nature protocols
https://doi.org/10.1038/s41596-024-01045-4

Protocol

Check for updates

CellChat for systematic analysis of cell–cell communication from single-cell transcriptomics

Suoqin Jin[1,2], Maksim V. Plikus[3,4] & Qing Nie[3,4,5]

Abstract

Recent advances in single-cell sequencing technologies offer an opportunity to explore cell–cell communication in tissues systematically and with reduced bias. A key challenge is integrating known molecular interactions and measurements into a framework to identify and analyze complex cell–cell communication networks. Previously, we developed a computational tool, named CellChat, that infers and analyzes cell–cell communication networks from single-cell transcriptomic data within an easily interpretable framework. CellChat quantifies the signaling communication probability between two cell groups using a simplified mass-action-based model, which incorporates the core interaction between ligands and receptors with multisubunit structure along with modulation by cofactors. Importantly, CellChat performs a systematic and comparative analysis of cell–cell communication using a variety of quantitative metrics and machine-learning approaches. CellChat v2 is an updated version that includes additional comparison functionalities, an expanded database of ligand–receptor pairs along with rich functional annotations, and an Interactive CellChat Explorer. Here we provide a step-by-step protocol for using CellChat v2 on single-cell transcriptomic data, including inference and analysis of cell–cell communication from one dataset and identification of altered intercellular communication, signals and cell populations from different datasets across biological conditions. The R implementation of CellChat v2 toolkit and its tutorials together with the graphic outputs are available at https://github.com/jinworks/CellChat. This protocol typically takes ~5 min depending on dataset size and requires a basic understanding of R and single-cell data analysis but no specialized bioinformatics training for its implementation.

Key points

- CellChat is a software package for systematic inference, quantitative analysis and intuitive visualization of cell–cell communication in an easily interpretable way from single-cell transcriptomic data; it also enables comparative analysis of intercellular communication across different conditions.

- CellChat v2 is an updated version that includes additional functionalities for comparative analysis and an expanded database of ligand–receptor pairs along with rich functional annotations.

Key references

Jin, S. et al. *Nat. Commun*. **12**, 1088 (2021): https://doi.org/10.1038/s41467-021-21246-9

Vu, R. et al. *Cell Rep*. **40**, 111155 (2022): https://doi.org/10.1016/j.celrep.2022.111155

[1]School of Mathematics and Statistics, Wuhan University, Wuhan, China. [2]Hubei Key Laboratory of Computational Science, Wuhan University, Wuhan, China. [3]NSF-Simons Center for Multiscale Cell Fate Research, University of California, Irvine, Irvine, CA, USA. [4]Department of Developmental and Cell Biology, University of California, Irvine, Irvine, CA, USA. [5]Department of Mathematics, University of California, Irvine, Irvine, CA, USA. e-mail: sqjin@whu.edu.cn; qnie@uci.edu

Introduction

Cell–cell communication orchestrates tissue organization. Recent advances in single-cell genomics offer unprecedented opportunities to systematically explore signaling mechanisms for cell fate decisions and their consequent tissue phenotypes. Using single-cell transcriptomic data and ligand–receptor (L–R) interaction information from prior knowledge, computational methods such as CellPhoneDB have been developed for inferring cell–cell communication between groups of cells[1–4]. However, a versatile and easy-to-use toolkit capable of systematic analysis and intuitive visualization of cell–cell communication as well as comparison analysis across biological conditions was still needed, so we developed CellChat to systematically and comprehensively infer and analyze cell–cell communication from single-cell transcriptomic data within an easily interpretable framework[5].

Development of the protocol

Comprehensive and accurate recapitulation of known molecular interactions is crucial for predicting biologically meaningful intercellular communications. We manually curated a literature-supported signaling molecule interaction database called CellChatDB[5], which considers several critical interaction mechanisms that are often neglected. Specifically, CellChatDB not only incorporates the multisubunit structure of L–R complexes but also accounts for soluble and membrane-bound stimulatory and inhibitory cofactors such as agonists, antagonists and coreceptors (Fig. 1). In addition, CellChatDB classifies each L–R pair into one of the functionally related signaling pathways (for example, WNT, BMP, CXCL and CCL) to construct cell–cell communication networks at a signaling pathway level, where each link of the network is computed by summing the interaction strengths of all associated L–R pairs. Such information allows the interpretation of inferred intercellular communications at a pathway scale. Moreover, the L–R pairs are categorized into different types, including 'Secreted Signaling', 'ECM-Receptor' and 'Cell–Cell Contact' (where ECM is extracellular matrix). The updated CellChat v2 expands upon the original CellChatDB database to include more than 1,000 protein and nonprotein interactions (for example, metabolic and synaptic signaling) based on the peer-reviewed literature and other existing databases such as CellPhoneDB[6] and NeuronChatDB[7]. In addition, CellChat v2 includes additional functional annotations of L–R pairs, such as UniProtKB keywords (including biological process, molecular function, functional class, disease and so on), subcellular location and relevance to neurotransmitter.

To quantify communication between two cell groups mediated by a given ligand and its cognate receptor, CellChat leverages the law of mass action to associate each interaction with an interaction score[5], which is calculated based on the average expression values of a ligand by one cell group and that of a receptor by another cell group, as well as their cofactors (Fig. 1). CellChat uses Hill functions in the simplified mass action model to reflect the saturation effect of the L–R binding. Significant interactions are identified based on a statistical test that randomly permutes the group labels of cells. When inferring cell–cell communication, CellChat computationally scales well with the number of cells and cell groups in the data, as reflected by the observed running time of ~15 min on a single cell atlas of adult human skin with ~300,000 cells (Fig. 2). It should be noted that the inferred signaling depends on the method for calculating average gene expression per cell group. To demonstrate this point, we used a human skin dataset from atopic dermatitis patients to compare the number of inferred interactions and the enriched signaling pathways when using 'triMean', 'truncatedMean' with 'trim = 0.1' and 'truncatedMean' with 'trim = 0.05', respectively (Procedure 1; Fig. 3a,b). The most stringent method, called 'triMean', produces fewer but stronger interactions, whereas the 'truncated Mean' method, with smaller values of 'trim' parameter (for example, 'trim = 0.1'), outputs more interactions, leading to the detection of weak signaling.

To obtain biological insights from many complicated cell–cell communication networks, CellChat employs quantitative analysis and machine learning approaches for various critical analysis tasks[5] (Fig. 1). First, to identify critical microenvironment components, CellChat determines major signaling sources and targets, as well as mediators and influencers within

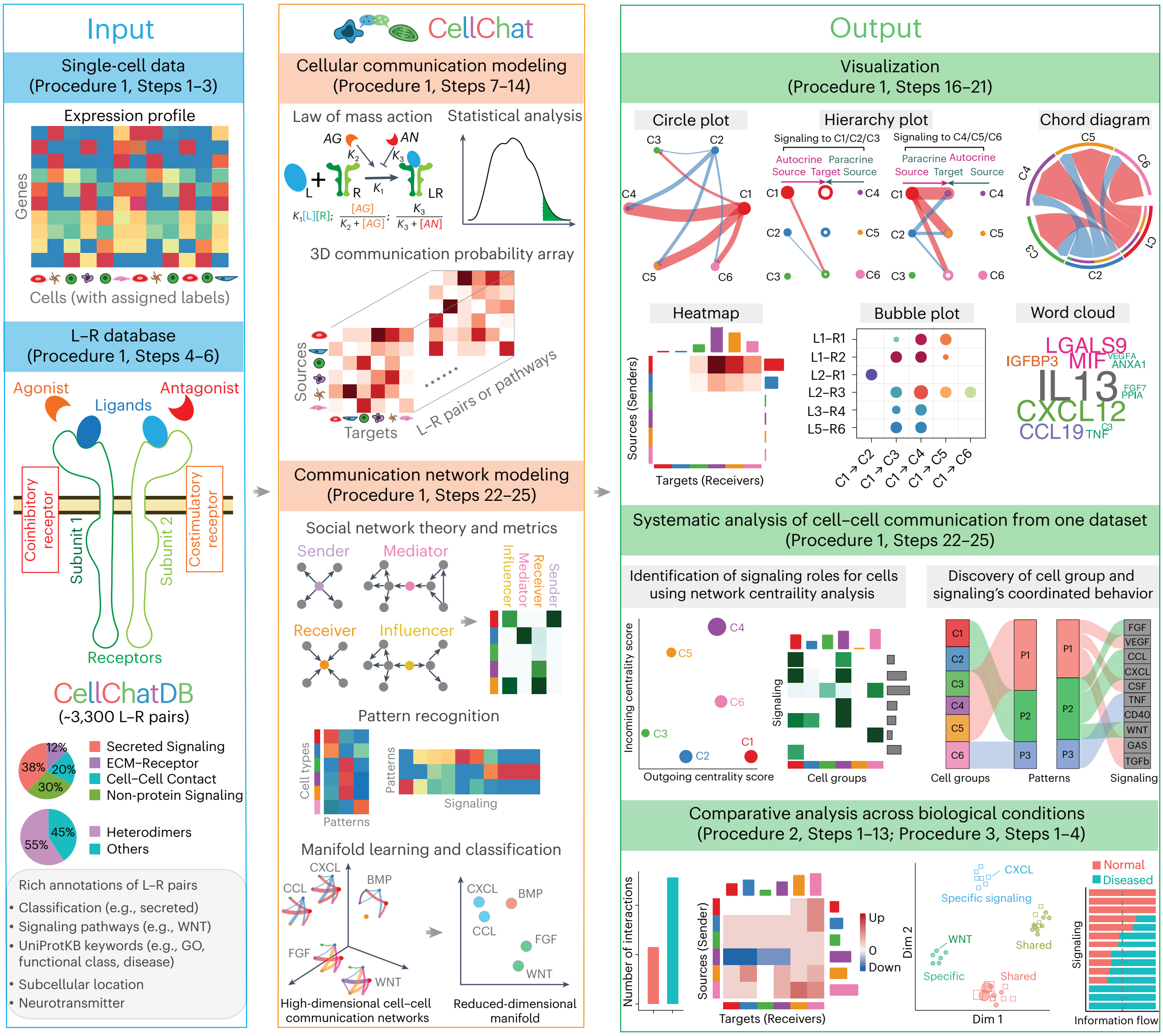


Fig. 1 | Overview of CellChat along with the procedure step numbers. Left: required input data and the L–R interaction database CellChatDB. CellChat's input data consist of gene expression data and cell group information. CellChatDB considers known composition of the L–R complexes, including complexes with multimeric ligands and receptors, as well as several cofactor types: soluble agonists, antagonists, costimulatory and coinhibitory membrane-bound receptors. Rich annotations of all L–R pairs are provided. Middle: CellChat models the communication probability based on the law of mass action and identifies significant communications using permutation tests. The inferred communication probabilities among all pairs of cell groups across all L–R pairs or signaling pathways are represented by a three-dimensional (3D) array. CellChat analyzes the inferred networks by leveraging social network metrics, pattern recognition methods, and manifold learning approaches. Right: CellChat offers several intuitive visualization outputs to facilitate data interpretation of different analytical tasks. In addition to analyzing individual datasets, CellChat also delineates signaling changes across different conditions.

a given signaling network using network centrality analysis. Second, to reveal how cells and signals coordinate together and to explore their communication patterns, CellChat predicts key incoming and outgoing signals for specific cell types, as well as coordinated responses among different cell types by leveraging pattern recognition approaches. Outgoing patterns reveal how sender cells (that is, cells acting as signal sources) coordinate with each other, as well as how they coordinate with certain signaling pathways to drive communication. Incoming patterns show how target cells (that is, cells acting as signal receivers) coordinate with each other to respond to incoming signals. Third, to predict signaling groups sharing similar communication

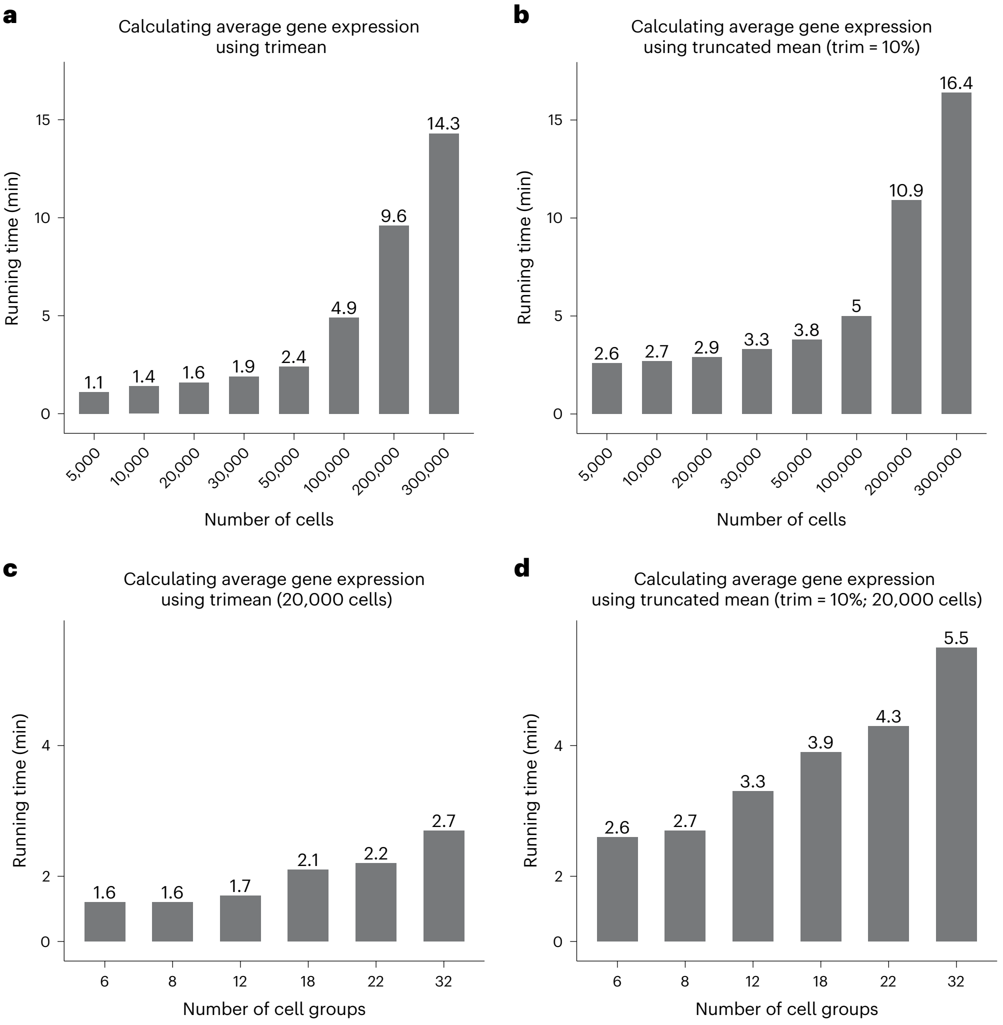


Fig. 2 | CellChat running time in relation to the increase of cell numbers and cell groups. a,**b**, Running time over different cell numbers in the data when calculating average gene expression per cell group using trimean (**a**) or 10% truncated mean (**b**). **c**,**d**, Running time over different numbers of cell groups in the data (no. 20,000 cells) when calculating average gene expression per cell group using trimean (**c**) or 10% truncated mean (**d**). Here, the running time is the total time when running Steps 1–8 and 11–14 in Procedure 1.

architecture and interpret the biological functions of poorly studied pathways, CellChat groups signaling pathways by defining similarity measures and performing manifold learning from both functional and topological perspectives[5].

