ccc和cnv例子

细胞间通讯 (Cell—cell communication, CCC) 是受生化信号调节的细胞间相互作用，能够调节单个细胞的生命过程和细胞间关系。

常用的CCC分析工具有CellChat、CellPhone、CellCall、NicheNet等。现利用CellChat、CellPhone、NicheNet软件分析举例。

常用的CNV分析工具有：copykat，inferCNV

文章：Revealing the transcriptional heterogeneity of organspecific metastasis in human gastric cancer using single-cell RNA Sequencing

DOI: 10.1002/ctm2.730

RESEARCH ARTICLE

CLINICAL AND TRANSLATIONAL MEDICINE

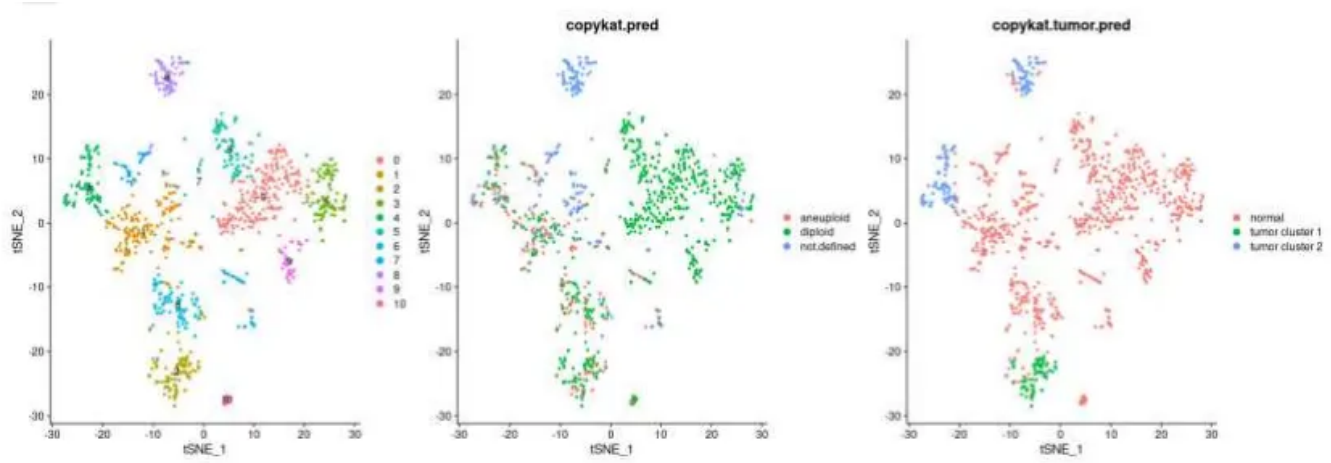
WILEY

Revealing the transcriptional heterogeneity of organ-specific metastasis in human gastric cancer using single-cell RNA Sequencing

Haiping Jiang[1] | Dingyi Yu[2] | Penghui Yang[2] | Rongfang Guo[2] | Mei Kong[3] | Yuan Gao[4] | Xiongfei Yu[5] | Xiaoyan Lu[2,6] | Xiaohui Fan[2,6,7]

