inducible coiled-coil 1
10344679 10344679	family with sequence similarity 150, member A
10344705 10344705	suppression of tumorigenicity 18
10344707 10344707	
10344713 10344713	protein-L-isoaspartate (D-aspartate) O-methyltransferase domain containing 1
10344715 10344715	S-adenosylhomocysteine hydrolase//predicted gene 4737
10344717 10344717	predicted gene, 30414
10344719 10344719	ring finger protein 7 pseudogene//ring finger protein 7
10344721 10344721	
10344723 10344723	ribosome biogenesis regulator 1
10344725 10344725	alcohol dehydrogenase, iron containing, 1
10344741 10344741	heterogeneous nuclear ribonucleoprotein A3 pseudogene//heterogeneous nuclear ribonucleoprotein A3
10344743 10344743	RIKEN cDNA 3110035E14 gene
10344750 10344750	serum/glucocorticoid regulated kinase 3
10344772 10344772	minichromosome maintenance domain containing 2
10344789 10344789	centrosome and spindle pole associated protein 1
10344797 10344797	centrosome and spindle pole associated protein 1
10344799 10344799	centrosome and spindle pole associated protein 1
10344801 10344801	centrosome and spindle pole associated protein 1
10344803 10344803	centrosome and spindle pole associated protein 1

Result

```

> exprs(gse)          # print the expression data.
   GSM6172191 GSM6172192 GSM6172193 GSM6172194 GSM6172195 GSM6172196 GSM6172197 GSM6172198 GSM6172199 GSM6172200
10344614 109.847 88.484 107.828 102.969 103.437 114.278 112.667 109.742 116.739 129.456
10344616 6.427 6.455 7.069 6.717 6.587 7.585 6.362 6.687 6.393 6.200
10344618 9.538 8.063 8.625 8.244 9.297 9.263 8.470 8.333 8.131 9.918
10344620 27.262 26.002 25.437 21.966 23.633 25.847 26.793 29.451 25.089 30.966
10344622 169.643 146.881 143.827 134.991 177.279 170.648 133.768 153.171 181.502 145.824
10344624 266.608 239.226 250.661 218.589 228.583 271.410 237.238 247.015 290.231 222.051
10344633 751.896 539.310 677.006 654.438 635.193 677.426 647.876 727.015 588.344 693.568
10344637 285.090 367.982 303.612 358.949 354.451 362.866 233.057 262.295 268.416 240.799
10344653 18.618 15.632 18.166 16.192 14.598 14.217 21.143 16.423 16.219 12.631
10344658 236.535 285.525 254.945 242.434 274.885 221.629 267.975 265.944 276.128 304.479
10344674 14.193 13.903 14.432 14.665 12.702 14.385 14.049 13.202 15.903 15.740
10344679 64.290 74.270 75.216 87.059 85.975 79.868 60.216 65.746 75.252 97.496
10344705 111.926 99.328 101.250 96.119 113.184 96.298 128.605 125.420 113.499 136.842
10344707 263.257 299.612 310.842 234.002 239.783 224.276 274.102 261.410 224.838 250.679
10344713 264.830 264.658 241.719 204.200 232.481 304.528 191.195 208.750 240.040 219.350
10344715 29.725 27.262 24.757 26.610 27.515 26.435 25.673 24.806 25.196 26.940
10344717 51.700 61.483 93.079 62.350 63.386 42.714 63.403 84.022 73.543 51.950
10344719 90.737 88.424 96.520 85.889 79.839 101.214 88.499 89.431 97.980 92.323
10344721 6.153 6.544 6.397 6.319 6.784 6.637 6.329 7.031 6.142 6.603
10344723 210.302 346.573 249.945 225.517 274.970 248.559 213.344 227.629 279.853 205.144
10344725 66.629 67.864 70.005 88.766 84.158 81.080 63.289 62.486 73.613 62.853
10344741 607.019 892.090 849.388 836.131 862.923 719.113 693.475 792.967 686.353 761.475

```

Result

Part 3. Group membership for all samples

```
# Part 3. Group membership for all samples.=====
print("Grouping the samples...")  
  
# Select one comparison between groups.  
cdc1_lps <- "XXXXXX111000XXXXXX"           # (cDC1 LPS vs cDC1 Ctrl)  
  
# Split Samples.-----  
sm1 <- strsplit(cdc1_lps, split = "")[[1]]
```

0 è la condizione che voglio
studiare e 1 è il mio controllo

La funzione strsplit() divide una stringa (groups) in base a un carattere specificato come separatore. Qui, il separatore è una stringa vuota (""), che indica di dividere il testo in **singoli caratteri**. Poiché strsplit() restituisce una lista (anche se contiene un solo elemento), si utilizza [[1]] per accedere al primo (e unico) elemento della lista, che è un vettore di caratteri.

Dopo aver selezionato il dataset con gse[[1]], l'oggetto diventa un **ExpressionSet**, un formato standard usato in R per analisi di dati di espressione genica, semplificando il lavoro con funzioni successive.

Part 4. Filter out excluded samples and create the Expression matrix

```

# Part 4. Filter out excluded samples and Create the expression matrix.=====
print("Filtering out excluded samples...")
sel <- which(sml != "X") # excluded samples marked as "X".
sml <- sml[sel]
gse <- gse[,sel]

head(gse)

# Create the expression matrix.-----
ex <- exprs(gse)
ex[which(ex <= 0)] <- NaN ← sostituisce valori uguali o inferiori a 0 con "NaN" (Not a Number), perché valori negativi o pari a zero non hanno alcun significato biologico in termini di espressione genica e possono falsare l'analisi.
head(ex)
hist(ex)

```

Questa sezione filtra i campioni che devono essere esclusi dall'analisi. Filtra gli indici dei campioni che non sono contrassegnati con "X". Questi sono i campioni da tenere per l'analisi. Aggiorna il vettore sml per includere solo i campioni selezionati. Aggiorna i dati (gse), mantenendo solo le colonne (campioni) selezionati. Mostra le prime righe dei dati per confermare che il filtro è stato applicato correttamente.

Creazione della matrice di espressione

Estrae la matrice delle espressioni geniche dai dati gse. Sostituisce tutti i valori minori o uguali a 0 con NaN (valori mancanti), poiché i valori negativi o nulli non sono validi per la log-trasformazione successiva. Crea un istogramma per visualizzare la distribuzione iniziale dei dati.

Matrice di Espressione

- L'expression matrix (matrice di espressione) è una rappresentazione fondamentale dei dati di espressione genica.
- Si tratta di una matrice in cui le **righe** rappresentano **geni o trascritti** e le **colonne** rappresentano **campioni o condizioni** sperimentali.

	Cell1	Cell2	...	CellN	campioni
trascritti	Gene1	3	2	.	13
	Gene2	2	3	.	1
	Gene3	1	14	.	18

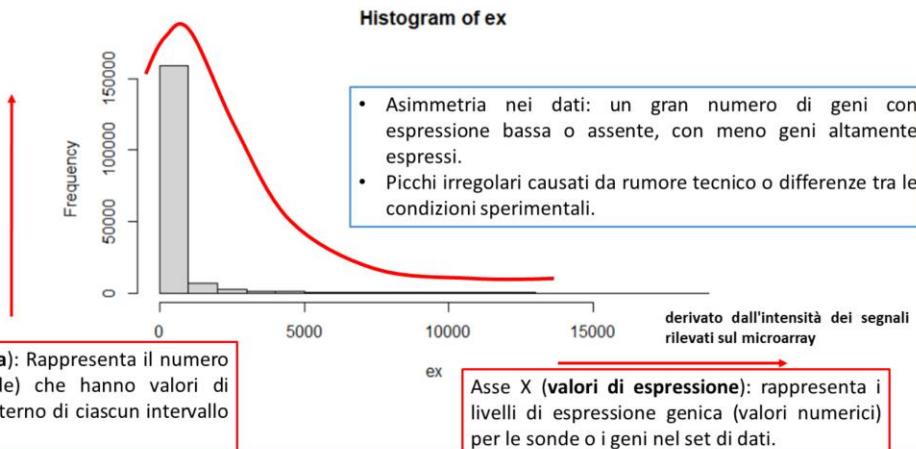
	GeneM	25	0	.	0

Questa matrice contiene i valori di espressione per ciascun gene o sonda sul microarray.

Per qualsiasi analisi omica che esegui, considera la matrice di espressione come il cuore dell'analisi.

Iistogramma - Espressione

L'istogramma rappresenta il numero di geni (o sonde) i cui valori di espressione rientrano in un certo intervallo.



La maggior parte dei geni generalmente mostra una bassa o nessuna espressione (segale vicino allo zero). Ciò è prevedibile, poiché solo una frazione dei geni è altamente espressa in un dato momento. Alcuni geni con valori di espressione molto elevati indicano geni che sono altamente espressi in determinate condizioni. I geni con valori eliminati a causa del comando `ex[quale(ex <= 0)] <- NaN` non compaiono nell'istogramma, il che potrebbe influenzare la forma finale della distribuzione.

Part 5. Log2 transformation and Normalization of the data

```
# Part 5. Log2 transformation and Normalization of the data.=====
print("Transforming and Normalizing data...")

# Box-and-whisker plot (all samples before normalization).-----
par(mar=c(7,4,2,1))
title <- paste ("GSE203450", "/", annotation(gse), sep ="")

# pdf("Microarray_data_before_transformation.pdf")
boxplot(ex, boxwex=0.7, notch=T, main=title, outline=FALSE, las=2)
# dev.off()
```

boxwex = Regola la larghezza relativa dei riquadri nel boxplot.
notch = rappresenta l'intervallo di confidenza attorno alla mediana.
main = Imposta il titolo del grafico.
outline = «outliers». comune nei microarray, ma non sempre rilevante.
las = Imposta l'orientamento delle etichette dell'asse X.

