

# Comparison analysis of multiple datasets using CellChat

Suoqin Jin

26 November, 2022

- Load the required libraries (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#load-the-required-libraries](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#load-the-required-libraries))
- Create a directory to save figures (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#create-a-directory-to-save-figures](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#create-a-directory-to-save-figures))
- Load CellChat object of each dataset and then merge together (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#load-cellchat-object-of-each-dataset-and-then-merge-together](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#load-cellchat-object-of-each-dataset-and-then-merge-together))
- Part I: Predict general principles of cell-cell communication (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#part-i-predict-general-principles-of-cell-cell-communication](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#part-i-predict-general-principles-of-cell-cell-communication))
  - Compare the total number of interactions and interaction strength (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#compare-the-total-number-of-interactions-and-interaction-strength](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#compare-the-total-number-of-interactions-and-interaction-strength))
  - Compare the number of interactions and interaction strength among different cell populations (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#compare-the-number-of-interactions-and-interaction-strength-among-different-cell-populations](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#compare-the-number-of-interactions-and-interaction-strength-among-different-cell-populations))
    - Differential number of interactions or interaction strength among different cell populations (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#differential-number-of-interactions-or-interaction-strength-among-different-cell-populations](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#differential-number-of-interactions-or-interaction-strength-among-different-cell-populations))
    - Differential number of interactions or interaction strength among different cell types (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#differential-number-of-interactions-or-interaction-strength-among-different-cell-types](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#differential-number-of-interactions-or-interaction-strength-among-different-cell-types))
  - Compare the major sources and targets in 2D space (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#compare-the-major-sources-and-targets-in-2d-space](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#compare-the-major-sources-and-targets-in-2d-space))
- Part II: Identify the conserved and context-specific signaling pathways (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#part-ii-identify-the-conserved-and-context-specific-signaling-pathways](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#part-ii-identify-the-conserved-and-context-specific-signaling-pathways))
  - Identify signaling networks with larger (or less) difference as well as signaling groups based on their functional/structure similarity (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#identify-signaling-networks-with-larger-or-less-difference-as-well-as-signaling-groups-based-on-their-functionalstructure-similarity](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#identify-signaling-networks-with-larger-or-less-difference-as-well-as-signaling-groups-based-on-their-functionalstructure-similarity))
    - Identify signaling groups based on their functional similarity (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#identify-signaling-groups-based-on-their-functional-similarity](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#identify-signaling-groups-based-on-their-functional-similarity))
    - Identify signaling groups based on structure similarity (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#identify-signaling-groups-based-on-structure-similarity](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#identify-signaling-groups-based-on-structure-similarity))
    - Compute and visualize the pathway distance in the learned joint manifold (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#compute-and-visualize-the-pathway-distance-in-the-learned-joint-manifold](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#compute-and-visualize-the-pathway-distance-in-the-learned-joint-manifold))
  - Identify and visualize the conserved and context-specific signaling pathways (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#identify-and-visualize-the-conserved-and-context-specific-signaling-pathways](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#identify-and-visualize-the-conserved-and-context-specific-signaling-pathways))
    - Compare the overall information flow of each signaling pathway (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#compare-the-overall-information-flow-of-each-signaling-pathway](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#compare-the-overall-information-flow-of-each-signaling-pathway))
    - Compare outgoing (or incoming) signaling associated with each cell population (//htmlpreview.github.io/?  
[https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#compare-outgoing-or-incoming-signaling-associated-with-each-cell-population](https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#compare-outgoing-or-incoming-signaling-associated-with-each-cell-population))

- Part III: Identify the upregulated and down-regulated signaling ligand-receptor pairs ([https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#part-iii-identify-the-upregulated-and-down-regulated-signaling-ligand-receptor-pairs](https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#part-iii-identify-the-upregulated-and-down-regulated-signaling-ligand-receptor-pairs))
  - Identify dysfunctional signaling by comparing the communication probabilities ([https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#identify-dysfunctional-signaling-by-comparing-the-communication-probabilities](https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#identify-dysfunctional-signaling-by-comparing-the-communication-probabilities))
  - Identify dysfunctional signaling by using differential expression analysis ([https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#identify-dysfunctional-signaling-by-using-differential-expression-analysis](https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#identify-dysfunctional-signaling-by-using-differential-expression-analysis))
- Part IV: Visually compare cell-cell communication using Hierarchy plot, Circle plot or Chord diagram ([https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#part-iv-visually-compare-cell-cell-communication-using-hierarchy-plot-circle-plot-or-chord-diagram](https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#part-iv-visually-compare-cell-cell-communication-using-hierarchy-plot-circle-plot-or-chord-diagram))
- Part V: Compare the signaling gene expression distribution between different datasets ([https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#part-v-compare-the-signaling-gene-expression-distribution-between-different-datasets](https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#part-v-compare-the-signaling-gene-expression-distribution-between-different-datasets))
- Save the merged CellChat object ([https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison\\_analysis\\_of\\_multiple\\_datasets.html#save-the-merged-cellchat-object](https://htmlpreview.github.io/?https://github.com/sqjin/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html#save-the-merged-cellchat-object))

This vignette shows how to apply CellChat to identify major signaling changes as well as conserved and context-specific signaling by joint manifold learning and quantitative contrasts of multiple cell-cell communication networks. We showcase CellChat's diverse functionalities by applying it to a scRNA-seq data on cells from two biological conditions: nonlesional (NL, normal) and lesional (LS, diseased) human skin from patients with atopic dermatitis. **These two datasets (conditions) have the same cell population compositions after joint clustering. If there are slightly or vastly different cell population compositions between different datasets, please check out another related tutorial.**

CellChat employs a top-down approach, i.e., starting with the big picture and then refining it in a greater detail on the signaling mechanisms, to identify signaling changes at different levels, including both general principles of cell-cell communication and dysfunctional cell populations/signaling pathways/ligand-receptors.

## Load the required libraries

```
library(CellChat)
library(patchwork)
```

## Create a directory to save figures

```
data.dir <- './comparison'
dir.create(data.dir)
setwd(data.dir)
```

## Load CellChat object of each dataset and then merge together

USERS need to run CellChat on each dataset separately and then merge different CellChat objects together. Please do `updateCellChat` if you have CellChat objects that are obtained using the earlier version (< 1.6.0).

```
# cellchat.NL <- readRDS(url("https://ndownloader.figshare.com/files/25954199"))
# cellchat.LS <- readRDS(url("https://ndownloader.figshare.com/files/25956518"))
cellchat.NL <- readRDS("/Users/jinsuoqin/Documents/CellChat/tutorial/cellchat_humanSkin_NL.rds")
cellchat.LS <- readRDS("/Users/jinsuoqin/Documents/CellChat/tutorial/cellchat_humanSkin_LS.rds")
cellchat.NL <- updateCellChat(cellchat.NL)
cellchat.LS <- updateCellChat(cellchat.LS)
object.list <- list(NL = cellchat.NL, LS = cellchat.LS)
cellchat <- mergeCellChat(object.list, add.names = names(object.list))
#> Merge the following slots: 'data.signalizing','images','net', 'netP','meta', 'idents', 'var.features' , 'DB', and 'LR'.
cellchat
#> An object of class CellChat created from a merged object with multiple datasets
#> 555 signaling genes.
#> 7563 cells.
#> CellChat analysis of single cell RNA-seq data!
```

# Part I: Predict general principles of cell-cell communication

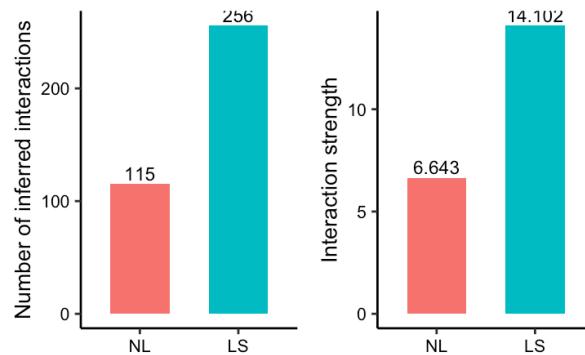
CellChat starts with the big picture to predict general principles of cell-cell communication. When comparing cell-cell communication among multiple biological conditions, it can answer the following biological questions:

- Whether the cell-cell communication is enhanced or not
- The interaction between which cell types is significantly changed
- How the major sources and targets change from one condition to another

## Compare the total number of interactions and interaction strength

To answer on question on whether the cell-cell communication is enhanced or not, CellChat compares the the total number of interactions and interaction strength of the inferred cell-cell communication networks from different biological conditions.

```
gg1 <- compareInteractions(cellchat, show.legend = F, group = c(1,2))
gg2 <- compareInteractions(cellchat, show.legend = F, group = c(1,2), measure = "weight")
gg1 + gg2
```



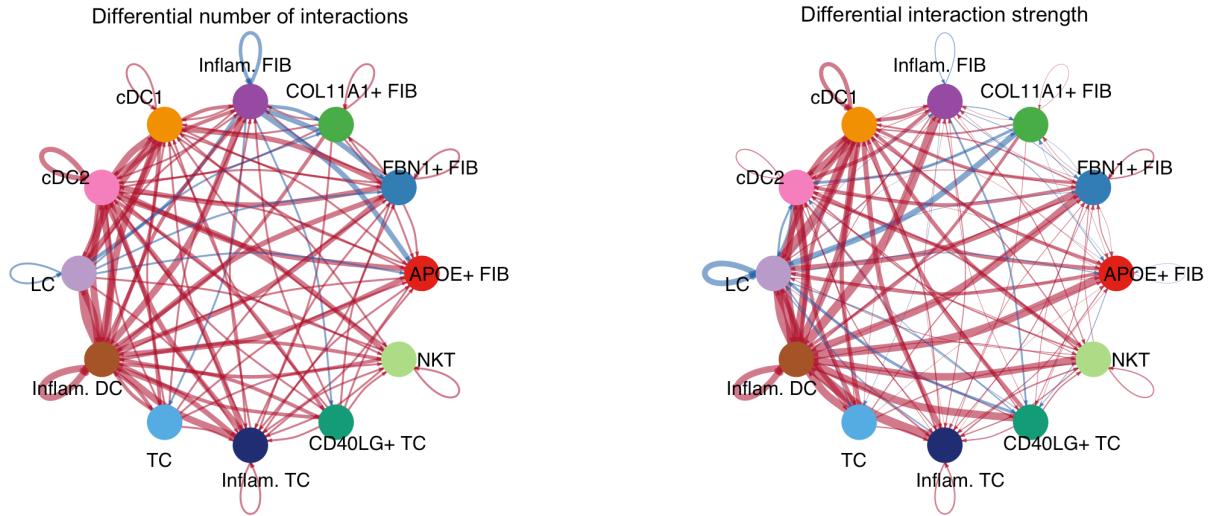
## Compare the number of interactions and interaction strength among different cell populations

To identify the interaction between which cell populations showing significant changes, CellChat compares the number of interactions and interaction strength among different cell populations.