To identify signaling changes across conditions, CellChat identifies altered signaling pathways and L–R pairs in terms of network architecture and interaction strength by performing joint manifold learning and information flow comparison analysis[5]. Compared to the original CellChat tool, CellChat v2 provides additional functionalities to allow systematic comparisons between multiple conditions. CellChat v2 first focuses on the overall signaling changes at the cell population level and then narrows down to altered signaling pathways and L–R pairs[5,8,9] (Fig. 1). Specifically, CellChat v2 identifies which interactions between two specific cell groups changed notably, as well as the cell group identities showing notable changes in sending or receiving signaling patterns across conditions. To identify substantially upregulated and downregulated L–R pairs across conditions, CellChat v2 combines cell–cell communication analysis with differential gene expression analysis and quantifies the enrichment of L–R pairs for each condition by defining an enrichment score[8].

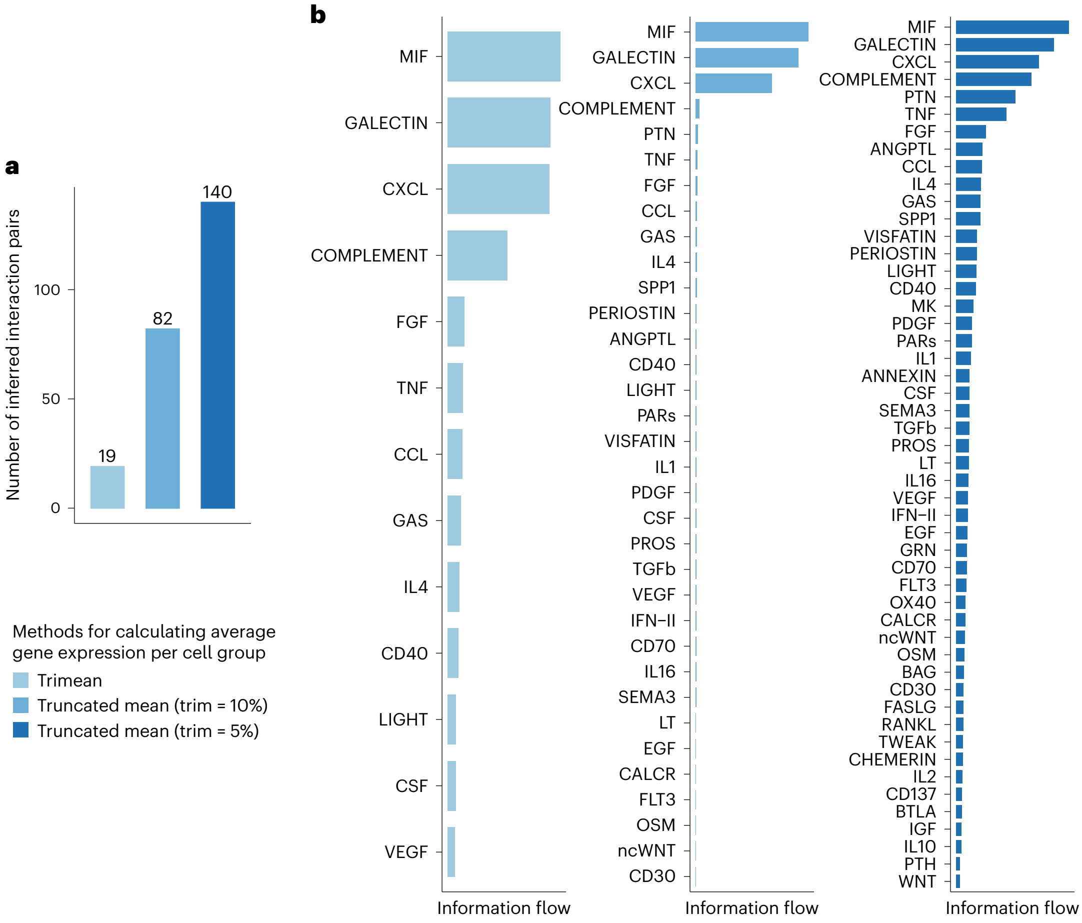
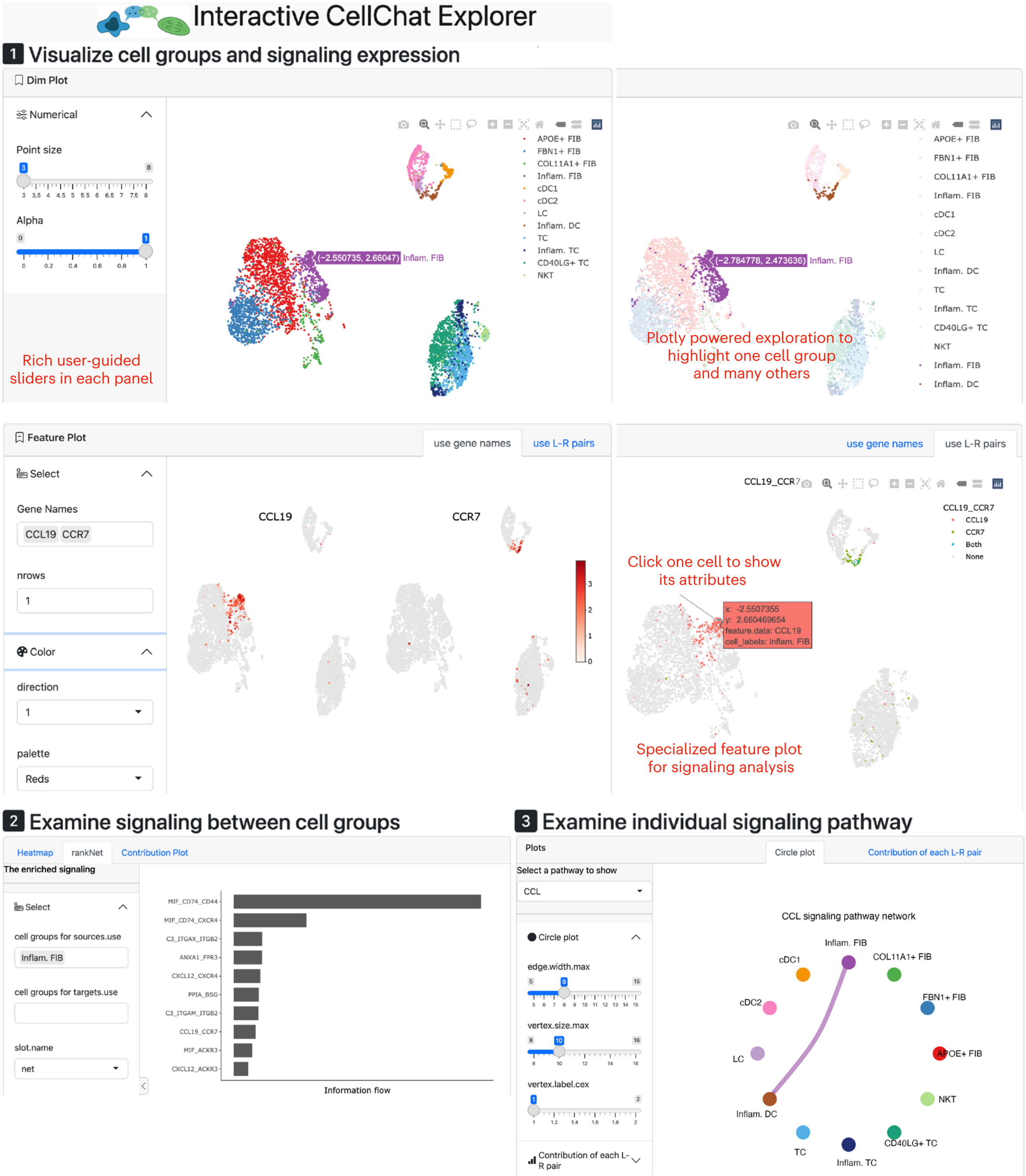


Fig. 3 | Comparison of the number of inferred L–R pairs and the identified signaling pathways when using different methods for calculating average gene expression per cell group. a, The number of inferred L–R pairs when using three different methods for calculating average gene expression per cell group, including trimean, 10% truncated mean and 5% truncated mean. **b**, The identified signaling pathways when using trimean, 10% truncated mean and 5% truncated mean. The most stringent method 'triMean' produces fewer but stronger interactions, while the 'truncated Mean' method with smaller values of 'trim' parameter enables the identification of weak signaling. This analysis is performed on a human skin dataset from atopic dermatitis patients with 5,011 cells and 12 cell groups.

Moreover, CellChat v2 offers an interactive web browser function to allow intuitive exploration and visualization of CellChat outputs (Fig. 4). To facilitate intuitive user-guided data interpretation, CellChat v2 provides a variety of visualization outputs, including circle plot, chord diagram, heatmap, hierarchy plot, bubble plot and word cloud (Fig. 1).

Comparison with other methods

Numerous computational tools have been developed to facilitate cell–cell communication exploration and analysis[2,10–18]. The cell–cell communication inference depends on the reference databases of known L–R interactions. The Python tool CellPhoneDB[12,19] is a pioneering method that considers multiple subunits of ligands and receptors to accurately represent known heteromeric molecular complexes. Two other R-based tools, CellChat[5] and ICELLNET[15], adopted the subunit architecture of heteromeric complexes and other tools have since followed their lead. Compared with CellPhoneDB and CellChat, which have over 2,000 L–R interactions, ICELLNET only has 380 interactions, resulting in partial characterization of signaling pathways. Recently, CellPhoneDB v4[20] added interactions of nonprotein molecules not directly encoded by genes, and NeuronChat[7] was designed specifically for neuron-to-neuron communication mediated by neurotransmitters. In CellChat v2, we add new literature-supported interactions,



Interactive CellChat Explorer

1 Visualize cell groups and signaling expression

Dim Plot

Numerical

Point size: 3

Alpha: 1

Rich user-guided sliders in each panel

Plotly powered exploration to highlight one cell group and many others

Feature Plot

Gene Names: CCL19 CCR7

nrows: 1

direction: 1

palette: Reds

Click one cell to show its attributes

x: -2.5507355
y: 2.660469654
feature.data: CCL19
cell_labels: Inflam. FIB

Specialized feature plot for signaling analysis

2 Examine signaling between cell groups

The enriched signaling

cell groups for sources.use: Inflam. FIB

cell groups for targets.use:

slot.name: net

3 Examine individual signaling pathway

Select a pathway to show: CCL

edge.width.max: 8

vertex.size.max: 10

vertex.label.cex: 1

Fig. 4 | Overview of the Interactive CellChat Explorer created by runCellChatApp function in the R package. To facilitate the exploration of cell–cell communication, CellChat allows the end-user to visualize and explore the data and the inferred signaling interactively. CellChat Explorer (1) visualizes cell groups and signaling expression, (2) examines the inferred signaling between different cell groups and (3) further visualizes the individual signaling pathway. Rich user-guided sliders are provided for flexible exploration, highlight and zoom-out of the related information of interest.

including both proteins and nonproteins acting as ligands, leading to a total of ~3,300 interactions for both mouse and human. Four unique features of CellChatDB v2 are:

1. Incorporation of soluble and membrane-bound stimulatory and inhibitory cofactors. This feature is considered because many pathways, such as BMP and WNT, are prominently modulated, positively or negatively, by their cofactors.
2. Categorization of L–R pairs into different types, including 'Secreted Signaling', 'ECM-Receptor', 'Cell–Cell Contact' and 'Non-protein Signaling'. This feature greatly facilitates cell–cell communication analysis within a particular type.
3. Classification of L–R pairs into functionally related signaling pathways. This feature provides useful insights into signaling mechanisms by examining cell–cell communication at a signaling pathway scale.
4. Rich annotations of each L–R pair. This feature is useful for selecting L–R pairs with similar biological functions and interpreting the downstream analysis.

Despite the adoption of different built-in L–R databases, current tools for cell–cell communication inference are all somewhat distinct in their performance, visualization outputs and downstream analysis. Two recent systematic evaluations of more than 15 cell–cell communication inference methods suggest CellChat is among the top-performing methods[11,18]. In addition to the high accuracy of cell–cell communication inference, CellChat offers a variety of visualization outputs that allow multiple intuitive user-guided interpretations of the complex cell–cell communication. Another key unique feature of CellChat is its ability to analyze the inferred cell–cell communications using a systems approach. Methods and concepts from social network analysis, pattern recognition and manifold learning are adapted to derive higher-order network information in an easily interpretable way. Moreover, CellChat is the pioneering method for the systematic comparison of communications inferred for different conditions, which is critically important for identifying altered signaling mechanisms responsible for cell fate decisions in single-cell studies. Afterwards, methods such as Connectome[16], Tensor-cell2cell[17] and multinichenetr[21] introduced functionalities for comparison across multiple conditions.

Applications of the method

So far, CellChat has been widely used in a broad range of biological systems to dissect signaling mechanisms during tissue homeostasis, development and disease[22]. In our original report[5], we applied CellChat to a small conditional RNA sequencing (scRNA-seq) dataset on mouse skin development and predicted a novel role of Edn3 signaling in stimulating the directed migration of melanocytes into placodes during hair follicle formation. Comparative analysis of nonlesional and lesional human skin from patients with atopic dermatitis using CellChat uncovered major signaling changes in response to disease. CCL19-CCR7 was identified as the most important signaling event activated in lesional skin, contributing to the communication from inflammatory fibroblasts to dendritic cells. Recently, we used CellChat to study aging-dependent dysregulations during skin wound healing in mice[8], showing system-level differences in the number, strength, route and signaling mediators of putative cell–cell communications in young versus aged skin wounds.