I dati grezzi presentano distribuzioni molto particolari, per cui a volte è difficile analizzarli e soprattutto creare modelli senza una qualche pre-elaborazione. L'obiettivo generale della trasformazione dei nostri dati è creare una distribuzione più normale (*gaussiana*), ovvero una curva a campana. Questa sezione trasforma e normalizza i dati per prepararli per un'analisi statistica più robusta.

Boxplot prima della normalizzazione

Creare un boxplot per ogni campione per visualizzare la distribuzione prima della normalizzazione.

Parametri: **boxwex=0.7**: Regola la larghezza delle box.

notch=T: Visualizza un'incisione (notch) per evidenziare la mediana.

outline=FALSE: Rimuove i valori anomali (outliers) dal boxplot.

las=2: Ruota le etichette degli assi.

Log2 transformation

Applica una trasformazione logaritmica in base 2 per stabilizzare la varianza e rendere i dati più simmetrici.

Boxplot dopo la trasformazione

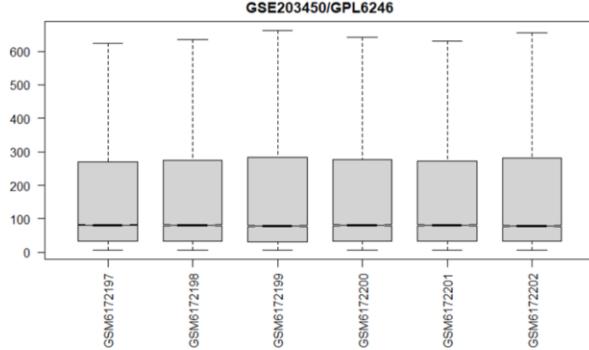
Ripete il boxplot, ma sui dati trasformati log2 per confrontare le distribuzioni.

Distribuzione dei valori

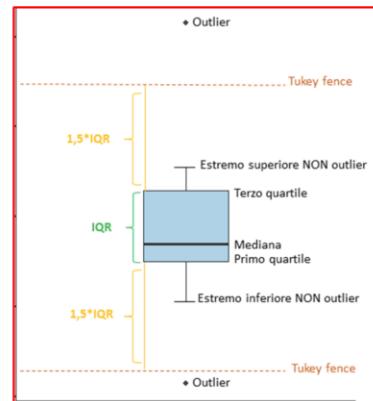
Genera un grafico delle densità per visualizzare la distribuzione dei valori dopo la trasformazione log2. Questo aiuta a verificare che i dati siano stati uniformati.

BoxPlot – before transformation

Il boxplot fornisce informazioni sulla mediana, sui quartili e sui valori anomali dei valori di espressione genica per ciascun campione.



È particolarmente utile per confrontare le distribuzioni tra diversi campioni e verificare eventuali errori sperimentali o la necessità di normalizzazione.

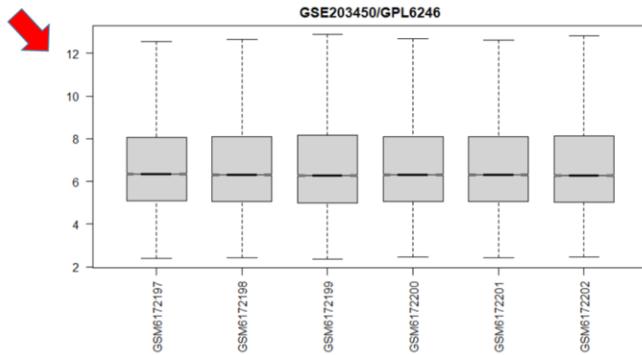


Nelle statistiche descrittive, un box plot o boxplot (noto anche come box and whisker plot) è un tipo di grafico spesso utilizzato nell'analisi delle variabili quantitative. I box plot mostrano visivamente la distribuzione dei dati numerici e l'asimmetria visualizzando i quartili (o percentili) e le medie dei dati. I box plot mostrano il riepilogo in cinque numeri di un set di dati, tra cui il punteggio minimo, il primo quartile (inferiore), la mediana, il terzo quartile (superiore) e il punteggio massimo.

Part 5. Log2 transformation and Normalization of the data

```
# Log2 transformation.-----  
exprs(gse) <- log2(ex)      # Log2 transform.  
  
# Box-and-whisker plot (after log2 transformation).-----  
par(mar=c(7,4,2,1))  
title <- paste ("GSE203450", "/", annotation(gse), sep = "")  
  
# pdf("Microarray_data_after_transformation.pdf")  
boxplot(exprs(gse), boxwex=0.7, notch=T, main=title, outline=FALSE, las=2)  
# dev.off()  
  
# Expression value distribution plot.-----  
par(mar=c(4,4,2,1))  
title <- paste ("GSE203450", "/", annotation(gse), " value distribution",  
              sep = "")  
  
# pdf("Densities_after_normalization.pdf")  
plotDensities(ex, main=title, legend=F)  
# dev.off()
```

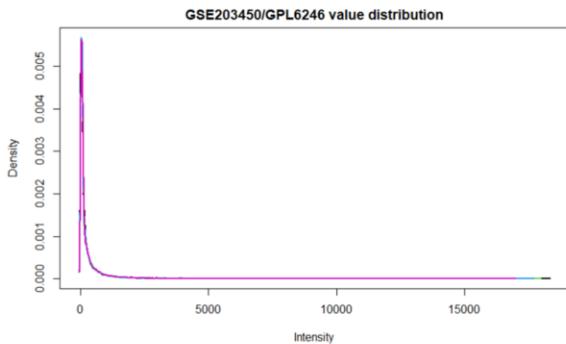
BoxPlot – after transformation



Trasformando i nostri dati non solo normalizziamo le osservazioni, ma anche i residui. La normalizzazione rende i modelli di training meno sensibili alla scala delle caratteristiche, quindi possiamo risolvere meglio i coefficienti.

Trasformando i nostri dati non solo normalizziamo le osservazioni, ma anche i residui.

Densità – Distribuzione



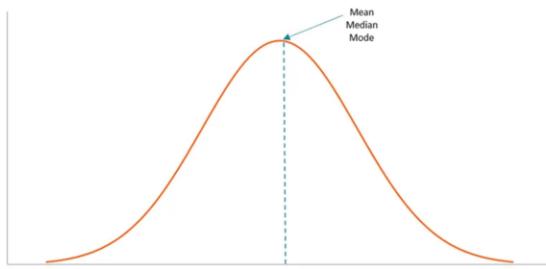
Il diagramma della densità è una visualizzazione che rappresenta la distribuzione di probabilità stimata dei valori di espressione genica per ciascun campione. In altre parole:

- Visualizza la frequenza relativa dei valori di espressione lungo una scala continua.
- Ogni riga del grafico corrisponde ad un campione (colonna della matrice ex).
- Il grafico della densità è un'alternativa fluida all'istogramma.

Result

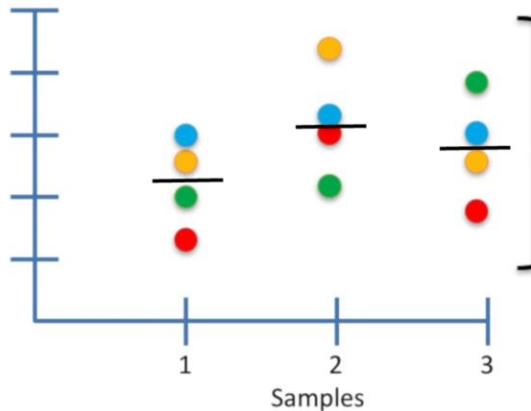
Normalizzazione - Quantile

- L'obiettivo generale della Normalizzazione dei nostri dati è creare una distribuzione più normale (*Gaussiana*), ovvero una curva a campana. **In generale, le distribuzioni normali tendono a produrre risultati migliori** in un modello perché ci sono osservazioni pressoché uguali sopra e sotto la media e la media e la mediana sono le stesse.
- I modelli vengono eseguiti presupponendo che i dati siano distribuiti normalmente.



La normalizzazione rende i modelli di training meno sensibili alla scala delle caratteristiche, quindi possiamo risolvere meglio i coefficienti. I coefficienti sono misure statistiche del grado in cui le variazioni del valore di una variabile prevedono la variazione del valore di un'altra variabile. La normalizzazione e il ridimensionamento sono due tipi di trasformazioni importanti nella pulizia dei dati.

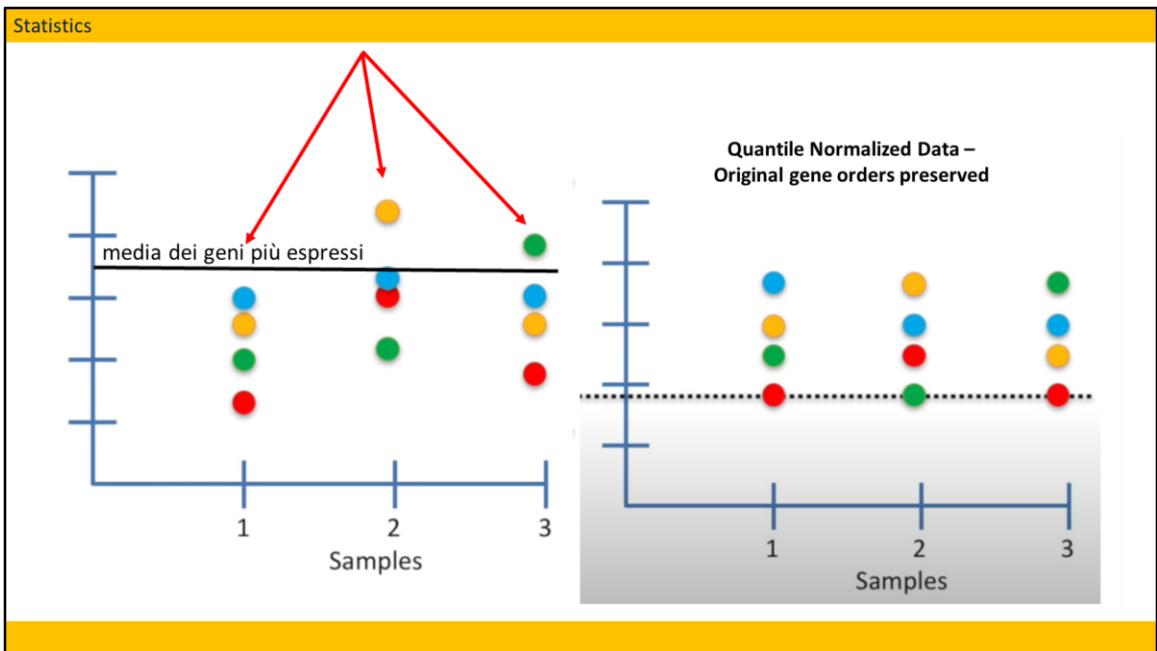
Raw Gene Expression Data



Here's our data. In this graph, each color represents a different gene.

