# 實作練習：以單細胞RNA定序資料為基礎之細胞間交互作用分析

# Hands-on practice: Cell-cell interaction/communication analysis based on single-cell RNA sequencing data

27 July 2023

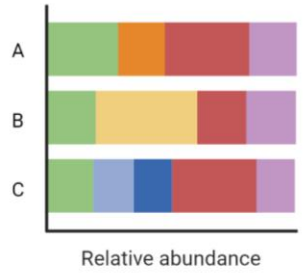
國家衛生研究院

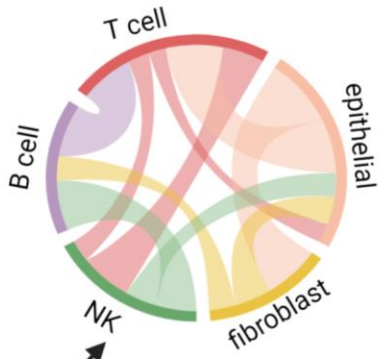
癌症研究所 江士昇 副研究員

群體健康科學研究所 吳怡儒 博士

# The application of single-cell RNA sequencing analysis

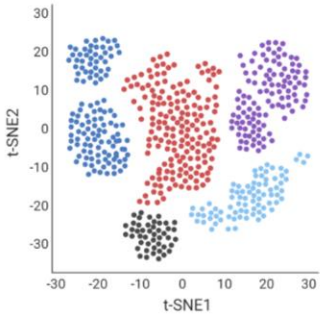
Cell type composition

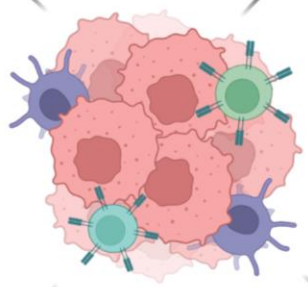


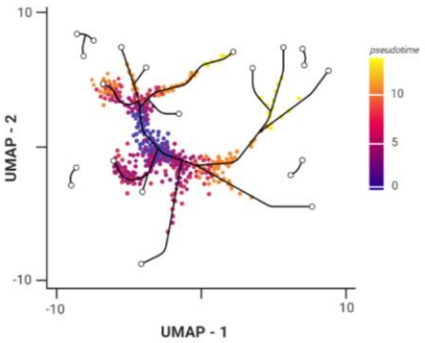




Cell-to-cell communication

Identify special cell populations

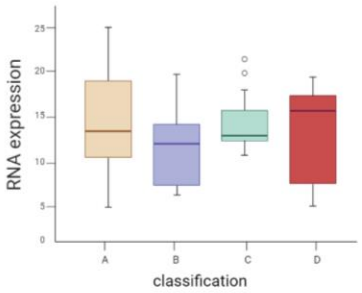


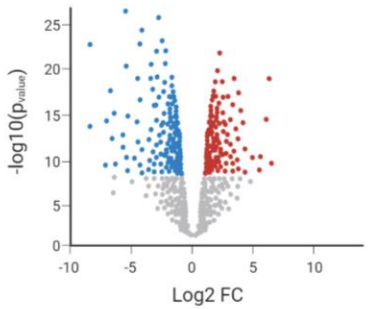



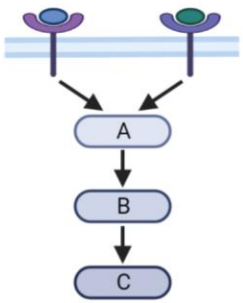



Trajectory analysis (Pseudotime)

Gene expression between classification



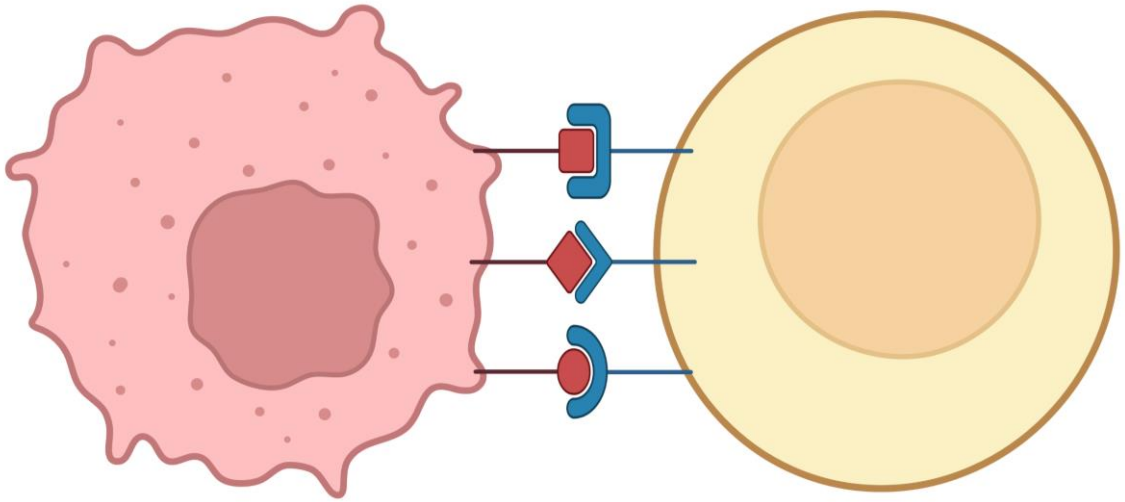

Pathway and functional annotation

## Inferring of cellular communication

Data preparation:

(1) Expression data (scRNA seq data)

(2) cell type information

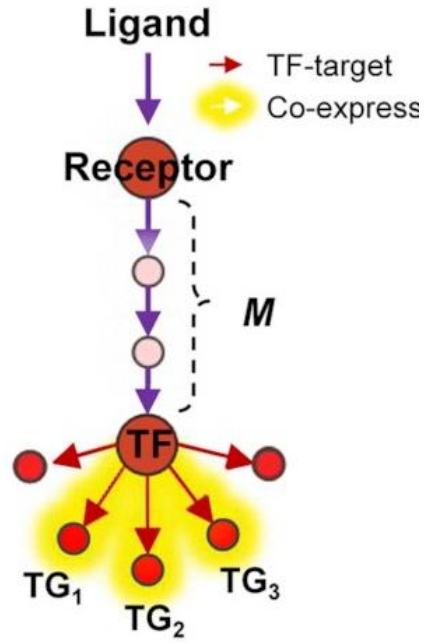


Software:

(1) R package (Cellcall, iTALK, CellChat)

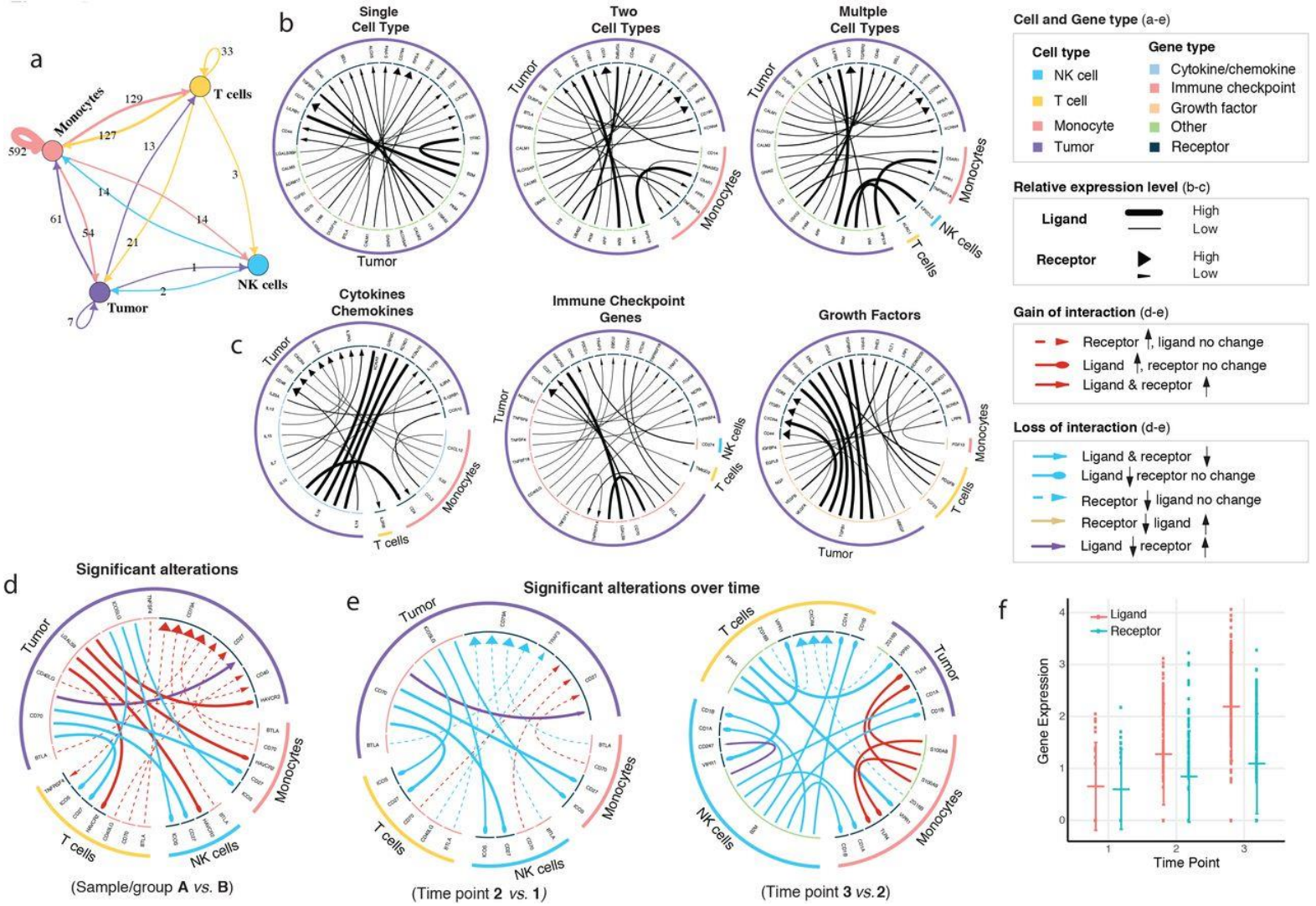
(2) Database of L-R pairs, pathways and transcription factors