Using CellChat, a previous study found a strong increase of key inflammatory pathways in the choroid-to-cortex network in patients with coronavirus disease 2019 (COVID-19) compared with control individuals[23]. Another study revealed increased interactions of CD163/LGMN-macrophages with myofibroblasts, fibroblasts and pericytes at later time points of COVID-19-induced 'acute respiratory distress syndrome'[24]. In a single-cell atlas of the adult human cerebral vasculature[25], CellChat analysis identified Nd2 as the strongest contributor to abnormal cell communications in arteriovenous malformations. Recently, state- and niche-dependent signaling pathways for reparative states in proximal and distal tubules have been identified by mining healthy and injured human kidney single-cell atlases[26]. Comparative analysis of *Gabbr1* mutant and control cortices from adult mice uncovered alterations in astrocyte–neuron communication[27]. CellChat has been used to predict a new role for a unique subset of cancer-associated fibroblasts in recruiting monocytes and neutrophils using in situ tumor arrays[28].

A study of PD1 blockade in mismatch repair-deficient colorectal cancer identified an interaction between CD4$^+$ T helper cells and germinal center B cells in antitumor immunity during immune checkpoint inhibitor treatment[29].

Limitations

It is possible that there are missing L–R interactions not covered in CellChatDB. Guidelines to update CellChatDB by adding user-defined L–R pairs or integrating other resources are provided in Box 1. There are several other limitations to the original CellChat and its updated version (v2), including the following:

- CellChat infers potential interactions between cell groups without considering heterogeneity within the defined cell groups. Users can refine cell grouping via subclustering analysis before applying CellChat.
- Like other methods, CellChat is limited to hypothesis generation and employs heuristics to guide the interpretation of cell–cell communication outputs. With limited benchmarking studies[10,11,18], the question of how to better validate the inferred signaling networks and their downstream gene outputs remains to be answered.
- Cross-condition analysis in CellChat is largely restricted to pairwise comparisons. Identification of signaling changes across multiple conditions and time series is valuable.
- For nonprotein-mediated cellular communication such as metabolic or synaptic signaling (where molecules are not directly encoded by genes measured in scRNA-seq), CellChat v2 approximately estimates the expression of ligands and receptors using the molecules' key mediators or enzymes. More sophisticated computational methods for estimating the expression of those signaling molecules could likely improve the inference accuracy.
- Given that cell–cell communication occurs within a short spatial distance and at the protein level, newly emerging data modalities (for example, spatially resolved transcriptomics[22,30,31] and single-cell multiomics such as single-cell proteomics[32] and epigenomics[33–35]) can be used to improve the inference of cell–cell communication. Recently, several methods have been developed for spatially resolved transcriptomics[4], such as SpaOTsc[36], SpaTalk[37], COMMOT[38], CellPhoneDB v3[13] and HoloNet[39], which are better at detecting spatially proximal cell–cell communication.
- CellChat employs a simplified mass-action-based model to quantify communication probability between a given ligand and its cognate receptor, and models with more biochemical details can potentially improve inference predictions. Finally, incorporation of the downstream signaling events of activated receptors on receiving cells could further improve the overall inference accuracy[40–44].

Overview of the procedure

Procedure 1 demonstrates the steps to run the CellChat package for inferring (Steps 1–15), visualizing (Steps 16–21) and analyzing (Steps 22–27) cell–cell communication from a single scRNA-seq dataset. Specifically, Procedure 1 includes the preprocessing of the input data (Steps 1–9) and the inference of cell–cell communication at both a L–R pair level and a signaling pathway level (Steps 10–15), the visualization of cell–cell communication networks of individual (Steps 16–19) and multiple (Step 20) signaling pathways or L–R pairs, the identification of the signaling roles and major contributing genes and pathways between cell groups (Steps 22–23), the analysis of global communication patterns (Step 24) and the manifold learning and classification analysis of signaling networks (Step 25), as well as the interactive exploration of the inferred cell–cell communication through a CellChat Shiny App (Step 26).

Procedure 2 demonstrates CellChat's ability to perform comparative analysis across different biological conditions by quantitative contrasts and joint manifold learning, including merging different CellChat objects together (Steps 1–4), detecting altered interactions and cell populations (Steps 5–9), altered signaling with distinct network architecture (Step 10) and interaction strength (Steps 11–13), as well as visually comparing the inferred cell–cell communication networks (Steps 14–15).

Procedure 3 briefly demonstrates how to apply CellChat to the comparative analysis of multiple conditions with differing cell type compositions (Steps 1–5).

BOX 1

Updating the L–R interaction database CellChatDB

In this box, we demonstrate the use of the function 'updateCellChatDB' to update the L–R interaction database CellChatDB by integrating new L–R pairs from other cell–cell communication analysis tools or utilizing a custom L–R interaction database.

Additional material:
Input data:

- Customized L–R pairs: a data frame with at least two columns named 'ligand' and 'receptor'. To infer cell–cell communication at a signaling pathway level, another column named 'pathway_name' must be provided, which classifies each L–R pair into one of known signaling pathways
- (Optional) Additional input files: (1) gene information: a data frame with one column named as 'Symbol'; (2) complex information: a data frame in which each row is the subunit information of either ligand or receptor; and (3) cofactor information: a data frame in which each row is the cofactor information of each pair

  ▲ **CRITICAL** Users can check the details of the required input data in the online tutorial (https://htmlpreview.github.io/?https://github.com/jinworks/CellChat/blob/master/tutorial/Update-CellChatDB.html), particularly the example codes on how to utilize other resources such as CellTalkDB and CellPhoneDB.

Procedure

▲ **CRITICAL** To demonstrate how to update the L–R interaction database, we use CellTalkDB[50] in human as an example. CellTalkDB can be downloaded from https://github.com/ZJUFanLab/CellTalkDB.

1. Load the customized L–R pairs by typing the following command in RStudio:

```
db.user <- readRDS("./CellTalkDB-master/database/human_lr_pair.rds")
```

2. (Optional) Load the gene information:

```
gene_info <- readRDS("./CellTalkDB-master/data/human_gene_info.rds")
```

3. (Optional) Modify the colnames if needed

```
colnames(db.user) <- plyr::mapvalues(colnames(db.user), from = c("ligand_gene_symbol","receptor_gene_
symbol","lr_pair"), to = c("ligand","receptor","interaction_name"), warn_missing = TRUE)
```

4. Create a new database by using the user-provided gene information (option A), create a new database by using the built-in gene information (option B) or integrate the customized L–R pairs into the built-in CellChatDB (option C).

   A. Create a new database by using the user-provided gene information:

```
db.new <- CellChat::updateCellChatDB(db = db.user, gene_info = gene_info)
```

   B. Create a new database by using the built-in gene information:

```
db.new <- CellChat::updateCellChatDB(db = db.user, gene_info = NULL, species_target = "human")
```

   C. Integrate the customized L–R pairs into the built-in CellChatDB:

```
db.new <- updateCellChatDB(db = db.user, merged = TRUE, species_target = "human")
```

5. Use this new database in the Procedure 1, Step 6 for CellChat analysis

```
cellchat@DB <- db.new
```

6. Save the new database for future use

```
save(db.new, file = "CellChatDB.human_user.rda")
```

Experimental design

RNA isolation and sequencing data

Although CellChat can, in principle, be used for any single-cell transcriptomics datasets, the quality of datasets directly affects the quality of CellChat outputs. First, having sufficient sequencing depth is critical to capturing gene expression of ligands and receptors. Expression levels are usually low for ligands during development, so sensitivity and depth of sequencing become particularly important for such cases. Second, batch effect may introduce output variability for any inference method, including CellChat. Whenever possible, it is important to use the same RNA isolation protocol for replicates and different conditions. To perform control analysis, we include several datasets that have been well explored using CellChat with known signaling events or pathways. New CellChat users are encouraged to first test their CellChat code on these datasets by comparing the outputs with the deposited cell–cell communication results.

Required input data

CellChat requires two user inputs: one is the gene expression data of cells and the other is the user-assigned cell labels. For the gene expression data matrix, genes should be in rows with rownames and cells in columns with colnames. Normalized data are required as input for CellChat analysis (for example, library-size normalization and then log-transformed with a pseudocount of 1). If users input raw count data, CellChat provides a '`normalizeData`' function for normalization. For the cell group information, a dataframe with rownames is required. Alternatively, users can use a Seurat, SingleCellExperiment or AnnData object as input.

Inference of cell–cell communication networks

To identify strong or weak cell–cell communications, users can modify the parameters '`type`' and '`trim`' in the function '`computeCommunProb`' when inferring cell–cell communication networks. The parameter '`type`' is the method for computing the average gene expression per cell group. By default, CellChat uses a statistically robust mean method by setting '`type = "triMean"`', producing fewer but stronger interactions. When setting '`type = "truncatedMean"`', a value should be assigned to another parameter '`trim`', producing more interactions. However, we find that CellChat performs well at predicting stronger interactions, which is helpful for identifying interactions for further experimental validations. The '`trimean`' approximates 25% truncated mean, implying that the average gene expression is zero if the percentage of expressing cells in one group is less than 25%. To identify weak signaling, users should use '`truncatedMean`'. In general, users can use 10% truncated mean by setting '`type = "truncatedMean"`' and '`trim = 0.1`'. To determine a proper value of '`trim`', CellChat provides a function '`computeAveExpr`', which can help to check the average expression of signaling genes of interest. Therefore, if well-known signaling events in the studied biological process are not predicted, users can try '`truncatedMean`' with lower values of '`trim`' to change the method for calculating the average gene expression per cell group.

Visualization of cell–cell communication networks

Upon inferring the cell–cell communication networks, CellChat provides various ways to visualize such networks, including hierarchical plots, circle plots, chord diagrams, heatmap and bubble plots. In hierarchical plots, circle plots and chord diagrams, edge colors are consistent with the sources as sender, and edge weights are proportional to the interaction strength. Thicker edge lines indicate a stronger signal. One can visualize the inferred communication network of signaling pathways using '`netVisual_aggregate`' and visualize the inferred communication networks of individual L–R pairs associated with that signaling pathway using '`netVisual_individual`'.

Hierarchical plots consist of two components: the left portion shows autocrine and paracrine signaling to certain cell groups of interest, and the right portion shows autocrine and paracrine signaling to the remaining cell groups in the dataset. Thus, a hierarchical plot provides an informative and intuitive way to visualize autocrine and paracrine signaling communications between cell groups of interest. In the hierarchical plot, solid and open circles represent the sources and targets, respectively.

In addition to creating a chord diagram using 'netVisual_aggregate' or 'netVisual_individual', CellChat provides another two functions with more adjustable parameters for better visualization. '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 L–Rs or signaling pathways (where each sector in the chord diagram is a ligand, receptor or signaling pathway). In the chord diagram, the inner thinner bar colors represent the targets that receive signals from the corresponding outer bar. The inner bar size is proportional to the signal strength received by the targets. Such an inner bar is helpful for interpreting the complex chord diagram.

Systematic analysis of cell–cell communication

To facilitate the interpretation of complex intercellular communication networks, CellChat quantitively measures networks through methods abstracted from graph theory, pattern recognition and manifold learning. It can determine major signaling sources and targets, as well as mediators and influencers within a given signaling network, using centrality measures from network analysis. It can also predict key incoming and outgoing signals for specific cell types as well as coordinated responses among different cell types by leveraging pattern recognition approaches. Finally, it can group signaling pathways by defining similarity measures and performing manifold learning from both functional and topological perspectives.

CellChat identifies dominant senders, receivers, mediators and influencers in the intercellular communication network using measures in weighted-directed networks, including out-degree, in-degree, flow betweenness and information centrality[5,45], respectively. In a weighted directed network with the weights as the computed communication probabilities, the outdegree (computed as the sum of communication probabilities of the outgoing signaling from a cell group) and the in-degree (computed as the sum of the communication probabilities of the incoming signaling to a cell group) can be used to identify the dominant cell senders and receivers of signaling networks, respectively. CellChat also provides another intuitive way to visualize the dominant senders (sources) and receivers (targets) in a two-dimensional (2D) space using the function 'netAnalysis_signalingRole_scatter'. In this plot, the x axis and y axis are, respectively, the total outgoing or incoming communication probability associated with each cell group. Dot size is proportional to the number of inferred links (both outgoing and incoming) associated with each cell group. The dot colors indicate different cell groups. The dot shapes indicate different categories of cell groups if the parameter 'group' is defined.

CellChat predict key incoming and outgoing signals for specific cell types using the function 'netAnalysis_signalingRole_heatmap'. In this heatmap, colorbar represents the relative signaling strength of a signaling pathway across cell groups. The top-colored bar plot shows the total signaling strength of a cell group by summarizing all signaling pathways displayed in the heatmap. The right bar plot shows the total signaling strength of a signaling pathway by summarizing all cell groups displayed in the heatmap.

CellChat employs a pattern recognition method to identify global communication patterns. For outgoing (or incoming) patterns, the cell group pattern indicates how these cell groups coordinate to send (or receive) signals and the signaling pathway pattern indicates how these signaling pathways work together to send (or receive) signals. To intuitively show the associations of latent patterns with cell groups and signaling pathways or L–R pairs, we used a river (alluvial) plot. As the number of patterns increases, there might be redundant patterns, making it difficult to interpret the communication patterns. In addition, CellChat also provides the function 'selectK' to infer the number of patterns, which is based on two metrics including Cophenetic and Silhouette. Both metrics measure the stability for a particular number of patterns based on a hierarchical clustering of the consensus matrix. A suitable number of patterns is the one at which Cophenetic and Silhouette values begin to drop suddenly.

CellChat can quantify the similarity between all significant signaling pathways and then group them based on their cellular communication network similarity. This analysis is helpful to predict putative functions of the poorly studied pathways based on their similarity to pathways with well-known functions. Signaling pathways can be grouped based on their functional similarity or structural similarity. A high degree of functional similarity indicates the major senders and receivers are similar and can be interpreted as the two signaling pathways or two L–R pairs exhibiting similar and/or redundant roles. A structural similarity relates to signaling network structure, without considering the similarity of senders and receivers. To obtain a manifold embedding of all inferred communication networks and further intuitively visualize these networks in a 2D space, we first compute the pairwise functional or topological similarity between any pair of inferred networks, then smooth the similarity matrix using a shared nearest-neighbor graph, and finally perform a uniform manifold approximation and projection (UMAP) on the smoothed similarity matrix.

Comparative analysis of cell–cell communication

CellChat provides versatile functionalities to allow systematic comparisons of cell–cell communication between different conditions. Here we present two examples of how we design the comparative analysis. CellChat shows the differential number of interactions or interaction strengths between pairs of scRNA-seq datasets in greater detail using the function '`netVisual_heatmap`'. In this heatmap, the top-colored bar plot represents the sum of each column of the absolute values displayed in the heatmap (incoming signaling). The right-colored bar plot represents the sum of each row of the absolute values (outgoing signaling). Therefore, the bar height indicates the degree of change in terms of the number of interactions or interaction strength between the two conditions. The colorbar indicates increased (or decreased) signaling in the second dataset compared to the first one.