- Ogni campione ha un valore medio diverso, suggerendo che dobbiamo **compensare le diverse intensità complessive della luce**.
- **Normalizzazione quantile (Quantile Normalization)** può correggere questo artefatto tecnico.

Quantile Normalization.



Quantile Normalization.

Part 6. Normalization log-ratios with limma

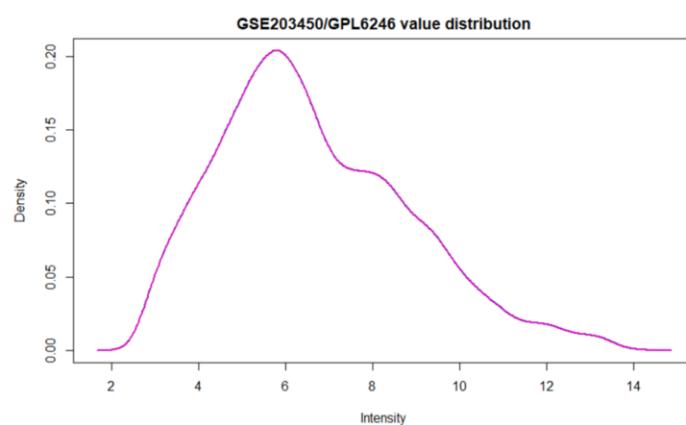
```
# Part 6. Normalization log-ratios with Limma.=====
print("Normalizing data with Limma...")
exprs(gse) <- normalizeBetweenArrays(exprs(gse))      # normalize data
gse
plotDensities(gse, main=title, legend=F)
```

Questa sezione utilizza il pacchetto *Limma* per normalizzare ulteriormente i dati.
normalizeBetweenArrays(exprs(gse)): Normalizza i dati per rendere le distribuzioni dei valori comparabili tra i campioni. Questa normalizzazione corregge eventuali differenze tecniche tra i campioni.

exprs(gse) <- normalizeBetweenArrays(exprs(gse)): Sostituisce i valori nella matrice di espressione con i valori normalizzati.

Statistics

Densità – Distribuzione



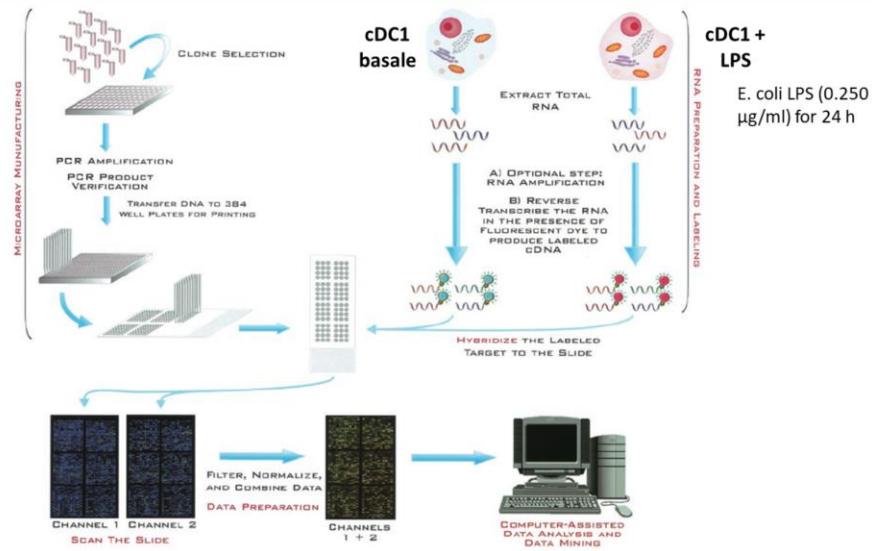
Result

Next lesson...

- PCA / UMAP
- Differential Expressed Genes (DEGs)
- Volcano Plot
- Enrichment Analysis
 - Gene Ontology
 - GSEA



Biological Sciences - Microarray



[Application of Microarrays to the Analysis of Gene Expression in Cancer](#)

L'analisi microarray comporta la rottura di una cellula, l'isolamento del suo contenuto genetico, l'identificazione di tutti i geni che sono attivati in quella particolare cellula e la generazione di un elenco di tali geni. L'analisi microarray del DNA è una tecnica che gli scienziati utilizzano per determinare se i geni sono attivati o disattivati.

Riepilogo – GEO2R Pipeline

GEO Datasets



- `getGEO()`

I dati GEO hanno quattro tipi di entità tra cui **GEO Platform (GPL)**, **GEO Sample (GSM)**, **GEO Series (GSE)** e curated **GEO DataSet (GDS)**.

Probe ID	Gene/transcript	Sample
> <code>exprs(gse)</code>		
		# print
10344614	109.847	88.484
10344616	6.427	6.455
10344618	9.538	8.063
10344620	27.262	26.002
10344622	169.643	146.881
10344624	266.608	239.226
10344633	751.896	539.310
10344637	285.090	367.982
10344653	18.618	15.632
10344658	236.535	285.525
		254.945

The NCBI Gene Expression Omnibus ([GEO](#)) is the largest public repository of high-throughput microarray experimental data. GEO data have four entity types including GEO Platform (GPL), GEO Sample (GSM), GEO Series (GSE) and curated GEO DataSet (GDS).

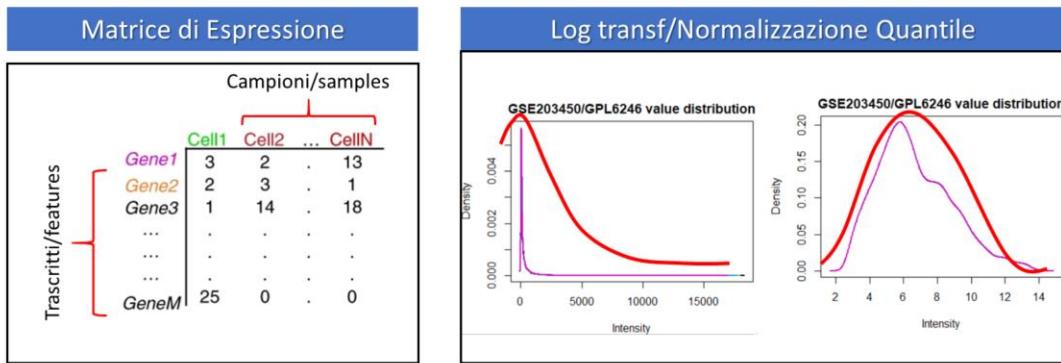
A Platform record describes the list of elements on the array in the experiment (e.g., cDNAs, oligonucleotide probesets). Each Platform record is assigned a unique and stable GEO accession number (GPLxxx).

A Sample record describes the conditions under which an individual Sample was handled, the manipulations it underwent, and the abundance measurement of each element derived from it. Each Sample record is assigned a unique and stable GEO accession number (GSMxxx).

A Series record defines a group of related Samples and provides a focal point and description of the whole study. Series records may also contain tables describing extracted data, summary conclusions, or analyses. Each Series record is assigned a unique GEO accession number (GSExxx).

A DataSet record (GDSxxx) represents a curated collection of biologically and statistically comparable GEO Samples. GEO DataSets (GDSxxx) are curated sets of GEO Sample data.

Riepilogo – GEO2R Pipeline



Irizarry RA, Hobbs B, Collin F, Beazer-Barclay YD, Antonellis KJ, Scherf U, Speed TP. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*. 2003 Apr;4(2):249-64.

Part 7. Assign samples to groups and set up design matrix

```
# Part 7. Assign samples to groups and set up design matrix.=====
# Define groups to compare.-----
groups <- make.names(c("cDC1_LPS", "cDC1_ctrl"))

gs <- factor(sml)           Converte sml in un fattore categorico
levels(gs) <- groups       Reimposta i nomi delle categorie nel fattore (gs)
gse$group <- gs
design <- model.matrix(~group + 0, gse)   Memorizza le informazioni sul gruppo sperimentale direttamente
colnames(design) <- levels(gs)             nell'oggetto gse

gse <- gse[complete.cases(exprs(gse)), ] # skip missing values
```

Questa parte del codice è una preparazione per l'analisi statistica differenziale con il pacchetto Limma. **Imposta i gruppi sperimentali**, associa i campioni a questi gruppi e crea una matrice di progettazione, essenziale per la modellazione lineare utilizzata nell'analisi.

Modello Matrice Design

cDC1_LPS	cDC1_ctrl
1	0
1	0
1	0
0	1
0	1
0	1

Notes: +0 = Indica che vogliamo una matrice senza l'intercetta (termine costante), creando colonne separate per ciascun gruppo sperimentale. Ogni colonna corrisponde a un gruppo, consentendo confronti diretti tra loro.
 Colnames = Definisce i nomi delle colonne della matrice di progettazione come gruppi sperimentali.