(Literature or experimental supported data and KEGG base)



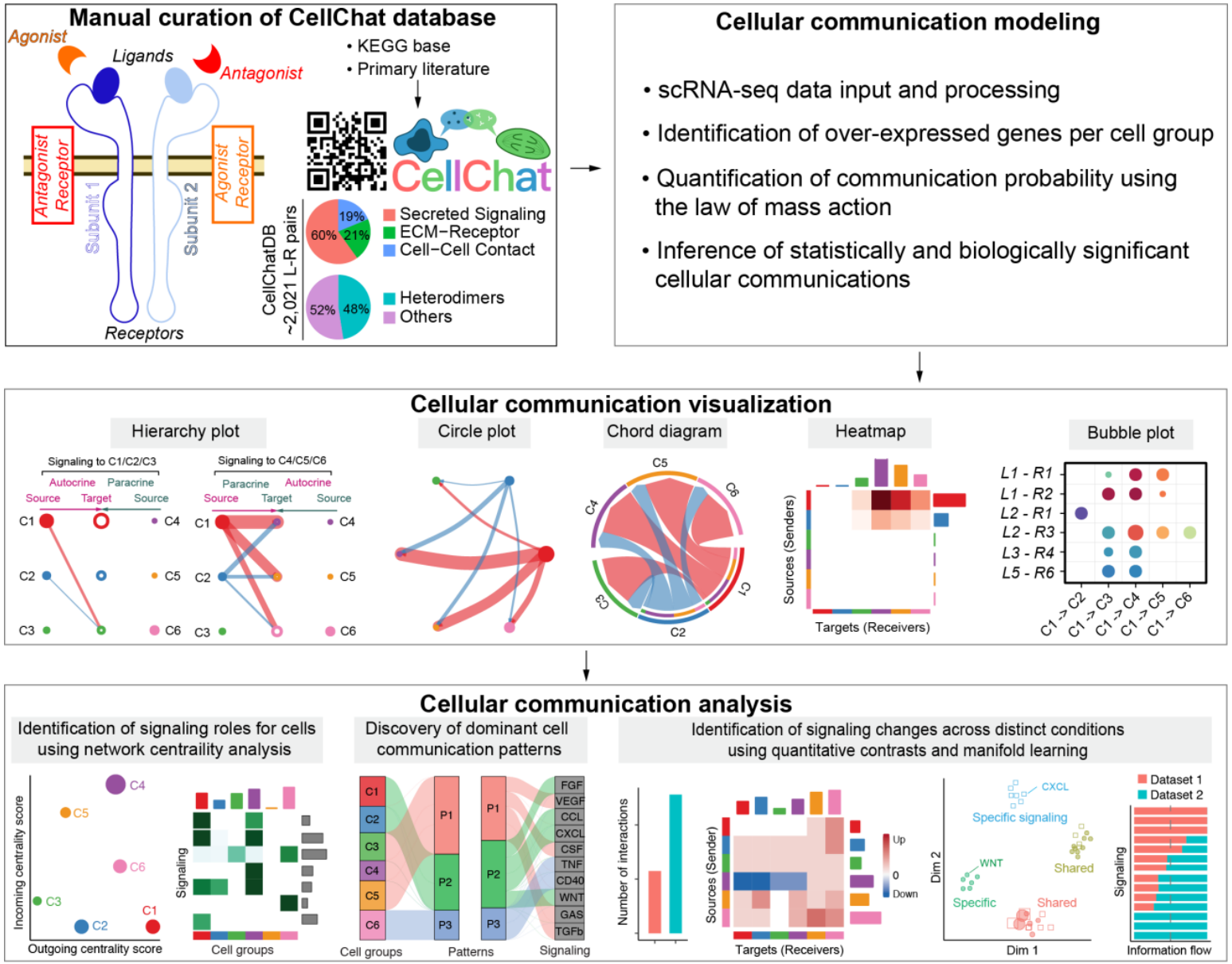
*bioRxiv* (2019) 507871:1–8. doi: 10.1101/507871

# iTALK: An R Package to Characterize and Illustrate Intercellular Communication.



*Nat Commun (IF: 14.92; Q1). 2021 Feb 17;12(1):1088.*

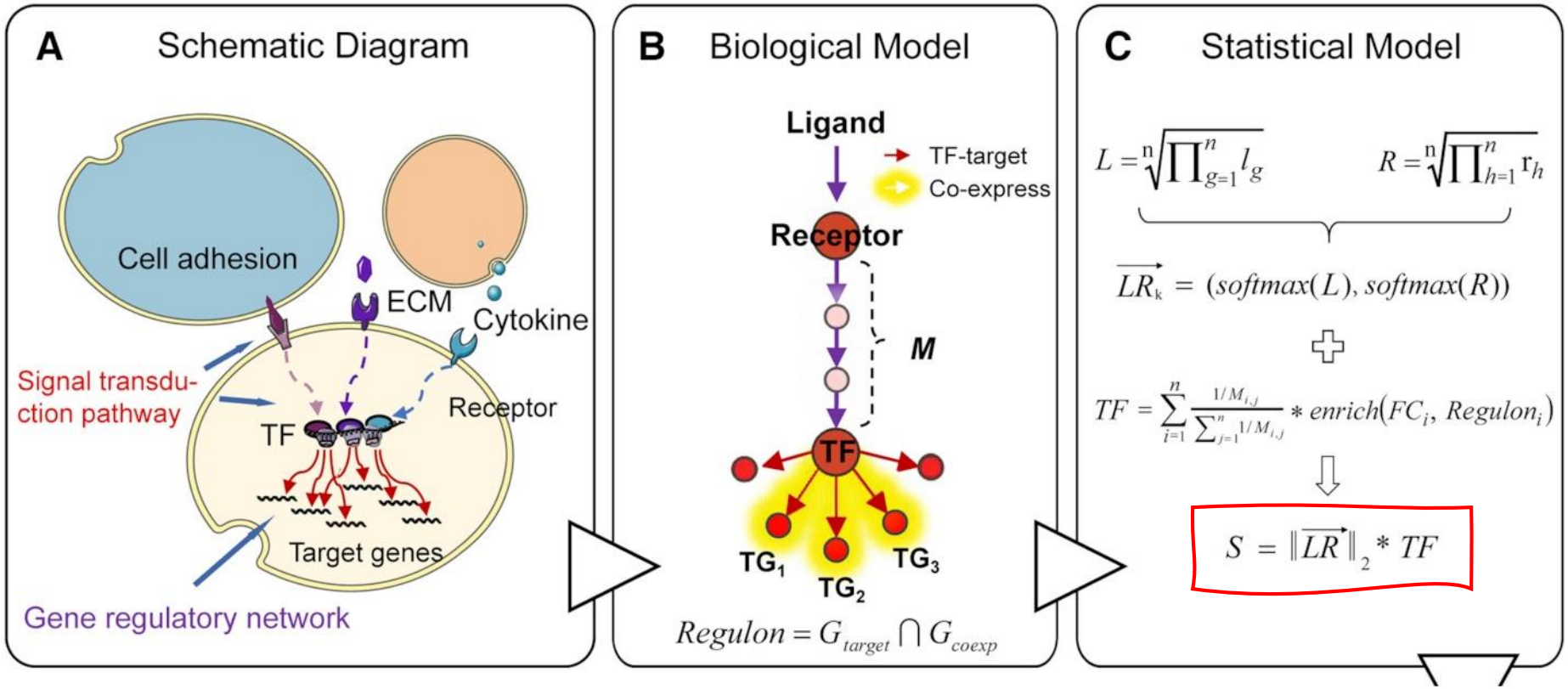
# Inference and analysis of cell-cell communication using CellChat