### Differential number of interactions or interaction strength among different cell populations

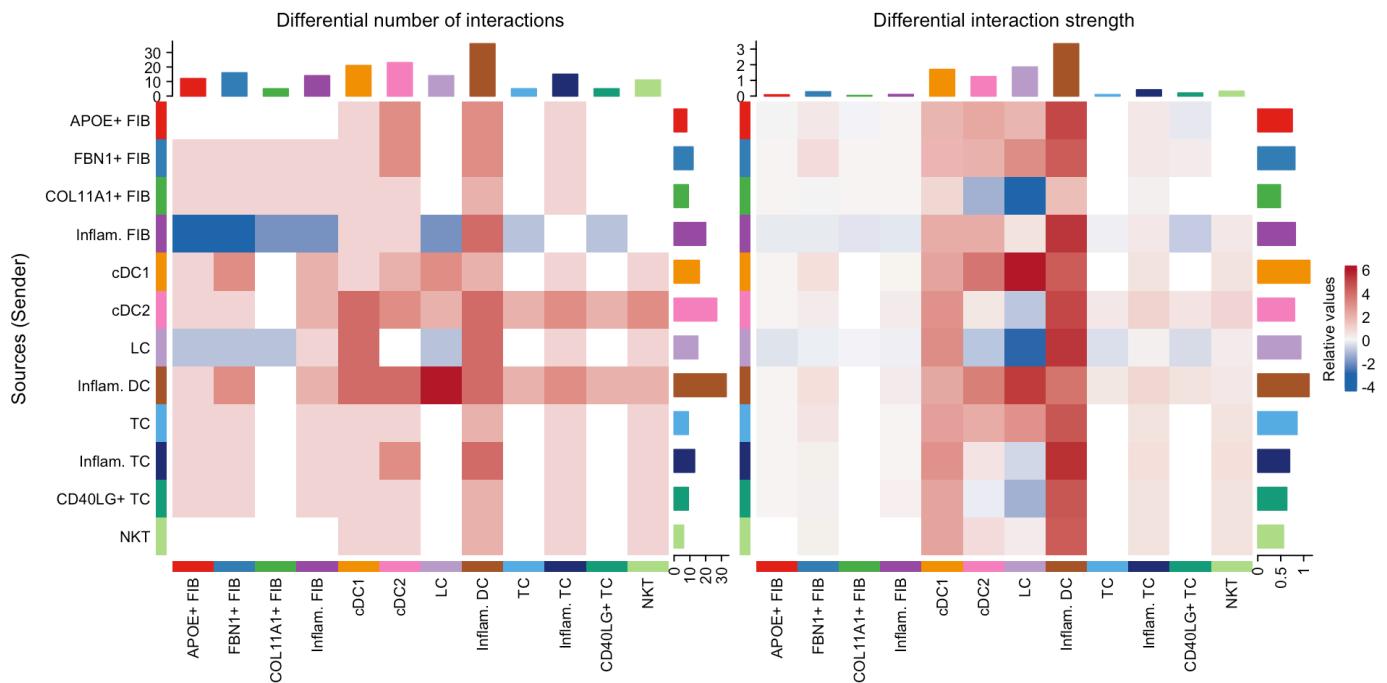
The differential number of interactions or interaction strength in the cell-cell communication network between two datasets can be visualized using circle plot, where red (or blue) colored edges represent increased (or decreased) signaling in the second dataset compared to the first one.

```
par(mfrow = c(1,2), xpd=TRUE)
netVisual_diffInteraction(cellchat, weight.scale = T)
netVisual_diffInteraction(cellchat, weight.scale = T, measure = "weight")
```



We can also show differential number of interactions or interaction strength in a greater details using a heatmap. The top colored bar plot represents the sum of column of values displayed in the heatmap (incoming signaling). The right colored bar plot represents the sum of row of values (outgoing signaling). In the colorbar, red (or blue) represents increased (or decreased) signaling in the second dataset compared to the first one.

```
gg1 <- netVisual_heatmap(cellchat)
#> Do heatmap based on a merged object
gg2 <- netVisual_heatmap(cellchat, measure = "weight")
#> Do heatmap based on a merged object
gg1 + gg2
```



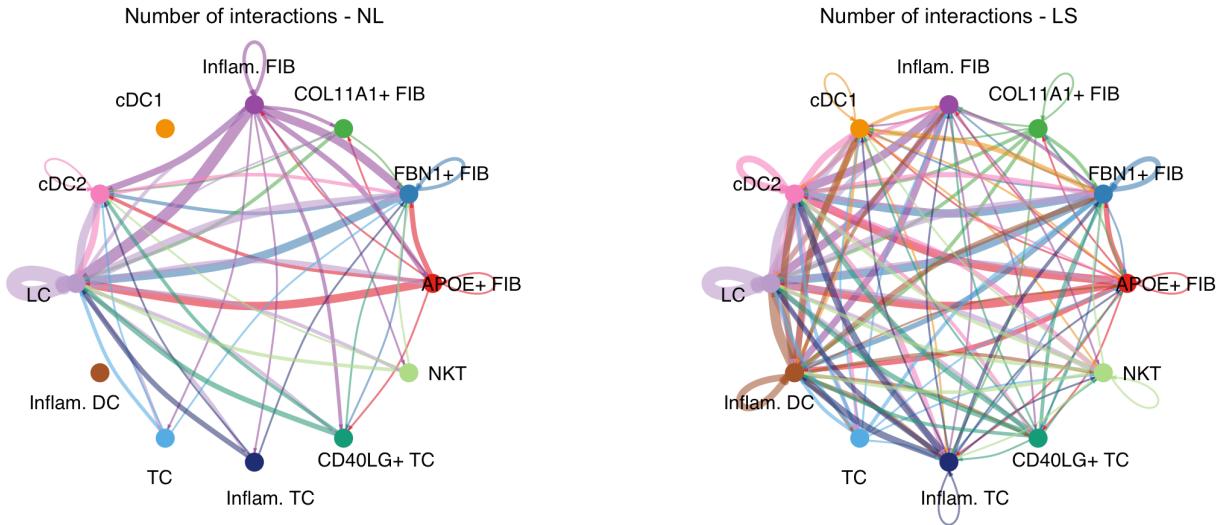
The differential network analysis only works for pairwise datasets. If there are more datasets for comparison, we can directly show the number of interactions or interaction strength between any two cell populations in each dataset.

To better control the node size and edge weights of the inferred networks across different datasets, we compute the maximum number of cells per cell group and the maximum number of interactions (or interaction weights) across all datasets.

```

weight.max <- getMaxWeight(object.list, attribute = c("idents", "count"))
par(mfrow = c(1,2), xpd=TRUE)
for (i in 1:length(object.list)) {
  netVisual_circle(object.list[[i]]@net$count, weight.scale = T, label.edge= F, edge.weight.max = weight.max[2], edg
e.width.max = 12, title.name = paste0("Number of interactions - ", names(object.list)[i]))
}

```



## Differential number of interactions or interaction strength among different cell types

To simplify the complicated network and gain insights into the cell-cell communication at the cell type level, we can aggregate the cell-cell communication based on the defined cell groups. Here we categorize the cell populations into three cell types, and then re-merge the list of CellChat object.

```

group.cellType <- c(rep("FIB", 4), rep("DC", 4), rep("TC", 4))
group.cellType <- factor(group.cellType, levels = c("FIB", "DC", "TC"))
object.list <- lapply(object.list, function(x) {mergeInteractions(x, group.cellType)})
cellchat <- mergeCellChat(object.list, add.names = names(object.list))
#> Merge the following slots: 'data.signalizing','images','net', 'netP','meta', 'idents', 'var.features' , 'DB', and
'LR'.

```

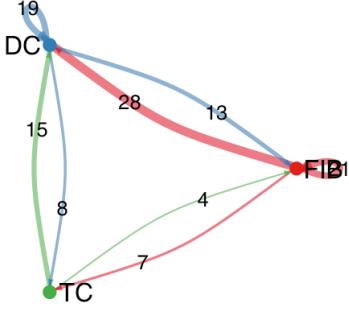
We then can show the number of interactions or interaction strength between any two cell types in each dataset.

```

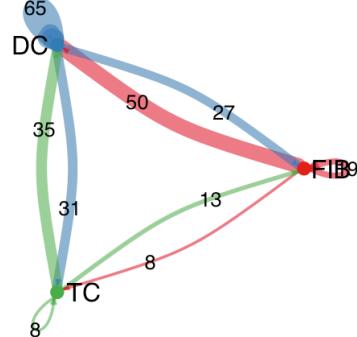
weight.max <- getMaxWeight(object.list, slot.name = c("idents", "net", "net"), attribute = c("idents","count", "coun
t.merged"))
par(mfrow = c(1,2), xpd=TRUE)
for (i in 1:length(object.list)) {
  netVisual_circle(object.list[[i]]@net$count.merged, weight.scale = T, label.edge= T, edge.weight.max = weight.max
[3], edge.width.max = 12, title.name = paste0("Number of interactions - ", names(object.list)[i]))
}

```

Number of interactions - NL



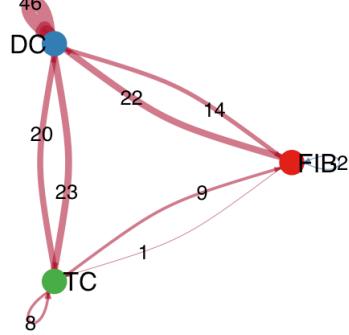
Number of interactions - LS



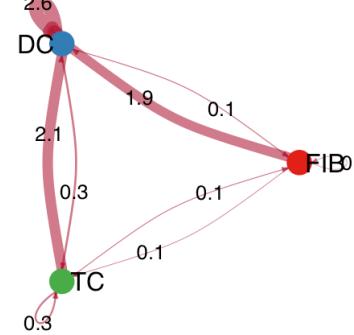
Similarly, we can also show the differential number of interactions or interaction strength between any two cell types using circle plot. Red (or blue) colored edges represent increased (or decreased) signaling in the second dataset compared to the first one.

```
par(mfrow = c(1,2), xpd=TRUE)
netVisual_diffInteraction(cellchat, weight.scale = T, measure = "count.merged", label.edge = T)
netVisual_diffInteraction(cellchat, weight.scale = T, measure = "weight.merged", label.edge = T)
```

Differential number of interactions



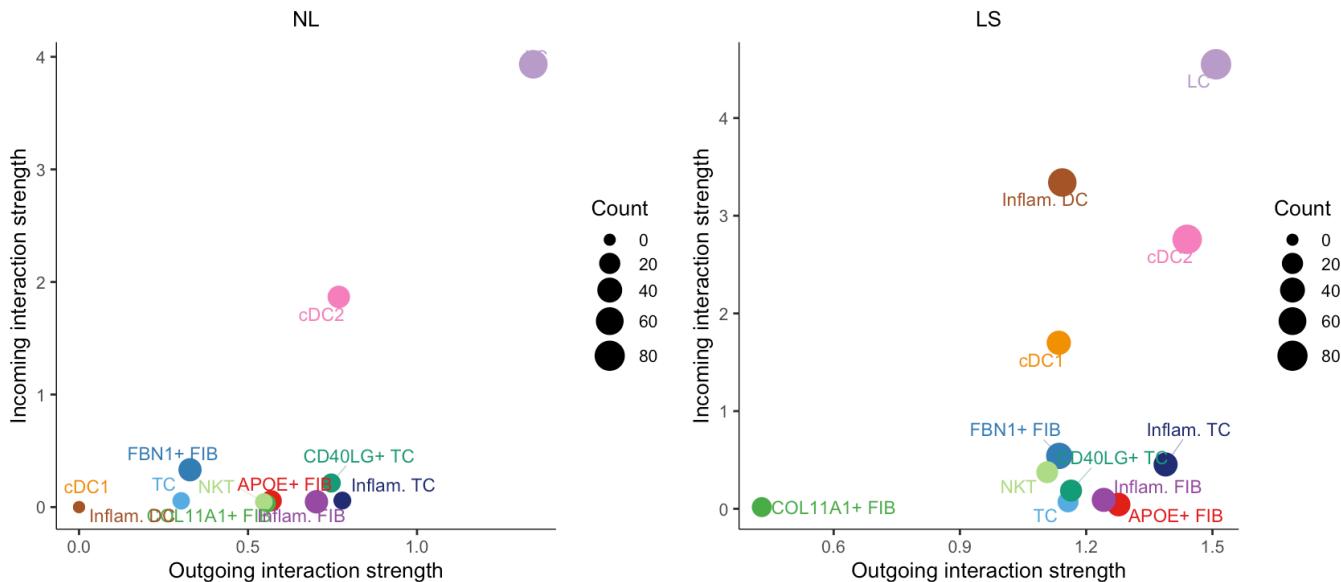
Differential interaction strength



## Compare the major sources and targets in 2D space

Comparing the outgoing and incoming interaction strength in 2D space allows ready identification of the cell populations with significant changes in sending or receiving signals between different datasets.