Part 7. Assign samples to groups and set up design matrix

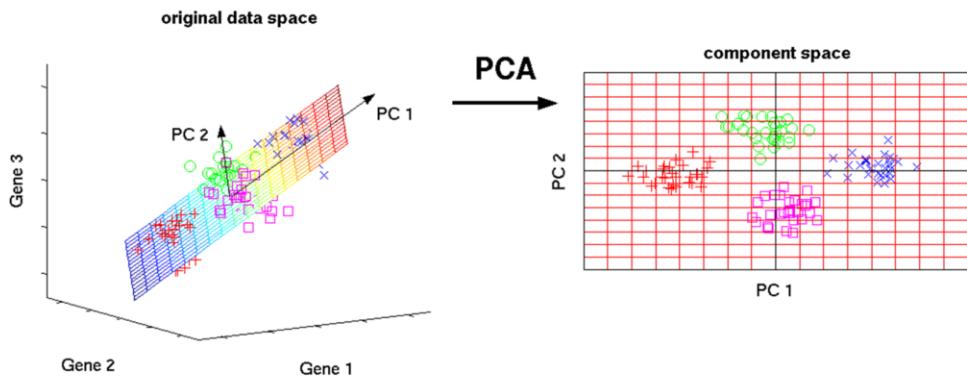
```
# UMAP plot (dimensionality reduction).-----
# library("maptools") # point labels without overlaps
library("umap")
library("car")

ex <- na.omit(ex) # eliminate rows with NAs
ex <- ex[!duplicated(ex), ] # remove duplicates

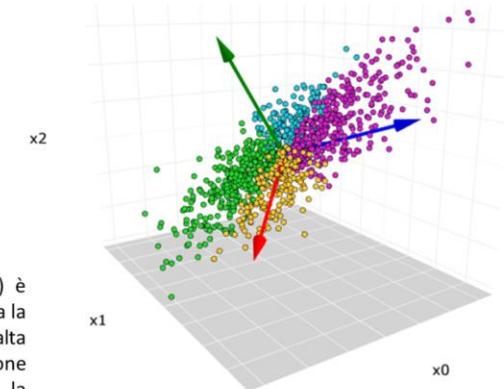
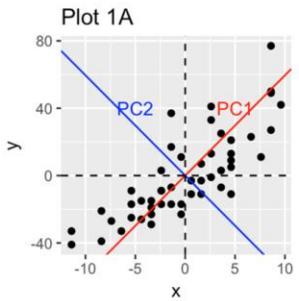
ump <- umap(t(ex), n_neighbors = 3, random_state = 123)
par(mar=c(3,3,2,6), xpd=TRUE)
plot(ump$layout, main="UMAP plot", nbrs=3, xlab="", ylab="", col=gs, pch=20, cex=1.5)
legend("topright", inset=c(-0.15,0), legend=levels(gs), pch=20,
       col=1:nlevels(gs), title="Group", pt.cex=1.5)
pointLabel(ump$layout, labels = rownames(ump$layout), method="SANN", cex=0.6)
```

Statistics

PCA (Principal Component Analysis) è una tecnica di riduzione dimensionale che trasforma dati complessi in un insieme di variabili principali (componenti) per catturare quanta più variazione possibile e facilitare la visualizzazione e l'analisi.

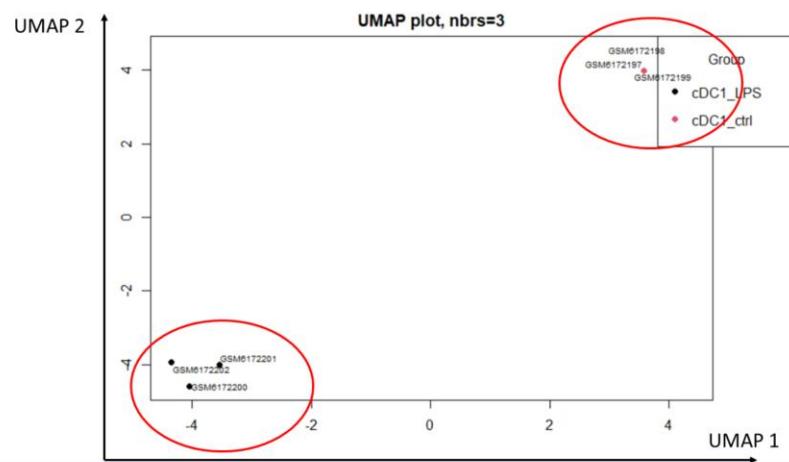


Statistics



UMAP (Uniform Manifold Approximation and Projection) è una tecnica di riduzione della dimensionalità che preserva sia la struttura locale che globale dei dati in spazi ad alta dimensionalità. È ampiamente utilizzato per la visualizzazione 2D o 3D, soprattutto nell'analisi di dati complessi come la genomica e l'apprendimento automatico, evidenziando cluster e modelli nei dati.

UMAP plot - riduzione della dimensionalità



Part 8. Differential Expression Genes

Precision Weight Calculation and Plotting the Mean-Variance Trend

Un oggetto contenente i dati sull'espressione genica.

La matrice di design per l'esperimento

```
# Part 8. Differential expression with Limma-Voom.
# Calculating the differential gene expression...
v <- vooma(gse, design, plot=T)
v$genes <- fData(gse) # attach gene annotations
```

Fit the Linear Model

```
# Fit linear model.
fit <- lmFit(v)
```

Questa funzione adatta un modello lineare ai dati (dopo la modellazione della varianza con vooma)

Part 8. Differential Expression Genes

Set Up Contrasts of Interest

Crea un contrasto, un confronto tra i due gruppi nell'esperimento. Il contrasto è scritto in un modo che sottrae il secondo gruppo dal primo (ad esempio, "gruppo1 - gruppo2").

```
# Set up contrasts of interest and recalculate model coefficients.
cts <- c(paste(groups[1], "-", groups[2], sep=""))
cont.matrix <- makeContrasts(contrasts=cts, levels=design)
fit2 <- contrasts.fit(fit, cont.matrix)
```

Questa funzione crea una matrice di contrasto dalle definizioni di contrasto (cts) basate sulla matrice di progettazione (design).

Questa funzione applica la moderazione empirica di Bayes agli errori standard del modello lineare.
Questo passaggio stabilizza le stime della varianza e migliora la potenza statistica dei test di ipotesi.

Compute Statistics and Table of Top Significant Genes

```
# Compute statistics and table of top significant genes.
fit2 <- eBayes(fit2, 0.01)
tT <- topTable(fit2, adjust="fdr", sort.by="B", number=250)
```

Bayesian statistics

Questa funzione applica la matrice di contrasto (cont.matrix) al modello lineare precedentemente adattato (fit).

Questa funzione estrae i geni maggiormente espressi in modo differenziale in base al modello adattato.

Inferenza Statistica

- Spesso capita di voler confermare i valori di una variabile in due popolazioni. In questo caso, assumendo che la variabile sia numerica continua, la domanda è: **la variabile ha una media significativamente diversa nelle due popolazioni?**

- **Verifica d'ipotesi**

- L'ipotesi Nulla (H_0)
- L'ipotesi Alternativa (H_1) -> La negazione dell'ipotesi Nulla

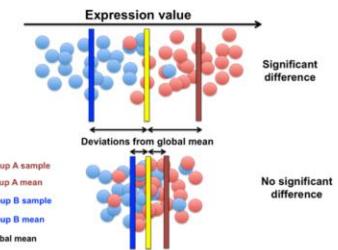
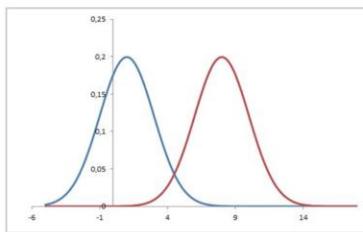
$$H_0 : \mu_1 = \mu_2$$

$$H_0 : \mu_1 \leq \mu_2$$

P-value

Livello di significatività (*p-value*)

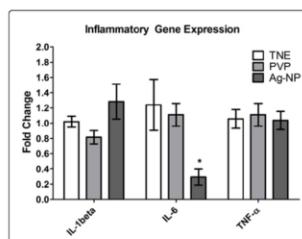
- Per avere maggiori informazioni sull'effettiva probabilità di osservare un certo valore del test nella distribuzione sotto H_0 , viene solitamente riportato il *p-value*.
- Il *p-value*, quindi, è una probabilità con valori che vanno da 0 a 1: valori piccoli del *p-value* portano a rifiutare H_0 .



P-value

Fold Change

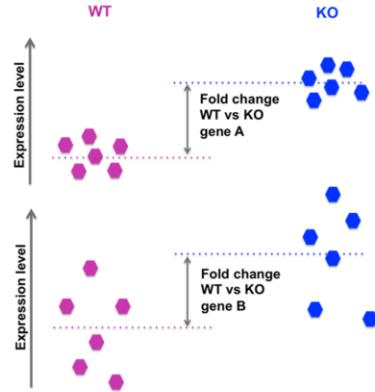
- Il fold change è un modo semplice per descrivere quanto un valore **cambia** rispetto a un riferimento.
 - Se il valore raddoppia rispetto al riferimento, il fold change è 2.
 - Se il valore si dimezza, il fold change è 0,5.
 - Un fold change di 1 significa che non c'è stato alcun cambiamento.



P-value

Log-Fold Change (*logFC*)

- Il fold change è spesso trasformato in scala logaritmica, come il log₂ fold change:
 - $\text{Log}_2(\text{Fold Change}) = 1 \rightarrow$ il gene ha **raddoppiato** l'espressione.
 - $\text{Log}_2(\text{Fold Change}) = -1 \rightarrow$ l'espressione si è **dimezzata**.
 - $\text{Log}_2(\text{Fold Change}) = 0 \rightarrow$ **nessun cambiamento** nell'espressione.