*Nucleic Acids Res (IF: 16.97). 2021 Sep 7;49(15):8520-8534.*

**CellCall: integrating paired ligand-receptor and transcription factor activities for cell-cell communication**

Database: (1) literature or experimental supported data and (2) KEGG



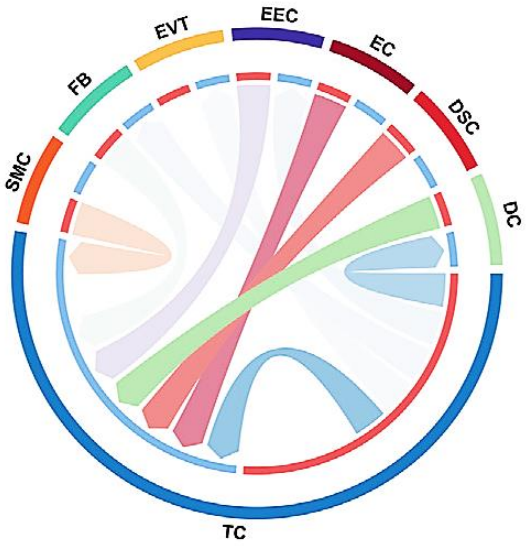

**S**: defined as the communication score of an L–R interaction

*Front Cell Dev Biol (IF: 6.68; Q2). 2021 Nov 5;9:770621.*

**Deciphering the Intercellular Communication Network of Peripartum Decidua that Orchestrates Deliver**

decidua before delivery (DBD)

decidua after delivery (DAD)