CellChat performs joint manifold learning and classification of all inferred communication networks across different conditions. The manifold embeddings are obtained by first computing the pairwise functional or topological similarity between any pair of inferred networks and then performing UMAP on a shared nearest neighbor-smoothed similarity matrix. UMAP is used for visualizing signaling relationships and interpreting our signaling outputs in an intuitive way without requiring classification of conditions. By quantifying the similarity between the cellular communication networks of signaling pathways across conditions, this analysis highlights the potentially altered signaling pathways. CellChat adopts the concept of network rewiring from network biology and is based on the hypothesis that the difference between different communication networks may affect biological processes across conditions. Furthermore, CellChat identifies the signaling networks with larger differences across conditions based on their Euclidean distance in the 2D UMAP space. CellChat computes and visualizes this Euclidean distance using the function '`rankSimilarity`'. Larger distance implies larger difference of the communication networks between two datasets in terms of either functional or structure similarity. CellChat only computes the distance of overlapping signaling pathways between two datasets. Those signaling pathways that are only identified in one dataset are not included in this analysis. If there are more than three datasets, you can do pairwise comparisons by modifying the parameter '`comparison`' in '`rankSimilarity`'.

Materials

Equipment

Hardware

- Any desktop workstation or laptop with an Internet connection is sufficient. This protocol was run on a MacBook Pro (MacOS Ventura Monterey, Version 13.5) with a 12-Core central processing unit (CPU) and 64 GB of random-access memory (RAM). For minimal performance, we recommend using a dual-core CPU with at least 16 GB of RAM for analyses

Software

- Operating system: Linux, Windows (10) or MacOS
- RStudio: an integrated development environment for R, which can be accessed at https://posit.co/download/rstudio-desktop/
- CellChat: the actively maintained open-source program is freely available at https://github.com/jinworks/CellChat

Data files

Required input data:

- Gene expression data matrix
- User-assigned cell labels

Example datasets: example datasets for running this protocol can be downloaded from the open-access repository figshare at https://figshare.com/projects/Example_data_for_cell-cell_communication_analysis_using_CellChat/157272.

Equipment setup

Installation of CellChat package

We recommend that users install CellChat and perform analysis in RStudio. In an RStudio environment, the following commands can be run from an R script or directly in the built-in R console. The R commands are the same on MacOS, Linux and Windows.

(Optional) Install RStudio. RStudio can be manually installed by downloading RStudio from its official website at https://posit.co/download/rstudio-desktop/.

1. (Optional) Install the devtools package from the Comprensive R Archive Network.

```
install.packages('devtools')
```

2. Install CellChat R packages from our GitHub repository by typing the following commands:

```
devtools::install_github("jinworks/CellChat")
```

Procedure 1: inferring cell–cell communication from a single scRNA-seq dataset

● **TIMING 4 min**

▲ **CRITICAL** Procedure 1 demonstrates the R commands needed to run the CellChat package for inferring and analyzing cell–cell communication from a single scRNA-seq dataset. The equivalent online version, along with the graphical plots, are available in the tutorial directory of the CellChat github repository (https://htmlpreview.github.io/?https://github.com/jinworks/CellChat/blob/master/tutorial/CellChat-vignette.html).

Data input and preprocessing

● **TIMING ~12 s**

▲ **CRITICAL** The example dataset containing single-cell data and cell metadata can be accessed directly from figshare via the following link: https://figshare.com/articles/dataset/scRNA-seq_data_of_human_skin_from_patients_with_atopic_dermatitis/24470719. Users can refer to the online tutorial of the CellChat github repository (https://htmlpreview.github.io/?https://github.com/jinworks/CellChat/blob/master/tutorial/Interface_with_other_single-cell_analysis_toolkits.html) for further details on preparing the input data for CellChat analysis.

1. Prepare the input data by following option A when the normalized count data and metadata are available, option B when the Seurat object is available, option C when the SingleCellExperiment object is available and option D when the Anndata object is available.

   (A) **Generate data input starting from a count data matrix:**

   (i) Upload the count data matrix in a .rda or other format:

```
load("./tutorial/data_humanSkin_CellChat.rda")
```

   (ii) Obtain the normalized data matrix:

```
data.input = data_humanSkin$data
```

(iii) Generate a data frame with rown ames containing cell meta data:

```
meta = data_humanSkin$meta
```

(iv) Subset the data from one condition for further analysis:

```
cell.use = rownames(meta)[meta$condition == "LS"]
data.input = data.input[, cell.use]
meta = meta[cell.use,]
```

(B) **Generate data input starting from a Seurat object:**

(i) Obtain the normalized data matrix:

```
data.input <- seurat_object[["RNA"]]@data
```

(ii) Generate a data frame with row names containing cell meta data:

```
labels <- Seurat::Idents(seurat.obj)
meta <- data.frame(labels = labels, row.names = names(labels))
```

(C) **Generate data input starting from a SingleCellExperiment object:**

(i) Obtain the normalized data matrix:

```
data.input <- SingleCellExperiment::logcounts(sce_object)
```

(ii) Generate a data frame with row names containing cell meta data:

```
meta <- as.data.frame(SingleCellExperiment::colData(sce_object))
meta$labels <- meta[["sce.clusters"]]
```

(D) **Generate data input starting from an Anndata object:**

(i) Upload the Anndata object using the anndata R package:

```
install.packages("anndata")
library(anndata)
ad <- read_h5ad("scanpy_object.h5ad")
```

(ii) Obtain the count data matrix:

```
counts <- t(as.matrix(ad$X))
```

(iii) Normalize the count data matrix:

```
data.input <- normalizeData(counts)
```

(iv) Generate a data frame with row names containing cell meta data:

```
meta <- ad$obs
meta$labels <- meta[["ad_clusters"]]
```

2. Using the '`createCellChat`' function and the input data files generated in Step 1, create a CellChat object by following option A if taking the digital gene expression matrix and cell label information as input, option B if taking a Seurat object as input, option C if taking a SingleCellExperiment object as input and option D if taking a AnnData object as input.

Users should refer to the 'Required input data' section in the 'Experimental design' section for further details.

(A) **Create a CellChat object from the digital gene expression matrix and cell label information**

```
library(CellChat)
cellchat <- createCellChat(object = data.input, meta = meta,
group.by = "labels")
```

(B) **Create a CellChat object from a Seurat object**

```
library(CellChat)
cellchat <- createCellChat(object = seurat.obj, group.by =
"ident", assay = "RNA")
```

(C) **Create a CellChat object from a SingleCellExperiment object**

```
library(CellChat)
cellchat <- createCellChat(object = sce.obj, group.by = "sce.clusters")
```

(D) **Create a CellChat object from an AnnData object**

(i) Convert the Anndata object to the SingleCellExperiment object using the zellkonverter R package:

```
sce <- zellkonverter::readH5AD(file = "adata.h5ad")
assayNames(sce)
```

(ii) Obtain the count data matrix:

```
counts <- assay(sce, "X")
```

(iii) Normalize the count data matrix and add a new assay entry 'logcounts' if not available:

```
logcounts(sce) <- normalizeData(counts)
```

(iv) Generate a CellChat object from a SingleCellExperiment object:

```
cellchat <- createCellChat(object = sce, group.by = "sce.clusters")
```

◆ **TROUBLESHOOTING**

3. (Optional) If cell meta information is not added when creating the CellChat object (Step 2A), use the '`addMeta`' function to add it and the '`setIdent`' function to assign the cell identities to each cell.

```
cellchat <- addMeta(cellchat, meta = meta)
cellchat <- setIdent(cellchat, ident.use = "labels")
```

4. Before running CellChat to infer cell–cell communication, select the L–R interaction database relevant to the study (for example, use the database `CellChatDB.human` when analyzing human samples or the database `CellChatDB.mouse` when analyzing mouse samples):

```
CellChatDB <- CellChatDB.human
showDatabaseCategory(CellChatDB)
dplyr::glimpse(CellChatDB$interaction)
```

5. Select the L–R pairs from the selected CellChatDB database in Step 4 for cell–cell communication analysis. Use option A to select a subset of the database, option B to exclude nonprotein signaling and option C to select all CellChatDB database.
   (A) Select a specific subset of the CellChatDB database

   ```
   CellChatDB.use <- subsetDB(CellChatDB, search = "Secreted
   Signaling")
   ```

   (B) Select all CellChatDB database except for nonprotein signaling

   ```
   CellChatDB.use <- subsetDB(CellChatDB)
   ```

   (C) Select all CellChatDB database

   ```
   CellChatDB.use <- CellChatDB
   ```

6. Set the selected database from Step 5 in the object and then subset the expression data matrix using genes relevant to the selected L–R pairs. An updated CellChat object is outputted with two updated slots 'DB' and 'data.signaling'.

   ```
   cellchat@DB <- CellChatDB.use
   cellchat <- subsetData(cellchat)
   ```

7. Identify over-expressed ligands or receptors in each cell group to infer the cell state-specific communications.

   ```
   future::plan("multisession", workers = 4)
   cellchat <- identifyOverExpressedGenes(cellchat)
   ```

8. For each overexpressed ligand and receptor obtained in Step 7, identify over-expressed L–R interactions if either its associated ligand or receptor is over expressed:

   ```
   cellchat <- identifyOverExpressedInteractions(cellchat)
   ```

9. (Optional) Smooth the data when analyzing single-cell data with shallow sequencing depth. The smoothed data could help to reduce the dropout effects of signaling genes, particularly for possible zero expression of subunits of ligands or receptors. Use a built-in protein–protein interaction (PPI) network from CellChat package to smooth the data:

   ```
   cellchat <- smoothData(cellchat, adj = PPI.human)
   ```

Inference of cell–cell communication networks

● **TIMING ~39 s**

10. (Optional) Check the average expression of signaling genes of interest to determine a proper value of 'trim' when well-known signaling events in the studied biological process are not predicted:

    ```
    computeAveExpr(cellchat, features = c("CXCL12","CXCR4"),
    type = "triMean")
    computeAveExpr(cellchat, features = c("CXCL12","CXCR4"),
    type = "truncatedMean", trim = 0.1)
    ```

11. Infer cell–cell communication at a L–R pair level. Users are now ready to infer cell–cell communication by using the following command:

    ```
    cellchat <- computeCommunProb(cellchat, type = "triMean", trim = NULL,
    raw.use = TRUE)
    ```

    ▲ **CRITICAL STEP** Important parameters of the '`computeCommunProb`' function are as follows:
    - `type`: the method for computing the average gene expression per cell group. By default, `type = "triMean"`, producing fewer but stronger interactions. When setting `type = "truncatedMean"`, a value should be assigned to the parameter '`trim`', producing more interactions.
    - `trim`: the fraction (0–0.25) of observations to be trimmed from each end before the mean is computed.
    - `raw.use`: whether to use the raw data (that is, '`object@data.signaling`') or the smoothed data (that is, '`object@data.smooth`'). The default is `TRUE`. Set '`raw.use = FALSE`' to use the smoothed data. When using the smoothed data, the number of inferred interactions clearly increases. However, generally, it only introduces very weak communications.
12. Filter the cell–cell communication, based on the number of cells in each group. By default, the minimum number of cells required in each cell group for cell–cell communication is 10.

    ```
    cellchat <- filterCommunication(cellchat, min.cells = 10)
    ```

13. Infer cell–cell communication at a signaling pathway level. CellChat computes the communication probability at the signaling pathway level by summarizing the communication probabilities of all L–R pairs associated with each signaling pathway. Note that the inferred intercellular communication network of each L–R pair and each signaling pathway is stored in the slots '`cellchat@net`' and '`cellchat@netP`', respectively.

    ```
    cellchat <- computeCommunProbPathway(cellchat)
    ```

14. Calculate the aggregated cell–cell communication network. CellChat calculates the aggregated cell–cell communication network by counting the number of links or summarizing the communication probability across all the cell groups (option A) or a subset of cell groups (option B).
    (A) Perform calculation across all the cell groups

    ```
    cellchat <- aggregateNet(cellchat)
    ```

    (B) Perform calculation across a subset of cell groups

    ```
    sources.use = c("FBN1+ FIB","APOE+ FIB","Inflam. FIB")
    targets.use = c("LC","Inflam. DC","cDC2","CD40LG+ TC")
    cellchat <- aggregateNet(cellchat, sources.use = sources.use,
    targets.use = targets.use)
    ```

15. Export the CellChat object together with the inferred cell–cell communication networks and save them as a .rds file.

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

    ■ **PAUSE POINT** The.rds files can be used later as input data for the visualization (Procedure 1, Steps 16–21) and the analysis (Procedure 1, Steps 22–27) of the cell–cell communication networks, as well as the comparison analysis of the cell–cell communication networks across biological conditions (Procedure 2, Steps 1–4).

Visualization of cell–cell communication networks

● TIMING ~4.8 s

Visualization of cell–cell communication networks of individual signaling pathways

● TIMING ~0.8 s

▲ CRITICAL Users can visualize the inferred communication network of signaling pathways using '`netVisual_aggregate`' (Step 16) and visualize the inferred communication network of individual L–R pairs associated with that signaling pathway using '`netVisual_individual`' (Step 17). All the signaling pathways showing significant communications can be accessed by '`cellchat@netP$pathways`'. Here, we take input of the CXCL signaling pathway as an example.

16. Using the .rds files from Step 15, visualize the inferred communication network of each signaling pathway using circle plot (option A), hierarchy plot (option B), chord diagram (option C) and heat map (option D).

 (A) **Circle plot**

 (i) Access all the signaling pathways showing significant communications:

```
pathways.show.all <- cellchat@netP$pathways
```

 (ii) Select one pathway:

```
pathways.show <- c("CXCL")
```

 (iii) Visualize the inferred communication network using the '`netVisual_aggregate`' function:

```
netVisual_aggregate(cellchat, signaling = pathways.show, layout
= "circle", color.use = NULL, sources.use = NULL, targets.use =
NULL, idents.use = NULL)
```

 ◆ TROUBLESHOOTING

 (B) **Hierarchy plot**

 (i) Access all the signaling pathways showing significant communications:

```
pathways.show.all <- cellchat@netP$pathways
```

 (ii) Select one pathway:

```
pathways.show <- c("CXCL")
```

 (iii) Visualize the inferred communication network using the '`netVisual_aggregate`' function. To study the cell–cell communication between fibroblasts and immune cells, define '`vertex.receiver`' as all fibroblast cell groups:

```
vertex.receiver = seq(1,4)
netVisual_aggregate(cellchat, signaling = pathways.show, layout
= "hierarchy", vertex.receiver = vertex.receiver)
```

 ▲ CRITICAL The key parameter for this plot is '`vertex.receiver`', a numeric vector giving the index of the cell groups as targets in the left part of the hierarchy plot.