copykat

用法：通过CopyKAT确定拷贝数变异，恶性上皮细胞根据拷贝数变异重聚类

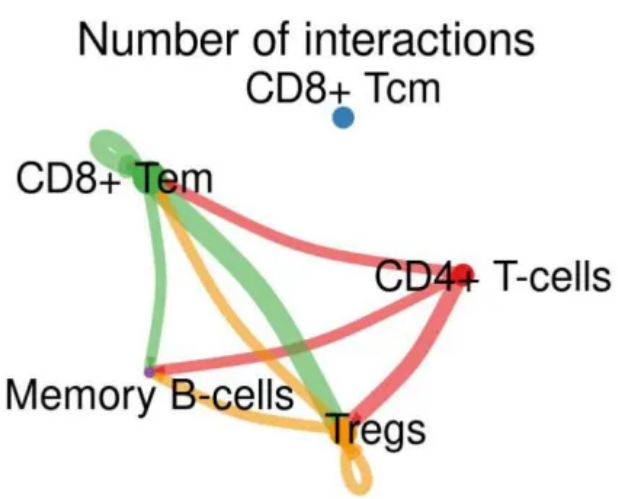


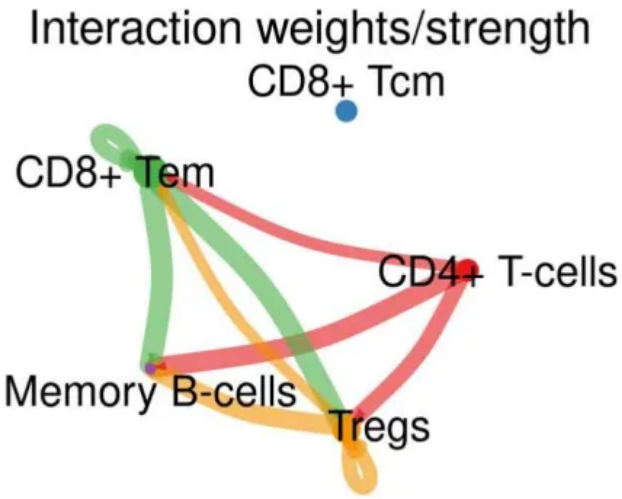
copykat Plain Text

```
# 清理环境并加载所需包
rm(list = ls())
options(stringsAsFactors = F)
library(Seurat)
library(ggplot2)
library(cowplot)
library(copykat)
library(parallelDist)

# 设置工作目录并读取数据
dir.create("8-CNV")
setwd("8-CNV")
sce <- readRDS("../combined.rds")
Idents(sce) <- sce$celltype

# 抽样，仅用于教程，避免流程运行过长
seurat_object <- subset(sce, downsample = 200)
table(Idents(seurat_object))

# 运行CopyKAT进行CNV分析并保存结果
counts <- as.matrix(seurat_object@assays$RNA$counts)
cnv <- copykat(rawmat = counts, ngene.chr = 5, sam.name = "test", n.cores = 8)
saveRDS(cnv, "cnv.rds")

# 读取CNV结果并进行可视化
seurat_object$CopyKAT <- cnv$prediction$copykat.pred
plot_list <- list(
  DimPlot(seurat_object, group.by = "celltype", label = T, reduction = 'tsne'),
  DimPlot(seurat_object, group.by = "CopyKAT", reduction = 'tsne') +
    scale_color_manual(values = c("#F8766D", '#02BFC4', "gray"))
)
ggsave("pred_mallignant.pdf", do.call(cowplot::plot_grid, plot_list), width = 12, height = 5)

# 肿瘤细胞亚群分析
pred.test <- data.frame(cnv$prediction)
CNA.test <- data.frame(cnv$CNAmat)
tumor.cells <- pred.test$cell.names[which(pred.test$copykat.pred == "aneuploid")]
tumor.mat <- CNA.test[, colnames(CNA.test) %in% tumor.cells]
hcc <- hclust(parallelDist::parDist(t(tumor.mat), threads = 4, method = "euclidean"), method = "ward.D2")
hc.umap <- cutree(hcc, 2)
```

```

# 投射CNV结果到单细胞聚类
standard10X <- function(dat, nPCs = 50, res = 1.0) {
  srat <- CreateSeuratObject(dat)
  srat <- NormalizeData(srat)
  srat <- ScaleData(srat)
  srat <- FindVariableFeatures(srat)
  srat <- RunPCA(srat)
  srat <- RunTSNE(srat, dims = seq(nPCs))
  srat <- FindNeighbors(srat, dims = seq(nPCs))
  srat <- FindClusters(srat, res = res)
  return(srat)
}

GC1 <- standard10X(counts, nPCs = 30, res = 0.8)
GC1$copykat.pred <- pred.test$copykat.pred
GC1$copykat.tumor.pred <- "normal"
GC1$copykat.tumor.pred[rownames(GC1@meta.data) %in% names(hc.umap[hc.umap == 1])] <- "tumor cluster 1"
GC1$copykat.tumor.pred[rownames(GC1@meta.data) %in% names(hc.umap[hc.umap == 2])] <- "tumor cluster 2"

plot_list2 <- list(
  DimPlot(GC1, label = T),
  DimPlot(GC1, group.by = "copykat.pred"),
  DimPlot(GC1, group.by = "copykat.tumor.pred")
)
do.call(cowplot::plot_grid, plot_list2)

# 标记基因表达可视化
FeaturePlot(GC1, features = c("PTPRC", "EPCAM"), order = T)

setwd('../')
```

CellChat


Number of interactions
CD8+ Tcm
CD8+ Tem
CD4+ T-cells
Memory B-cells
Tregs


Interaction weights/strength
CD8+ Tcm
CD8+ Tem
CD4+ T-cells
Memory B-cells
Tregs

cellchat Plain Text

```
# 清理环境并加载所需包
rm(list = ls())
library(Seurat)
library(SeuratObject)
library(ggplot2)
library(cowplot)
library(dplyr)
library(CellChat)
library(tidyverse)
library(ggalluvial)

# 设置工作目录
dir.create("10-cellchat")
setwd('10-cellchat/')
sce <- readRDS("../9-T/T_sce_celltype.rds")

# 设置细胞标识
Idents(sce) <- sce$singleR

# 创建CellChat对象
cellchat <- createCellChat(sce@assays$RNA$data, meta = sce@meta.data, grou
p.by = "singleR")
CellChatDB <- CellChatDB.human
CellChatDB.use <- subsetDB(CellChatDB, search = "Secreted Signaling")
cellchat@DB <- CellChatDB.use

# 预处理表达数据
cellchat <- subsetData(cellchat)
cellchat <- identifyOverExpressedGenes(cellchat)
cellchat <- identifyOverExpressedInteractions(cellchat)
cellchat <- projectData(cellchat, PPI.human)

# 细胞通讯预测
cellchat <- computeCommunProb(cellchat)
df.net <- subsetCommunication(cellchat, slot.name = 'net')
df.net2 <- subsetCommunication(cellchat, signaling = c('MIF'))
cellchat <- computeCommunProbPathway(cellchat)
df.netp <- subsetCommunication(cellchat, slot.name = 'netP')

# 细胞互作关系展示
cellchat <- aggregateNet(cellchat)
groupSize <- as.numeric(table(cellchat@idents))

# 可视化细胞亚群间的互作数量与概率
par(mfrow = c(1, 1), xpd = TRUE)
```

```
netVisual_circle(cellchat@net$count, vertex.weight = groupSize, weight.scale = TRUE, label.edge = FALSE, title.name = 'Number of interactions')

par(mfrow = c(1, 1), xpd = TRUE)
netVisual_circle(cellchat@net$weight, vertex.weight = groupSize, weight.scale = TRUE, label.edge = FALSE, title.name = 'Interaction weights/strength')

# 检查单个细胞亚群的互作信号强度
mat <- cellchat@net$weight
par(mfrow = c(3, 4), xpd = TRUE)
for (i in 1:nrow(mat)) {
  mat2 <- matrix(0, nrow = nrow(mat), ncol = ncol(mat), dimnames = dimnames(mat))
  mat2[i, ] <- mat[i, ]
  netVisual_circle(mat2, vertex.weight = groupSize, weight.scale = TRUE, edge.weight.max = max(mat), title.name = rownames(mat)[i])
}

saveRDS(cellchat, file = "cellchat.rds")

# 可视化特定信号通路'MIF'
pathways.show <- c('MIF')

par(mfrow = c(1, 1))
netVisual_aggregate(cellchat, layout = 'hierarchy', signaling = pathways.show, vertex.receiver = c(1, 2, 3))

par(mfrow = c(1, 1))
netVisual_aggregate(cellchat, layout = 'circle', signaling = pathways.show)

par(mfrow = c(1, 1))
netVisual_aggregate(cellchat, layout = 'chord', signaling = pathways.show)

par(mfrow = c(1, 1))
netVisual_heatmap(cellchat, signaling = pathways.show, color.heatmap = c("white", "#b2182b"))

# 配受体对的贡献分析
netAnalysis_contribution(cellchat, signaling = pathways.show)
pairLR.CXCL <- extractEnrichedLR(cellchat, signaling = pathways.show)
LR.show <- pairLR.CXCL[1,]

# 细胞亚群间多个信号通路的可视化
netVisual_bubble(cellchat, sources.use = 4, targets.use = c(1, 2))
plotGeneExpression(cellchat, signaling = 'MIF', type = 'violin')
```