TC: T cells

SMC: Smooth muscle cells

FB: Fibroblasts

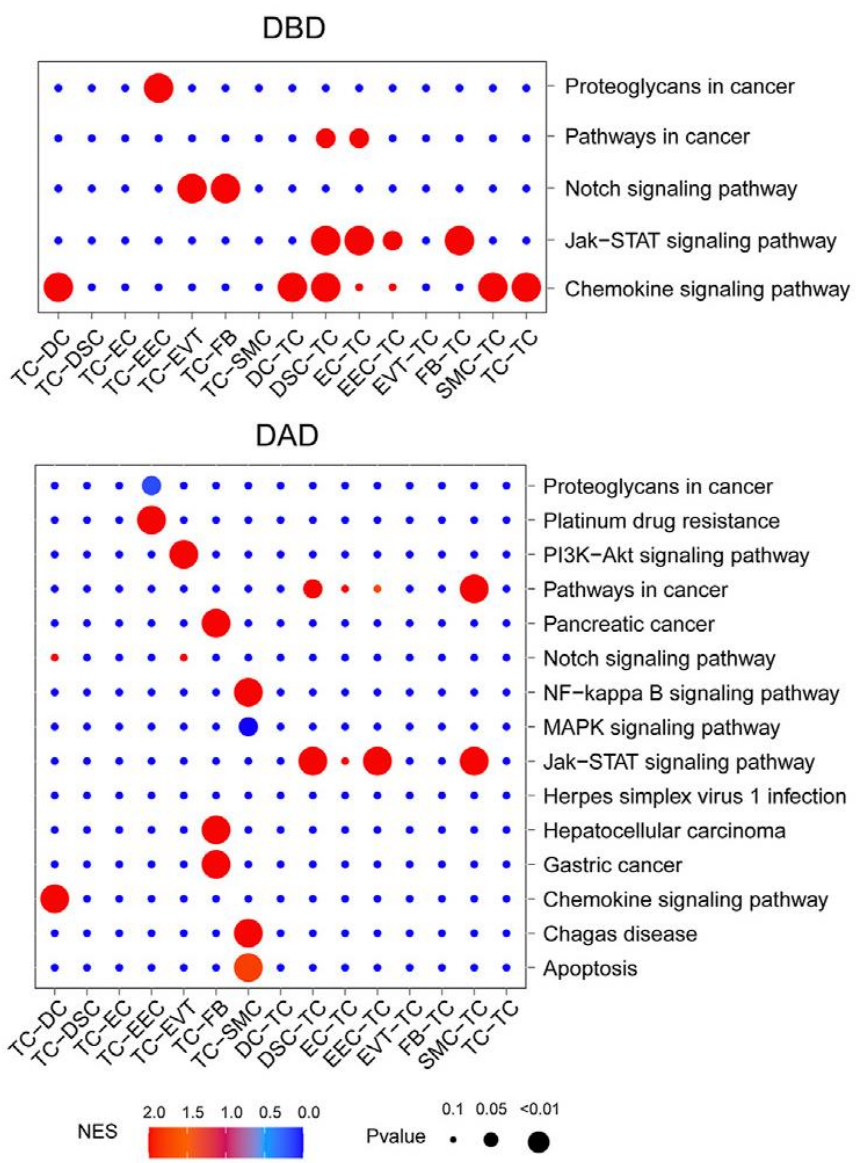
EVT: extravillous trophoblasts

EEC: endometrial cells

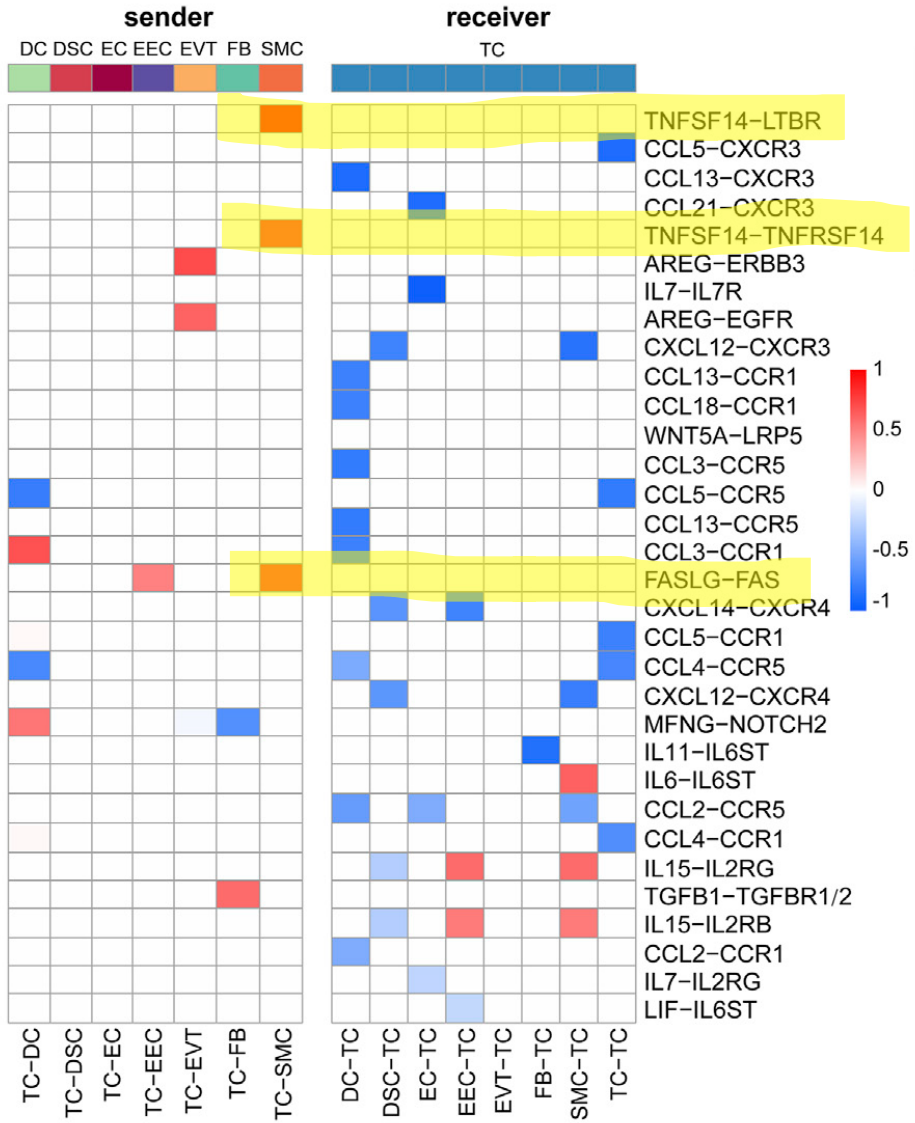
EC: endothelial cells

DSC: decidual stromal cells

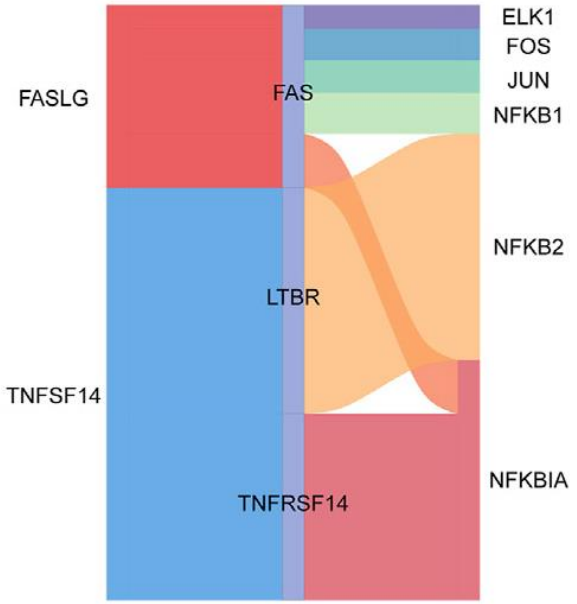
DC: Dendritic cells



A



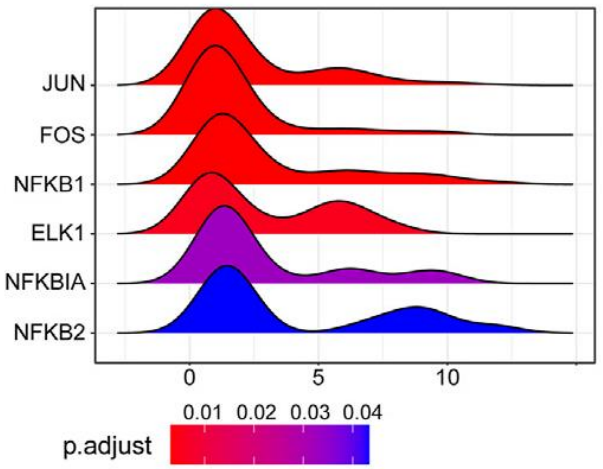

B




Ligand (SMC) Receptor (T cell) TF (T cell)

C

T cell

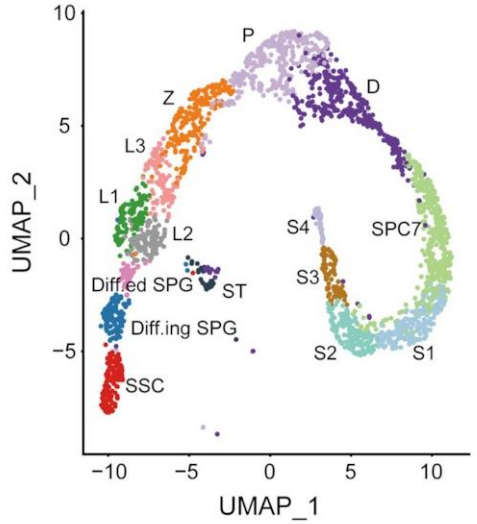



*Front Cell Dev Biol. 2021; 9: 770621.*

*Nucleic Acids Res (IF: 16.97). 2021 Sep 7;49(15):8520-8534.*

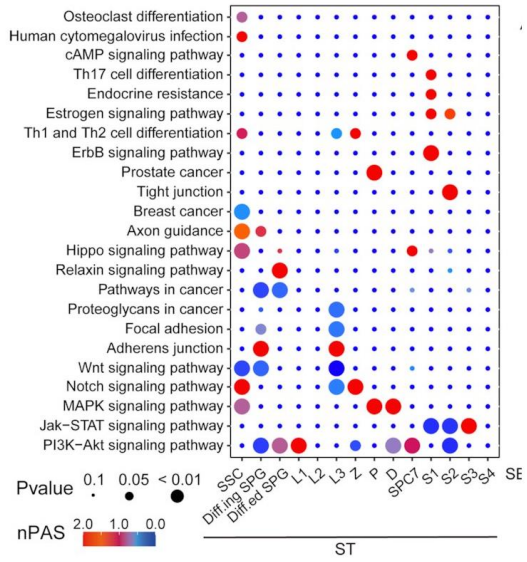
# CellCall: integrating paired ligand-receptor and transcription factor activities for cell-cell communication

## A. Cell typing

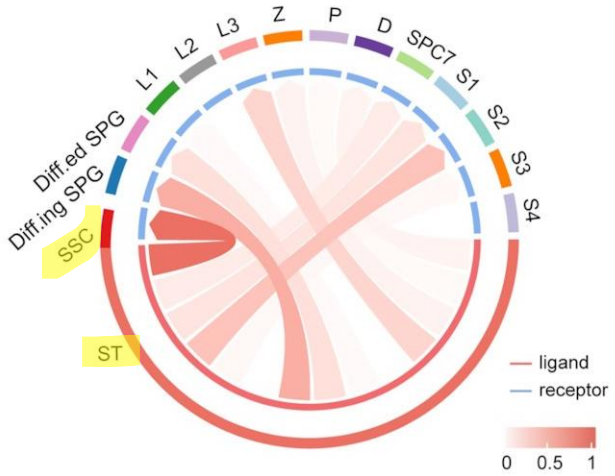



Diff.ing SPG: differentiating spermatogonia
Diff.ed SPG: differentiated spermatogonia
L: leptotene spermatocytes
Z: zygotene
P: pachytene
D: diplotene
SPC7: spermatocyte 7
S: spermatids
ST: sertoli cells
SSC: spermatogonial stem cells