 (C) **Chord diagram**

 (i) Create a chord diagram using the universal function '`netVisual_aggregate`'.

```
pathways.show <- c("CXCL")
par(mfrow=c(1,1))
netVisual_aggregate(cellchat, signaling = pathways.show,
layout = "chord")
```

(ii) Customize a chord diagram using 'netVisual_chord_cell' to flexibly visualize cell–cell communication with different purposes and at different levels. For example, define a named char vector 'group' to create multiple-group chord diagram, for example, grouping cell clusters into different cell types.

```
par(mfrow=c(1,1))
group.cellType <- c(rep("FIB", 4), rep("DC", 4), rep("TC", 4))
names(group.cellType) <- levels(cellchat@idents)
netVisual_chord_cell(cellchat, signaling = pathways.show, group
= group.cellType, title.name = paste0(pathways.show, " signaling
network"))
```

(D) **Heat map plot**

```
pathways.show <- c("CXCL")
par(mfrow=c(1,1))
netVisual_heatmap(cellchat, signaling = pathways.show,
color.heatmap = "Reds")
```

Visualization of cell–cell communication networks of individual L–R pairs

● TIMING ~0.5 s

▲ CRITICAL CellChat can compute the contribution of each associated L–R pair within a particular signaling pathway (Step 17) and then visualize the cell–cell communication mediated by a single L–R pair.

17. Using the .rds files from Step 15, compute and visualize the contribution of each associated L–R pair within a particular signaling pathway:

```
netAnalysis_contribution(cellchat, signaling = pathways.show)
```

18. Extract all the significant L–R pairs for a given signaling pathway:

```
pairLR.CXCL <- extractEnrichedLR(cellchat, signaling = pathways.show,
geneLR.return = FALSE)
```

19. Select one L–R pair to visualize the inferred cell–cell communication network using the 'netVisual_individual' function. Users can also visualize the inferred network using other functions (Step 16), such as 'netVisual_chord_cell' and 'netVisual_heatmap'.

```
LR.show <- pairLR.CXCL[1,]
netVisual_individual(cellchat, signaling = pathways.show, pairLR.use =
LR.show, layout = "circle")
```

Visualization of cell–cell communication mediated by multiple L–R or signaling pathways

● TIMING ~2 s

20. Using the .rds files from Step 15, visualize cell–cell communication mediated by multiple L–R or signaling pathways. Visualize all the significant interactions using a bubble plot (option A) or chord diagram (option B).

(A) **Visualize the inferred significant interactions using a Bubble plot**

(i) Show all the significant interactions from some cell groups defined by 'sources.use' to other cell groups defined by 'targets.use'. By default, the *x* axis first sorts cell group pairs based on the appearance of signaling sources in 'sources.use', and then based on the appearance of signaling targets in 'targets.use'. To change this order, refer to (iv–vii).

```
netVisual_bubble(cellchat, sources.use = 4, targets.use =
c(5:11), remove.isolate = FALSE)
```

(ii) Show all the significant interactions associated with certain signaling pathways defined by 'signaling':

```
netVisual_bubble(cellchat, sources.use = 4, targets.use =
c(5:11), signaling = c("CCL","CXCL"), remove.isolate = FALSE)
```

(iii) Show all the significant interactions associated with certain L–R pairs defined by 'pairLR.use':

```
pairLR.use <- extractEnrichedLR(cellchat, signaling =
c("CCL","CXCL","FGF"))
netVisual_bubble(cellchat, sources.use = c(3,4), targets.use =
c(5:8), pairLR.use = pairLR.use, remove.isolate = TRUE)
```

(iv) Show all the significant interactions by sorting cell group pairs based on the defined 'targets.use'

```
netVisual_bubble(cellchat, targets.use = c("LC","Inflam.
DC","cDC2","CD40LG+ TC"), pairLR.use = pairLR.use, remove.
isolate = TRUE, sort.by.target = T)
```

(v) Show all the significant interactions by sorting cell group pairs based on the defined 'sources.use':

```
netVisual_bubble(cellchat, sources.use = c("FBN1+ FIB","APOE+
FIB","Inflam. FIB"), pairLR.use = pairLR.use, remove.isolate =
TRUE, sort.by.source = T)
```

(vi) Show all the significant interactions by sorting cell group pairs based on the defined 'sources.use' and then 'targets.use':

```
netVisual_bubble(cellchat, sources.use = c("FBN1+ FIB",
"APOE+ FIB","Inflam. FIB"), targets.use = c("LC","Inflam.
DC","cDC2","CD40LG+ TC"), pairLR.use = pairLR.use, remove.
isolate = TRUE, sort.by.source = T, sort.by.target = T)
```

(vii) Show all the significant interactions by sorting cell group pairs based on the defined 'targets.use' and then 'sources.use':

```
netVisual_bubble(cellchat, sources.use = c("FBN1+ FIB",
"APOE+ FIB","Inflam. FIB"), targets.use = c("LC","Inflam.
DC","cDC2","CD40LG+ TC"), pairLR.use = pairLR.use, remove.isolate
= TRUE, sort.by.source = T, sort.by.target = T, sort.by.source.
priority = FALSE)
```

▲ **CRITICAL STEP** Important parameters of the `netVisual_bubble` function are as follows:

- slot.name: the slot name of object: 'netP' or 'net'. Use 'netP' to analyze cell–cell communication at the level of signaling pathways and 'net' to analyze cell–cell communication at the level of L–R pairs
- sources.use: a vector giving the index or the name of source cell groups
- targets.use: a vector giving the index or the name of target cell groups
- signaling: a character vector giving the name of signaling pathways of interest
- pairLR.use: a data frame consisting of one column named either 'interaction_name' or 'pathway_name', defining the interactions of interest and the order of L–R on the *y* axis
- remove.isolate: whether to remove the entire empty columns, that is, communication between certain cell groups
- sort.by.source, sort.by.target, sort.by.source.priority: reorder the interacting cell pairs

(B) **Visualize the inferred significant interactions using a chord diagram**

(i) Show all the L–R mediated interactions sending from 'Inflam.FIB' defined by 'sources.use':

```
netVisual_chord_gene(cellchat, sources.use = 4, targets.use =
c(5:11), lab.cex = 0.5,legend.pos.y = 30)
```

(ii) Show all the L–R mediated interactions received by 'Inflam.DC' defined by 'targets.use':

```
netVisual_chord_gene(cellchat, sources.use = c(1,2,3,4),
targets.use = 8, legend.pos.x = 15)
```

(iii) Show all the L–R mediated interactions associated with certain signaling pathways defined by 'signaling':

```
netVisual_chord_gene(cellchat, sources.use = c(1,2,3,4),
targets.use = c(5:11), signaling = c("CCL","CXCL"),
legend.pos.x = 8)
```

(iv) Show all the signaling pathways mediated interactions by setting 'slot.name' as 'netP':

```
netVisual_chord_gene(cellchat, sources.use = c(1,2,3,4),
targets.use = c(5:11), slot.name = "netP", legend.pos.x = 10)
```

▲ **CRITICAL STEP** Important parameters of the '`netVisual_chord_gene`' function are as follows:

- slot.name: the slot name of object: 'netP' or 'net'. Use 'netP' to visualize cell–cell communication at the level of signaling pathways and 'net' to visualize cell–cell communication at the level of L–R pairs
- signaling: a character vector giving the name of signaling networks
- pairLR.use: a data frame consisting of one column named either 'interaction_name' or 'pathway_name', defining the interactions of interest
- net: a data frame consisting of the interactions of interest. 'net' needs to have at least three columns: 'source', 'target' and 'interaction_name' when visualizing links at the level of ligands/receptors; 'source', 'target' and 'pathway_name' when visualizing links at the level of signaling pathway; 'interaction_name' and 'pathway_name' must be the matched names in 'CellChatDB$interaction'
- sources.use: a vector giving the index or the name of source cell groups

- targets.use: a vector giving the index or the name of target cell groups
- color.use: colors for the cell groups
- lab.cex: font size for the text
- small.gap: small gap between sectors; if the gene names are overlapping, users can adjust the argument 'small.gap' by decreasing their values
- big.gap: gap between the different sets of sectors, which are defined in the 'group' parameter

Visualization of signaling gene expression distribution

● TIMING ~1.5 s

21. Visualize signaling gene expression distribution. Once the Seurat R package has been installed, CellChat can plot the gene expression distribution of signaling genes related to L–R pairs or signaling pathways using a Seurat wrapper function '`plotGeneExpression`' (option A). This function provides three types of visualization, including '`violin`', '`dot`' and '`bar`'. Alternatively, users can extract the signaling genes related to the inferred L–R pairs or signaling pathway using the function '`extractEnrichedLR`' and then plot gene expression using Seurat or other packages like Scanpy (option B).

    (A) **Visualize signaling gene expression using CellChat built-in function**

    ```
    plotGeneExpression(cellchat, signaling = "CXCL", enriched.only = TRUE,
    type = "violin")
    ```

    (B) **Visualize signaling gene expression using Seurat package**

    ```
    genes.use <- extractEnrichedLR(cellchat, signaling = "CXCL",
    geneLR.return = TRUE)$geneLR
    Seurat::VlnPlot(seu_obj, features = genes.use)
    ```

Systematic analysis of cell–cell communication

● TIMING ~3 min

Identify the signaling roles and major contributing signaling events of cell groups

● TIMING 2 s

22. Compute the network centrality scores of the inferred cell–cell communication network.

    ```
    cellchat <- netAnalysis_computeCentrality(cellchat, slot.name = "netP")
    ```

    Important parameters of '`netAnalysis_computeCentrality`' are as follows:

    - slot.name: the slot name of object that is used to compute centrality measures of signaling networks. Setting `slot.name = "netP"` to compute the network centrality scores at the level of signaling pathways and setting `slot.name = "net"` to compute the network centrality scores at the level of L–R pairs.

23. Identify the signaling roles of cell groups by visualizing the centrality scores on a heat map (option A) and a 2D plot (option B). Alternatively, identify the major contributing signaling events (that is, which signals contribute the most to outgoing or incoming signaling of certain cell groups) by following option C.

    (A) **Visualize the network centrality scores on a heat map**

    ```
    netAnalysis_signalingRole_network(cellchat, signaling = pathways.show,
    width = 8, height = 2.5, font.size = 10)
    ```

    (B) **Visualize dominant senders (sources) and receivers (targets) in a 2D space**

    ```
    netAnalysis_signalingRole_scatter(cellchat)
    ```

Important parameters of 'netAnalysis_signalingRole_scatter' are as follows:

- signaling: a char vector to specify signaling pathway names of interest. signaling = NULL: signaling role analysis on the aggregated cell–cell communication network from all signaling pathways
- color.use: defining the color for each cell group
- slot.name: the slot name of object that is used to compute centrality measures of signaling networks
- group: a vector to categorize the cell groups, for example, categorize the cell groups into two major categories: immune cells and fibroblasts
- dot.size: a range defining the size of the symbol, which is proportional to the number of inferred links (both outgoing and incoming) associated with each cell group
- x.measure: The measure used as *x* axis. This measure should be one of 'names(slot(object, slot.name)$centr[[1]])' computed from 'netAnalysis_computeCentrality'. Default = "outdeg" is the weighted outgoing links (i.e., outgoing interaction strength). If setting as "outdeg_unweighted", it represents the total number of outgoing signaling
- y.measure: The measure used as *y* axis. This measure should be one of 'names(slot(object, slot.name)$centr[[1]])' computed from 'netAnalysis_computeCentrality'. Default = "indeg" is the weighted incoming links (i.e., incoming interaction strength). If setting as "indeg_unweighted", it represents the total number of incoming signaling

(C) **Identify the major contributing signaling events of each cell group**

(i) Identify the major outgoing signaling events

```
ht1 <- netAnalysis_signalingRole_heatmap(cellchat, pattern =
"outgoing")
ht1
```

(ii) Identify the major incoming signaling events

```
ht2 <- netAnalysis_signalingRole_heatmap(cellchat, pattern =
"incoming")
ht2
```

(iii) Show the major outgoing and incoming signaling events together

```
ht1 + ht2
```

Important parameters of 'netAnalysis_signalingRole_heatmap' are as follows:

- signaling: a character vector giving the name of signaling networks
- pattern: 'outgoing', 'incoming' or 'all'. When pattern = "all", it aggregates the strength of outgoing and incoming signaling events together
- slot.name: the slot name of object: 'netP' or 'net'. Use 'netP' to analyze cell–cell communication at the level of signaling pathways and 'net' to analyze cell–cell communication at the level of L–R pairs
- color.use: the character vector defining the color of each cell group

Analysis of global communication patterns

● TIMING 115 s

24. Identify global communication patterns to explore how multiple cell groups and signaling events coordinate together. In addition to exploring detailed communications for individual pathways (Steps 16–23), an important question is how multiple cell groups and signaling pathways coordinate to function. Follow option A to explore outgoing signaling

patterns and reveal how the sender cells (that is, cells as signal source) coordinate with each other and with certain signaling pathways to drive communication. Follow option B to explore incoming signaling patterns and to show how the target cells (that is, cells as signal receivers) coordinate with each other and with certain signaling pathways to respond to incoming signals.

(A) **Identify and visualize outgoing communication patterns of secreting cells**

(i) (Optional) Infer the number of outgoing communication patterns

```
library(NMF)
library(ggalluvial)
selectK(cellchat, pattern ="outgoing")
```

(ii) Identify outgoing communication patterns via matrix factorization of outgoing communication probability

```
nPatterns = 6
cellchat <- identifyCommunicationPatterns(cellchat, pattern
="outgoing", k = nPatterns)
```

(iii) Visualize the associations of latent patterns with cell groups and signaling pathways

```
netAnalysis_river(cellchat, slot.name = "netP", pattern
="outgoing", cutoff = 0.5)
```

Important parameters of ‘`netAnalysis_river`’ are as follows:

- slot.name: the slot name of object: ‘netP’ or ‘net’. Use ‘netP’ to analyze cell–cell communication at the level of signaling pathways and ‘net’ to analyze cell–cell communication at the level of L–R pairs;
- pattern: ‘outgoing’ or ‘incoming’;
- cutoff: the threshold for filtering out weak links.

(iv) Visualize the direct associations of cell groups and signaling pathways:

```
netAnalysis_dot(cellchat, slot.name = "netP", pattern
="outgoing", cutoff = NULL, color.use = NULL, dot.size =
c(1, 3))
```

Important parameters of ‘`netAnalysis_dot`’ are as follows:

- cutoff: the threshold for filtering out weak links. Default is 1/R where R is the number of latent patterns.
- color.use: the character vector defining the color of each cell group
- dot.size: a range defining the size of the symbol. This dot size is proportional to the contribution score of each cell group to each signaling pathway.