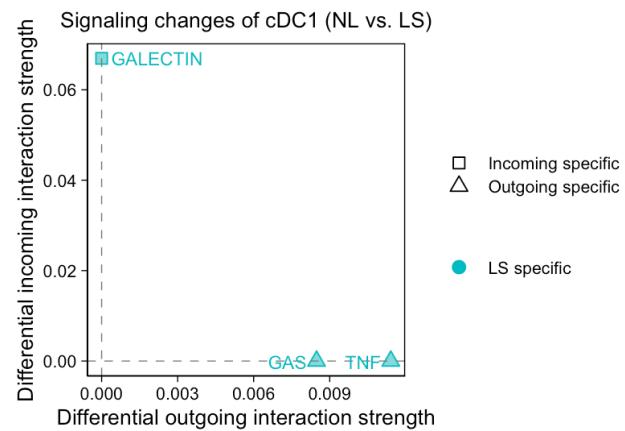
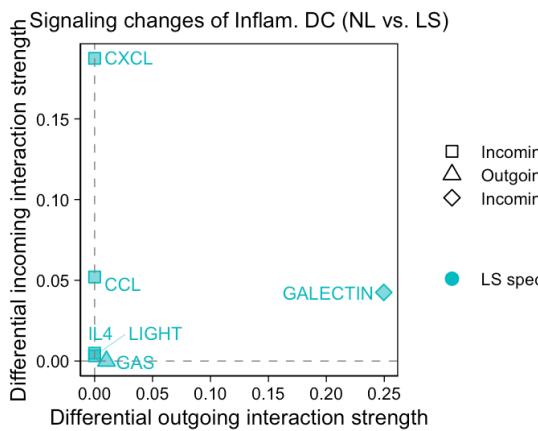
```
num.link <- sapply(object.list, function(x) {rowSums(x@net$count) + colSums(x@net$count)-diag(x@net$count)})
weight.MinMax <- c(min(num.link), max(num.link)) # control the dot size in the different datasets
gg <- list()
for (i in 1:length(object.list)) {
  gg[[i]] <- netAnalysis_signalingRole_scatter(object.list[[i]], title = names(object.list)[i], weight.MinMax = weight.MinMax)
}
## Signaling role analysis on the aggregated cell-cell communication network from all signaling pathways
## Signaling role analysis on the aggregated cell-cell communication network from all signaling pathways
patchwork::wrap_plots(plots = gg)
```



From the scatter plot, we can see that Inflam.DC and cDC1 emerge as one of the major source and targets in LS compared to NL. Fibroblast populations also become the major sources in LS.

Furthermore, we can identify the specific signaling changes of Inflam.DC and cDC1 between NL and LS. ## Identify signaling changes associated with one cell group

```
gg1 <- netAnalysis_signalingChanges_scatter(cellchat, idents.use = "Inflam. DC", signaling.exclude = "MIF")
## Visualizing differential outgoing and incoming signaling changes from NL to LS
## The following `from` values were not present in `x`: 0
## The following `from` values were not present in `x`: 0, -1
gg2 <- netAnalysis_signalingChanges_scatter(cellchat, idents.use = "cDC1", signaling.exclude = c("MIF"))
## Visualizing differential outgoing and incoming signaling changes from NL to LS
## The following `from` values were not present in `x`: 0, 2
## The following `from` values were not present in `x`: 0, -1
patchwork::wrap_plots(plots = list(gg1, gg2))
```



## Part II: Identify the conserved and context-specific signaling pathways

CellChat then can identify signaling networks with larger (or less) difference, signaling groups, and the conserved and context-specific signaling pathways based on their cell-cell communication networks among multiple biological conditions.

**Identify signaling networks with larger (or less) difference as well as signaling groups based on their functional/structure similarity**

CellChat performs joint manifold learning and classification of the inferred communication networks based on their functional and topological similarity. NB: Such analysis is applicable to more than two datasets.

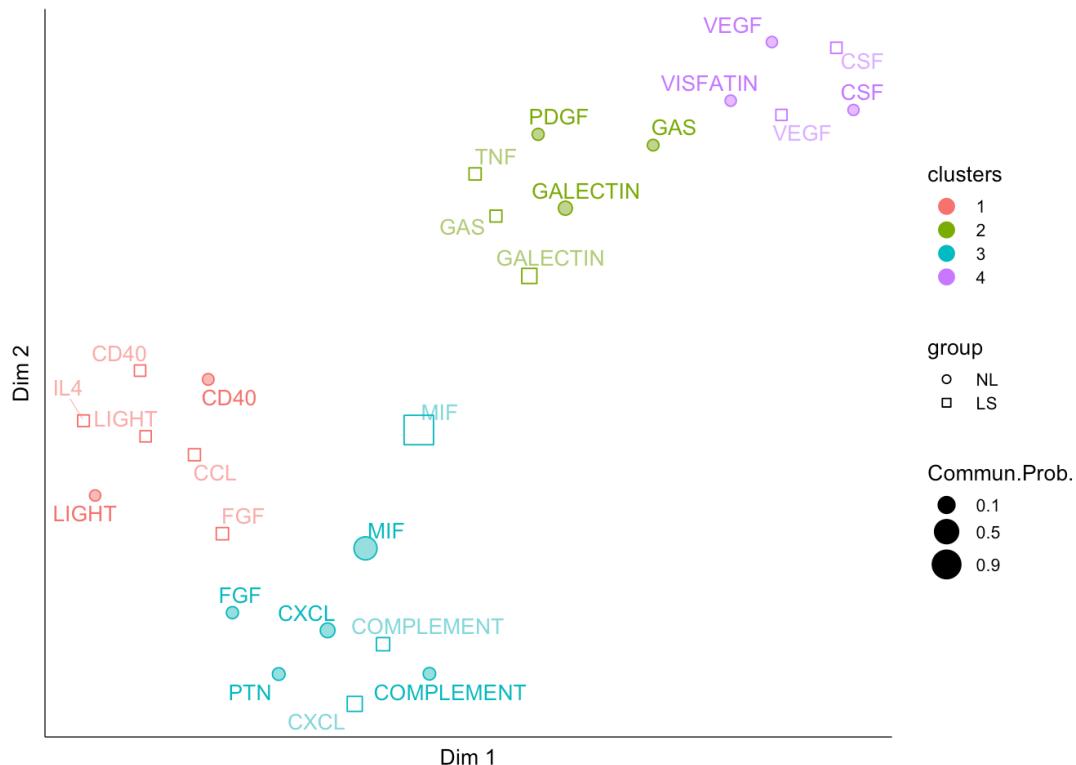
**Functional similarity:** High degree of functional similarity indicates major senders and receivers are similar, and it can be interpreted as the two signaling pathways or two ligand-receptor pairs exhibit similar and/or redundant roles. **NB:** Functional similarity analysis is not applicable to multiple datasets with different cell type composition.

**Structural similarity:** A structural similarity was used to compare their signaling network structure, without considering the similarity of senders and receivers. **NB:** Structural similarity analysis is applicable to multiple datasets with the same cell type composition or the vastly different cell type composition.

Here we can run the manifold and classification learning analysis based on the functional similarity because the two datasets have the same cell type composition.

## Identify signaling groups based on their functional similarity

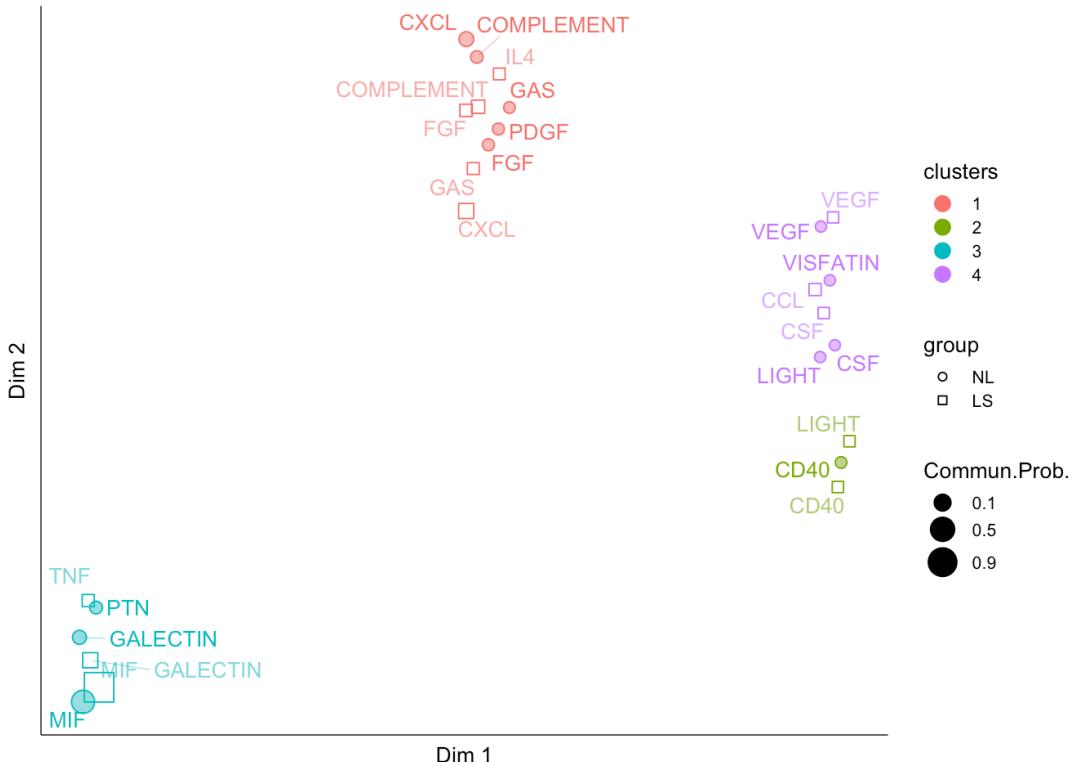
```
cellchat <- computeNetSimilarityPairwise(cellchat, type = "functional")
#> Compute signaling network similarity for datasets 1 2
cellchat <- netEmbedding(cellchat, type = "functional")
#> Manifold learning of the signaling networks for datasets 1 2
cellchat <- netClustering(cellchat, type = "functional")
#> Classification learning of the signaling networks for datasets 1 2
# Visualization in 2D-space
netVisual_embeddingPairwise(cellchat, type = "functional", label.size = 3.5)
#> 2D visualization of signaling networks from datasets 1 2
```