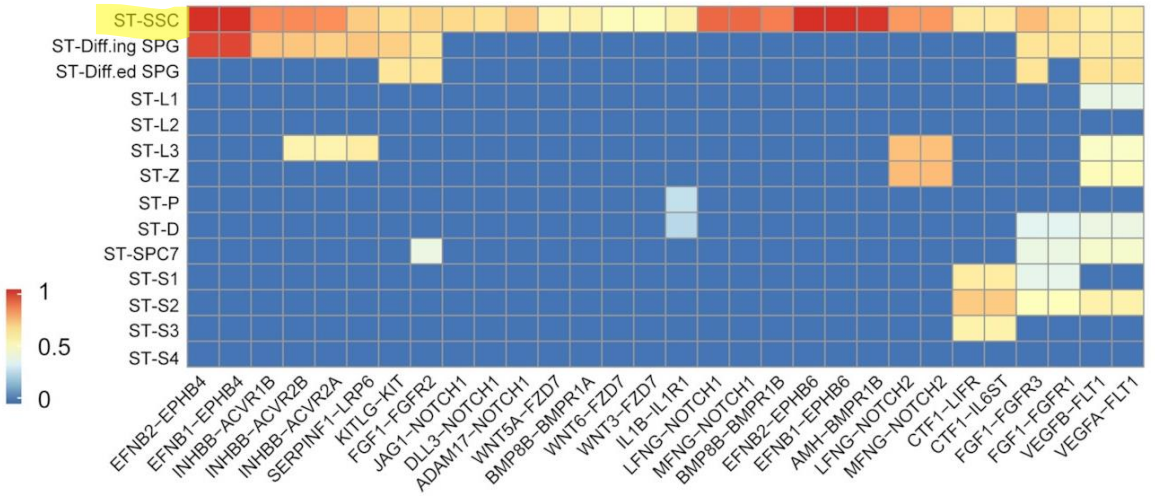
## C. Pathway analysis



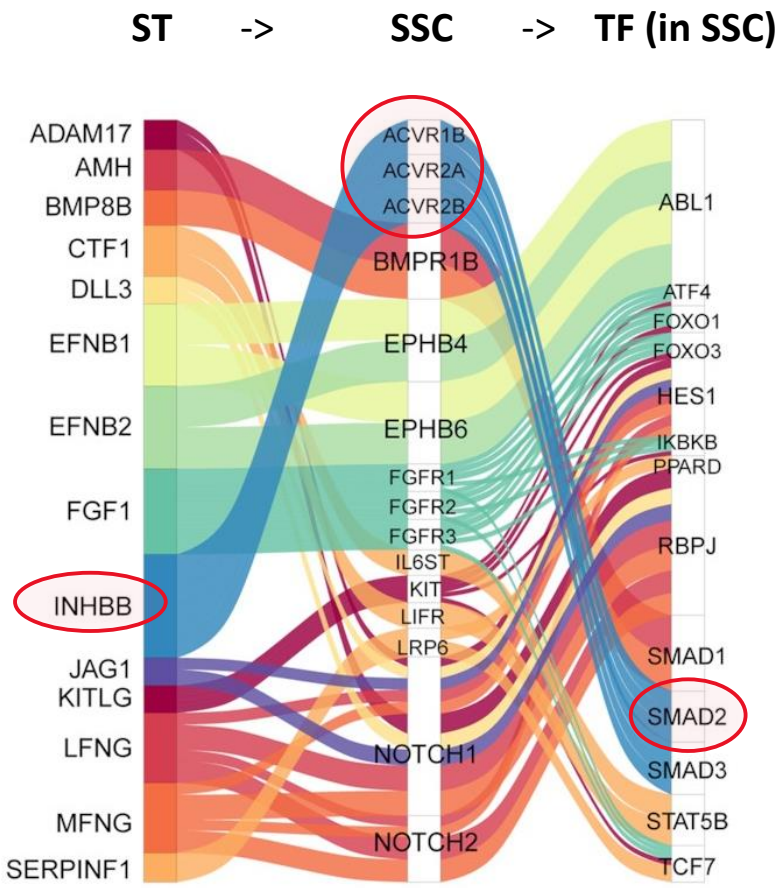

## B. Cell interaction



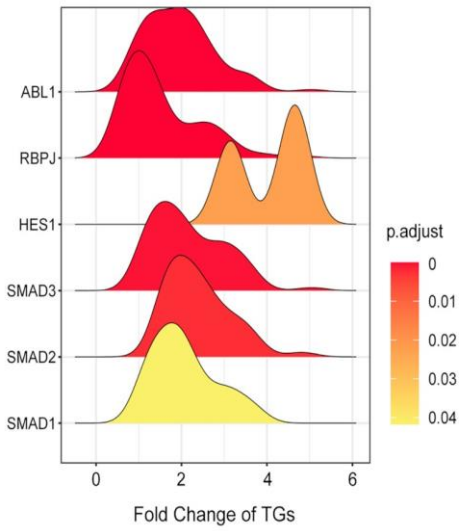

## D. L-R pairs



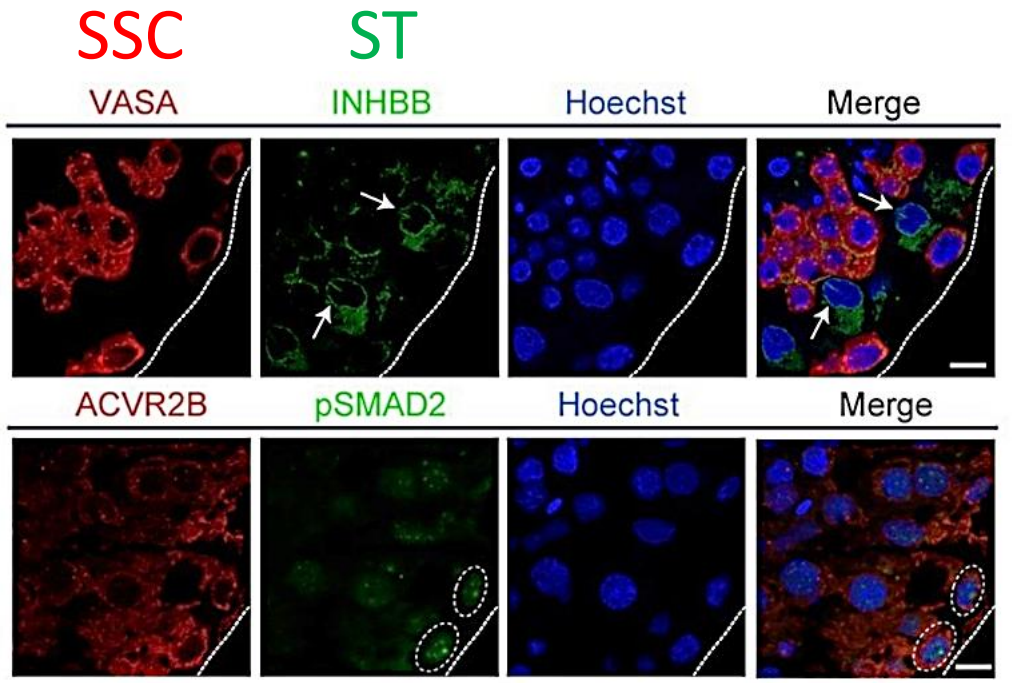
**A. Sankey plot of cell communication**




**B. Expression level of TFs in SSC (received cell)**




**C. Validation by IF staining**



# Hands-on practice: Cell-cell interaction/communication analysis based on single-cell RNA sequencing data

- R and Rtools installation
- Install R packages
- Upload single-cell RNA seq data
- Run the “Cellcall”
- Cell interaction, L-R pairs and downstream transcription factor

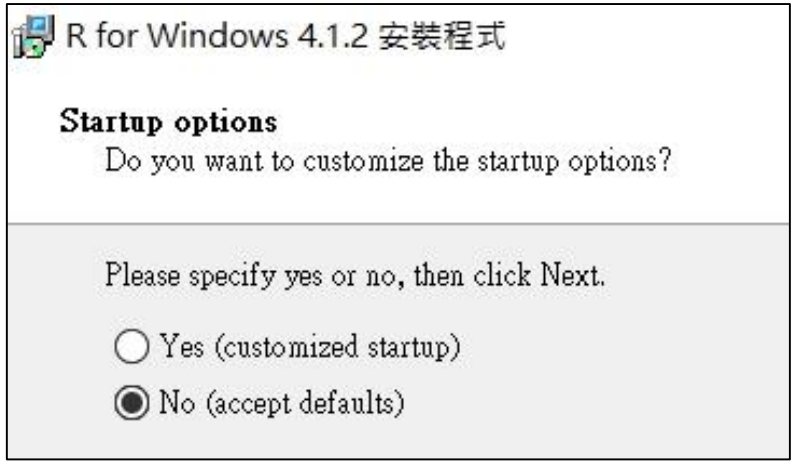
# Install R vision 4.1.2

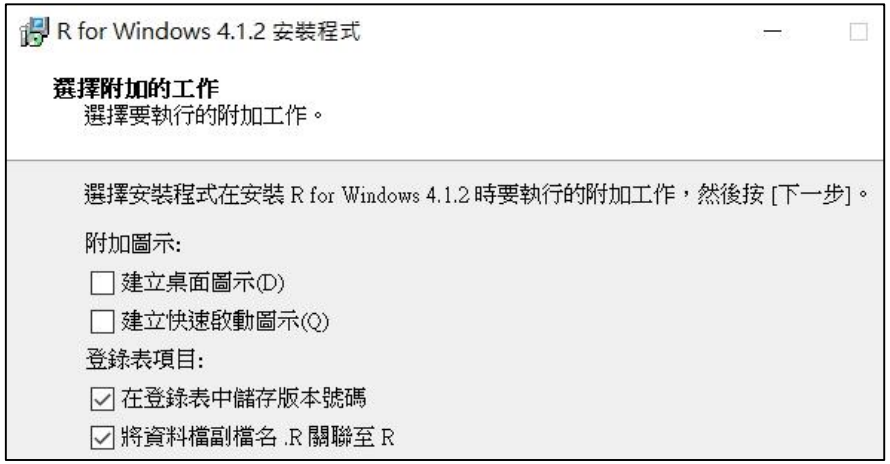
https://cran.r-project.org/bin/windows/base/old/4.1.2/









# Install RTools

https://cran.r-project.org/bin/windows/Rtools/

**RTools: Toolchains for**

Choose your version of Rtools:

| | |
|---|---|
| RTools 4.3 | for R versions from 4.3.0 (R-devel) |
| RTools 4.2 | for R versions 4.2.x (R-release) |
| RTools 4.0 | for R from version 4.0.0 to 4.1.3 |
| old versions of RTools | for R versions prior to 4.0.0 |

Note that Rtools is only needed build R packages with C/C++/Fortran code from source. By default, R for Windows installs the precompil which you do not need Rtools.