(B) **Identify and visualize incoming communication patterns of target cells**

(i) (Optional) Infer the number of incoming communication patterns

```
selectK(cellchat, pattern = "incoming")
```

(ii) Identify outgoing communication patterns via matrix factorization of outgoing communication probability

```
nPatterns = 3
cellchat <- identifyCommunicationPatterns(cellchat, pattern =
"incoming", k = nPatterns)
```

(iii) Visualize the associations of latent patterns with cell groups and signaling pathways

```
netAnalysis_river(cellchat, pattern = "incoming")
```

(iv) Visualize the direct associations of cell groups and signaling pathways:

```
netAnalysis_dot(cellchat, pattern = "incoming")
```

Manifold and classification learning analysis of signaling networks

● **TIMING 35 s**

25. Perform manifold and classification learning analysis of signaling networks to group all significant signaling pathways based on their cellular communication network similarity. Signaling pathways can be grouped based on their functional similarity by following option A, or based on their structural similarity by following option B. Functional similarity analysis is not applicable to multiple datasets with different cell type compositions, whereas structural similarity analysis is applicable to multiple datasets either with the same cell type composition or with vastly different cell type compositions. More detailed information is available in our previous study[5].

(A) **Functional similarity analysis**

(i) Compute the functional similarity between any pair of inferred networks

```
cellchat <- computeNetSimilarity(cellchat, type = "functional")
```

(ii) Perform manifold learning of inferred communication networks

```
cellchat <- netEmbedding(cellchat, type = "functional")
```

(iii) Perform clustering of inferred communication networks

```
cellchat <- netClustering(cellchat, type = "functional")
```

(iv) Visualize inferred communication networks in a 2D space

```
netVisual_embedding(cellchat, type = "functional",
label.size = 3.5)
```

(v) (Optional) Zoom in each group of signaling pathways in a 2D space

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

(B) **Structure similarity analysis**

```
cellchat <- computeNetSimilarity(cellchat, type = "structural")
cellchat <- netEmbedding(cellchat, type = "structural")
cellchat <- netClustering(cellchat, type = "structural")
netVisual_embedding(cellchat, type = "structural", label.size = 3.5)
# netVisual_embeddingZoomIn(cellchat, type = "structural", nCol = 2)
```

26. Explore cell–cell communication interactively through a CellChat Shiny App. For CellChat analysis of single-cell transcriptomics, make sure the '`cellchat@dr`' contains a low-dimensional space of the data such as '`umap`' and '`tsne`' to produce the feature plot of signaling genes.

(i) (Optional) Add a new low-dimensional space of the data if not available

```
cell.embeddings <- read.table("./cellEmbeddings_umap.txt",
row.names = 1, header = T, sep = "\t")
cellChat <- addReduction(object = cellchat, dr = cell.embeddings,
dr.name = "umap")
```

(ii) Run the CellChat Shiny App

```
runCellChatApp(cellchat)
```

27. Export the CellChat object as .rds file as follows:

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

■ **PAUSE POINT** Users can store the .rds files for later use.

Procedure 2: comparative analysis of cell–cell communication from pairs of scRNA-seq datasets

● **TIMING** 30 s

▲ **CRITICAL** In Procedure 2, we showcase CellChat's diverse functionalities for identifying major signaling changes across different biological conditions by quantitative contrasts and joint manifold learning. Here, this ability of CellChat has been demonstrated by applying it to two scRNA-seq datasets 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 different cell population compositions between different conditions, users should refer to Procedure 3.

▲ **CRITICAL** The equivalent online version and the graphical plots are available in the github repository (https://htmlpreview.github.io/?https://github.com/jinworks/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets.html).

Load the CellChat object of each dataset and merge them

● **TIMING** ~3 s

1. Generate a CellChat object for each dataset from NL (or LS) condition, as discussed in Procedure 1, Steps 1–15.
2. (Optional) If the CellChat objects are obtained using the earlier version (<1.6.0), update by running the function 'updateCellChat'.
3. Merge multiple CellChat objects for comparison analysis.

```
library(CellChat)
library(patchwork)
cellchat.NL <- readRDS("./tutorial/cellchat_humanSkin_NL.rds")
cellchat.LS <- readRDS("./tutorial/cellchat_humanSkin_LS.rds")
object.list <- list(NL = cellchat.NL, LS = cellchat.LS)
cellchat <- mergeCellChat(object.list, add.names = names(object.list))
```

4. Export the merged CellChat object and the list of the two separate objects as .RData or .rds files for later use:

```
save(object.list, file = "cellchat_object.list_humanSkin_NL_LS.RData")
save(cellchat, file = "cellchat_merged_humanSkin_NL_LS.RData")
```

■ **PAUSE POINT** The exported data files can be further processed for data visualization at a later date.

Identify altered interactions and cell populations

● **TIMING** ~2 s

▲ **CRITICAL** CellChat employs a top-down approach (that is, starting with the big picture and then refining it in greater detail on the signaling mechanisms) to identify signaling changes at different levels, including altered interactions, cell populations, signaling pathways and L–R pairs.

5. Establish whether the cell–cell communication is enhanced or not by comparing the total number of interactions (option A) and the interaction strength (option B) of the inferred cell–cell communication networks from different biological conditions.

   (A) **Comparing the total number of interactions**

   ```
   gg1 <- compareInteractions(cellchat, show.legend = F,
   group = c(1,2))
   gg1
   ```

   (B) **Comparing the total interaction strength**

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

6. Identify substantially altered interactions between cell populations by comparing the number of interactions and interaction strength among different cell populations. Visualize differential interactions with a circle plot (option A) or a heat map (option B). Options A and B are recommended when working with pairwise datasets. If comparing more than two datasets, use Option C to generate multiple circle plots showing the number of interactions or interaction strength per dataset. Alternatively, examine the differential number of interactions or interaction strength among coarse cell types by aggregating the cell–cell communications based on the defined cell groups (option D).

   (A) **Circle plot showing the differential number of interactions or interaction strengths among different cell populations across two datasets**

   (i) Examine the differential number of interactions

   ```
   netVisual_diffInteraction(cellchat, weight.scale = T)
   ```

   (ii) Examine the differential interaction strengths

   ```
   netVisual_diffInteraction(cellchat, weight.scale = T,
   measure = "weight")
   ```

   ◆ **TROUBLESHOOTING**

   (B) **Heat map showing the differential number of interactions or interaction strengths among different cell populations across two datasets**

   (i) Examine the differential number of interactions

   ```
   netVisual_heatmap(cellchat)
   ```

   (ii) Examine the differential interaction strengths

   ```
   netVisual_heatmap(cellchat, measure = "weight")
   ```

7. **Circle plot showing the number of interactions or interaction strengths among different cell populations across multiple datasets**

   (i) Compute the maximum number of cells per cell group and the maximum number of interactions across all datasets

```
weight.max <- getMaxWeight(object.list, attribute =
c("idents","count"))
```

(ii) Examine the number of interactions between any two cell populations in each dataset

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

8. **Circle plot showing the differential number of interactions or interaction strengths among coarse cell types**
▲ **CRITICAL** To simplify the complicated network and gain insights into the cell–cell communication at the cell type level, CellChat aggregates the cell–cell communication based on the defined cell groups.

(i) Categorize the cell populations into three major cell types

```
group.cellType <- c(rep("FIB", 4), rep("DC", 4), rep("TC", 4))
group.cellType <- factor(group.cellType, levels = c("FIB",
"DC", "TC"))
```

(ii) Remerge the list of CellChat objects based on the defined major cell types

```
object.list <- lapply(object.list, function(x)
{mergeInteractions(x, group.cellType)})
cellchat <- mergeCellChat(object.list, add.names =
names(object.list))
```

(iii) Examine the number of interactions between any two major cell types in each dataset

```
weight.max <- getMaxWeight(object.list, slot.name = c("idents",
"net", "net"), attribute = c("idents","count", "count.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]))
}
```

(iv) Examine the differential number of interactions between any two cell types

```
netVisual_diffInteraction(cellchat, weight.scale = T, measure =
"count.merged", label.edge = T)
```

(v) Examine the differential interaction strengths between any two cell types

```
netVisual_diffInteraction(cellchat, weight.scale = T, measure =
"weight.merged", label.edge = T)
```

9. Compare major sources and targets in a 2D space. Identify cell populations with notable changes in sending or receiving signals between different datasets by following option A, or identify the signaling changes of specific cell populations by following option B.
   (A) **Identify cell populations with notable changes in sending or receiving signals**
      (i) Compute the maximum and minimum number of interactions across all 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))
```

      (ii) Visualize the number of outgoing and incoming interactions of each cell population in a 2D space. See Step 23 in Procedure 1 for the detailed description of the important parameters of '`netAnalysis_signalingRole_scatter`'

```
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)
}
patchwork::wrap_plots(plots = gg)
```

◆ **TROUBLESHOOTING**

   (B) **Identify the signaling changes of specific cell populations**

```
netAnalysis_signalingChanges_scatter(cellchat, idents.use =
"Inflam. DC", signaling.exclude = "MIF")
```

Identify altered signaling with distinct network architecture

● **TIMING ~15 s**

10. Identify signaling pathways with larger cell–cell communication network differences across different conditions based on the functional or structure similarity. More detailed information of the functional and structural similarity is described in Procedure 1, Step 25.
      (i) Compute the functional similarity between any pair of inferred networks

```
cellchat <- computeNetSimilarityPairwise(cellchat, type =
"functional")
```

      (ii) Perform joint manifold learning of inferred communication networks across different conditions

```
cellchat <- netEmbedding(cellchat, type = "functional")
```

      (iii) Perform joint clustering of inferred communication networks across different conditions

```
cellchat <- netClustering(cellchat, type = "functional")
```

      (iv) Visualize inferred communication networks in a 2D space

```
netVisual_embedding(cellchat, type = "functional", label.size = 3.5)
```

      (v) (Optional) Zoom in each group of signaling pathways in a 2D space

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

(vi) Compute and visualize the pathway distance in the learned joint manifold

```
rankSimilarity(cellchat, slot.name = "netP", type =
"functional", comparison1 = NULL, comparison2 = c(1,2))
```

Important parameters of '`rankSimilarity`' are as follows:

- slot.name: 'netP' or 'net', the slot name of object that is used to compute the distance of signaling networks; setting `slot.name = "netP"` to compare the distance of signaling networks at the level of signaling pathways and setting `slot.name = "net"` to compute the distance of signaling networks at the level of L–R pairs.
- type: 'functional' or 'structural'. 'functional' (or 'structural') means calculation of network differences based on the functional (or structure) similarity.
- comparison1: a numerical vector giving the datasets for comparison. This should be the same as 'comparison' in 'computeNetSimilarityPairwise';
- comparison2: a numerical vector with two elements giving the datasets for comparison.

If there are more than 2 datasets defined in 'comparison1', 'comparison2' can be defined to indicate which two datasets used for computing the distance. For example, comparison2 = c(1,3) indicates the first and third datasets defined in 'comparison1' will be used for comparison.

Identify altered signaling with distinct interaction strength

● **TIMING ~4 s**

11. By comparing the information flow/interaction strength of each signaling pathway, CellChat identifies signaling pathways that: (1) turn off, (2) decrease, (3) turn on or (4) increase, by changing their information flow at one condition as compared with another condition. Identify altered signaling pathways (or L–R pairs) with distinct interaction strength based on the overall information flow (option A) or based on the outgoing (or incoming) signaling patterns (option B).

(A) **Compare the overall information flow of each signaling pathway or L–R pair**

(i) Compare the information flow for each signaling pathway using a stacked bar chart

```
rankNet(cellchat, slot.name = "netP", mode = "comparison",
measure = "weight", sources.use = NULL, targets.use = NULL,
stacked = T, do.stat = FALSE)
```

(ii) Compare the information flow for each signaling pathway by performing a paired Wilcoxon test

```
rankNet(cellchat, mode = "comparison", measure = "weight",
stacked = T, do.stat = TRUE)
```

(iii) Compare the information flow for each L–R pair by performing a paired Wilcoxon test

```
rankNet(cellchat, slot.name = "net", mode = "comparison",
measure = "weight", stacked = T, do.stat = TRUE)
```

(iv) Compare the information flow for each signaling pathway using a grouped bar chart

```
rankNet(cellchat, mode = "comparison", measure = "weight",
stacked = F, do.stat = FALSE)
```

▲**CRITICAL** Important parameters of the `rankNet` function are as follows:

- slot.name: the slot name of object. Setting `slot.name = "netP"` to compare the information flow for each signaling pathway and setting `slot.name = "net"` to compute the information flow for each L–R pair
- measure: 'weight' or "count". Setting `measure = "weight"` to compare the total interaction weights (strength) and setting `measure = "count"` to compare the number of interactions
- mode: 'comparison' or 'single'. Setting `mode = " comparison"` to perform comparison analysis across different datasets; and setting `mode = "single"` to rank the enriched signaling in one dataset
- comparison: a numerical vector giving the datasets for comparison; a single value means ranking for only one dataset and two values means comparison for two datasets
- color.use: the character vector defining the colors of bar charts
- sources.use: a vector giving the index or the name of source cell groups
- targets.use: a vector giving the index or the name of target cell groups
- stacked: whether to plot the stacked bar plot or not. Default = TRUE
- do.stat: whether to do a paired Wilcoxon test to determine whether there is significant difference between two datasets. Default = FALSE
- signaling.type: a char vector giving the types of signaling from the four categories: 'Secreted Signaling', 'ECM-Receptor', 'Cell–Cell Contact' and 'Non-protein Signaling'

(B) **Compare outgoing (or incoming) signaling patterns associated with each cell population**

▲**CRITICAL** The above '`rankNet`' analysis summarizes the information from the outgoing and incoming signaling together. CellChat can also compare the outgoing (or incoming) signaling pattern between two datasets, allowing to identify signaling pathways/L–R that exhibit different signaling patterns.

(i) Load the required R package for generating heat map plots

```
library(ComplexHeatmap)
```

(ii) Combine all the identified signaling pathways from different datasets

```
pathway.union <- union(object.list[[1]]@netP$pathways,
object.list[[2]]@netP$pathways)
```

(iii) Assign the contribution of signaling pathways to each cell group within each dataset in terms of outgoing interaction strengths

```
ht1 = netAnalysis_signalingRole_heatmap(object.list[[1]],
pattern = "outgoing", signaling = pathway.union, title =
names(object.list)[1], width = 5, height = 6)
ht2 = netAnalysis_signalingRole_heatmap(object.list[[2]],
pattern = "outgoing", signaling = pathway.union, title =
names(object.list)[2], width = 5, height = 6)
```

(iv) Compare the heat map plots side by side for different datasets