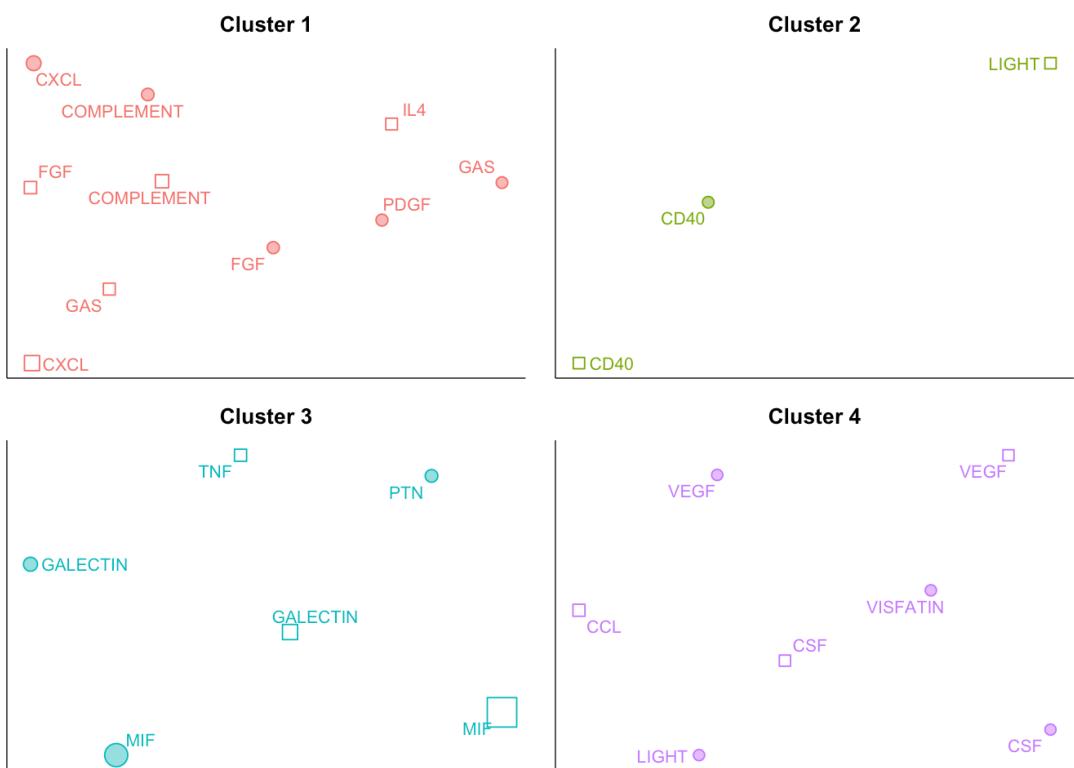
```
# netVisual_embeddingZoomIn(cellchat, type = "functional", nCol = 2)
```

## Identify signaling groups based on structure similarity

```
cellchat <- computeNetSimilarityPairwise(cellchat, type = "structural")
#> Compute signaling network similarity for datasets 1 2
cellchat <- netEmbedding(cellchat, type = "structural")
#> Manifold learning of the signaling networks for datasets 1 2
cellchat <- netClustering(cellchat, type = "structural")
#> Classification learning of the signaling networks for datasets 1 2
# Visualization in 2D-space
netVisual_embeddingPairwise(cellchat, type = "structural", label.size = 3.5)
#> 2D visualization of signaling networks from datasets 1 2
```



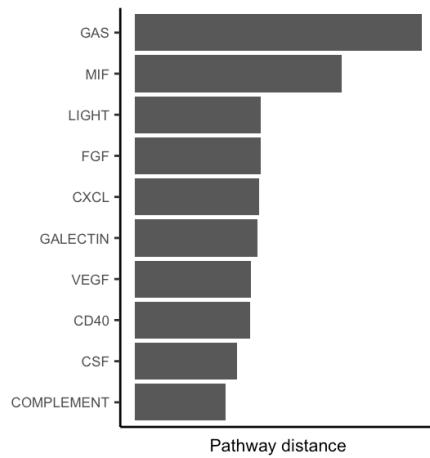
```
netVisual_embeddingPairwiseZoomIn(cellchat, type = "structural", nCol = 2)
#> 2D visualization of signaling networks from datasets 1 2
```



## Compute and visualize the pathway distance in the learned joint manifold

We can identify the signaling networks with larger (or less) difference based on their Euclidean distance in the shared two-dimensions space. Larger distance implies larger difference of the communication networks between two datasets in terms of either functional or structure similarity. **NB:** We only compute the distance of overlapped signaling pathways between two datasets. Those signaling pathways that are only identified in one dataset are not considered here. If there are more than three datasets, one can do pairwise comparisons by defining comparison in the function rankSimilarity .

```
rankSimilarity(cellchat, type = "functional")
#> Compute the distance of signaling networks between datasets 1 2
```



## Identify and visualize the conserved and context-specific signaling pathways

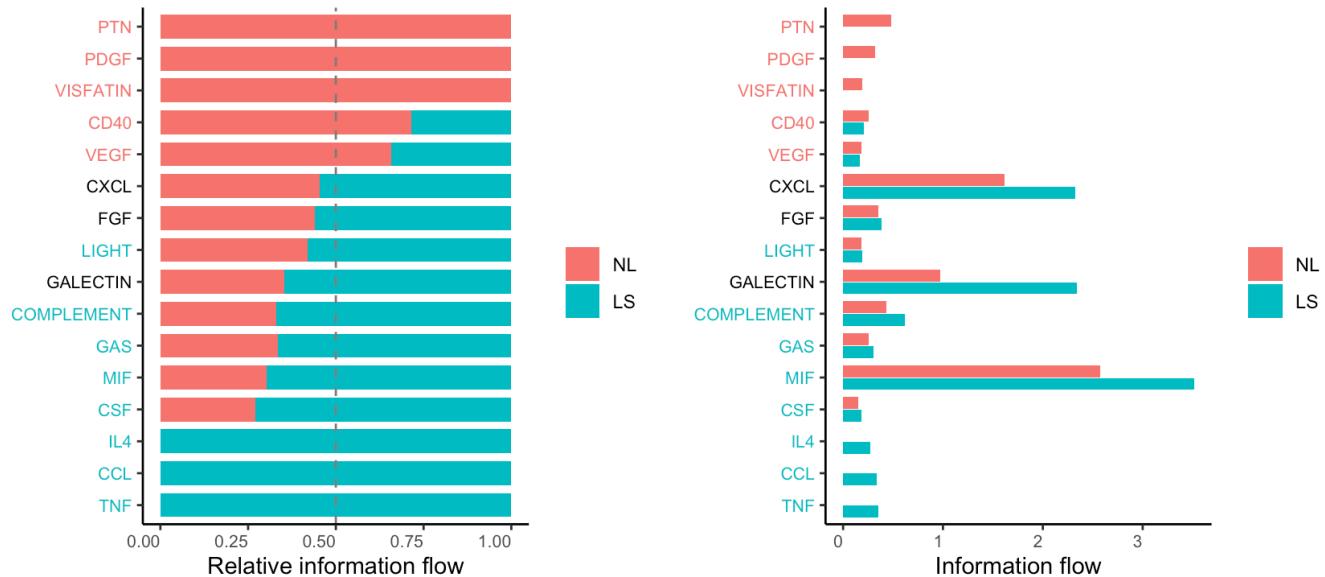
By comparing the information flow/interaction strength of each signaling pathway, we can identify signaling pathways, (i) turn off, (ii) decrease, (iii) turn on or (iv) increase, by change their information flow at one condition as compared to another condition.

### Compare the overall information flow of each signaling pathway

We can identify the conserved and context-specific signaling pathways by simply comparing the information flow for each signaling pathway, which is defined by the sum of communication probability among all pairs of cell groups in the inferred network (i.e., the total weights in the network).

This bar graph can be plotted in a stacked mode or not. Significant signaling pathways were ranked based on differences in the overall information flow within the inferred networks between NL and LS skin. The top signaling pathways colored red are enriched in NL skin, and these colored green were enriched in the LS skin.

```
gg1 <- rankNet(cellchat, mode = "comparison", stacked = T, do.stat = TRUE)
gg2 <- rankNet(cellchat, mode = "comparison", stacked = F, do.stat = TRUE)
gg1 + gg2
```



### Compare outgoing (or incoming) signaling associated with each cell population

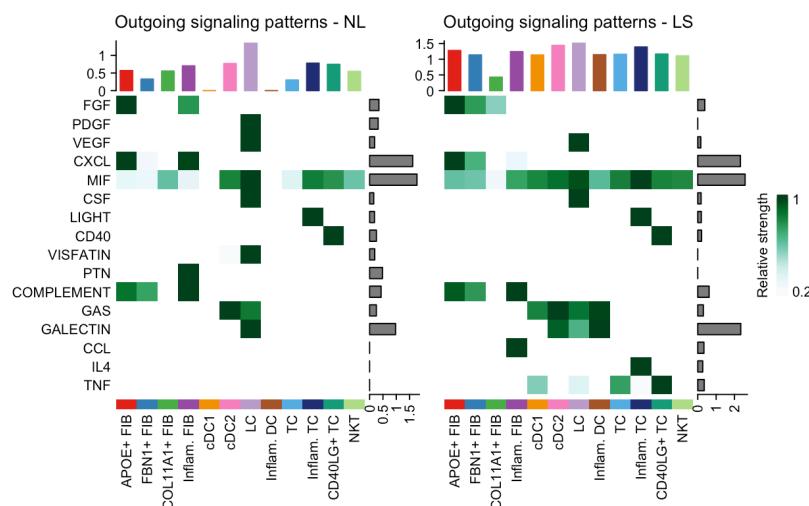
The above analysis summarize the information from the outgoing and incoming signaling together. We can also compare the outgoing (or incoming) signaling pattern between two datasets, allowing to identify signaling pathways/ligand-receptors that exhibit different signaling patterns.