To use rtools, download the installer from CRAN:

- On Windows 64-bit: rtools40-x86_64.exe (includes both i386 and x64 compilers). Permanent url: rtools40-x86_64.exe.
- On Windows 32-bit: rtools40-i686.exe (i386 compilers only). Permanent url: rtools40-i686.exe.

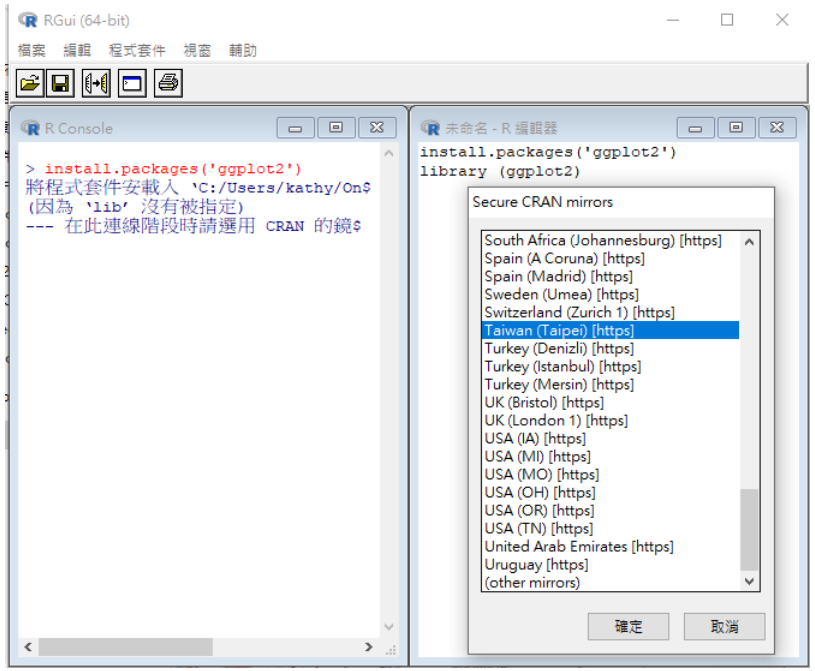
# R commands







把code貼在R編輯器中，按 **Ctrl + R** 執行




# Install

```
# install "devtools"
install.packages("devtools", type = "win.binary")
library(devtools)

#install "Seurat" 目前是Seurat 5 version
install.packages('Seurat')
library(Seurat)

#install "ggplot2"
install.packages('ggplot2')
library (ggplot2)

# install "patchwork"
devtools::install_github("thomasp85/patchwork")
library(patchwork)
```

# Install

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

# install "ComplexHeatmap"
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("ComplexHeatmap")

# install “enrichplot"
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("enrichplot")

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

BiocManager::install("DOSE")
```

```
> BiocManager::install("clusterProfiler")
'getOption("repos")' replaces Bioconductor standard reposit
details

replacement repositories:
    CRAN: https://cran.csie.ntu.edu.tw

Bioconductor version 3.14 (BiocManager 1.30.18), R 4.1.2 (2
Installation paths not writeable, unable to update packages
  path: C:/Program Files/R/R-4.1.2/library
  packages:
    boot, class, cluster, codetools, foreign, lattice, MASS,
    rpart, spatial, survival
Old packages: 'BiocManager', 'cachem', 'cli', 'digest', 'dp
  'httpuv', 'magrittr', 'processx', 'ps', 'ranger', 'Rcpp',
Update all/some/none? [a/s/n]: a
```

Update all/some/none? 選 **a**

# Install

```
#install "cellcall"
devtools::install_github("ShellyCoder/cellcall")
        #Downloading GitHub repo ShellyCoder/cellcall@HEAD
        #These packages have more recent versions available.
        #It is recommended to update all of them.
        #Which would you like to update?
        #1: All
        #2: CRAN packages only
        #3: None
        #4: fs (1.5.2 -> 1.6.1) [CRAN]
        #Enter one or more numbers, or an empty line to skip updates:
        #選擇"1"

#checking packages have been installed
library(Seurat)
library (ggplot2)
library(patchwork)
library (cellcall)
```

```
>devtools::install_github("ShellyCoder/cellcall")

Downloading GitHub repo ShellyCoder/cellcall@HEAD
These packages have more recent versions available.
It is recommended to update all of them.
Which would you like to update?
1: All
2: CRAN packages only
3: None
4: fs (1.5.2 -> 1.6.1) [CRAN]
Enter one or more numbers, or an empty line to skip updates:
```

```
rm(list=ls(all=TRUE))
library(Seurat)
library (ggplot2)
library(patchwork)
library (cellcall)

#upload data
setwd("D:\\CellCall")
data0 <- readRDS("demo_immune.rds")
data0

colnames (data0@meta.data)
table (data0@meta.data$CellType, data0@meta.data$sample)

a<- data0@meta.data[,"sample"] %in% c('ade')
sum(a)
cells.use <- row.names(data0@meta.data)[a==TRUE]
data <- subset(data0, cells=cells.use)
data
```

```
> rm(list=ls(all=TRUE))
> library(Seurat)
> library (ggplot2)
> library(patchwork)
> library (cellcall)
>
> #upload data
> setwd(                )
> data0<- readRDS("demo_immune          .rds")
> data0
An object of class Seurat
23148 features across 1772 samples within 1 assay
Active assay: RNA (23148 features, 0 variable features)
>
> colnames (data0@meta.data)
[1] "orig.ident"   "nCount_RNA"   "nFeature_RNA" "CellType"     "sample"
> table (data0@meta.data$CellType, data0@meta.data$sample)

             ade nor
  DC           3  84
  Macrophage 298 480
  NK          29  98
  T cell     402 378
>
> a<- data0@meta.data[,"sample"] %in% c('ade')
> sum(a)
[1] 732
> cells.use <- row.names(data0@meta.data)[a==TRUE]
> data <- subset(data0, cells=cells.use)
> data
An object of class Seurat
23148 features across 732 samples within 1 assay
Active assay: RNA (23148 features, 0 variable features)
```

檔案路徑 / 等於 \\
引號 “ ” 等於 ‘ ‘

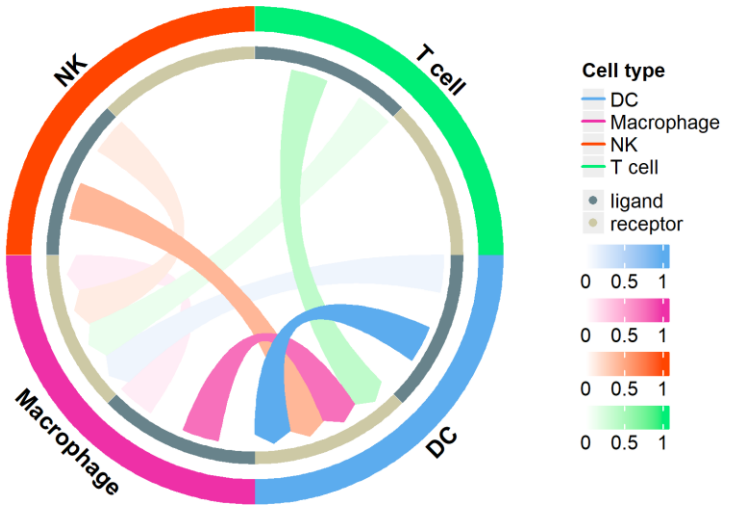
```
mt <- CreateObject_fromSeurat(Seurat.object=data,
                              slot="counts",
                              cell_type="CellType",
                              data_source="UMI",
                              scale.factor = 10^6,
                              Org = "Homo sapiens")