P-value

Part 9. Summarize test results

Adjusted P-Values and Histogram, Summarize Test Results

p-value = significatività statistica
Fold-change = entità del cambiamento

```
# Part 9. Visualize adj p-values, Venn diagram and QQ plot.=====
print("Visualizing adj p-values and venn diagram...")

# Build histogram of P-values for all genes. Normal test.-----
# assumption is that most genes are not differentially expressed.
tT2 <- topTable(fit2, adjust="fdr", sort.by="B", number=Inf)
hist(tT2$adj.P.Val, col = "grey", border = "white", xlab = "P-adj",
     ylab = "Number of genes", main = "P-adj value distribution")
```

```
# Summarize test results as "up", "down" or "not expressed".-----
dT <- decideTests(fit2, adjust.method="fdr", p.value=0.05, lfc=0.5)
```

Classifica i geni in categorie in base alle soglie statistiche

LogFC threshold

L'output, dT, è una matrice in cui ogni elemento indica se un gene è:

- 1: Upregulated.
- -1: Downregulated.
- 0: Not significantly differentially expressed.

Part 10. DEGs List

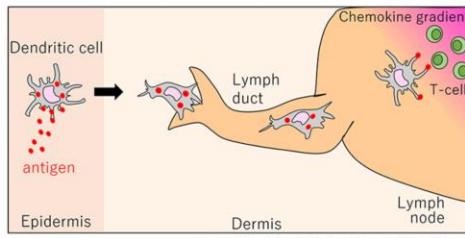
Creating the DEGs List and Annotating Differential Expression Status

```
# Part 10. DEGs list.=====
print("Creating a DEGs list...")
library(tidyverse)
DEG <- tT2[c(3,1,22,26)] %>%
  setNames(c("gene","id","logFC","padj"))

# DEGs selection (Define groups and cut-off points of logFC and padj).-----
DEG$Enriched <- "NS"
DEG$Enriched[DEG$logFC > 0.5 & DEG$padj < 0.05] <- "cDC1_LPS"
DEG$Enriched[DEG$logFC < -0.5 & DEG$padj < 0.05] <- "cDC1_ctrl"
#
```

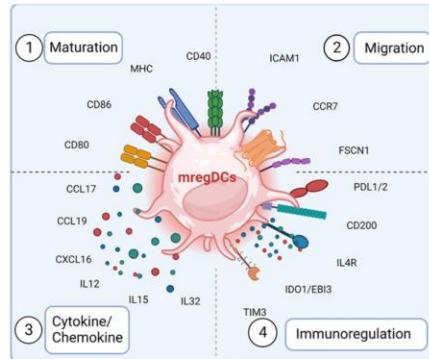
Biological Sciences

- Later, around 4 h after LPS activation, DCs show recovery of migratory ability and start to progressively lose their antigen uptake function until the mature stage in which they show poor antigen uptake and migratory activity (Granucci et al., 1999).



<https://bsw3.naist.jp/eng/bsedge/0011.html>

Granucci F, Ferrero E, Foti M, Aggujaro D, Vettoretti K, Ricciardi-Castagnoli P. Early events in dendritic cell maturation induced by LPS. *Microbes Infect*. 1999 Nov;1(13):1079-84.

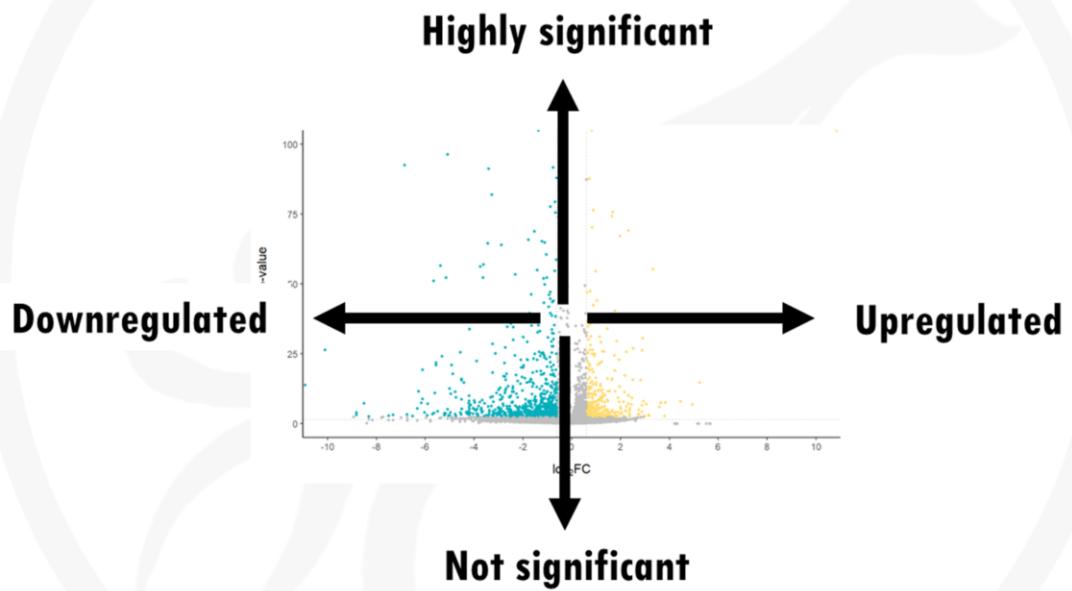


Clinical & Translational Med, Volume: 13, Issue: 2, First published: 17 February 2023, DOI: (10.1002/ctm2.1199)

Part 11. Volcano plot Representation ($\log FC = |0.5|$)

```
# a) Select genes to highlight.=====
genes <- DEG %>%
  filter(gene %in% c("Il6", "Il1b", "Tnf", "Cd86", "Cd80", "Clec9a", "Xcr1"))

# visualization Volcano Plot.=====
ggplot(data = DEG,
       aes(x = logFC,
           y = -log10(padj))) +
  geom_point(aes(colour = Enriched),
             alpha = 0.5,
             shape = 16,
             size = 1) +
  geom_point(data = genes,
             shape = 21,
             size = 2,
             fill = "black",
             colour = "black") +
  theme_classic() +
  geom_hline(yintercept = -log10(0.05),
             linetype = "dashed") +
  geom_vline(xintercept = c(log2(0.7071), log2(1.4142)),
             linetype = "dashed") +
  geom_label_repel(data = genes, # Add labels last to appear as the top layer
                  max.overlaps = Inf,
                  aes(label = gene, fontface = 'italic',
                      force = 1, nudge_y = 0.5) +
  scale_colour_manual(values = cols) +
  # scale_x_continuous(breaks = c(seq(-4, 5, 2)),
  #                     limits = c(-4, 5)) +
  ggtitle("BMDCs/cDC1-ctrl versus cDC1-LPS enriched genes")
```



Enrichment Analysis

- L'analisi di arricchimento è un metodo che identifica se insiemi predefiniti di geni o tratti sono significativamente sovrarappresentati in un elenco di geni, aiutando a interpretare processi o funzioni biologici associati a condizioni specifiche.

Differenza essenziale:

Gene Ontology (GO): si concentra sull'associazione funzionale di uno specifico sottoinsieme di geni (come quelli espressi in modo differenziale) con funzioni o processi.

GSEA: esamina tutti i geni e identifica set di geni predefiniti arricchiti in un contesto più ampio, anche senza utilizzare un limite rigoroso.

Part 12. GO (Gene Ontology) Enrichment Analysis

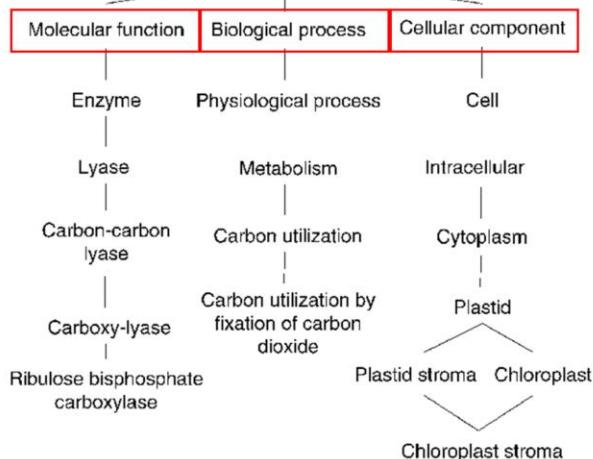
```
# Part 12. GO (Gene Ontology) enrichment Analysis.=====
# BiocManager::install("DOSE")
library(DOSE)

# BiocManager::install("clusterProfiler")
library(clusterProfiler)

# BiocManager::install("org.Mm.eg.db")
library(org.Mm.eg.db)

# Select only group 2 for analysis.-----
DEG_LPS <- filter(DEG, Enriched == "cDC1_LPS")
DEG_LPS$entrez = mapIds(org.Mm.eg.db,
                        keys=as.character(DEG_LPS$gene),
                        column = "ENTREZID",
                        keytype = "SYMBOL",
                        multivals = "first")
DEG_LPS <- DEG_LPS$entrez # Create a list with genes.
go_cDC1_LPS <- enrichGO(gene = DEG_LPS,
                         orgDb = org.Mm.eg.db,
                         pvalueCutoff = 0.05,
                         qvalueCutoff = 0.05,
                         ont="all", #BP, CC, MF or all
                         readable = T)
```