We can combine all the identified signaling pathways from different datasets and thus compare them side by side, including outgoing signaling, incoming signaling and overall signaling by aggregating outgoing and incoming signaling together. NB: rankNet also shows the comparison of overall signaling, but it does not show the signaling strength in specific cell populations.

```

library(ComplexHeatmap)
#> Loading required package: grid
#> =====
#> ComplexHeatmap version 2.10.0
#> Bioconductor page: http://bioconductor.org/packages/ComplexHeatmap/
#> Github page: https://github.com/jokergoo/ComplexHeatmap
#> Documentation: http://jokergoo.github.io/ComplexHeatmap-reference
#>
#> If you use it in published research, please cite:
#> Gu, Z. Complex heatmaps reveal patterns and correlations in multidimensional
#> genomic data. Bioinformatics 2016.
#>
#> The new InteractiveComplexHeatmap package can directly export static
#> complex heatmaps into an interactive Shiny app with zero effort. Have a try!
#>
#> This message can be suppressed by:
#> suppressPackageStartupMessages(library(ComplexHeatmap))
#> =====
i = 1
# combining all the identified signaling pathways from different datasets
pathway.union <- union(object.list[[i]]@netP$pathways, object.list[[i+1]]@netP$pathways)
ht1 = netAnalysis_signalingRole_heatmap(object.list[[i]], pattern = "outgoing", signaling = pathway.union, title = names(object.list)[i], width = 5, height = 6)
ht2 = netAnalysis_signalingRole_heatmap(object.list[[i+1]], pattern = "outgoing", signaling = pathway.union, title = names(object.list)[i+1], width = 5, height = 6)
draw(ht1 + ht2, ht_gap = unit(0.5, "cm"))

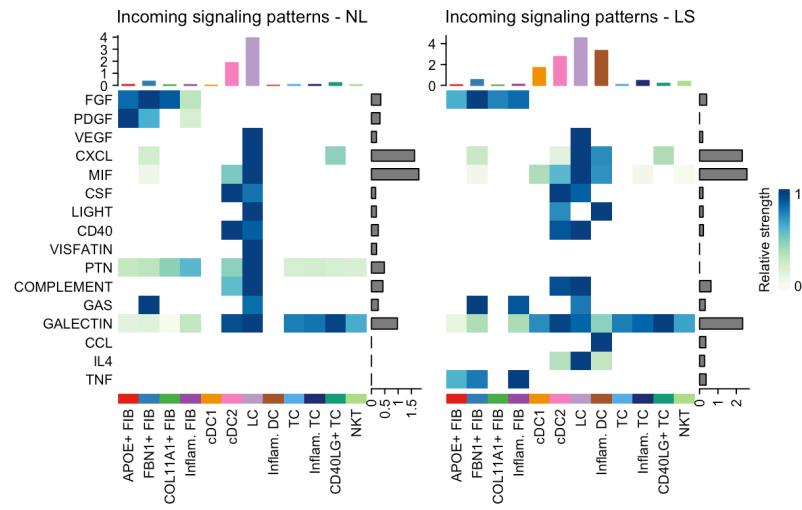
```



```

ht1 = netAnalysis_signalingRole_heatmap(object.list[[i]], pattern = "incoming", signaling = pathway.union, title = names(object.list)[i], width = 5, height = 6, color.heatmap = "GnBu")
ht2 = netAnalysis_signalingRole_heatmap(object.list[[i+1]], pattern = "incoming", signaling = pathway.union, title = names(object.list)[i+1], width = 5, height = 6, color.heatmap = "GnBu")
draw(ht1 + ht2, ht_gap = unit(0.5, "cm"))

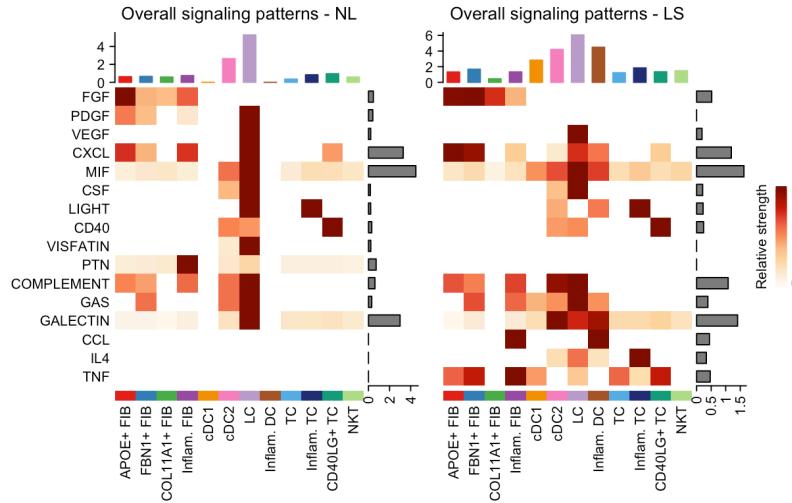
```



```

ht1 = netAnalysis_signalingRole_heatmap(object.list[[i]], pattern = "all", signaling = pathway.union, title = names(object.list)[i], width = 5, height = 6, color.heatmap = "OrRd")
ht2 = netAnalysis_signalingRole_heatmap(object.list[[i+1]], pattern = "all", signaling = pathway.union, title = names(object.list)[i+1], width = 5, height = 6, color.heatmap = "OrRd")
draw(ht1 + ht2, ht_gap = unit(0.5, "cm"))

```

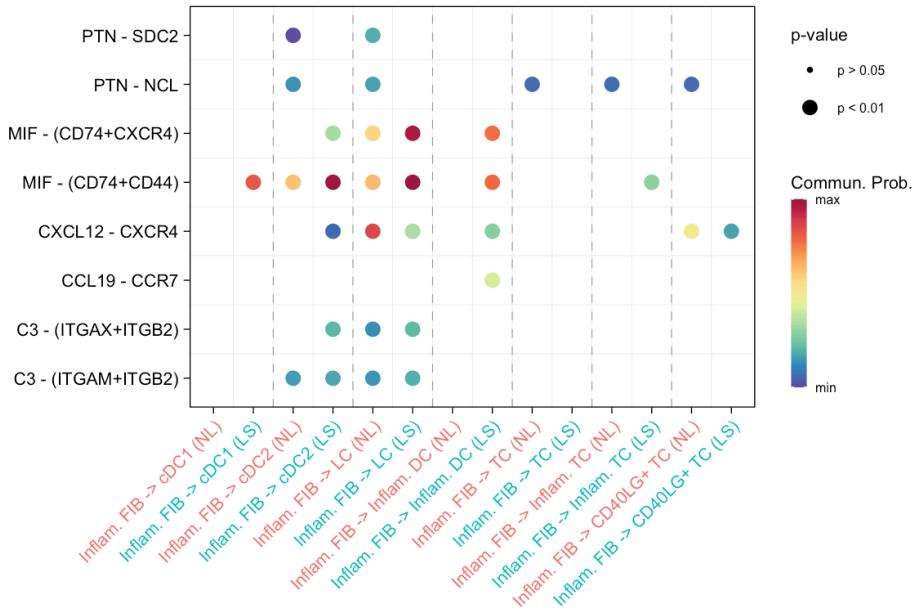


## Part III: Identify the upregulated and down-regulated signaling ligand-receptor pairs

**Identify dysfunctional signaling by comparing the communication probabilities**

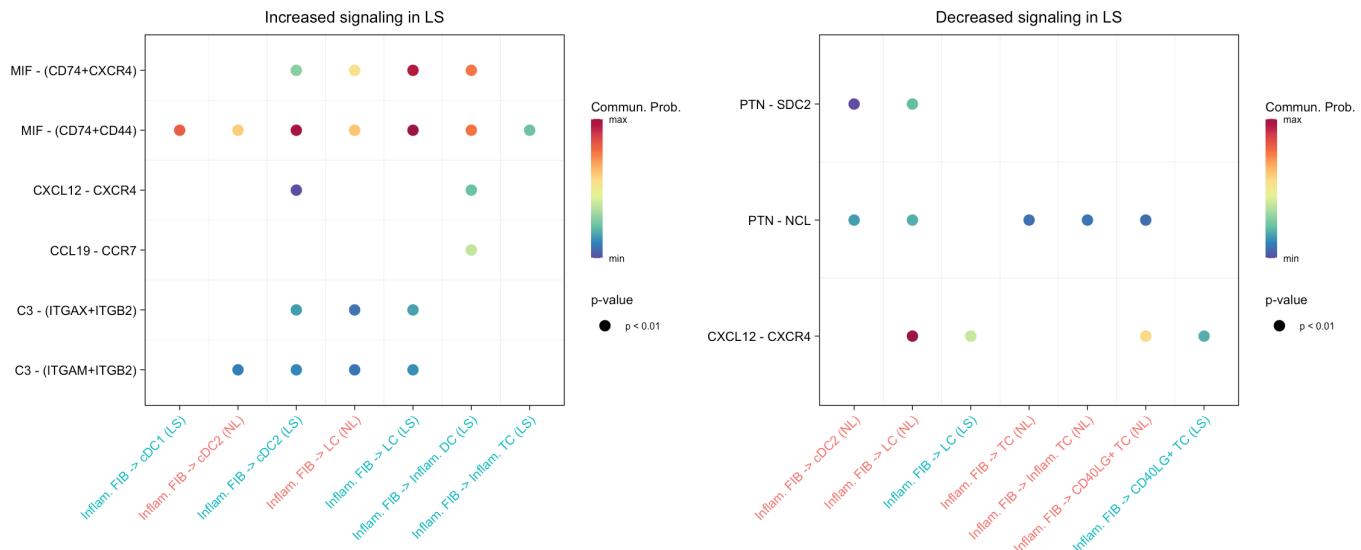
We can compare the communication probabilities mediated by ligand-receptor pairs from some cell groups to other cell groups. This can be done by setting `comparison` in the function `netVisual_bubble`.

```
netVisual_bubble(cellchat, sources.use = 4, targets.use = c(5:11), comparison = c(1, 2), angle.x = 45)
#> Comparing communications on a merged object
```



Moreover, we can identify the upregulated (increased) and down-regulated (decreased) signaling ligand-receptor pairs in one dataset compared to the other dataset. This can be done by specifying `max.dataset` and `min.dataset` in the function `netVisual_bubble`. The increased signaling means these signaling have higher communication probability (strength) in one dataset compared to the other dataset.

```
gg1 <- netVisual_bubble(cellchat, sources.use = 4, targets.use = c(5:11), comparison = c(1, 2), max.dataset = 2, title.name = "Increased signaling in LS", angle.x = 45, remove.isolate = T)
#> Comparing communications on a merged object
gg2 <- netVisual_bubble(cellchat, sources.use = 4, targets.use = c(5:11), comparison = c(1, 2), max.dataset = 1, title.name = "Decreased signaling in LS", angle.x = 45, remove.isolate = T)
#> Comparing communications on a merged object
gg1 + gg2
```



NB: The ligand-receptor pairs shown in the bubble plot can be accessed via `signaling.LSIIncreased = gg1$data`.