mt <- TransCommuProfile(object = mt,
                        pValueCor =0.05
                        CorValue = 0.1,
                        topTargetCor=1,
                        p.adjust = 0.05,
                        use.type="median",
                        probs = 0.9,
                        method="mean",
                        IS_core = TRUE,
                        Org = 'Homo sapiens')
```

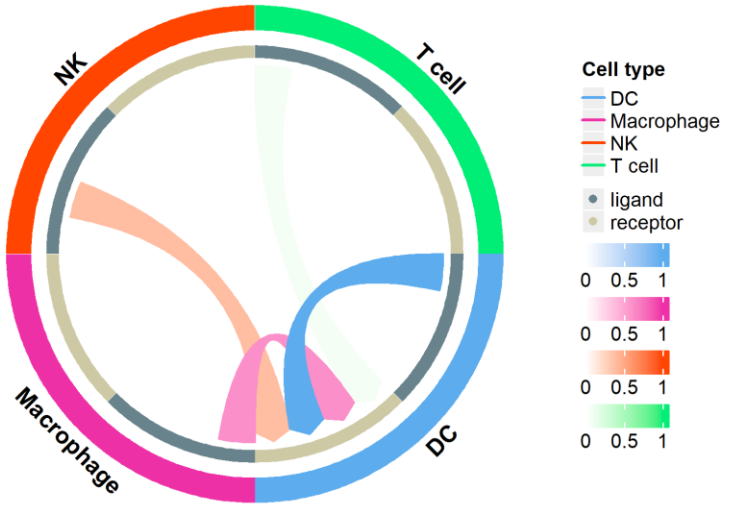
```
cell_color <-
data.frame(color=c("steelblue2","maroon2","orangered","springgreen2"),
stringsAsFactors = FALSE)
rownames(cell_color) <- c('DC','Macrophage','NK','T cell')

ViewInterCircos(object = mt, font = 2, cellColor = cell_color,
        lrColor = c("#68838B", "lemonchiffon3"),
        arr.type = "big.arrow",arr.length =0.04,
        trackhight1 = 0.1,
        slot="expr_l_r_log2_scale",
        linkcolor.from.sender = TRUE,
        linkcolor = NULL, gap.degree = 0.08,
        order.vector=c('DC','Macrophage','NK','T cell'),
        trackhight2 = 0.05, track.margin2 = c(0.02,0.05), DIY = FALSE)
```

**Normal**




**Tumor**



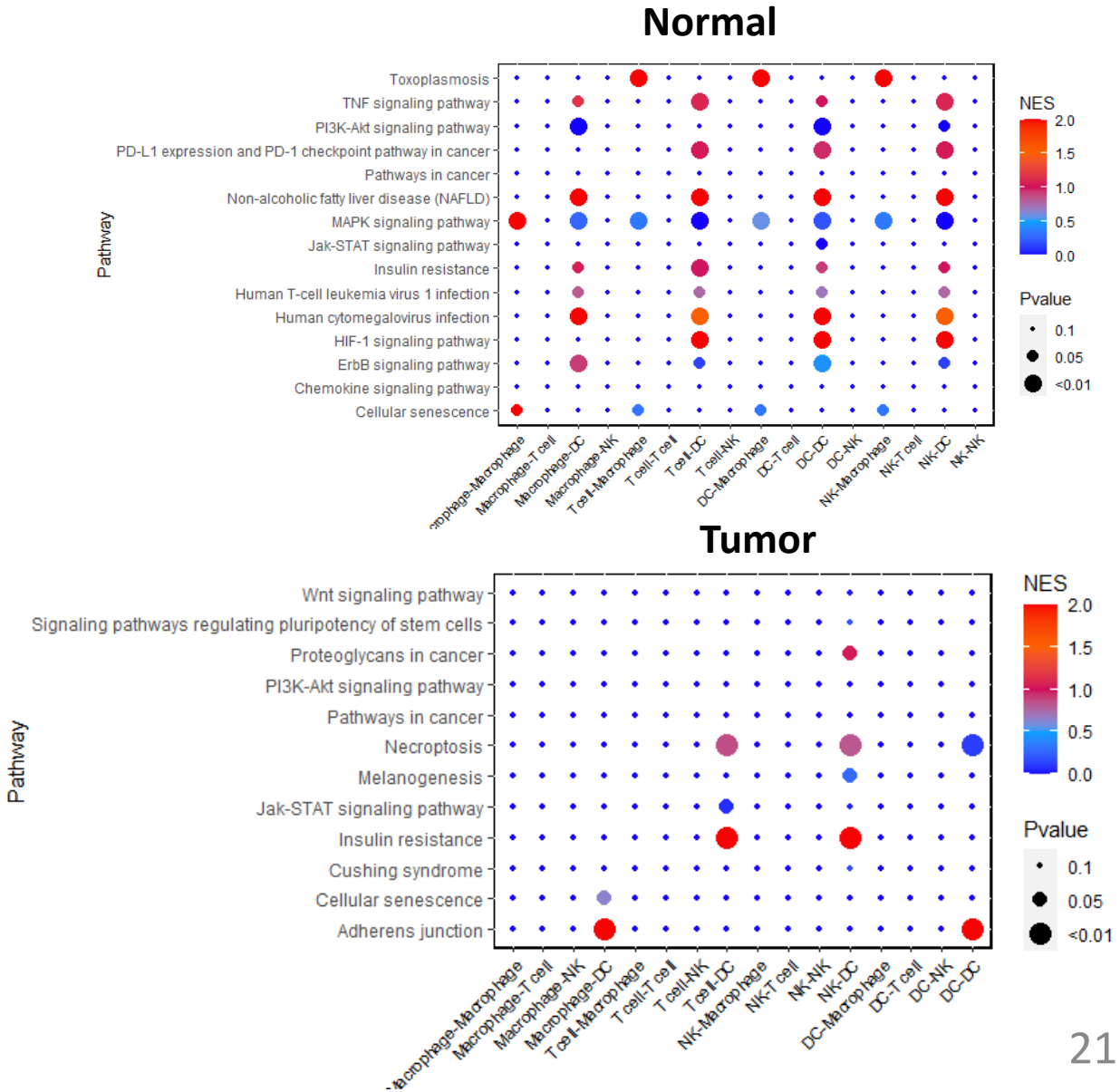
**Applications of cell-to-cell interaction:**

1. Microenvironment in normal and tumor tissue
2. Pathological classification (TNM stage)
3. Before and after drug treatment
4. Co-culture system
5. Tissue development (with trajectory analysis)

```
n <- mt@data$expr_l_r_log2_scale

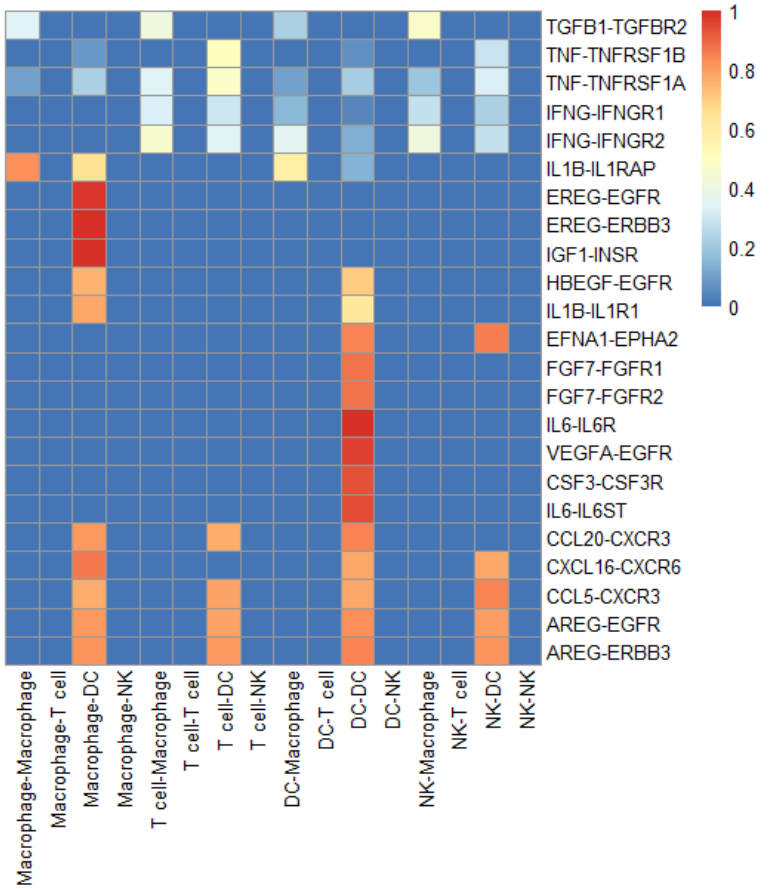
pathway.hyper.list <- lapply(colnames(n), function(i){
   print(i)
   tmp <- getHyperPathway(data = n, object = mt, cella_cellb = i,
Org="Homo sapiens")
   return(tmp)  })