文章：Delineating the dynamic evolution from preneoplasia to invasive lung adenocarcinoma by integrating single-cell RNA sequencing and spatial transcriptomics

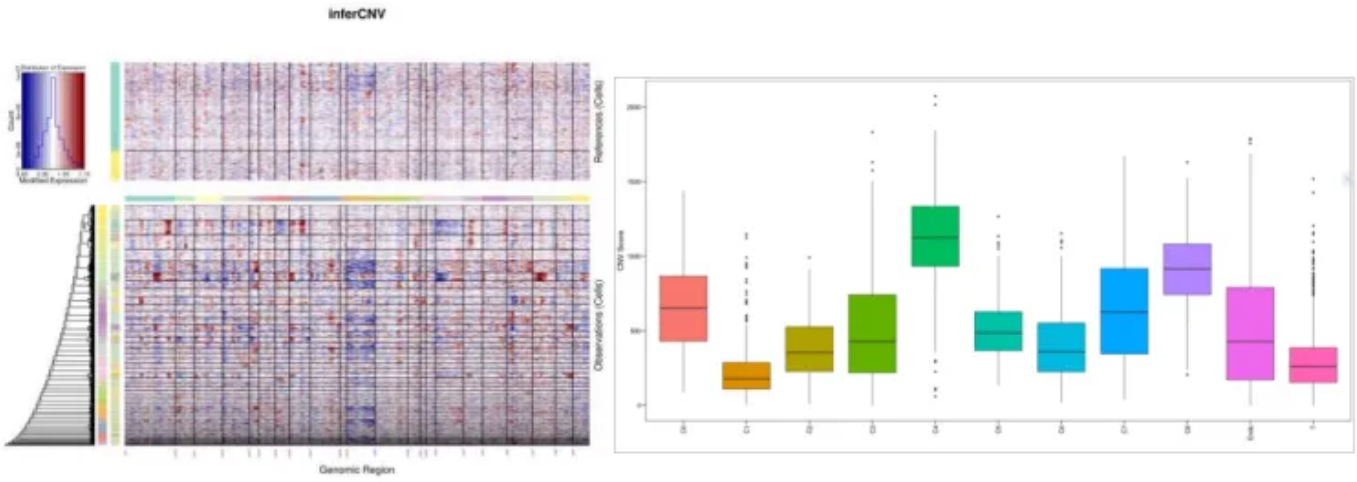
Article | Open access | Published: 25 November 2022

Delineating the dynamic evolution from preneoplasia to invasive lung adenocarcinoma by integrating single-cell RNA sequencing and spatial transcriptomics

Jianfei Zhu, Yue Fan, Yanlu Xiong, Wenchen Wang, Jiakuan Chen, Yanmin Xia, Jie Lei ✉, Li Gong ✉, Shiquan Sun ✉ & Tao Jiang ✉

Experimental & Molecular Medicine **54**, 2060–2076 (2022) | Cite this article

inferCNV



```
# 清理环境并加载所需包
rm(list = ls())
options(stringsAsFactors = F)
library(Seurat)
library(ggplot2)
library(cowplot)
library(dplyr)
library(infercnv)

# 设置工作目录
dir.create("7-epi")
setwd("7-epi")
set.seed(12345)

# 读取数据并筛选上皮细胞
sce.all = readRDS("../3-Celltype/sce_celltype.rds")
sce1 = sce.all[, sce.all$celltype == 'Epithelial']
sce1 = JoinLayers(sce1)

# 数据标准化与降维
sce = NormalizeData(sce1, normalization.method = "LogNormalize", scale.fa
ctor = 1e4)
sce = FindVariableFeatures(sce, selection.method = "vst", nfeatures = 200
0)
sce = ScaleData(sce)
sce = RunPCA(sce, features = VariableFeatures(sce))
sce = FindNeighbors(sce, dims = 1:15)
sce = FindClusters(sce, resolution = 0.1)
sce = RunTSNE(sce, dims = 1:15, do.fast = TRUE)
sce = RunUMAP(sce, dims = 1:5, do.fast = TRUE)

# UMAP可视化
DimPlot(sce, label = TRUE, cols = mycolors) + ggsave('umap_by_celltype.pd
f', width = 9, height = 7)

# 标记基因表达可视化
genes_to_check = toupper(c('TM4SF1', 'CRABP2', 'UBE2C', 'TOP2A', 'MKI6
7', 'CAV1', 'CLDN18', 'CAPS', 'SCGB1A1'))
DotPlot(sce, features = unique(genes_to_check), assay = 'RNA') + coord_fl
ip() + ggsave('check_markers.pdf', height = 9, width = 7)

# 准备CNV分析输入文件
dir.create("CNV")
setwd("CNV")