## Identify dysfunctional signaling by using differential expression analysis

The above method for identifying the upregulated and down-regulated signaling is performed by comparing the communication probability between two datasets for each L-R pair and each pair of cell groups. Alternatively, we can identify the upregulated and down-regulated signaling ligand-receptor pairs based on the differential gene expression analysis. Specifically, we perform differential expression analysis between two biological conditions (i.e., NL and LS) for each cell group, and then obtain the upregulated and down-regulated signaling based on the fold change of ligands in the sender cells and receptors in the receiver cells. Such analysis can be done as follows.

```

# define a positive dataset, i.e., the dataset with positive fold change against the other dataset
pos.dataset = "LS"

# define a char name used for storing the results of differential expression analysis
features.name = pos.dataset

# perform differential expression analysis
cellchat <- identifyOverExpressedGenes(cellchat, group.dataset = "datasets", pos.dataset = pos.dataset, features.name = features.name, only.pos = FALSE, thresh.pc = 0.1, thresh.fc = 0.1, thresh.p = 1)
#> Use the joint cell labels from the merged CellChat object

# map the results of differential expression analysis onto the inferred cell-cell communications to easily manage/subset the ligand-receptor pairs of interest
net <- netMappingDEG(cellchat, features.name = features.name)

# extract the ligand-receptor pairs with upregulated ligands in LS
net.up <- subsetCommunication(cellchat, net = net, datasets = "LS", ligand.logFC = 0.2, receptor.logFC = NULL)
# extract the ligand-receptor pairs with upregulated ligands and upregulated receptors in NL, i.e., downregulated in LS
net.down <- subsetCommunication(cellchat, net = net, datasets = "NL", ligand.logFC = -0.1, receptor.logFC = -0.1)

```

Since the signaling genes in the `net.up` and `net.down` might be complex with multi-subunits, we can do further deconvolution to obtain the individual signaling genes.

```

gene.up <- extractGeneSubsetFromPair(net.up, cellchat)
gene.down <- extractGeneSubsetFromPair(net.down, cellchat)

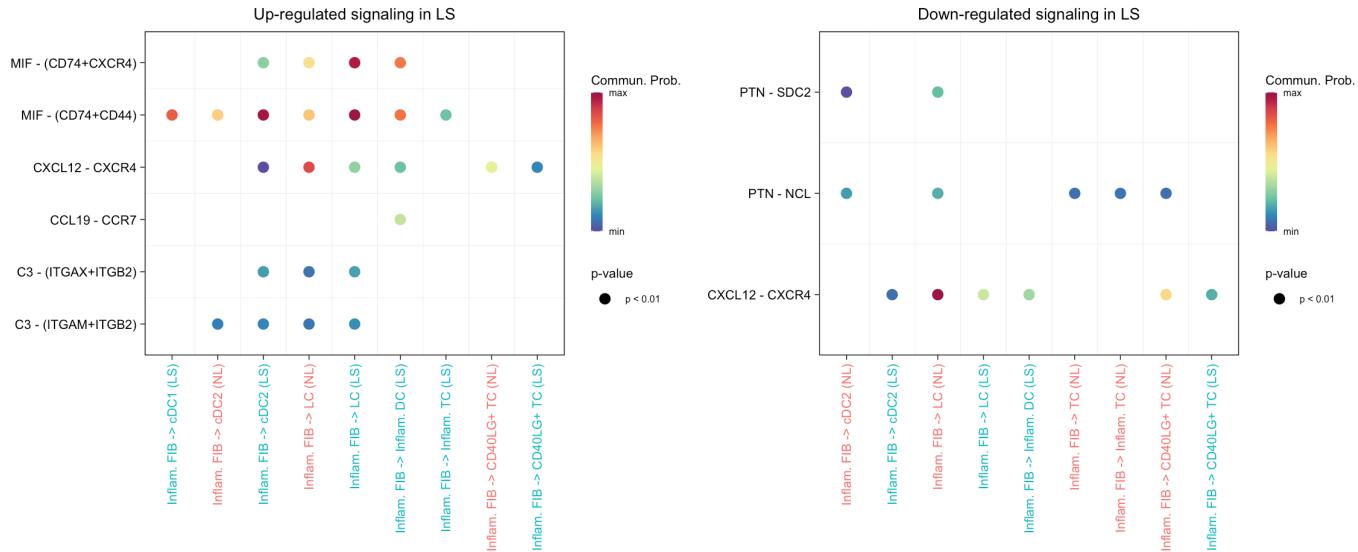
```

We then visualize the upregulated and down-regulated signaling ligand-receptor pairs using bubble plot or chord diagram.

```

pairLR.use.up = net.up[, "interaction_name", drop = F]
gg1 <- netVisual_bubble(cellchat, pairLR.use = pairLR.use.up, sources.use = 4, targets.use = c(5:11), comparison = c(1, 2), angle.x = 90, remove.isolate = T, title.name = paste0("Up-regulated signaling in ", names(object.list)[2]))
#> Comparing communications on a merged object
pairLR.use.down = net.down[, "interaction_name", drop = F]
gg2 <- netVisual_bubble(cellchat, pairLR.use = pairLR.use.down, sources.use = 4, targets.use = c(5:11), comparison = c(1, 2), angle.x = 90, remove.isolate = T, title.name = paste0("Down-regulated signaling in ", names(object.list)[2]))
#> Comparing communications on a merged object
gg1 + gg2

```



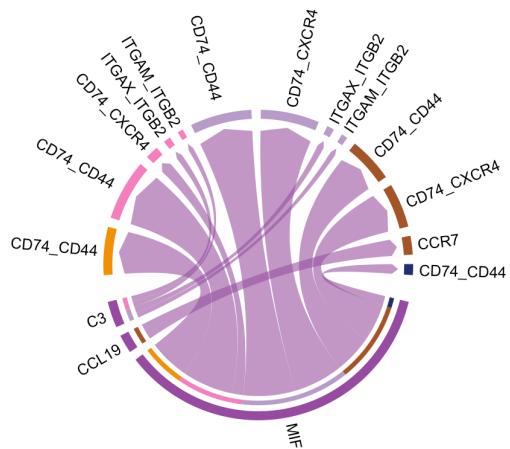
Visualize the upregulated and down-regulated signaling ligand-receptor pairs using Chord diagram

```

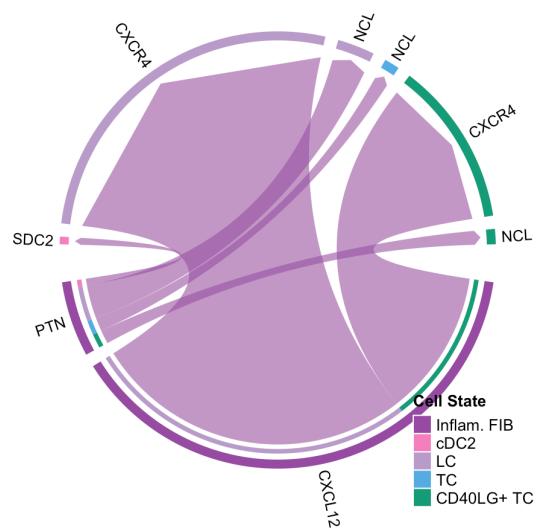
# Chord diagram
par(mfrow = c(1,2), xpd=TRUE)
netVisual_chord_gene(object.list[[2]], sources.use = 4, targets.use = c(5:11), slot.name = 'net', net = net.up, lab.cex = 0.8, small.gap = 3.5, title.name = paste0("Up-regulated signaling in ", names(object.list)[2]))
netVisual_chord_gene(object.list[[1]], sources.use = 4, targets.use = c(5:11), slot.name = 'net', net = net.down, lab.cex = 0.8, small.gap = 3.5, title.name = paste0("Down-regulated signaling in ", names(object.list)[2]))

```

Up-regulated signaling in LS



Down-regulated signaling in LS



Visualize the enriched ligands, signaling, or ligand-receptor pairs in one condition compared to another condition using wordcloud

```
# visualize the enriched ligands in the first condition  
computeEnrichmentScore(net.down, species = 'human')
```



```
# visualize the enriched ligands in the second condition  
computeEnrichmentScore(net.up, species = 'human')
```

C3 MIF  
 CCL19 TNF  
 IL13 FGF7  
 LGALS9  
 CXCL12

## Part IV: Visually compare cell-cell communication using Hierarchy plot, Circle plot or Chord diagram

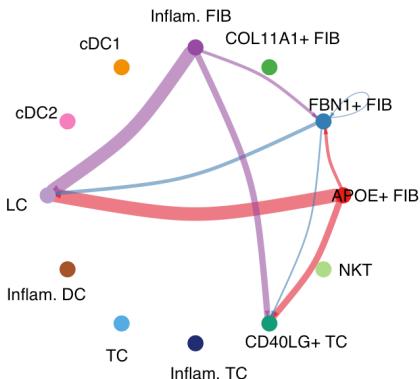
Similar to the CellChat analysis of individual dataset, we can visualize the cell-cell communication network using Hierarchy plot, Circle plot or Chord diagram.

**Edge color/weight, node color/size/shape:** In all visualization plots, edge colors are consistent with the sources as sender, and edge weights are proportional to the interaction strength. Thicker edge line indicates a stronger signal. In the **Hierarchy plot** and **Circle plot**, circle sizes are proportional to the number of cells in each cell group. In the hierarchy plot, solid and open circles represent source and target, respectively. In the **Chord diagram**, the inner thinner bar colors represent the targets that receive signal from the corresponding outer bar. The inner bar size is proportional to the signal strength received by the targets. Such inner bar is helpful for interpreting the complex chord diagram. Note that there exist some inner bars without any chord for some cell groups, please just ignore it because this is an issue that has not been addressed by circlize (<https://github.com/jokergoo/circlize>) package.

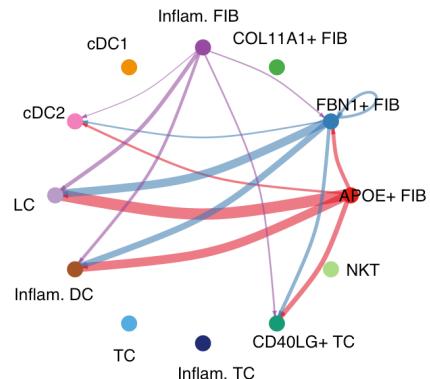
```

pathways.show <- c("CXCL")
weight.max <- getMaxWeight(object.list, slot.name = c("netP"), attribute = pathways.show) # control the edge weights
across different datasets
par(mfrow = c(1,2), xpd=TRUE)
for (i in 1:length(object.list)) {
  netVisual_aggregate(object.list[[i]], signaling = pathways.show, layout = "circle", edge.weight.max = weight.max
[1], edge.width.max = 10, signaling.name = paste(pathways.show, names(object.list)[i]))
}
  
```

CXCL NL signaling pathway network



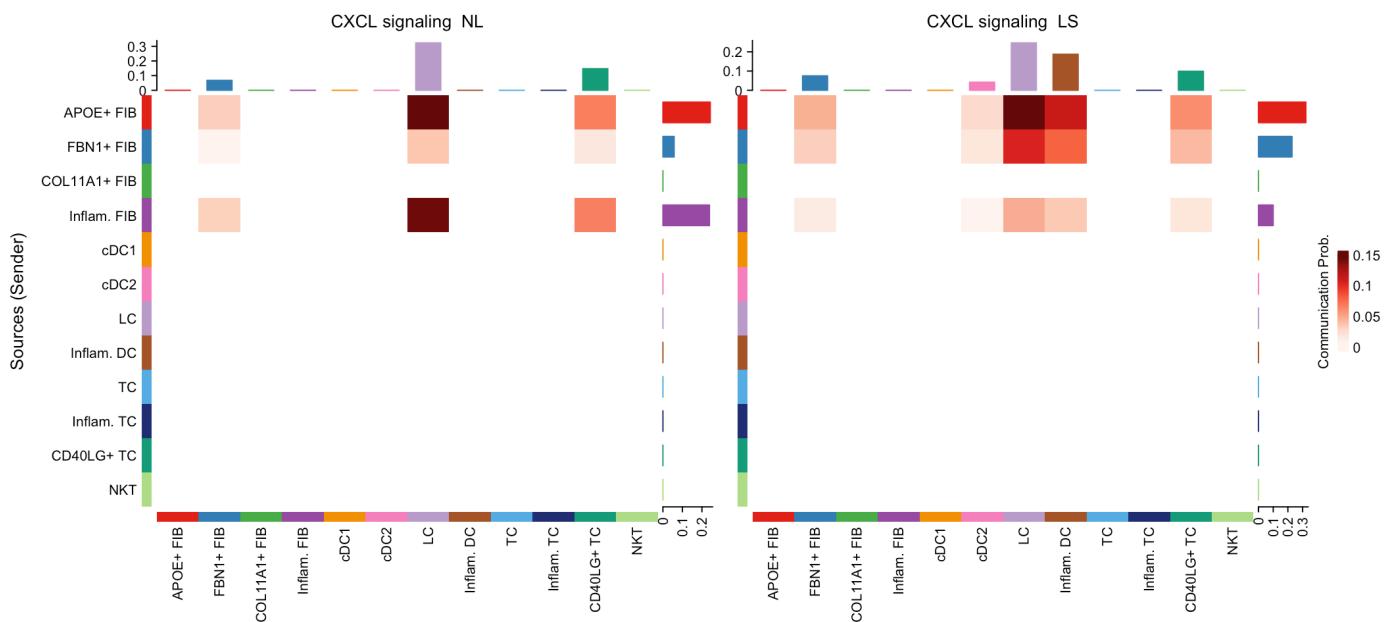
CXCL LS signaling pathway network



```

pathways.show <- c("CXCL")
par(mfrow = c(1,2), xpd=TRUE)
ht <- list()
for (i in 1:length(object.list)) {
  ht[[i]] <- netVisual_heatmap(object.list[[i]], signaling = pathways.show, color.heatmap = "Reds", title.name = paste
e(pathways.show, "signaling ",names(object.list)[i]))
}
#> Do heatmap based on a single object
#>
#> Do heatmap based on a single object
ComplexHeatmap::draw(ht[[1]] + ht[[2]], ht_gap = unit(0.5, "cm"))

```

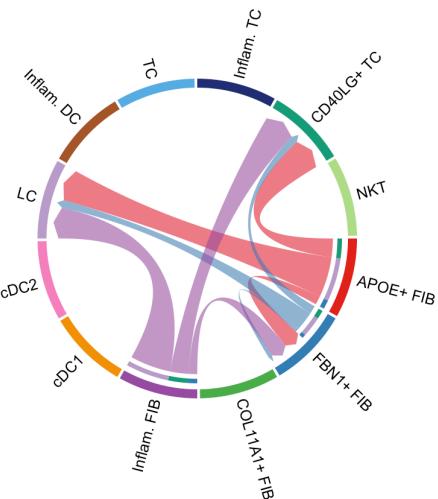


```

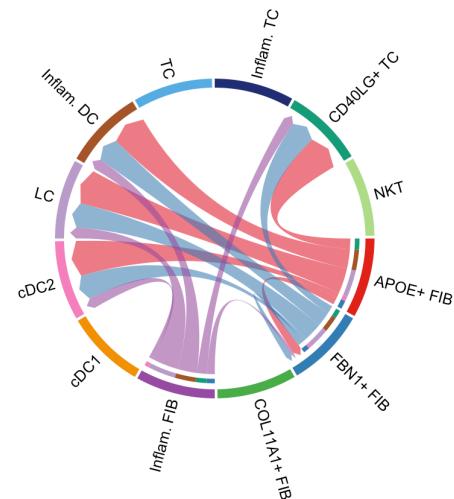
# Chord diagram
pathways.show <- c("CXCL")
par(mfrow = c(1,2), xpd=TRUE)
for (i in 1:length(object.list)) {
  netVisual_aggregate(object.list[[i]], signaling = pathways.show, layout = "chord", signaling.name = paste(pathway
s.show, names(object.list)[i]))
}

```

CXCL NL signaling pathway network



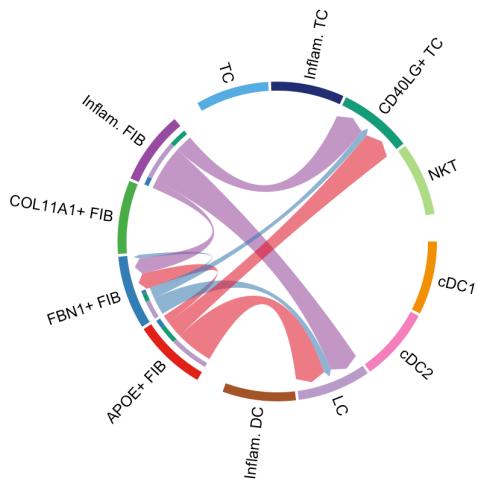
CXCL LS signaling pathway network



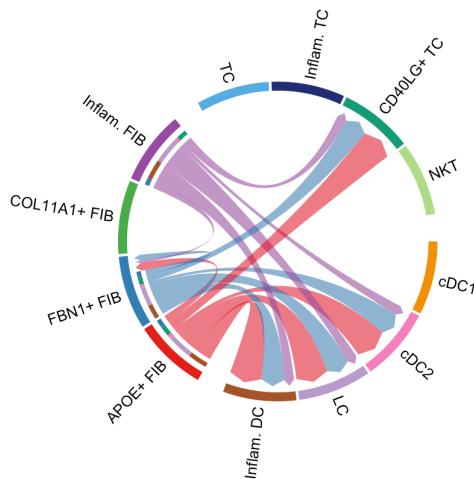
For the chord diagram, CellChat has an independent function `netVisual_chord_cell` to flexibly visualize the signaling network by adjusting different parameters in the circlize (<https://github.com/jokergoo/circlize>) package. For example, we can define a named char vector `group` to create multiple-group chord diagram, e.g., grouping cell clusters into different cell types.

```
# Chord diagram
group.cellType <- c(rep("FIB", 4), rep("DC", 4), rep("TC", 4)) # grouping cell clusters into fibroblast, DC and TC cells
names(group.cellType) <- levels(object.list[[1]]@idents)
pathways.show <- c("CXCL")
par(mfrow = c(1,2), xpd=TRUE)
for (i in 1:length(object.list)) {
  netVisual_chord_cell(object.list[[i]], signaling = pathways.show, group = group.cellType, title.name = paste0(pathways.show, " signaling network - ", names(object.list)[i]))
}
#> Plot the aggregated cell-cell communication network at the signaling pathway level
#> Plot the aggregated cell-cell communication network at the signaling pathway level
```

CXCL signaling network - NL



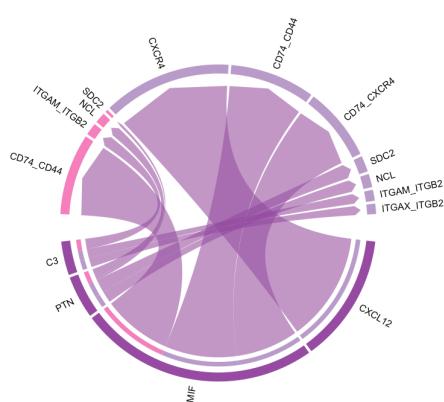
CXCL signaling network - LS



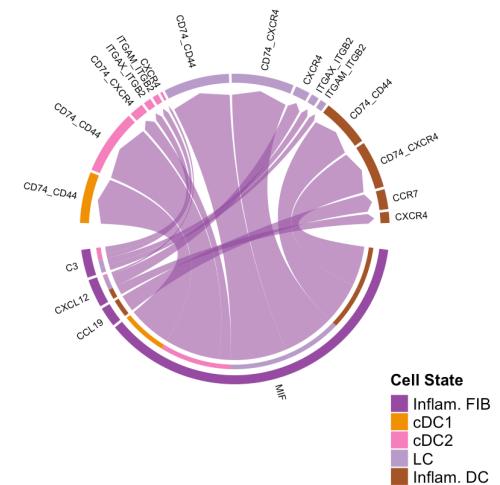
Using chord diagram, CellChat provides two functions `netVisual_chord_cell` and `netVisual_chord_gene` for visualizing cell-cell communication with different purposes and different levels. `netVisual_chord_cell` is used for visualizing the cell-cell communication between different cell groups (where each sector in the chord diagram is a cell group), and `netVisual_chord_gene` is used for visualizing the cell-cell communication mediated by multiple ligand-receptors or signaling pathways (where each sector in the chord diagram is a ligand, receptor or signaling pathway.)

```
par(mfrow = c(1, 2), xpd=TRUE)
# compare all the interactions sending from Inflam.FIB to DC cells
for (i in 1:length(object.list)) {
  netVisual_chord_gene(object.list[[i]], sources.use = 4, targets.use = c(5:8), lab.cex = 0.5, title.name = paste0("Signaling from Inflam.FIB - ", names(object.list)[i]))
}
```

Signaling from Inflam.FIB - NL

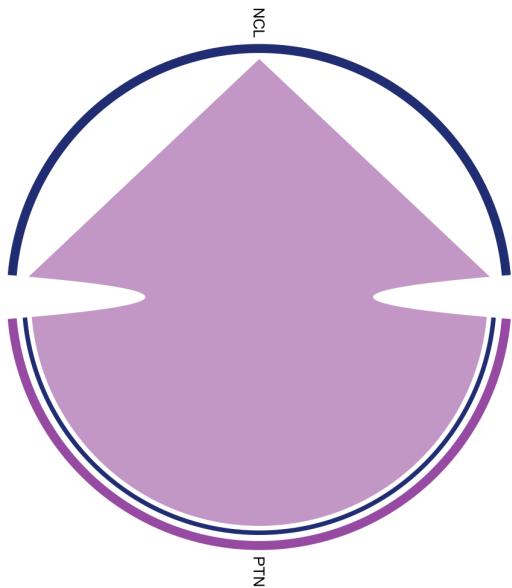


Signaling from Inflam.FIB - LS

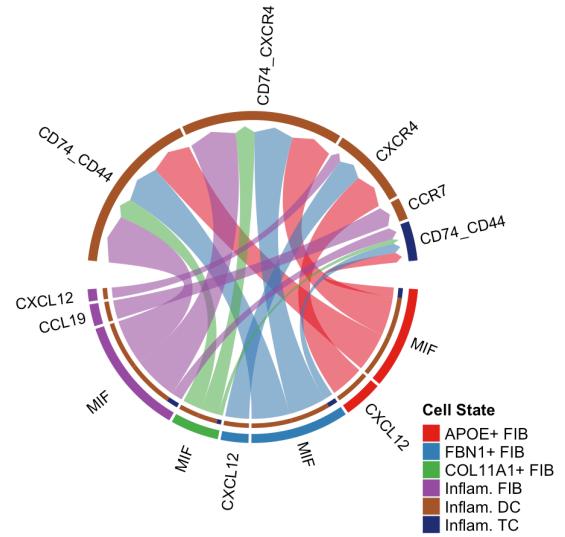


```
# compare all the interactions sending from fibroblast to inflammatory immune cells
par(mfrow = c(1, 2), xpd=TRUE)
for (i in 1:length(object.list)) {
  netVisual_chord_gene(object.list[[i]], sources.use = c(1,2, 3, 4), targets.use = c(8,10), title.name = paste0("Signaling received by Inflam.DC and .TC - ", names(object.list)[i]), legend.pos.x = 10)
}
```