myPub.df <- getForBubble(pathway.hyper.list,
cella_cellb=colnames(n))
p <- plotBubble(myPub.df)
p
```

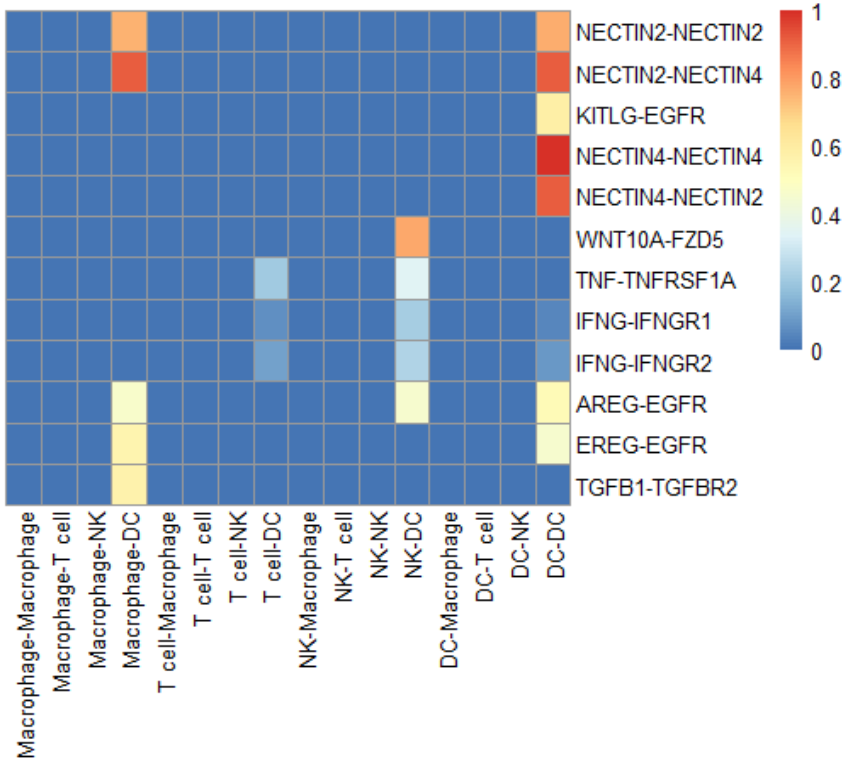


```
viewPheatmap(object = mt, slot="expr_l_r_log2_scale",
show_rownames = T,show_colnames = T,treeheight_row=0, treeheight_col=10,
cluster_rows = T,cluster_cols = T,fontsize = 10,angle_col = "90")
```

**Normal**



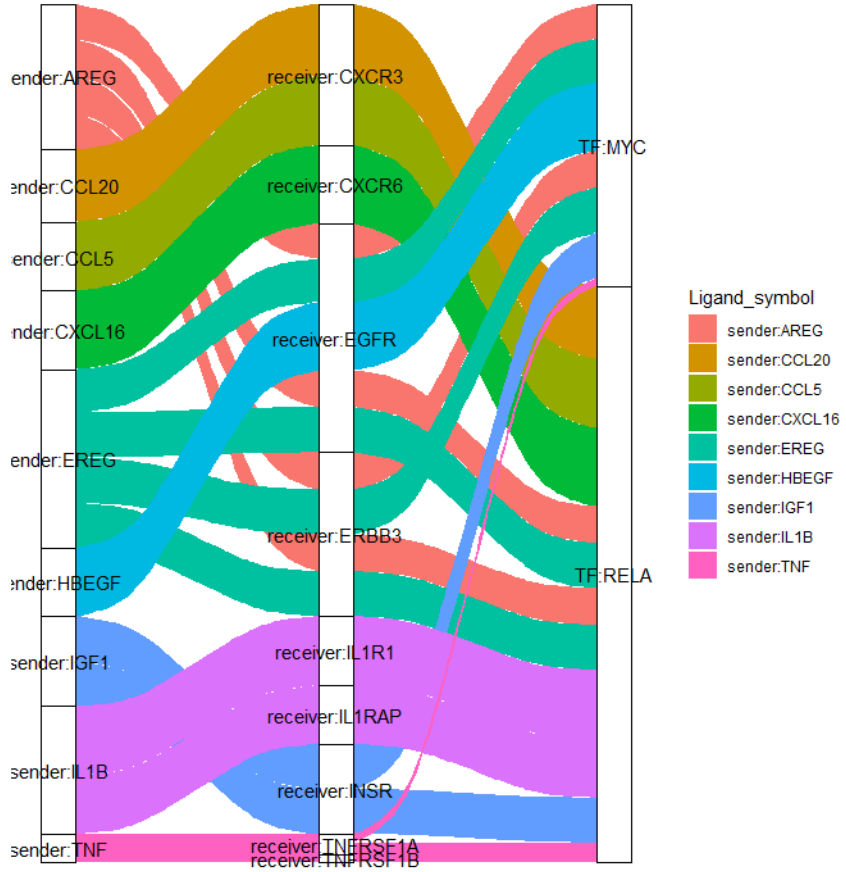

**Tumor**



```
mt <- LR2TF(object = mt, sender_cell="Macrophage", recevier_cell="DC",
            slot="expr_l_r_log2_scale", org="Homo sapiens")
 head(mt@reductions$sankey)
if(!require(networkD3)){
BiocManager::install("networkD3")}
library(networkD3)
library(magrittr)
library(dplyr)
tmp <- mt@reductions$sankey
tmp1 <- dplyr::filter(tmp, weight1>0)
tmp.df <- trans2tripleScore(tmp1)
head(tmp.df)

mycol.vector = NULL
elments.num <-  tmp.df %>% unlist %>% unique %>% length()
mycol.vector.list <- rep(mycol.vector, times=ceiling(elments.num/length(mycol.vector)))

sankey_graph(df = tmp.df, axes=1:3, mycol = mycol.vector.list[1:elments.num],
 nudge_x = NULL, font.size = 4, boder.col="white", isGrandSon = TRUE)
```




| | Ligand | Receptor | TF | value |
|---|---|---|---|---|
| 1 | sender:NECTIN2 | receiver:NECTIN4 | TF:CTNNB1 | 0.9138468 |
| 2 | sender:NECTIN2 | receiver:NECTIN2 | TF:CTNNB1 | 0.7518381 |
| 3 | sender:TGFB1 | receiver:TGFBR2 | TF:RB1 | 0.5631452 |
| 4 | sender:EREG | receiver:EGFR | TF:RXRA | 0.5509881 |
| 5 | sender:AREG | receiver:EGFR | TF:RXRA | 0.4606457 |

```
TF.tf <- names(mt@data$gsea.list$DC@geneSets)
TF.tf

egmt <- mt@data$gsea.list$DC
egmt.df <- data.frame(egmt)
head(egmt.df[,1:6])
flag.index <- which(egmt.df$pvalue < 0.05)
ridgeplot.DIY(x=egmt, fill="pvalue", showCategory=flag.index,
core_enrichment = T, orderBy = "pvalue", decreasing = TRUE)
```

```
> TF.tf <- names(mt@data$gsea.list$DC@geneSets)
> TF.tf
[1] "CTNNB1" "FOXO1"  "LEF1"   "RB1"    "RXRA"   "SREBF1" "STAT4"
>
> egmt <- mt@data$gsea.list$DC
> egmt.df <- data.frame(egmt)
> head(egmt.df[,1:6])
           ID Description setSize enrichmentScore      NES       pvalue
SREBF1 SREBF1      SREBF1     307        0.9204249 1.352337 1.000000e-10
STAT4   STAT4       STAT4     276        0.9223197 1.351613 1.000000e-10
CTNNB1 CTNNB1      CTNNB1     410        0.8603253 1.264993 6.522712e-09
RB1         RB1         RB1     170        0.9029795 1.316360 4.043646e-08
FOXO1     FOXO1       FOXO1     118        0.9181456 1.328885 1.095413e-06
LEF1       LEF1        LEF1     270        0.8611627 1.260690 4.463256e-06
```