```

```r
epi_sce = subset(sce, downsample = 200)
epi_sce$cnv <- paste0("C", Idents(epi_sce))

Endothelial = sce.all[, sce.all$celltype == 'Endothelial']
T = subset(sce.all[, sce.all$celltype == 'T&NK'], downsample = 500)

spike_mat1 = as.data.frame(Endothelial[["RNA"]]$counts)
spike_mat2 = as.data.frame(T[["RNA"]]$counts)
epiMat = as.data.frame(epi_sce[["RNA"]]$counts)

ids = intersect(rownames(epiMat), rownames(spike_mat1))
this_dat = cbind(epiMat[ids,], spike_mat1[ids,], spike_mat2[ids,])

groupinfo = data.frame(v1 = colnames(this_dat),
                       v2 = c(epi_sce$cnv,
                              rep('spike-1', ncol(spike_mat1)),
                              rep('ref-1', ncol(spike_mat1)),
                              rep('spike-2', ncol(spike_mat2)),
                              rep('ref-2', ncol(spike_mat2))))
write.table(groupinfo, file = 'groupFiles.txt', sep = '\t', quote = F, col.names = F, row.names = F)

# 准备基因信息
library(AnnoProbe)
geneInfor = annoGene(rownames(this_dat), "SYMBOL", 'human')
geneInfor = geneInfor[!duplicated(geneInfor[, 1]), ][order(geneInfor$chr, geneInfor$start), c(1, 4:6)]
write.table(geneInfor, file = 'geneFile.txt', sep = '\t', quote = F, col.names = F, row.names = F)
write.table(this_dat, file = 'expFile.txt', sep = '\t', quote = F, row.names = T)

# 运行inferCNV
infercnv_obj = CreateInfercnvObject(raw_counts_matrix = 'expFile.txt',
                                    annotations_file = 'groupFiles.txt',
                                    delim = "\t",
                                    gene_order_file = 'geneFile.txt',
                                    ref_group_names = c('ref-1', 'ref-2'))

infercnv::run(infercnv_obj, cutoff = 0.1, out_dir = "infercnv_output", cluster_by_groups = TRUE, hclust_method = "ward.D2", plot_steps = TRUE)

# 应用inferCNV结果
infer_CNV_obj <- readRDS('infercnv_output/run.final.infercnv_obj')
expr <- infer_CNV_obj@expr.data
data_cnv <- as.data.frame(expr)
```

```
# CNV打分
ref_expr = expr[, c(infer_CNV_obj@reference_grouped_cell_indices$`ref-1`, infer_CNV_obj@reference_grouped_cell_indices$`ref-2`)]
mean_ref = mean(rowMeans(ref_expr))
sd_ref = mean(apply(ref_expr, 1, sd))
down = mean_ref - 2 * sd_ref
up = mean_ref + 2 * sd_ref
oneCopy = up - down

cnv_score_table <- expr
cnv_score_table[expr > 0 & expr < down - 2 * oneCopy] <- 2  # complete loss
cnv_score_table[expr >= down - 2 * oneCopy & expr < down - oneCopy] <- 1  # loss of one copy
cnv_score_table[expr >= down & expr < up] <- 0  # neutral
cnv_score_table[expr >= up & expr <= up + oneCopy] <- 1  # addition of one copy
cnv_score_table[expr > up + oneCopy] <- 2  # addition of more than one copy

# 保存结果
save(groupinfo, geneInfor, data_cnv, epi_sce, file = 'infercnv.Rdata')
```

除了上述例子外，一些软件的官方网站也能够很好的帮助我们学习。

CellphoneDB

安装 Plain Text

```
创建 python=>3.8 环境

使用 conda：conda create -n cpdb python=3.8

使用 virtualenv：python -m venv cpdb

激活环境

使用 conda：source activate cpdb

使用 virtualenv：source cpdb/bin/activate

安装 CellphoneDBpip install cellphonedb

设置 Jupyter 笔记本的内核。

安装 ipython 内核：.pip install -U ipykernel

将环境添加为 jupyter 内核：。python -m ipykernel install --user --name 'cpdb'

打开/启动 Jupyter 并选择创建的内核。
```

下载数据库 Plain Text

```
from cellphonedb.utils import db_utils

cpdb_target_dir = "./cellphonedb_data"  # 目标目录
cpdb_version = "v2.0.0"  # 指定版本

db_utils.download_database(cpdb_target_dir, cpdb_version)

```

运行CellphoneDB Plain Text

```
##使用统计方法运行分析
cellphoneDB method statistical_analysis test_meta.txt test_counts.txt
##使用文本文件
cellphoneDB method analysis test_meta.txt test_counts.txt
##输入文件h5ad
cellphonedb method analysis test_meta.txt test_counts.h5ad
```

PlantPhoneDB

1. 如何构建植物配体–受体库
2. 如何定义配体–受体互作的score
3. 如何删选显著的配体–受体互作对，或者定义相应的统计学参数pvalue

拟南芥数据集 Plain Text

```
install.packages(c("devtools", "Seurat", "tidyverse", "ggplot2", "ggsci", "pheatmap", "ggpubr", "RColorBrewer", "patchwork", "lsa", "viridis", "hrbrthemes", "circlize", "chorddiag", "ggplotify", "data.table", "parmigene", "infotheo", "igraph", "cowplot", "grid", "dplyr"))
library(devtools)
install_github("Jasonxu0109/PlantPhoneDB")
library(Seurat)

library(tidyverse)

library(ggplot2)

library(ggsci)

library(ggpubr)

library(pheatmap)

library(RColorBrewer)

library(patchwork)

library(lsa)

library(viridis)

library(hrbrthemes)

library(circlize)

library(chorddiag)

library(ggplotify)

library(data.table)

library(parmigene)

library(readxl)

library(infotheo)

library(igraph)

library(muxViz)
```

```
library(rgl)

library(tidyverse)

library(dplyr)
pbmc <- readRDS("pbmc3k_final.rds")

pbmc@meta.data$labels <- Idents(pbmc)



control.data <- Read10X(data.dir = "control/filtered_feature_bc_matrix/")

control<- CreateSeuratObject(counts = control.data, project = "control",
min.cells = 3, min.features = 200)

control <- subset(control, subset = nFeature_RNA > 200 & nCount_RNA > 100
0)

control <- SCTransform(control, verbose = FALSE)







heat.data <- Read10X(data.dir = "heat/filtered_feature_bc_matrix/")

heat<- CreateSeuratObject(counts = heat.data, project = "heat", min.cell
s = 3, min.features = 200)

heat <- subset(heat, subset = nFeature_RNA > 200 & nCount_RNA > 1000)

heat <- SCTransform(heat, verbose = FALSE)
datasets <- c(control,heat)

features <- SelectIntegrationFeatures(object.list = datasets, nfeatures
= 8000)

datasets <- PrepSCTIntegration(object.list = datasets, anchor.features =
features, verbose = TRUE)

datasets <- lapply(X = datasets, FUN = RunPCA, verbose = FALSE, features
= features)

anchors <- FindIntegrationAnchors(object.list = datasets, normalization.m
ethod = "SCT",anchor.features = features, verbose = TRUE, reference=1,red
uction = "cca")
```

```
83
84  objs <- IntegrateData(anchorset = anchors, normalization.method = "SCT",
    verbose = TRUE)
85
86
87
88
89
90
91  objs <- RunPCA(objs, verbose = FALSE, approx = FALSE, npcs = 50)
92
    objs <- RunUMAP(objs, reduction = "pca", dims = 1:50, umap.method = "umap
    -learn", metric = "correlation")
93
94  objs <- RunTSNE(objs, reduction = "pca",dims = 1:50,tsne.method = "Rtsn
    e")
95
96
97  objs <- FindNeighbors(objs, reduction = "pca",dims = 1:50)
98
99  objs <- FindClusters(objs, resolution = 0.5, algorithm = 2)
100 DefaultAssay(objs) <- 'SCT'
101
102 objs<- PrepSCTFindMarkers(objs,assay = "SCT", verbose = TRUE)
103
104 DEG <- FindAllMarkers(objs,
105
106                       logfc.threshold=0.25,
107
108                       min.diff.pct = 0.25,
109
110                       max.cells.per.ident = 10000,
111
112                       only.pos=T)
113
114
115 mark_gene <- DEG %>% mutate(avg_logFC=avg_log2FC) %>% filter(p_val_adj<0.
    05)
116
117
118
119
120
121 signature <- readxl::read_excel('../ath_doi_202104.xlsx')
122
123 sig_gene  <- signature %>% as.data.frame() %>% filter(Tissue=="Root") %
    >% mutate(V1=`Cell Type`,V2=Cell_Marker) %>% unique(.) %>% select(V1,V2)
124 new.cluster.ids <- c("Columella","Vasculature","Cortex","Endodermis","Ste
    le","Trichoblast","Non-hair cell","Non-hair cell","Xylem pole pericycl
```

```
e","Phloem pole pericycle","Trichoblast","Xylem","Lateral root cap","Non-hair cell","Phloem companion cell","Root cap","Protoxylem")

names(new.cluster.ids) <- levels(objs)

objs <- RenameIdents(objs, new.cluster.ids)

saveRDS(objs,"objs.rds")
pdf("pic1.pdf")

options(repr.plot.width=6, repr.plot.height=5)

pic1 <- DimPlot(objs,group.by = 'labels', label=TRUE, label.size = 6, reduction='umap',cols=mycolor)+NoLegend()+ggtitle("")

pic1
objs@meta.data$treatment <- objs@meta.data$orig.ident


##聚类
pdf("pics1.pdf")

options(repr.plot.width=6, repr.plot.height=5)

pics1 <- DimPlot(objs,group.by = 'labels', split.by= 'treatment', label=TRUE, label.size = 6,cols=mycolor,ncol =1)+NoLegend()+ggtitle("")

pics1
#计算各cell类型的比例
pdf("pics2.pdf")

options(repr.plot.width=10, repr.plot.height=4)

pics2 <- objs@meta.data %>%

        ggplot(aes(treatment,fill=labels,color=I('white')))+

        geom_bar(position = "fill")+

        coord_flip()+

        theme_bw()+

        ylab("")+

        theme(panel.grid.major = element_blank(),

              panel.grid.minor = element_blank(),
```

```
                panel.border=element_blank(),
                axis.title=element_text(size=7.82,face="bold"),
                axis.text=element_text(size=7.82,color='black'),
                legend.text=element_text(size=7.82),
                plot.title = element_text(size = 7.82, face = "bold"),
                axis.line=element_line(color='black'),
                legend.title = element_text(size = 7.82))+
        scale_y_continuous(position = "right",expand = c(0,0))+
        scale_fill_manual(values=mycolor)

pics2
##运用fisher检验热处理下细胞响应胁迫的偏好性（基于测试数据）
tbl <- table(objs@meta.data$labels,objs@meta.data$treatment)

res = chisq.test(tbl)
expected = res$expected

roe = tbl/expected
##marker gene展示
meta <- objs@meta.data %>% select(seurat_clusters,labels) %>% unique()

mark_gene <- DEG %>% mutate(avg_logFC=avg_log2FC) %>%
    filter(p_val_adj<0.05 & avg_log2FC>1.5 & gene %in% sig_gene$V2) %>%
    inner_join(meta,c("cluster" = "seurat_clusters")) %>%
    arrange(p_val)




expr <- AverageExpression(objs,assays ='SCT')

expr <- expr$SCT

expr <- expr[mark_gene$gene,]    //取出自己想展示的mark gene
```

```
217
218
219
//expr <- expr[,c('Trichoblast','Lateral root','Cortex','Atrichoblast','E
ndodermis','Xylem','Meristem','Phloem','Pericycle')]
220
221
222
223
pdf("pic3.pdf")
224
225
options(repr.plot.width=8, repr.plot.height=5)
226
227
pic3 <- pheatmap(t(expr), scale="column",angle_col=90,cluster_rows = F,cl
uster_cols = T,show_colnames=F)
228
229
pic3
230
##细胞通讯分析
231
source("PlantPhoneDB-main/R/CCI_circle.r")
232
233
source("PlantPhoneDB-main/R/CCI_network.r")
234
235
source("PlantPhoneDB-main/R/heatmap_count.r")
236
237
source("PlantPhoneDB-main/R/LR_pathway.r")
238
239
source("PlantPhoneDB-main/R/LRscore.r")
240
##加载拟南芥配体—受体对
241
load("PlantPhoneDB-main/LR_pair_ath.RDa")
242
243
LR_pair <- LR_pair %>% filter(source!="orthologs") %>% select(Ligands, Re
ceptors) %>% unique()
244
##计算配体—受体互作的分数
245
objs_heat <- subset(objs, treatment!="Control")
246
247
Heat <- LRscore(objs_heat@assays$SCT@data, LRdb=LR_pair, cluster = Idents
(objs_heat), min.pct = 0.1,iterations=100, method='Average')
248
249
## 获得每个细胞类型再control和heat处理之间的差异基因
250
DE_test <- NULL
251
252
for(i in unique(objs$labels))
253
254
{
255
256
    i
257
258
    objs_flt <- subset(objs,labels==i)
259
260
    Idents(objs_flt) <- objs_flt$treatment
```

```
    objs_flt <- PrepSCTFindMarkers(objs_flt ,assay = "SCT", verbose = TRUE)

    degs <- FindAllMarkers(objs_flt,logfc.threshold=0.25,min.diff.pct = 0.25,max.cells.per.ident = 10000)

    if(length(degs)!=0)

    {

        degs <- degs %>% mutate(avg_logFC=avg_log2FC, labels=i) %>% filter(p_val_adj<0.05) %>% mutate(label_gene=paste0(labels,"_",gene))

        DE_test <- rbind(DE_test,degs)

    }

}
##识别热胁迫下显著差异的配体-受体互作对
Heat_sig <- Heat %>% filter(Pvalue<0.05) %>%

    mutate(Ligands_cell_gene=paste0(Ligands_cell,"_",Ligands), Receptors_cell_gene=paste0(Receptors_cell,"_",Receptors)) %>%

    filter(Ligands_cell_gene %in% DE_test$label_gene | Receptors_cell_gene %in% DE_test$label_gene)  %>%

    select(-Ligands_cell_gene,-Receptors_cell_gene)







interaction_count <- Heat_sig %>% group_by(Ligands_cell,Receptors_cell) %>% summarise(Number=n(),.groups = 'drop')



sum(interaction_count$Number)

interaction_count %>% mutate(Type=ifelse(Ligands_cell==Receptors_cell,"Autocrine","Paracrine")) %>% group_by(Type) %>% summarise(Number=sum(Number))
```

```
Autocrine <- interaction_count[interaction_count$Ligands_cell==interaction_count$Receptors_cell,]

Paracrine <- interaction_count[interaction_count$Ligands_cell!=interaction_count$Receptors_cell,]
##top10的配体-受体对在不同细胞互作之间的heatmap
Top10 <- Heat_sig %>%

    arrange(desc(Score)) %>%

    select(LR_pair) %>%

    unique() %>%

    head(10) %>%

    inner_join(Heat) %>%

    select(LR_pair,Cell_pair,Score) %>%

    spread(.,Cell_pair,Score) %>%

    replace(is.na(.), 0)







rownames(Top10) <- Top10$LR_pair

Top10 <- Top10[,-1]

Top10 <- t(Top10)

Top10 <- apply(Top10,2,function(x){x/max(x)})
pic6 <- pheatmap(Top10, scale="none",angle_col=45,fontsize_row=4,cluster_rows = T,cluster_cols = F,show_colnames=T)
##构建细胞之间的信号通路
geneSet <- fread('../plantGSAD/Ara_ALL.KEGG.txt')

geneSet$Gene <- toupper(geneSet$Gene)



CellA <- "Trichoblast"

CellB <- "Cortex"
```

```
lr <- subset(Heat_sig,Ligands_cell==CellA & Receptors_cell==CellB )

pathway_result2 <- LR_pathway(lr, objs_heat, CellA, CellB, neighbor=2, geneSet)

pathway_result2$FDR <- p.adjust(pathway_result2$Pvalue,method='BH')

pathway_result2 <- pathway_result2[order(pathway_result2$Pvalue),]
```