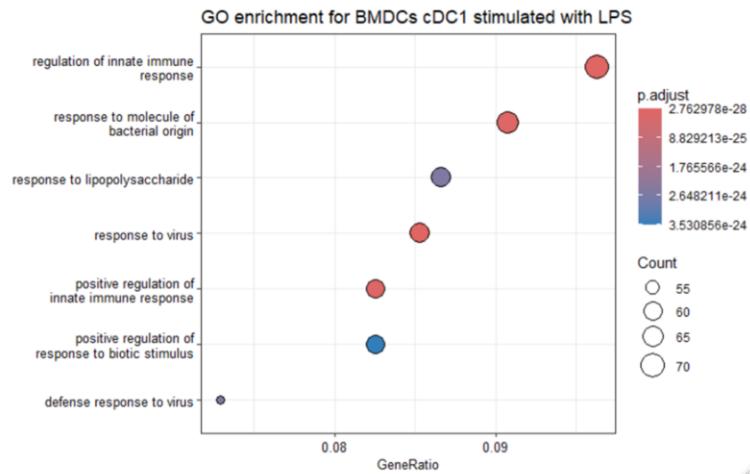
Gene Ontology



Gerarchico

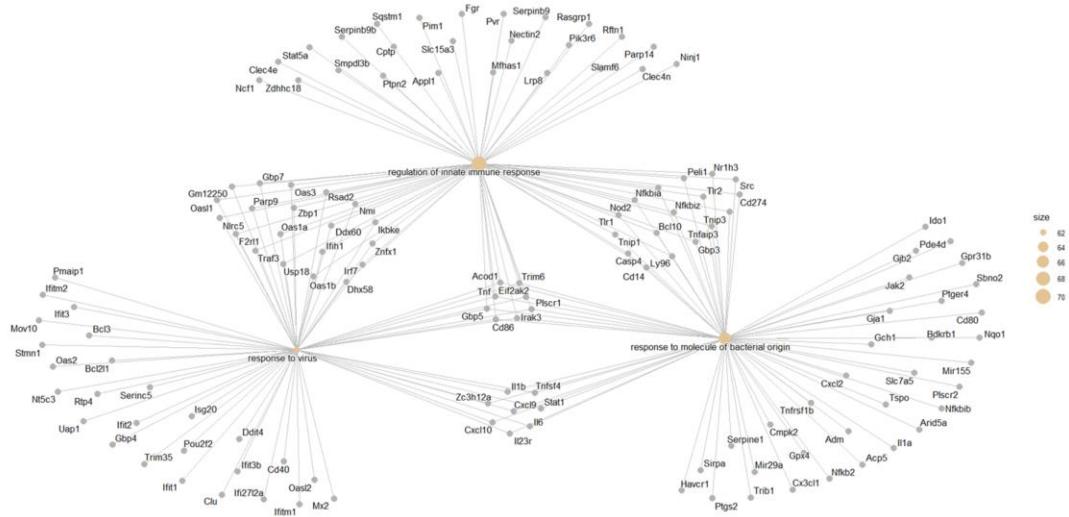
Gene Ontology: Un sistema di classificazione che raggruppa i geni in base alle loro funzioni biologiche, alla posizione cellulare o al ruolo molecolare.

- **Obiettivo:** Identificare quali processi biologici, componenti cellulari o funzioni molecolari sono associati a un insieme di geni.
- **Input:** un elenco di geni precedentemente selezionati (ad esempio: geni espressi in modo differenziale).
- **Risultato:** termini GO (funzioni o processi) arricchiti, cioè più rappresentati in questo elenco di geni di quanto ci si aspetterebbe per caso.



Category Network Plot - Cnetplot

GO enrichment for BMDCs cDC1 stimulated with LPS



Part 13. Gene Set Enrichment Analysis

```

# Part 13. GSEA=====
print("Gene Set Enrichment Analysis...")

DEG$entrez = mapIds(org.Mm.eg.db, keys=as.character(DEG$gene), column = "ENTREZID",
                     keytype = "SYMBOL", multivals = "first")
genelist <- DEG$entrez # Create a list with genes.

rnk_gsea <- dplyr::bind_cols(DEG$gene,
                             as.numeric(-log10(DEG$padj)) *
                               sign(DEG$logFC))
# rnk_gsea <- rnk_gsea[order(rnk_gsea[,2], decreasing = TRUE),]
rnk_gsea <- rnk_gsea[!is.na(rnk_gsea[,2]),]
rnk_gsea_clean <- rnk_gsea[rnk_gsea[,2] != -Inf,]
rnk_gsea_clean <- rnk_gsea_clean[rnk_gsea_clean[,2] != Inf,]

write.table(rnk_gsea_clean, file = "BMDC_DC1_ctrl_vs_LPS.rnk",
            quote = FALSE,
            row.names = FALSE,
            col.names = FALSE,
            sep = "\t") # load this file into the GSEA program.

# End of script.
#####

```

Sept3	-3.83629341624994
Parvg	3.83629341624994
Il2re	3.68019522299916
Ccr4te	3.68019522299916
Ccr2	-3.68019522299916
G53n01106RER	3.68019522299916
Trip3	3.68019522299916
Tas	3.68019522299916
Opqr3	3.62576295323193
Tmef2f1	-3.62576295323193
F638028010Rlik	-3.62576295323193
Cacnate	-3.62576295323193
Hrpz	3.62576295323193
Cdkn1a	3.62576295323193
Gbp7	3.62103317804584
Srgap3	-3.62103317804584
Oas12	3.62103317804584
Lpar4	3.62103317804584
Usp14	3.62103317804584
Sigmar1	3.62103317804584
Tmjd3///Adora3	-3.62103317804584
Srgap3	-3.62103317804584
Opqr2	-3.62103317804584
Pipr4	3.62103317804584
Ifit1	3.62103317804584
Tbxas1	-3.5240065795282
Arsh	-3.52711110616329
Cdkn1	3.51728416281519
Sugd2	3.38200377391616
Wdr41	3.38218734696675
Ickf4	3.38218734696675
Itgb7	3.35015437873727
Hepacam2	-3.31851934032522
Edil3	3.31851934032522
Tnfaf9	3.31613944397327
Gsn	-3.31613944397327
Htr4	-3.31613944397327

GSEA
Gene Set Enrichment Analysis

- [GSEA Home](#)
- [Downloads](#)
- [Molecular Signatures Database](#)
- [Documentation](#)
- [Contact](#)
- [Team](#)

UC San Diego

BROAD INSTITUTE

Overview

Gene Set Enrichment Analysis (GSEA) is a computational method that determines whether *a priori* defined set of genes shows statistically significant, concordant differences between two biological states (e.g. phenotypes).

- ▶ [Download](#) the GSEA software and additional resources to analyze, annotate and interpret enrichment results.
- ▶ [Explore the Molecular Signatures Database \(MSigDB\)](#), a collection of annotated gene sets for use with GSEA software.
- ▶ [View documentation](#) describing GSEA and MSigDB.
- ▶ View guidelines for [using RNA-seq datasets with GSEA](#).
- ▶ Use the [GenePattern](#) platform to run analyses, including [classical GSEA](#) and a variation designed for single-sample analysis ([ssGSEA](#)).

What's New

9-Aug-2024: MSigDB 2024.1 provides collection updates for GO, Reactome, WikiPathways, and more along with numerous new set additions for Human and Mouse Databases. Additionally, gene data has been updated to Ensembl 112. See the [release notes](#) for details.

Molecular Profile Data

Run GSEA

Gene Set Database

Enriched Sets

License Terms

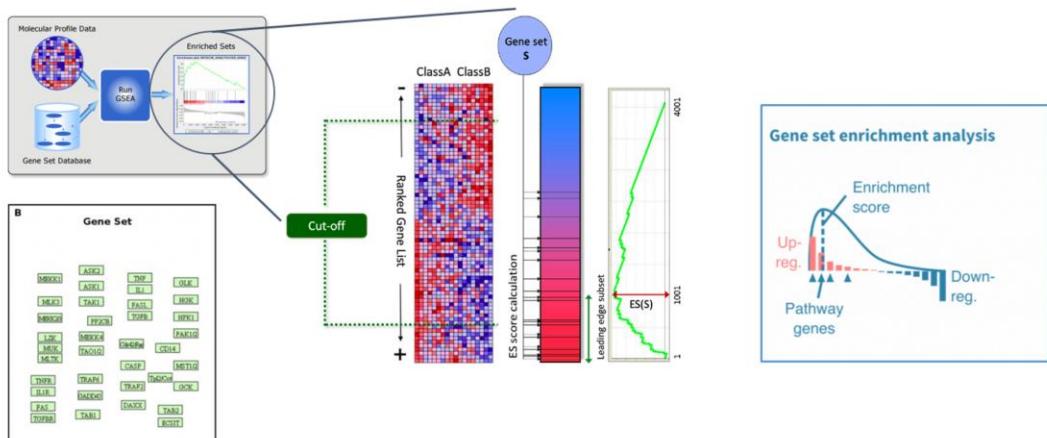
GSEA and MSigDB are available for use under these license terms.

Please register to download the GSEA software, access our web tools, and view the MSigDB gene sets. After registering, you can log in at any time using your email address. Registration is free. Its only purpose is to help us track usage for reports to our funding agencies.