```
draw(ht1 + ht2, ht_gap = unit(0.5, "cm"))
```

▲**CRITICAL** Important parameters of the '`netAnalysis_signalingRole_heatmap`' function are as follows:

- signaling: a character vector giving the names of signaling networks of interest
- pattern: this parameter can be set as 'outgoing', 'incoming' or 'all'. When pattern = "all", CellChat aggregates the outgoing and incoming signaling strength together

- slot.name: the slot name of object that is used to examine the signaling patterns at the level of signaling pathways (`slot.name = "netP"`) or L–R pairs (`slot.name = "net"`)
- color.use: the character vector defining the color of each cell group

12. Identify dysfunctional signaling by comparing the communication probabilities (option A) or by differential expression analysis (option B).
    (A) **Identify dysfunctional signaling by comparing the communication probabilities.**
    (i) Compare the communication probabilities mediated by L–R pairs from certain cell groups to other cell groups

```
netVisual_bubble(cellchat, sources.use = 4, targets.use =
c(5:11), comparison = c(1, 2), angle.x = 45)
```

(ii) Identify the up-regulated (that is, increased) L–R pairs in the second dataset compared with the first dataset

```
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)
```

(iii) Identify the down-regulated (i.e., decreased) L–R pairs in the second dataset compared with the first dataset

```
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)
```

▲**CRITICAL** Important parameters of the ‘`netVisual_bubble`’ function for the comparison analysis are as follows:

- sources.use: a vector giving the index or the name of source cell groups
- targets.use: a vector giving the index or the name of target cell groups
- comparison: a numerical vector giving the datasets for comparison in the merged object; e.g., comparison = c(1,2)
- group: a numerical vector giving the group information of different datasets; e.g., group = c(1,2,2)
- max.dataset: a scale, keeping the communications with highest probability in max.dataset (i.e., certain condition)
- min.dataset: a scale, keeping the communications with lowest probability in min.dataset
- color.text.use: whether to color the xtick labels according to the dataset origin when doing comparison analysis
- color.text: the colors for xtick labels according to the dataset origin when doing comparison analysis

(B) Identify dysfunctional signaling by using differential expression analysis. CellChat performs differential expression analysis between two biological conditions (that is, NL and LS) for each cell group, and then obtains the up-regulated and down-regulated interactions based on the fold change of ligands in the sender cells and receptors in the receiver cells. Such analysis can be done as follows:
(i) Define a positive dataset (that is, the dataset with positive fold change against the other dataset) and a variable name used for storing the results of differential expression analysis

```
pos.dataset = "LS"
features.name = pos.dataset
```

(ii) Perform differential expression analysis for each cell group

```
cellchat <- identifyOverExpressedGenes(cellchat, group.dataset =
"datasets", pos.dataset = pos.dataset, features.name = features.
name, only.pos = FALSE, thresh.pc = 0.1, thresh.fc = 0.05)
```

(iii) (Optional) Perform differential expression analysis by ignoring cell group information

```
cellchat <- identifyOverExpressedGenes(cellchat, group.dataset =
"datasets", pos.dataset = pos.dataset, features.name = features.
name, only.pos = FALSE, thresh.pc = 0.1, thresh.fc = 0.05,
group.DE.combined = TRUE)
```

(iv) Map the results of differential expression analysis onto the inferred cell–cell communications to easily subset the L–R pairs of interest

```
net <- netMappingDEG(cellchat, features.name = features.name)
```

(v) Extract the L–R pairs with upregulated ligands in LS

```
net.up <- subsetCommunication(cellchat, net = net, datasets =
"LS",ligand.logFC = 0.05, receptor.logFC = NULL)
```

(vi) Extract the L–R pairs with upregulated ligands and upregulated recetptors in NL, that is,downregulated in LS

```
net.down <- subsetCommunication(cellchat, net = net, datasets =
"NL",ligand.logFC = -0.05, receptor.logFC = NULL)
```

(vii) (Optional) Perform further deconvolution to obtain the individual signaling genes

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

(viii) (Optional) Find all the significant outgoing/incoming/both signaling according to the customized features and cell groups of interest

```
df <- findEnrichedSignaling(object.list[[2]], features =
c("CCL19", "CXCL12"), idents = c("Inflam. FIB", "COL11A1+ FIB"),
pattern ="outgoing")
```

13. Visualize the upregulated signaling events identified from the above Step 12B(v) and downregulated signaling events identified from the above Step 12B(vi) using a bubble plot (option A), chord diagram (option B) or wordcloud (option C). The bubble plot is useful to compare the interaction strengths between pairs of cell groups across multiple datasets in the same plot, the chord diagram is useful to show the interactions and the associated ligands/receptors in one dataset, and the wordcloud is useful to highlight the ligands of the dysfunctional signaling in one dataset.

(A) **Visualize dysfunctional signaling using a bubble plot**

(i) Find all L–R pairs with upregulated signaling strength in the second dataset

```
pairLR.use.up = net.up[, "interaction_name", drop = F]
```

(ii) Visualize the upregulated signaling in the second dataset

```
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]))
```

(iii) Find all L–R pairs with downregulated signaling strength in the second dataset

```
pairLR.use.down = net.down[, "interaction_name", drop = F]
```

(iv) Visualize the downregulated signaling in the second dataset

```
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]))
```

(v) Visualize both the upregulated and downregulated signaling in the second dataset

```
gg1 + gg2
```

(B) **Visualize dysfunctional signaling using a chord diagram**

(i) Visualize the upregulated signaling in the second dataset

```
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]))
```

(ii) Visualize the downregulated signaling in the second dataset

```
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]))
```

(C) **Visualize dysfunctional signaling using a wordcloud plot**

(i) Visualize the enriched ligands in the second dataset

```
computeEnrichmentScore(net.up, species = 'human')
```

(ii) Visualize the enriched ligands in the first dataset

```
computeEnrichmentScore(net.down, species = 'human')
```

Visually compare inferred cell–cell communication networks

● **TIMING** ~6 s

▲ **CRITICAL** Here, we briefly show two examples of how to visually compare inferred cell–cell communication networks using circle plots or heat map plots. More details on alternative visualization options (for example, hierarchy plots and chord diagrams) can be found in Procedure 1, Step 16.

14. Visualize inferred cell–cell communication networks using circle plots (option A) or heat maps (option B).

(A) **Visualize inferred cell–cell communication networks using circle plots**

(i) Compute the maximum interaction strength of a signaling pathway across all datasets

```
pathways.show <- c("CXCL")
weight.max <- getMaxWeight(object.list, slot.name = c("netP"),
attribute = pathways.show)
```

(ii) Visually compare the inferred cell–cell communication network between 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]))
}
```

(B) **Visualize inferred cell–cell communication networks using heat map plots**

(i) Generate a heat map plot of the inferred cell–cell communication network from each dataset

```
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(pathways.
show, "signaling ",names(object.list)[i]))
}
```

(ii) Compare the heat map plots side by side for different datasets

```
ComplexHeatmap::draw(ht[[1]] + ht[[2]], ht_gap = unit(0.5, "cm"))
```

15. Plot the gene expression distribution of signaling genes related to L–R pairs or signaling pathway using the Seurat wrapper function ‘`plotGeneExpression`’.

(i) (Optional) Specify the order of the datasets to appear in the plot

```
cellchat@meta$datasets = factor(cellchat@meta$datasets, levels =
c("NL", "LS"))
```

(ii) Generate a violin plot to compare the gene expression distribution across different datasets

```
plotGeneExpression(cellchat, signaling = "CXCL", split.by =
"datasets", colors.ggplot = T, type = "violin")
```

16. Export the merged CellChat object and the list of the two separate objects as .RData or .rds files.

```
save(object.list, file = "cellchat_object.list_humanSkin_NL_LS.RData")
save(cellchat, file = "cellchat_merged_humanSkin_NL_LS.RData")
```

■ **PAUSE POINT** Users can save the .RData or .rds files for later use.

Procedure 3: comparative analysis of multiple datasets with differing cell type compositions

● **TIMING** ~5 s

▲ **CRITICAL** Procedure 3 demonstrates how to apply CellChat to the comparative analysis of multiple conditions with differing cell type compositions. The equivalent online version along with the graphical plots are available in the github repository (https://htmlpreview.github.io/?https://github.com/jinworks/CellChat/blob/master/tutorial/Comparison_analysis_of_multiple_datasets_with_different_cellular_compositions.html).

▲ **CRITICAL** For datasets with differing cell type (group) compositions, CellChat adjusts the cell groups to the same cell type compositions across all datasets using the function '`liftCellChat`' and then performs comparative analysis as the joint analysis of datasets with the same cell type compositions. Here, we take an example of comparative analysis of two embryonic mouse skin scRNA-seq datasets from days E13.5 and E14.5. There are 11 skin cell populations shared between E13.5 and E14.5 and two additional populations (that is, dermal DC and pericytes) specific to E14.5. Therefore, we lift up the cell groups from E13.5 to the same cell type compositions as E14.5.

1. Load the generated .rds CellChat object of each dataset, which is obtained as described in Procedure 1, Steps 1–15.
2. (Optional) If the CellChat objects are obtained using the earlier version (< 1.6.0), update by running the function '`updateCellChat`'.

```
cellchat.E13 <- readRDS("./tutorial/cellchat_embryonic_E13.rds")
cellchat.E13 <- updateCellChat(cellchat.E13)
cellchat.E14 <- readRDS("./tutorial/cellchat_embryonic_E14.rds")
cellchat.E14 <- updateCellChat(cellchat.E14)
```

3. Lift up CellChat objects and merge them together. Since there are two additional populations specific to E14.5 compared with E13.5, we lift up '`cellchat.E13`' by lifting up the cell groups to the same cell group compositions as E14.5. '`liftCellChat`' only updates the slot related to cell–cell communication network, including slots '`object@net`', '`object@netP`' and '`object@idents`'.

```
group.new = levels(cellchat.E14@idents) # Define the cell labels to
lift up
cellchat.E13 <- liftCellChat(cellchat.E13, group.new)
object.list <- list(E13 = cellchat.E13, E14 = cellchat.E14)
cellchat <- mergeCellChat(object.list, add.names = names(object.list),
cell.prefix = TRUE)
```

4. Once the CellChat object are lifted up and merged together, perform comparative visualization and analysis of cell–cell communication as described for the comparative analysis of multiple datasets with the same cell type compositions (see Procedure 2, Steps 5–15). Below is an example of how to compare the inferred cell–cell communication networks using circle plot:
   (i) Compute the maximum interaction strength of a signaling pathway across all datasets

   ```
   pathways.show <- c("WNT")
   weight.max <- getMaxWeight(object.list, slot.name = c("netP"),
   attribute = pathways.show)
   ```

   (ii) Visually compare the inferred cell–cell communication network between 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]))
    }
```

5. Export the merged CellChat object and the list of the two separate objects as .RData or .rds files as follows:

