

```

In [ ]: custom_heatmap = function(object,signaling = NULL, pattern = c("outgoing","incoming"),
                                color.use = NULL, color.heatmap = 'Blues', pathway

                                title = NULL, width = 8, height = 20, font.size =
                                row_clusters = NULL, column_clusters = NULL, seed

pattern <- match.arg(pattern)
set.seed(seed)
#####
#####construct matrix#####
#####
centr <- slot(object, slot.name)$centr
outgoing <- matrix(0, nrow = nlevels(object@idents), ncol = length(centr))
incoming <- matrix(0, nrow = nlevels(object@idents), ncol = length(centr))
dimnames(outgoing) <- list(levels(object@idents), names(centr))
dimnames(incoming) <- dimnames(outgoing)
for (i in 1:length(centr)) {
  outgoing[,i] <- centr[[i]]$outdeg
  incoming[,i] <- centr[[i]]$indeg
}
if (pattern == "outgoing") {
  mat <- t(outgoing)
  legend.name <- "Outgoing"
} else if (pattern == "incoming") {
  mat <- t(incoming)
  legend.name <- "Incoming"
} else if (pattern == "all") {
  mat <- t(outgoing+ incoming)
  legend.name <- "Overall"
}
if (!is.null(signaling)) {
  mat1 <- mat[rownames(mat) %in% signaling, , drop = FALSE]
  mat <- matrix(0, nrow = length(signaling), ncol = ncol(mat))
  idx <- match(rownames(mat1), signaling)
  mat[idx[!is.na(idx)], ] <- mat1
  dimnames(mat) <- list(signaling, colnames(mat1))
}
mat.ori <- mat
mat <- sweep(mat, 1L, apply(mat, 1, max), '/', check.margin = FALSE)

#####
###setup legend and colors###
#####
if (is.null(title)) {
  title <- paste0(legend.name, " signaling patterns")
} else {
  title <- paste0(paste0(legend.name, " signaling patterns"), " - ",title)
}
if (min(mat, na.rm = T) == max(mat, na.rm = T)) {
  legend.break <- max(mat, na.rm = T)
} else {
  legend.break <- c(round(min(mat, na.rm = T), digits = 1), round(max(mat, na.rm = T), digits = 1))
}

#heatmap color
color.heatmap.use = grDevices::colorRampPalette((RColorBrewer::brewer.pal(11,"Blues")))

```

```

color.heatmap.use[1] = '#ffffff' #set 0 point to white

#####
#####heatmap annotations#####
#####
###Rows###
#Left annotation
library(dplyr)
pathway_annotations = distinct(data.frame(pathway = cellchat@DB$interact
                                           annotation = cellchat@DB$interact
                                           pathway_annotations[pathway_annotations$pathway %in%
                                                                pathway_annotations[duplicated(pathway_annotations$pathway)
                                                                pathway_annotations = distinct(pathway_annotations)
pathway_annotations = pathway_annotations[match(row.names(mat), pathway_annot
rownames(pathway_annotations) <- NULL

pathway.col = pathway.col[pathway_annotations$annotation]

row_anno = HeatmapAnnotation("Signaling Type" = pathway_annotations$anno
                             col = list("Signaling Type" = pathway.col),
                             which = "row",
                             show_legend = TRUE, show_annotation_name =
                             simple_anno_size = grid::unit(0.1, "cm"))

#right annotation
pSum <- rowSums(mat.ori)
pSum.original <- pSum
pSum <- -1/log(pSum)
pSum[is.na(pSum)] <- 0
idx1 <- which(is.infinite(pSum) | pSum < 0)
if (length(idx1) > 0) {
  values.assign <- seq(max(pSum)*1.1, max(pSum)*1.5, length.out = length(idx1))
  position <- sort(pSum.original[idx1], index.return = TRUE)$ix
  pSum[idx1] <- values.assign[match(1:length(idx1), position)]
}
ha1 = rowAnnotation(Strength = anno_barplot(pSum, border = FALSE), show_

###columns###
if (is.null(color.use)) {
  color.use <- scPalette(length(colnames(mat)))
}
df<- data.frame(group = colnames(mat)); rownames(df) <- colnames(mat)
names(color.use) <- colnames(mat)

#bottom annotation
col_annotation <- HeatmapAnnotation(df = df, col = list(group = color.us
show_legend = FALSE, show_annotation
simple_anno_size = grid::unit(0.2, "

#top annotation
ha2 = HeatmapAnnotation(Strength = anno_barplot(colSums(mat.ori), border

#####
#####plotting heatmap#####
#####
ht1 = Heatmap(mat, col = color.heatmap.use, na_col = "white", name = "Re

```

```

right_annotation = ha1,
left_annotation = row_anno,
bottom_annotation = col_annotation, top_annotation = ha2,
#cluster_rows=hclust(dist(mat.noNA),method = 'average'),
#cluster_columns =hclust(dist(t(mat.noNA)),method = 'average')
row_km = row_clusters, row_km_repeats = 5,
column_km = column_clusters, column_km_repeats = 5,
border = TRUE,
row_names_side = "left",row_names_rot = 0,row_names_gp = gpar(
width = unit(width, "cm"), height = unit(height, "cm"),
column_title = title,column_title_gp = gpar(fontsize = font.si
heatmap_legend_param = list(title_gp = gpar(fontsize = 8, font
border = NA, at = legend.break,
legend_height = unit(20, "mm"),l
)
return(ht1)
}

```