数据库：https://jasonxu.shinyapps.io/PlantPhoneDB/

下载Oryza sativa配受体文件，rds格式

单细胞数据：https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM4363201

水稻数据集 Plain Text

```
"barcodes.tsv.gz"
"features.tsv.gz"
"matrix.mtx.gz"#单细胞数据三个文件名称改为标准格式
# 如果没有这些包，这需要提前安装
install.packages(c("devtools", "Seurat", "tidyverse", "ggplot2", "ggsci", "pheatmap","ggpubr", "RColorBrewer", "patchwork", "lsa", "viridis", "hrbrthemes", "circlize","chorddiag", "ggplotify", "data.table", "parmigene", "infotheo", "igraph", "cowplot","grid", "dplyr")) ##Installs devtools and the PlantPhoneDB CRAN dependancies

# 安装PlantPhoneDB
library(devtools)
install_github("Jasonxu0109/PlantPhoneDB")

# 载入需要的R包
# Load the required packages
library(Seurat)
library(sceasy)
library(data.table)
library(readxl)
library(tidyverse)
library(anndata)
library(SeuratDisk)
library(PlantPhoneDB)
library(UniprotR)
library(riceidconverter)
seod <- Read10X('D:\\data') # 读取我们刚刚下载的单细胞数据


# 数据格式看一下行列内容，对于我们这个项目主要看下水稻的基因名规律：
seod[1:4,1:4]
4 x 4 sparse Matrix of class "dgCMatrix"
               AAACCTGAGCTTTGGT-1 AAACCTGAGGCTACGA-1 AAACCTGAGTAGTGCG-1
Add_EVM0003044                  .                  .                  .
EVM0007658                      .                  .                  .
LOC_Os02g38589                  .                  .                  .
LOC_Os02g38598                  .                  .                  .
               AAACCTGAGTTAAGTG-1
Add_EVM0003044                  .
EVM0007658                      .
LOC_Os02g38589                  .
LOC_Os02g38598                  .

seo <- CreateSeuratObject(seod) # 创建Seurt对象

```

```
# Warning: Feature names cannot have underscores ('_'), replacing with da
shes ('-')
seo <-subset(seo,cells=sample(colnames(seo),5000))  # 仅作为展示，随机取一部
分细胞，避免笔记本跑不动

seo

#An object of class Seurat
#39217 features across 5000 samples within 1 assay
#Active assay: RNA (39217 features, 0 variable features)
seo <- SCTransform(seo, verbose = FALSE)
seo <- RunPCA(seo, verbose = FALSE, approx = FALSE, npcs = 50) %>%
       RunUMAP(reduction = "pca", dims = 1:50,  metric = "correlation")
%>%
       FindNeighbors(reduction = "pca",dims = 1:50) %>%
       FindClusters(resolution = 0.5, algorithm = 2)
DimPlot(objs,label = T,label.size = 8)
#计算亚群间的差异基因
DefaultAssay(objs) <- 'SCT'
seo <- PrepSCTFindMarkers(object = seo)
DEG <- FindAllMarkers(seo,
                      logfc.threshold=0.25,
                      min.diff.pct = 0.25,
                      max.cells.per.ident = 10000,
                      only.pos=T)



mark_gene <- DEG %>%
    mutate(avg_logFC=avg_log2FC) %>%
    filter(p_val_adj<0.05)

# 查看差异基因
head(mark_gene)

               p_val avg_log2FC pct.1 pct.2 p_val_adj cluster          g
ene
LOC-Os01g11600     0   2.347920 0.869 0.248         0       0 LOC-Os01g11
600
LOC-Os05g25920     0   1.873972 0.986 0.719         0       0 LOC-Os05g25
920
LOC-Os09g31040     0   1.766442 0.828 0.335         0       0 LOC-Os09g31
040
LOC-Os04g52504     0   1.556508 0.993 0.674         0       0 LOC-Os04g52
504
LOC-Os09g29960     0   1.533655 0.836 0.318         0       0 LOC-Os09g29
960
```

```
LOC-Os09g23590     0   1.394198 0.966 0.521         0       0 LOC-Os09g23
590

               avg_logFC
LOC-Os01g11600  2.347920
LOC-Os05g25920  1.873972
LOC-Os09g31040  1.766442
LOC-Os04g52504  1.556508
LOC-Os09g29960  1.533655
LOC-Os09g23590  1.394198
load("D:\\data/LR_pair_osa.RDa")

head(LR_pair_osa)
  Ligands Receptors    source     Organism
1  Q69T51    D7UPN3 orthologs Oryza sativa
2  Q69T51    Q6ZD33 orthologs Oryza sativa
3  Q69T51    Q0J431 orthologs Oryza sativa
4  Q8GVT9    D7UPN3 orthologs Oryza sativa
5  Q8GVT9    Q6ZD33 orthologs Oryza sativa
6  Q8GVT9    Q0J431 orthologs Oryza sativa
table(LR_pair_osa$Ligands  %in% rownames(seo))

FALSE
 3762



table(LR_pair_osa$Receptors  %in% rownames(seo))

FALSE
 3762
#因配受体数据中的ID为uniprot格式，可在https://www.uniprot.org/id-mapping上进行
ID转换
> convID <- read.table('D:\\data\\uniprot-compressed_true_download_true_f
ormat_tsv-2023.05.05-09.48.37.34.tsv',header = T)
> head(convID)
    From           To
1 Q69T51 Os06g0208800
2 Q8GVT9 Os07g0287400
3 Q9SDJ5 Os01g0257100
4 Q2QP50 Os12g0541700
5 Q67VB1 Os06g0239100
6 Q6Z7S6 Os02g0740700
#再次转换ID
convtomsufun<-function(rapid){
    consumid<-RiceIDConvert(rapid,'RAP',toType = 'MSU')
    if(length(consumid[,2]) > 1 | length(consumid[,2]) ==1){
    return(str_replace(substring(consumid[,2][1],1,14),'_','-'))} else{re
turn(NULL)}

```

```
}

convID$MUSID<-unlist(map(convID$To,function(x){convtomsufun(x)}))

head(convID)

    From           To          MUSID
1 Q69T51 Os06g0208800 LOC-Os06g10660
2 Q8GVT9 Os07g0287400 LOC-Os07g18750
3 Q9SDJ5 Os01g0257100 LOC-Os01g15320
4 Q2QP50 Os12g0541700 LOC-Os12g35670
5 Q67VB1 Os06g0239100 LOC-Os06g13180
6 Q6Z7S6 Os02g0740700 LOC-Os02g50730

table(  convID$MUSID %in% rownames(seo) )

FALSE  TRUE
  137   417
#install.packages('hash')
library(hash)
idhash <- hash( convID$From, convID$MUSID) # 建立二者之间的hash关系。

converhash <- function(x){if(x %in% convID$From ){return(values(idhash,key=x))} else{return('NULL')}}


LR_pair_osa$Ligands_n<-unlist(map(LR_pair_osa$Ligands,function(x){converhash(x)}))
LR_pair_osa$Receptors_n<-unlist(map(LR_pair_osa$Receptors,function(x){converhash(x)}))

head(LR_pair_osa)

  Ligands Receptors    source     Organism      Ligands_n    Receptors_n
1  Q69T51    D7UPN3 orthologs Oryza sativa LOC-Os06g10660           NULL
2  Q69T51    Q6ZD33 orthologs Oryza sativa LOC-Os06g10660 LOC-Os08g42580
3  Q69T51    Q0J431 orthologs Oryza sativa LOC-Os06g10660 LOC-Os08g42580
4  Q8GVT9    D7UPN3 orthologs Oryza sativa LOC-Os07g18750           NULL
5  Q8GVT9    Q6ZD33 orthologs Oryza sativa LOC-Os07g18750 LOC-Os08g42580
6  Q8GVT9    Q0J431 orthologs Oryza sativa LOC-Os07g18750 LOC-Os08g42580
##构建配受体所需的格式数据
LR_pair <- data.frame(Ligands = LR_pair_osa$Ligands_n,Receptors=LR_pair_osa$Receptors_n)
head(LR_pair) # 按道理来讲，只有配体或者只有受体的行也应该过滤掉，我们这里并没有过滤，请注意。
         Ligands      Receptors
1 LOC-Os06g10660           NULL
2 LOC-Os06g10660 LOC-Os08g42580
```

```
3 LOC-Os06g10660 LOC-Os08g42580
4 LOC-Os07g18750           NULL
5 LOC-Os07g18750 LOC-Os08g42580
6 LOC-Os07g18750 LOC-Os08g42580

# 全剧最重要的函数
LRp <- LRscore(seo@assays$SCT@data, LRdb=LR_pair, cluster = Idents(seo),
min.pct = 0.1,iterations=100, method='Average')

 head(LRp) # 最重要的数据
         Ligands      Receptors Ligands_cell Receptors_cell Ligands_expr
1 LOC-Os02g50730 LOC-Os04g41030            0              0    2.6109641
2 LOC-Os06g13180 LOC-Os04g41030            0              0    1.5481818
3 LOC-Os01g16430 LOC-Os01g07200            0              0    0.3593471
4 LOC-Os03g05730 LOC-Os06g13180            0              0    0.3965515
5 LOC-Os03g05730 LOC-Os02g50730            0              0    0.3965515
6 LOC-Os05g46200 LOC-Os06g13180            0              0    0.2344154
  Receptors_expr     Score        Pvalue      Type
1      0.3012195 1.4560918 8.613986e-144 Autocrine
2      0.3012195 0.9247006 1.124863e-109 Autocrine
3      0.2242606 0.2918038  1.000000e+00 Autocrine
4      1.5481818 0.9723666  1.216372e-30 Autocrine
5      2.6109641 1.5037578 2.260378e-133 Autocrine
6      1.5481818 0.8912986  7.672429e-89 Autocrine
                        LR_pair Cell_pair
1 LOC-Os02g50730->LOC-Os04g41030      0->0
2 LOC-Os06g13180->LOC-Os04g41030      0->0
3 LOC-Os01g16430->LOC-Os01g07200      0->0
4 LOC-Os03g05730->LOC-Os06g13180      0->0
5 LOC-Os03g05730->LOC-Os02g50730      0->0
6 LOC-Os05g46200->LOC-Os06g13180      0->0
##LRscre源代码
# Generated from function body. Editing this file has no effect.
function (expr, LRdb, cluster = NULL, min.pct = 0.1, method = "LRscore",
    iterations = 10, seed = 123, ...)
{
    expr <- as.data.frame(expr) #速度应该比较慢，可以考虑加入一个pseudocell函数
    expr <- expr[rowSums(expr) > 0, ]
    u <- sum(expr)/(nrow(expr) * ncol(expr))
    expr_flt <- function(expr, LR_gene, ident = "NK") {
        tmp <- expr[rownames(expr) %in% LR_gene, new_cluster ==
            ident]
        pct <- apply(tmp, 1, function(x) {
            sum(x > 0)
        })
        pct <- pct/ncol(tmp)
        tmp <- tmp[pct > min.pct, ]
        tmp <- apply(tmp, 1, mean)
```

```
        return(tmp)
    }
    score <- function(l, r, u = NULL, method = "LRscore") {
        if (method == "LRscore") {
            if (!is.null(u)) {
                s <- (l * r)^(1/2)/((l * r)^(1/2) + u)
            }
            else {
                print("u is NULL")
            }
        }
        else if (method == "WeightProduct") {
            s <- l * r
            z_scores <- (s - mean(s))/sd(s)
            s <- (z_scores - min(z_scores))/(max(z_scores) -
                min(z_scores))
        }
        else if (method == "Average") {
            s <- (l + r)/2 # 最快的方法
        }
        else if (method == "Product") {
            s <- (l * r)
        }
        return(as.numeric(s))
    }
    LR_result <- NULL
    set.seed(seed)
    for (Ligands_cell in unique(cluster)) {
        for (Receptors_cell in unique(cluster)) {
            new_cluster <- cluster
            tmp.L <- expr_flt(expr, LRdb$Ligands, Ligands_cell)
            tmp.R <- expr_flt(expr, LRdb$Receptors, Receptors_cell)
            tmp.LR <- LRdb[LRdb$Receptors %in% names(tmp.R) &
                LRdb$Ligands %in% names(tmp.L), ]
            if (nrow(tmp.LR) <= 0) {
                next
            }
            tmp.LR$Ligands_cell <- Ligands_cell
            tmp.LR$Receptors_cell <- Receptors_cell
            tmp.LR$Ligands_expr <- tmp.L[tmp.LR$Ligands]
            tmp.LR$Receptors_expr <- tmp.R[tmp.LR$Receptors]
            tmp.LR$Score <- score(l = tmp.L[tmp.LR$Ligands],
                r = tmp.R[tmp.LR$Receptors], u = u, method)
            Score <- matrix(0, nrow(tmp.LR), iterations)
            if (method %in% c("Product", "Average")) {
                for (i in 1:iterations) {
                  cols <- sample(1:ncol(expr), ncol(expr))
                  new_cluster <- cluster[cols]
```

```
                    sample.L <- expr_flt(expr, LRdb$Ligands, Ligands_cell)
                    sample.R <- expr_flt(expr, LRdb$Receptors,
                      Receptors_cell)
                    L <- sample.L[tmp.LR$Ligands]
                    L[is.na(L)] <- 0
                    R <- sample.R[tmp.LR$Receptors]
                    R[is.na(R)] <- 0
                    SS <- score(l = L, r = R, u = u, method)
                    Score[, i] <- SS
                  }
```