Citing GSEA

Non gerarchico

Gene Set Enrichment Analysis (GSEA)

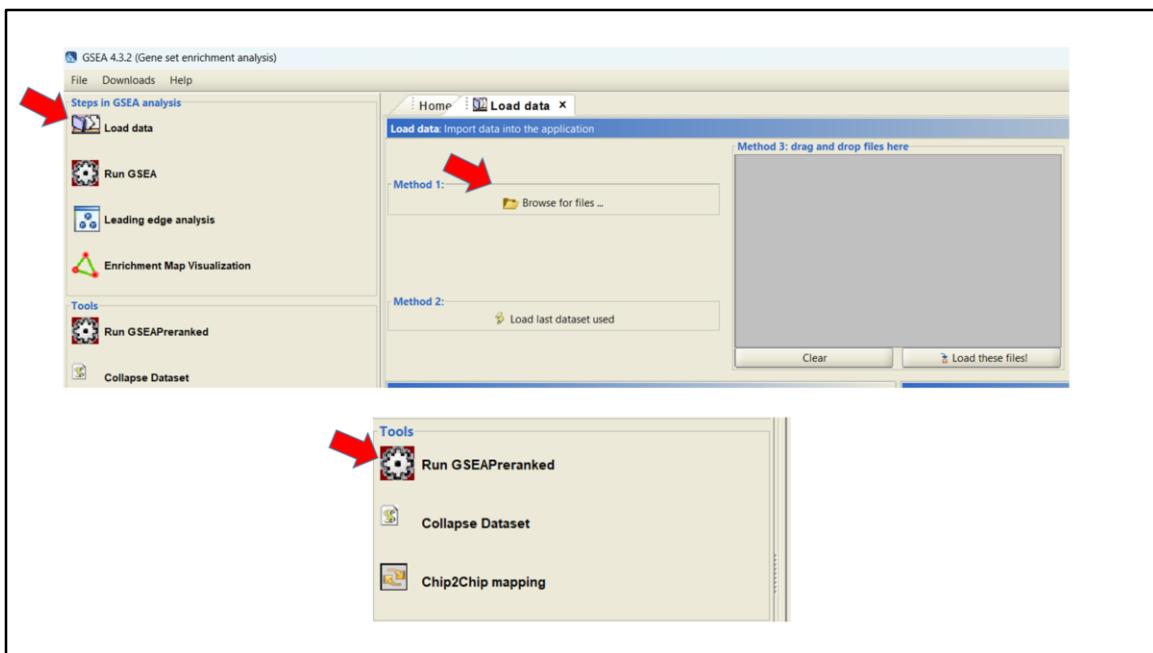


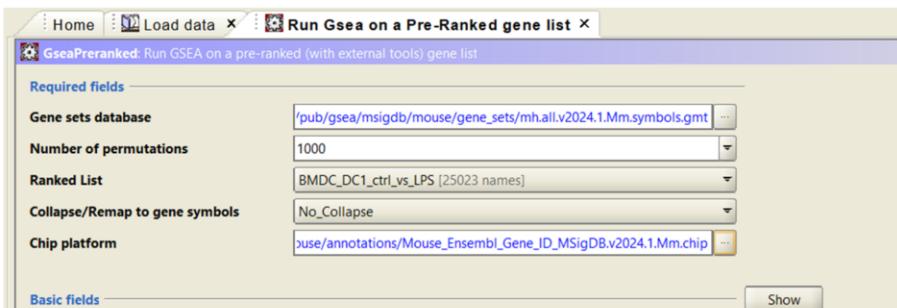
Che cos'è: un metodo che analizza l'elenco completo dei geni classificati in base all'espressione (ad esempio, dal più al meno regolamentato) per identificare se gruppi di geni predefiniti (set di geni) sono arricchiti a un'estremità della classifica.

Obiettivo: rilevare associazioni significative anche in geni che non verrebbero identificati in analisi basate su un'unica soglia (come "differenziatamente espressi").

Input: l'intero elenco di geni ordinato in base a qualche misura, come la variazione di log₂ volte o le statistiche di significatività.

Risultato: insiemi di geni (set genetici) che sono costantemente più o meno espressi in una condizione.





Collapse: Quando sono presenti più sonde o trascritti mappati sullo stesso gene, GSEA applica un criterio per ridurre questi dati (ad esempio selezionando il valore di espressione più alto o utilizzando una media) e associa il valore finale al gene corrispondente.

Premere
RUN

No collapse: Ogni ID di sonda o trascrizione viene trattato individualmente. Ciò può portare a più voci per lo stesso gene se ad esso sono associati più sonde o trascrizioni.

GSEA Report for Dataset BMDC_DC1_ctrl_vs_LPS

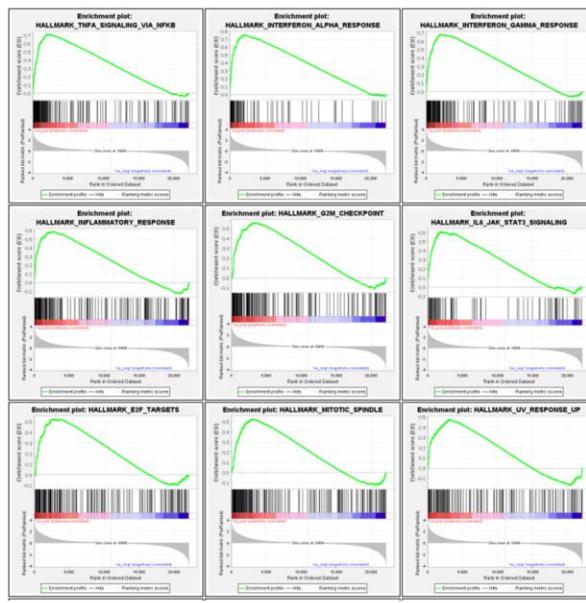
Enrichment in phenotype: na

- 40 / 50 gene sets are upregulated in phenotype **na_pos**
- 31 gene sets are significant at FDR < 25%
- 20 gene sets are significantly enriched at nominal pvalue < 1%
- 23 gene sets are significantly enriched at nominal pvalue < 5%
- [Snapshot](#) of enrichment results
- Detailed [enrichment results in html](#) format
- Detailed [enrichment results in TSV](#) format (tab delimited text)
- [Guide to](#) interpret results

Enrichment in phenotype: na

- 10 / 50 gene sets are upregulated in phenotype **na_neg**
- 1 gene sets are significantly enriched at FDR < 25%
- 0 gene sets are significantly enriched at nominal pvalue < 1%
- 2 gene sets are significantly enriched at nominal pvalue < 5%
- [Snapshot](#) of enrichment results
- Detailed [enrichment results in html](#) format
- Detailed [enrichment results in TSV](#) format (tab delimited text)
- [Guide to](#) interpret results

Table: Snapshot of enrichment results



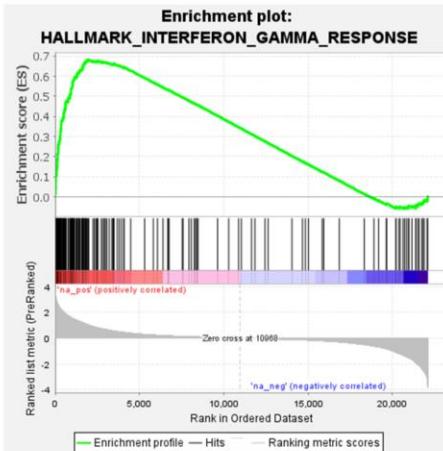


Table: GSEA Results Summary

Dataset	BMDC_DC1_ctrl_vs_LPS
Phenotype	NoPhenotypeAvailable
Upregulated in class	na_pos
GeneSet	HALLMARK_INTERFERON_GAMMA_RESPONSE
Enrichment Score (ES)	0.68222123
Normalized Enrichment Score (NES)	2.372906
Nominal p-value	0.0
FDR q-value	0.0
FWER p-Value	0.0



Gene Set Enrichment Analysis

GSEA Home

Downloads

Molecular Signatures Database

Documentation

Contact

Team

MsigDB Home

Human Collections

- ▶ About
- ▶ Browse
- ▶ Search
- ▶ Investigate
- ▶ Gene Families

Mouse Collections

- ▶ About
- ▶ Browse
- ▶ Search
- ▶ Investigate

Help

UC San Diego

MSigDB
Molecular Signatures
Database

Overview

The Molecular Signatures Database (MSigDB) is a resource of tens of thousands of annotated gene sets for use with GSEA software, divided into Human and Mouse collections. From this web site, you can

- ▶ Examine a gene set and its annotations. See, for example, the HALLMARK_APOPTOSIS human gene set page.
- ▶ Browse gene sets by name or collection.
- ▶ Search for gene sets by keyword.
- ▶ Investigate gene sets:
 - ▶ Compute overlaps between your gene set and gene sets in MSigDB.
 - ▶ Categorize members of a gene set by gene families.
 - ▶ View the expression profile of a gene set in a provided public expression compendia.
 - ▶ Investigate the gene set in the online biological network repository NDE
- ▶ Download gene sets.

License Terms

GSEA and MSigDB are available for use under these license terms.

Molecular Signatures Database

Human Collections

H hallmark gene sets are coherently expressed signatures derived by aggregating many MSigDB gene sets to represent well-defined biological states or processes.

C1 positional gene sets corresponding to human chromosome cytogenetic bands.

C2 curated gene sets from online pathway databases, publications in PubMed, and knowledge of domain experts.

C3 regulatory target gene sets based on gene target predictions for microRNA seed sequences and predicted transcription factor binding sites.

C4 computational gene sets defined by mining large collections of cancer-oriented expression data.

C5 ontology gene sets consist of genes annotated by the same ontology term.

C6 oncogenic signature gene sets defined directly from microarray gene expression data from cancer gene perturbations.

C7 immunologic signature gene sets represent cell states and perturbations within the immune system.

C8 cell type signature gene sets curated from cluster markers identified in single-cell sequencing studies of human tissue.



GSEA
Gene Set Enrichment Analysis

GSEA Home Downloads Molecular Signatures Database Documentation Contact Team

MSigDB Home

Human Collections

- ▶ About
- ▶ Browse
- ▶ Search
- ▶ Investigate
- ▶ Gene Families

Mouse Collections

- ▶ About
- ▶ Browse
- ▶ Search
- ▶ Investigate

Help

Browse Mouse Gene Sets

UC San Diego 

Gene set name: (Enter full or partial name)

By first letter: [A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

By collection: [about the MSigDB Mouse collections]

- ▶ **MH** (orthology-mapped hallmark gene sets, 50 gene sets)
- ▶ **M1** (positional gene sets, 341 gene sets)
 - ▶ by chromosome: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 X Y MT
- ▶ **M2** (curated gene sets, 2710 gene sets)
 - ▶ **CGP** (chemical and genetic perturbations, 980 gene sets)
 - ▶ **CP** (canonical pathways, 1730 gene sets)
 - ▶ **CP:BIOCARTA** (BioCarta gene sets, 252 gene sets)
 - ▶ **CP:REACTOME** (Reactome gene sets, 1289 gene sets)
 - ▶ **CP:WIKIPATHWAYS** (WikiPathways gene sets, 189 gene sets)
 - ▶ **M3** (regulatory gene sets, 2047 gene sets)
 - ▶ **GTRD** (GTRD transcription factor targets, 279 gene sets)
 - ▶ **MIRDB** (miRDB microRNA targets, 1768 gene sets)
 - ▶ **M5** (ontology gene sets, 10678 gene sets)

- » GO:BP (GO biological process, 7713 gene sets)
- » GO:CC (GO cellular component, 1028 gene sets)
- » GO:MF (GO molecular function, 1845 gene sets)

- » MPT (Mouse Phenotype Ontology MP Tumor, 92 gene sets)

- » MB (cell type signature gene sets, 233 gene sets)

Click on a gene set name to view its gene set page.

[Back to Top](#)

GOBP_ACUTE_INFLAMMATORY_RESPONSE_TO_ANTIGENIC_STIMULUS	GOBP_NEGATIVE_REGULATION_OF_INFLAMMATORY_RESPONSE_TO_ANTIGENIC_STIMULUS	GOBP_T_CELL_ACTIVATION_VIA_T_CELL_RECECTOR_CONTACT_WITH_ANTIGEN_BOUND_TO_MHC_MOLECULE_ON_ANTIGEN_PRESENTING_CELL
GOBP_ANTIGEN_PROCESSING_AND_PRESENTATION	GOBP_PEPTIDE_ANTIGEN_ASSEMBLY_WITH_MHC_CLASS_I_PROTEIN_COMPLEX	GOBP_T_CELL_ANTIGEN_PROCESSING_AND_PRESENTATION
GOBP_ANTIGEN_PROCESSING_AND_PRESENTATION_OF_ENDOGENOUS_ANTIGEN	GOBP_PEPTIDE_ANTIGEN_ASSEMBLY_WITH_MHC_CLASS_II_PROTEIN_COMPLEX	GOBP_TOLERANCE_INDUCTION_TO_SELF_ANITIGEN
GOBP_ANTIGEN_PROCESSING_AND_PRESENTATION_OF_EXOGENOUS_ANTIGEN	GOBP_PEPTIDE_ANTIGEN_ASSEMBLY_WITH_MHC_PROTEIN_COMPLEX	GOLDRATH_ANTIGEN_RESPONSE
GOBP_ANTIGEN_PROCESSING_AND_PRESENTATION_OF_EXOGENOUS_PEPTIDE_ANTIGEN	GOBP_POSITIVE_REGULATION_OF_ACUTE_INFAMMATORY_RESPONSE_TO_ANTIGENIC_STIMULUS	GOMF_ANTIGEN_BINDING
GOBP_ANTIGEN_PROCESSING_AND_PRESENTATION_OF_EXOGENOUS_PEPTIDE_ANTIGEN	GOBP_POSITIVE_REGULATION_OF_ANTIGENIC_STIMULUS	GOMF_PEPTIDE_ANTIGEN_BINDING
VIA_MHC_CLASS_I	GOBP_POSITIVE_REGULATION_OF_ANTIGEN_PRESENTATION	GOmf_PROTEIN_ANTIGEN_BINDING
GOBP_ANTIGEN_PROCESSING_AND_PRESENTATION_OF_EXOGENOUS_PEPTIDE_ANTIGEN_VIA_MHC_CLASS_II	GOBP_POSITIVE_REGULATION_OF_ANTIGEN_PRESENTATION	REACTOME_ANTIGEN_ACTIVATES_B_CELL_RECECTOR_BCR_LEADING_TO_GENERATION_OF_SECOND_MESSENGER
GOBP_ANTIGEN_PROCESSING_AND_PRESENTATION_OF_EXOGENOUS_PEPTIDE_ANTIGEN_VIA_MHC_CLASS_II	GOBP_POSITIVE_REGULATION_OF_DENDRITIC_CELL_ANTIGEN_PROCESSING_AND_PRESENTATION	REACTOME_ANTIGEN_PRESENTATION_FOLDING_ASSEMBLY_AND_PEPTIDE_LOADING_OF_CLASS_I_MHC
GOBP_ANTIGEN_PROCESSING_AND_PRESENTATION_OF_EXOGENOUS_PEPTIDE_ANTIGEN_VIA_MHC_CLASS_I	GOBP_POSITIVE_REGULATION_OF_INFAMMATORY_RESPONSE_TO_ANTIGENIC_STIMULUS	REACTOME_ANTIGEN_PROCESSING_CROSS_PRESENTATION
GOBP_ANTIGEN_PROCESSING_AND_PRESENTATION_OF_EXOGENOUS_PEPTIDE_OR_POLYSACCHARIDE	GOBP_REGULATION_OF_ACUTE_INFAMMATORY_RESPONSE_TO_ANTIGENIC_STIMULUS	REACTOME_ANTIGEN_PROCESSING_UBIQUITINATION_PROTEASOME_DEGRADATION
ANTIGEN_VIA_MHC_CLASS_II	GOBP_REGULATION_OF_ANTIGEN_PROCESSING	REACTOME_CLASS_I_MHC_MEDIATED_ANTIGEN_PROCESSING_PRESENTATION
GOBP_ANTIGEN_PROCESSING_AND_PRESENTATION_OF_EXOGENOUS_PEPTIDE_OR_POLYSACCHARIDE_VIA_MHC_CLASS_I	GOBP_REGULATION_OF_ANTIGEN_PRESENTATION	REACTOME_CROSS_PRESENTATION_OF_PARTICULATE_EXOGENOUS_ANTIGENS_PHAGOSOMES
GOBP_ANTIGEN_RECECTOR_MEDIATED_SIGNALING		

Mouse Gene Set: GOBP_REGULATION_OF_ANTIGEN_PROCESSING_AND_PRESENTATION

For the Human gene set with the same name, see [GOBP_REGULATION_OF_ANTIGEN_PROCESSING_AND_PRESENTATION](#)

Standard name	GOBP_REGULATION_OF_ANTIGEN_PROCESSING_AND_PRESENTATION
Systematic name	MM4309
Brief description	Any process that modulates the frequency, rate, or extent of antigen processing and presentation. [GOC:add]
Full description or abstract	
Collection	M5: Ontology GO: Gene Ontology GO:BP: GO Biological Process
Source publication	
Exact source	GO:0002577
Related gene sets	
External links	http://amigo.geneontology.org/amigo/term/GO:0002577
Filtered by similarity 	
Source species	<i>Mus musculus</i>
Contributed by	Gene Ontology (Gene Ontology Consortium)
Source platform or identifier namespace	Mouse_NCBI_Gene_ID
Dataset references	
Download gene set	format: grp gmt xml json TSV metadata (show collections to investigate for overlap with this gene set)
Compute overlaps 	NG-CHM interactive heatmaps (Please note that clustering takes a few seconds) Mouse Transcriptomic BodyMap compendium 
Compendia expression profiles 	Legacy heatmaps (PNG) Mouse Transcriptomic BodyMap compendium

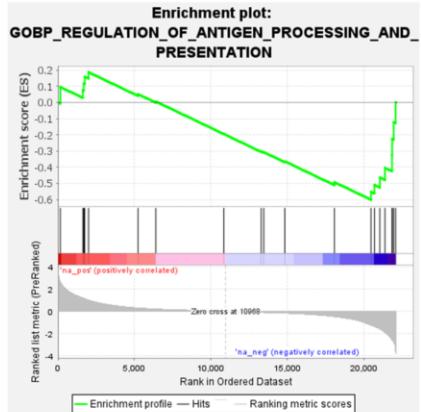
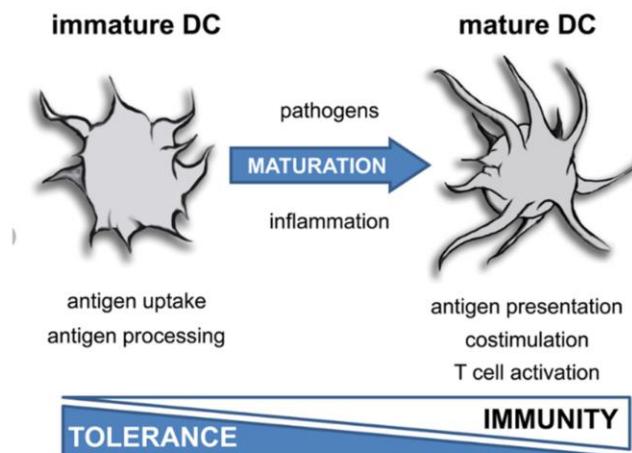


Table: GSEA Results Summary

Dataset	BMDC_DC1_ctrl_vs_LPS
Phenotype	NoPhenotypeAvailable
Upregulated in class	na_pos
GeneSet	GOBP_REGULATION_OF_ANTIGEN_PROCESSING_AND_PRESENTATION
Enrichment Score (ES)	-0.60123724
Normalized Enrichment Score (NES)	-1.4314585
Nominal p-value	0.037950665
FDR q-value	0.037950665
FWER p-Value	0.02

Conclusion



Troubleshooting - packages

1) Verifica la versione R:
`version()`

2) Errore nel compilare il codice fonte:
Windows: **Installa Rtools**.
Mac: **Installa Xcode Command Line Tools**.

3) Verificare che ci siano dipendenze ausenti:
`install.packages(c("curl", "jsonlite"))`

4) Errori nella connessione con il CRAN:
`chooseCRANmirror() # choose another repository`
`install.packages("httr2")`

5) Errore di autorizzazioni: #admin
`install.packages("httr2",`
`lib = "percorso/della/cartella/con/permesso")`

6) Installazione manuale: #RTolls
Download .tar.gz from CRAN
`install.packages("percorso/del/httr2.tar.gz",`
`repository = NULL, tipo = "fonte")`