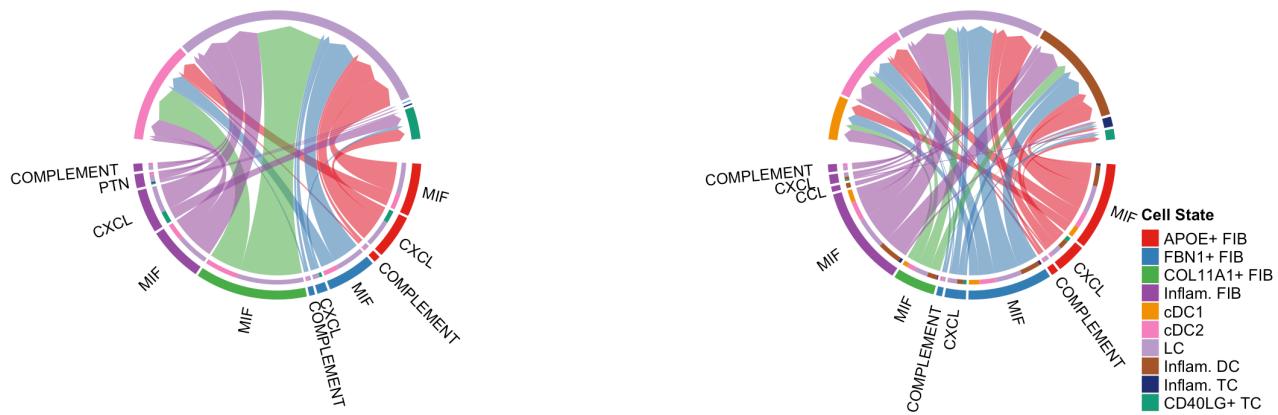
Signaling received by Inflam.DC and .TC - NL



Signaling received by Inflam.DC and .TC - LS



```
# show all the significant signaling pathways from fibroblast to immune cells
par(mfrow = c(1, 2), xpd=TRUE)
for (i in 1:length(object.list)) {
  netVisual_chord_gene(object.list[[i]], sources.use = c(1,2,3,4), targets.use = c(5:11), slot.name = "netP", title.name = paste0("Signaling pathways sending from fibroblast - ", names(object.list)[i]), legend.pos.x = 10)
}
```

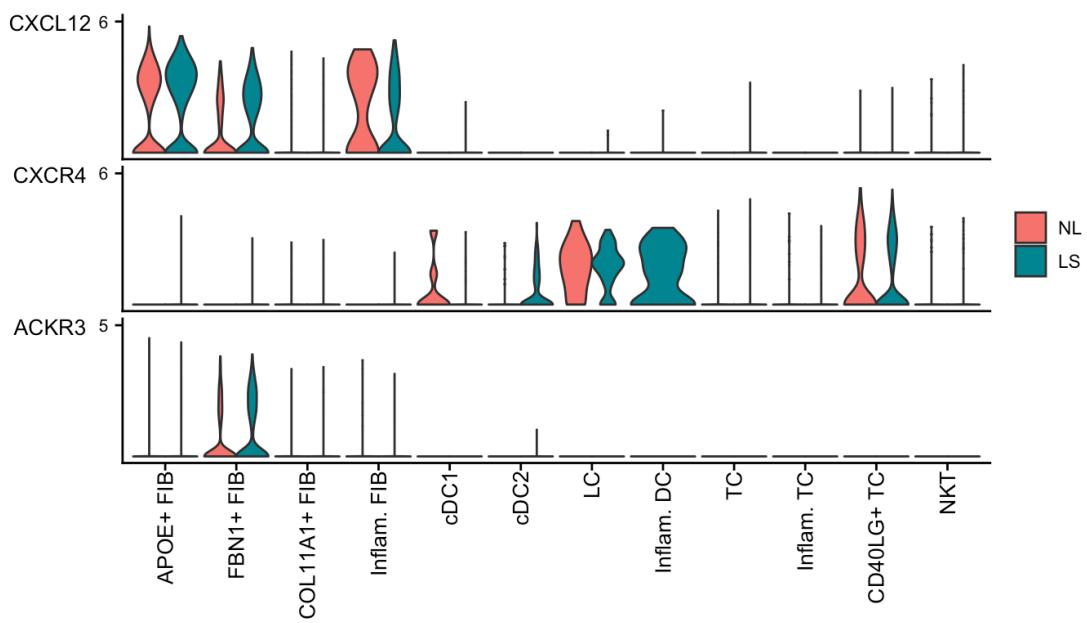


NB: Please ignore the note when generating the plot such as "Note: The first link end is drawn out of sector 'MIF'.". If the gene names are overlapped, you can adjust the argument `small.gap` by decreasing the value.

## Part V: Compare the signaling gene expression distribution between different datasets

We can plot the gene expression distribution of signaling genes related to L-R pairs or signaling pathway using a Seurat wrapper function `plotGeneExpression`.

```
cellchat@meta$datasets = factor(cellchat@meta$datasets, levels = c("NL", "LS")) # set factor level
plotGeneExpression(cellchat, signaling = "CXCL", split.by = "datasets", colors.ggplot = T)
#> The default behaviour of split.by has changed.
#> Separate violin plots are now plotted side-by-side.
#> To restore the old behaviour of a single split violin,
#> set split.plot = TRUE.
#>
#> This message will be shown once per session.
#> Scale for 'y' is already present. Adding another scale for 'y', which will
#> replace the existing scale.
#> Scale for 'y' is already present. Adding another scale for 'y', which will
#> replace the existing scale.
#> Scale for 'y' is already present. Adding another scale for 'y', which will
#> replace the existing scale.
```



## Save the merged CellChat object

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

```
sessionInfo()
#> R version 4.1.2 (2021-11-01)
#> Platform: x86_64-apple-darwin17.0 (64-bit)
#> Running under: macOS Big Sur 10.16
#>
#> Matrix products: default
#> BLAS: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
#> LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
#>
#> locale:
#> [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
#>
#> attached base packages:
#> [1] grid      stats     graphics  grDevices utils      datasets  methods
#> [8] base
#>
#> # other attached packages:
#> [1] ComplexHeatmap_2.10.0 patchwork_1.1.1      CellChat_1.6.0
#> [4] Biobase_2.54.0        BiocGenerics_0.40.0   ggplot2_3.3.5
#> [7] igraph_1.3.4         dplyr_1.0.7
#>
#> # loaded via a namespace (and not attached):
#> [1] backports_1.4.1      circlize_0.4.13    systemfonts_1.0.2
#> [4] NMF_0.23.0          plyr_1.8.6       lazyeval_0.2.2
#> [7] splines_4.1.2       BiocParallel_1.28.3  listenv_0.8.0
#> [10] scattermore_0.7    ggnetwork_0.5.10   gridBase_0.4-7
#> [13] digest_0.6.29     foreach_1.5.1     htmltools_0.5.2
#> [16] magick_2.7.3      ggalluvial_0.12.3  fansi_0.5.0
#> [19] magrittr_2.0.1    tensor_1.5       cluster_2.1.2
#> [22] doParallel_1.0.16 ROCR_1.0-11      sna_2.6
#> [25] globals_0.14.0    wordcloud_2.6     matrixStats_0.61.0
#> [28] svglite_2.0.0     spatstat.sparse_2.1-0 colorspace_2.0-2
#> [31] ggrepel_0.9.1     xfun_0.33       crayon_1.4.2
#> [34] jsonlite_1.7.2    spatstat.data_2.1-2 zoo_1.8-9
#> [37] survival_3.2-13   iterators_1.0.13   glue_1.6.0
#> [40] polyclip_1.10-0   registry_0.5-1    gtable_0.3.0
#> [43] leiden_0.3.9     GetoptLong_1.0.5   car_3.0-12
#> [46] future.apply_1.8.1 shape_1.4.6     abind_1.4-5
#> [49] scales_1.1.1     DBI_1.1.2       rngtools_1.5.2
#> [52] rstatix_0.7.0    miniUI_0.1.1.1   Rcpp_1.0.7
#> [55] viridisLite_0.4.0 xtable_1.8-4     clue_0.3-60
#> [58] spatstat.core_2.3-2 reticulate_1.22  stats4_4.1.2
#> [61] htmlwidgets_1.5.4   httr_1.4.2      FNN_1.1.3
#> [64] RColorBrewer_1.1-2 ellipsis_0.3.2   Seurat_4.0.6
#> [67] ica_1.0-2        pkgconfig_2.0.3  farver_2.1.0
#> [70] uwot_0.1.11     deldir_1.0-6    sass_0.4.0
#> [73] utf8_1.2.2      here_1.0.1     later_1.3.0
#> [76] tidyselect_1.1.1  labeling_0.4.2   rlang_0.4.12
#> [79] reshape2_1.4.4   munsell_0.5.0   tools_4.1.2
#> [82] generics_0.1.1   statnet.common_4.5.0 broom_0.7.10
#> [85] ggridges_0.5.3  evaluate_0.17   stringr_1.4.0
#> [88] fastmap_1.1.0   goftest_1.2-3   yaml_2.2.1
#> [91] knitr_1.40      fitdistrplus_1.1-6 purrr_0.3.4
#> [94] RANN_2.6.1      nlme_3.1-153   pbapply_1.5-0
#> [97] future_1.23.0   mime_0.12      compiler_4.1.2
#> [100] plotly_4.10.0  png_0.1-7     ggsignif_0.6.3
#> [103] spatstat.utils_2.3-0 tibble_3.1.6   bslib_0.3.1
#> [106] stringi_1.7.6  highr_0.9      RSpectra_0.16-0
#> [109] lattice_0.20-45 Matrix_1.3-4    vctrs_0.3.8
#> [112] pillar_1.6.4   lifecycle_1.0.1  spatstat.geom_2.3-1
#> [115] lmtest_0.9-39   jquerylib_0.1.4  GlobalOptions_0.1.2
#> [118] RcppAnnoy_0.0.19 BiocNeighbors_1.12.0 data.table_1.14.2
#> [121] cowplot_1.1.1   irlba_2.3.5    httpuv_1.6.4
#> [124] R6_2.5.1       promises_1.2.0.1 network_1.17.1
#> [127] gridExtra_2.3   KernSmooth_2.23-20 IRanges_2.28.0
#> [130] parallelly_1.30.0 codetools_0.2-18 MASS_7.3-54
#> [133] assertthat_0.2.1 pkgmaker_0.32.2   rprojroot_2.0.2
#> [136] rjson_0.2.20    withr_2.4.3   SeuratObject_4.0.4
```

```
#> [139] sctransform_0.3.2      S4Vectors_0.32.3      mgcv_1.8-38
#> [142] parallel_4.1.2        rpart_4.1-15         tidyverse_1.1.4
#> [145] coda_0.19-4          rmarkdown_2.17       carData_3.0-4
#> [148] Rtsne_0.15           ggpubr_0.4.0         shiny_1.7.1
```