```
save(object.list, file = "cellchat_object.list_embryonic_E13_E14.RData")
save(cellchat, file = "cellchat_merged_embryonic_E13_E14.RData")
```

■ **PAUSE POINT** The merged CellChat object and the list of the two separate objects can be stored for later use.

Troubleshooting

Troubleshooting advice can be found in Table 1.

Table 1 | Troubleshooting table

Step	Problem	Possible reason	Solution
Procedure 1, Step 2D	RStudio encounters FATAL ERROR when starting from an AnnData object	Compatibility issue of AnnData object between Python and R	Install the anndata R package. In addition, ensure that the required data files 'data.input' and 'meta' or the SingleCellExperiment object are saved in the user's local computer and then try to reload them for CellChat analysis
Procedure 1, Step 16	Error with 'Length of new attribute value...' when using circle plot	Possible issue of the igraph version	User can try degrade igraph form 1.4.0 to 1.3.5; or update the object using updateCellchat(); or reinstall the CellChat R package
Procedure 2, Step 9	Error in 'netAnalysis_computeCentrality' when using the merged CellChat object	'netAnalysis_computeCentrality' was not run on each individual CellChat object	Run 'netAnalysis_computeCentrality' on each CellChat object in the 'object.list' separately and then run the function 'mergeCellChat'

Timing

Procedure 1, Steps 1–27; inferring cell–cell communication from a single scRNA-seq dataset: ~4 min
Steps 1–9, data input and preprocessing: ~12 s
Steps 10–15, inference of cell–cell communication networks: ~39 s
Steps 16–21 visualization of cell–cell communication networks: ~4.8 s
Steps 22–27, systematic analysis of cell–cell communication: ~3 min

Procedure 2, Steps 1–15; comparative analysis of cell–cell communication from pairs of scRNA-seq datasets: 30s
Steps 1–4, load CellChat objects of each dataset and merge them together: ~3 s
Steps 5–9, identify altered interactions and cell populations: ~2 s
Step 10, identify altered signaling with distinct network architecture: ~15 s
Steps 11–13, identify altered signaling with distinct interaction strength: ~4 s
Steps 14–16, visually compare inferred cell–cell communication networks: ~6 s

Procedure 3, Steps 1–5; comparative analysis of multiple datasets with differing cell type compositions: ~5 s

Anticipated results

Running CellChat's inference (Procedure 1, Step 11) of the L–R pair-mediated cell–cell communication produces the communication probability (that is, interaction strength) array and the corresponding P value array, which can be accessed by `object@net$prob` and `object@net$pval`, respectively. Both arrays are three dimensional, where the first, second and third dimensions represent sources, targets and L–R pairs, respectively (Fig. 1). For example, given an inferred cell–cell communication probability array $P(x,y,z)$, $P(x_i,y_j,z_k)$ is the communication probability from cell group x_i to cell group y_j for a given L–R pair z_k. CellChat also infers signaling pathway-mediated cell–cell communication (Procedure 1, Step 13), where the communication probability array can be accessed by `object@netP$prob`. CellChat provides several ways to visualize the inferred cell–cell communication network, including circle plot, hierarchical plot, chord diagram, heatmap and bubble plot. Importantly, users can visualize inferred communication networks of an individual L–R pair, a signaling pathway as well as multiple L–R pairs or signaling pathways. To facilitate the interpretation of the inferred intercellular communication networks within one condition and across different conditions (Fig. 1), CellChat v2 can (1) identify signaling roles of cell groups as well as the major contributing signaling within a given signaling network; (2) predict key incoming and outgoing signals for specific cell types as well as global communication patterns on how multiple cell types and signaling pathways coordinate together; (3) group signaling pathways from both functional and topological perspectives; (4) identify major signaling changes and altered cell populations across different biological conditions using various quantitative metrics and differential expression analysis; and (5) perform comparison analysis across different conditions with differing cell type compositions.

As shown in Fig. 3, the inferred cell–cell communications depend on the method for computing average expression per cell group. The '`triMean`' method produces fewer but stronger interactions, while the '`truncatedMean`' method with a smaller value of the '`trim`' parameter (for example, '`trim = 0.1`') enables the identification of weak signaling. Therefore, if known signaling is not observed, users can use '`truncatedMean`' with lower values of '`trim`' to change the method for calculating the average gene expression per cell group.

Finally, CellChat v2 allows users to visualize and explore the cell–cell communication analysis interactively by defining various analysis parameters (Fig. 4). Briefly, it can visualize cell groups and signaling expression, examine the inferred signaling between different cell groups, and further visualize the individual signaling pathway. A rich user-guided sliders in each panel are provided for flexible exploration, highlight and zoom out of the related information of interest.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The example datasets analyzed in this protocol are all publicly available. The scRNA-seq data used in Procedure 1 and Procedure 2 are available at the Gene Expression Omnibus under accession GSE147424[46]. The scRNA-seq data used in Procedure 3 are available at the Gene Expression Omnibus under accession GSM3453535, GSM3453536, GSM3453537 and GSM3453538[47]. The scRNA-seq data used in Fig. 2 are available at ArrayExpress database under accession E-MTAB-8142[48]. In addition, all the preprocessed datasets and CellChat objects required to reproduce this protocol are publicly available at https://figshare.com/projects/Example_data_for_cell-cell_communication_analysis_using_CellChat/157272 (ref. 49).

Code availability

CellChat v2 is publicly available as an R package. Source codes of the R package and this protocol have been deposited at the GitHub repository (https://github.com/jinworks/CellChat).

Received: 31 July 2023; Accepted: 27 June 2024;
Published online: 16 September 2024

References

1. Shao, X., Lu, X., Liao, J., Chen, H. & Fan, X. New avenues for systematically inferring cell-cell communication: through single-cell transcriptomics data. *Protein Cell* **11**, 866–880 (2020).
2. Armingol, E., Officer, A., Harismendy, O. & Lewis, N. E. Deciphering cell–cell interactions and communication from gene expression. *Nat. Rev. Genet.* **22**, 71–88 (2021).
3. Almet, A. A., Cang, Z., Jin, S. & Nie, Q. The landscape of cell–cell communication through single-cell transcriptomics. *Curr. Opin. Syst. Biol.* **26**, 12–23 (2021).
4. Armingol, E., Baghdassarian, H. M. & Lewis, N. E. The diversification of methods for studying cell-cell interactions and communication. *Nat. Rev. Genet.* **25**, 381–400 (2024).
5. Jin, S. et al. Inference and analysis of cell–cell communication using CellChat. *Nat. Commun.* **12**, 1088 (2021).
6. Kanemaru, K. et al. Spatially resolved multiomics of human cardiac niches. *Nature* **619**, 801–810 (2023).
7. Zhao, W., Johnston, K. G., Ren, H., Xu, X. & Nie, Q. Inferring neuron-neuron communications from single-cell transcriptomics through NeuronChat. *Nat. Commun.* **14**, 1128 (2023).
8. Vu, R. et al. Wound healing in aged skin exhibits systems-level alterations in cellular composition and cell–cell communication. *Cell Rep.* **40**, 111155 (2022).
9. Hao, M., Zou, X. & Jin, S. Identification of intercellular signaling changes across conditions and their influence on intracellular signaling response from multiple single-cell datasets. *Front. Genet.* **12**, 751158 (2021).
10. Dimitrov, D. et al. Comparison of methods and resources for cell–cell communication inference from single-cell RNA-Seq data. *Nat. Commun.* **13**, 3224 (2022).
11. Liu, Z., Sun, D. & Wang, C. Evaluation of cell–cell interaction methods by integrating single-cell RNA sequencing data with spatial information. *Genome Biol.* **23**, 218 (2022).
12. Efremova, M., Vento-Tormo, M., Teichmann, S. A. & Vento-Tormo, R. CellPhoneDB: inferring cell-cell communication from combined expression of multi-subunit ligand–receptor complexes. *Nat. Protoc.* **15**, 1484–1506 (2020).
13. Garcia-Alonso, L. et al. Mapping the temporal and spatial dynamics of the human endometrium in vivo and in vitro. *Nat. Genet.* **53**, 1698–1711 (2021).
14. Hou, R., Denisenko, E., Ong, H. T., Ramilowski, J. A. & Forrest, A. R. R. Predicting cell-to-cell communication networks using NATMI. *Nat. Commun.* **11**, 5011 (2020).
15. Noel, F. et al. Dissection of intercellular communication using the transcriptome-based framework ICELLNET. *Nat. Commun.* **12**, 1089 (2021).
16. Raredon, M. S. B. et al. Computation and visualization of cell–cell signaling topologies in single-cell systems data using Connectome. *Sci. Rep.* **12**, 4187 (2022).
17. Armingol, E. et al. Context-aware deconvolution of cell-cell communication with Tensor-cell2cell. *Nat. Commun.* **13**, 3665 (2022).
18. Luo, J., Deng, M., Zhang, X. & Sun, X. ESICCC as a systematic computational framework for evaluation, selection, and integration of cell-cell communication inference methods. *Genome Res.* **33**, 1788–1805 (2023).
19. Vento-Tormo, R. et al. Single-cell reconstruction of the early maternal-fetal interface in humans. *Nature* **563**, 347–353 (2018).
20. Garcia-Alonso, L. et al. Single-cell roadmap of human gonadal development. *Nature* **607**, 540–547 (2022).
21. Browaeys, R. et al. MultiNicheNet: a flexible framework for differential cell-cell communication analysis from multi-sample multi-condition single-cell transcriptomics data. Preprint at *bioRxiv* https://doi.org/10.1101/2023.06.13.544751 (2023).
22. Jin, S. & Ramos, R. Computational exploration of cellular communication in skin from emerging single-cell and spatial transcriptomic data. *Biochem. Soc. Trans.* **50**, 297–308 (2022).
23. Yang, A. C. et al. Dysregulation of brain and choroid plexus cell types in severe COVID-19. *Nature* **595**, 565–571 (2021).
24. Wendisch, D. et al. SARS-CoV-2 infection triggers profibrotic macrophage responses and lung fibrosis. *Cell* **184**, 6243–6261 e6227 (2021).
25. Winkler, E. A. et al. A single-cell atlas of the normal and malformed human brain vasculature. *Science* **375**, eabi7377 (2022).
26. Lake, B. B. et al. An atlas of healthy and injured cell states and niches in the human kidney. *Nature* **619**, 585–594 (2023).
27. Cheng, Y. T. et al. Inhibitory input directs astrocyte morphogenesis through glial GABA(B) R. *Nature* **617**, 369–376 (2023).
28. Ortiz-Munoz, G. et al. In situ tumour arrays reveal early environmental control of cancer immunity. *Nature* **618**, 827–833 (2023).
29. Li, J. et al. Remodeling of the immune and stromal cell compartment by PD-1 blockade in mismatch repair-deficient colorectal cancer. *Cancer Cell* **41**, 1152–1169 e1157 (2023).
30. Longo, S. K., Guo, M. G., Ji, A. L. & Khavari, P. A. Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics. *Nat. Rev. Genet.* **22**, 627–644 (2021).
31. Wang, X., Almet, A. A. & Nie, Q. The promising application of cell-cell interaction analysis in cancer From single-cell and spatial transcriptomics. *Semin. Cancer Biol.*, (2023).
32. Perkel, J. M. Single-cell proteomics takes centre stage. *Nature* **597**, 580–582 (2021).
33. Jin, S., Zhang, L. & Nie, Q. scAI: an unsupervised approach for the integrative analysis of parallel single-cell transcriptomic and epigenomic profiles. *Genome Biol.* **21**, 25 (2020).
34. Zhang, L., Zhang, J. & Nie, Q. DIRECT-NET: an efficient method to discover cis-regulatory elements and construct regulatory networks from single-cell multiomics data. *Sci. Adv.* **8**, eabl7393 (2022).
35. Troulé, K. et al. CellPhoneDB v5: inferring cell-cell communication from single-cell multiomics data. Preprint at https://doi.org/10.48550/arXiv.2311.04567 (2023).
36. Cang, Z. & Nie, Q. Inferring spatial and signaling relationships between cells from single cell transcriptomic data. *Nat. Commun.* **11**, 2084 (2020).
37. Shao, X. et al. Knowledge-graph-based cell–cell communication inference for spatially resolved transcriptomic data with SpaTalk. *Nat. Commun.* **13**, 4429 (2022).
38. Cang, Z. et al. Screening cell–cell communication in spatial transcriptomics via collective optimal transport. *Nat. Methods* **20**, 218–228 (2023).
39. Li, H. et al. Decoding functional cell-cell communication events by multi-view graph learning on spatial transcriptomics. *Brief. Bioinform.* **24**, bbad359 (2023).
40. Wang, S., Karikomi, M., MacLean, A. L. & Nie, Q. Cell lineage and communication network inference via optimization for single-cell transcriptomics. *Nucleic Acids Res.* **47**, e66 (2019).
41. Browaeys, R., Saelens, W. & Saeys, Y. NicheNet: modeling intercellular communication by linking ligands to target genes. *Nat. Methods* **17**, 159–162 (2020).
42. Hu, Y., Peng, T., Gao, L. & Tan, K. CytoTalk: de novo construction of signal transduction networks using single-cell transcriptomic data. *Sci. Adv.* **7**, eabf1356 (2021).
43. Zhang, Y. et al. CellCall: integrating paired ligand–receptor and transcription factor activities for cell-cell communication. *Nucleic Acids Res.* **49**, 8520–8534 (2021).
44. Cheng, J., Zhang, J., Wu, Z. & Sun, X. Inferring microenvironmental regulation of gene expression from single-cell RNA sequencing data using scMLnet with an application to COVID-19. *Brief. Bioinform.* **22**, 988–1005 (2021).
45. Landherr, A., Friedl, B. & Heidemann, J. A critical review of centrality measures in social networks. *Bus. Inform. Syst. Eng.* **2**, 371–385 (2010).
46. He, H. et al. Single-cell transcriptome analysis of human skin identifies novel fibroblast subpopulation and enrichment of immune subsets in atopic dermatitis. *J. Allergy Clin. Immunol.* **145**, 1615–1628 (2020).
47. Gupta, K. et al. Single-cell analysis reveals a hair follicle dermal niche molecular differentiation trajectory that begins prior to morphogenesis. *Dev. Cell* **48**, 17–31 e16 (2019).
48. Reynolds, G. et al. Developmental cell programs are co-opted in inflammatory skin disease. *Science* **371**, eaba6500 (2021).
49. Jin, S. Datasets and CellChat objects for running cell-cell communication analysis using CellChat. *Figshare* https://figshare.com/projects/Example_data_for_cell-cell_communication_analysis_using_CellChat/157272 (2023).
50. Shao, X. et al. CellTalkDB: a manually curated database of ligand–receptor interactions in humans and mice. *Brief. Bioinform.* **22**, bbaa269 (2021).

Acknowledgements

The work was partly supported by a Major Research Plan of the National Natural Science Foundation of China (grant no. 92374108 to S.J.), National Science Foundation grants DMS1763272, MCB2028424 and CBET2134916, a grant from the Simons Foundation (594598 to Q.N.), National Institutes of Health grants R01AR079150 and R01DE030565, and a Chan Zuckerberg Initiative grant (AN-0000000062). We thank Y. Su in the School of Mathematics and Statistics at Wuhan University for the help in implementing the R Shiny app.

Author contributions

S.J. and Q.N. designed the project. S.J. implemented the codes and developed the protocol. S.J. and Q.N. drafted the manuscript. S.J., Q.N. and M.V.P. edited and approved the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41596-024-01045-4.

Correspondence and requests for materials should be addressed to Suoqin Jin or Qing Nie.

Peer review information *Nature Protocols* thanks Xiaohui Fan, Xiang Liu, Jing Su and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permissions information is available at www.nature.com/reprints.

Related links

Key references using this protocol

Jin, S. et al. *Nat. Commun.* **12**, 1088 (2021): https://doi.org/10.1038/s41467-021-21246-9

Vu, R. et al. *Cell Rep.* **40**, 111155 (2022): https://doi.org/10.1016/j.celrep.2022.111155 nature portfolio

Corresponding author(s): Qing Nie

Last updated by author(s): May 12, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
□	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☒	□	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☒	□	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	□	A description of all covariates tested
☒	□	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☒	□	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☒	□	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	□	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	□	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	□	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	All data analyzed within this manuscript are publicly available. No additional software was used for the data collection process.
Data analysis	The CellChat v2 software was developed and is publicly available at GitHub (https://github.com/jinworks/CellChat). The data analysis procedures using CellChat are documented in detail in the Procedure section of the manuscript. Other data analysis was performed using R (v4.3.1), Python (v3.11.0), and Seurat (4.4.0).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The example datasets analyzed in this protocol are all publicly available. The scRNA-seq data used in Procedure 1 and Procedure 2 are available at the Gene Expression Omnibus (GEO) under accession GSE14742446. The scRNA-seq data used in Procedure 3 are available at the GEO under accession GSM3453535, GSM3453536, GSM3453537, and GSM345353847. The scRNA-seq data used in Fig. 2 are available at ArrayExpress database under accession E-MTAB-814248. In addition, all the preprocessed datasets and CellChat objects required to reproduce this protocol are publicly available at https://figshare.com/projects/Example_data_for_cell-cell_communication_analysis_using_CellChat/157272.

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	*Use the terms sex (biological attribute) and gender (shaped by social and cultural circumstances) carefully in order to avoid confusing both terms. Indicate if findings apply to only one sex or gender; describe whether sex and gender were considered in study design whether sex and/or gender was determined based on self-reporting or assigned and methods used. Provide in the source data disaggregated sex and gender data where this information has been collected, and consent has been obtained for sharing of individual-level data; provide overall numbers in this Reporting Summary. Please state if this information has not been collected. Report sex- and gender-based analyses where performed, justify reasons for lack of sex- and gender-based analysis.*
Population characteristics	*Describe the covariate-relevant population characteristics of the human research participants (e.g. age, genotypic information, past and current diagnosis and treatment categories). If you filled out the behavioural & social sciences study design questions and have nothing to add here, write "See above."*
Recruitment	*Describe how participants were recruited. Outline any potential self-selection bias or other biases that may be present and how these are likely to impact results.*
Ethics oversight	*Identify the organization(s) that approved the study protocol.*

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences   ☐ Behavioural & social sciences   ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Only publicly available data was used. Therefore, no statistical methods were used to predetermine sample size.
Data exclusions	Only publicly available data was used. Therefore, no exclusion was applied to the data.
Replication	Only publicly available data was used. Therefore, no replication was needed.
Randomization	Only publicly available data was used. Therefore, no randomization protocol was required.
Blinding	Only publicly available data was used. Therefore, no blinding was performed.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging