nature protocols

PROTOCOL

https://doi.org/10.1038/s41596-020-0292-x

CellPhoneDB: inferring cell–cell communication from combined expression of multi-subunit ligand–receptor complexes

Mirjana Efremova[1], Miquel Vento-Tormo[2], Sarah A. Teichmann[1,3] and Roser Vento-Tormo[1*]

Cell–cell communication mediated by ligand–receptor complexes is critical to coordinating diverse biological processes, such as development, differentiation and inflammation. To investigate how the context-dependent crosstalk of different cell types enables physiological processes to proceed, we developed CellPhoneDB, a novel repository of ligands, receptors and their interactions. In contrast to other repositories, our database takes into account the subunit architecture of both ligands and receptors, representing heteromeric complexes accurately. We integrated our resource with a statistical framework that predicts enriched cellular interactions between two cell types from single-cell transcriptomics data. Here, we outline the structure and content of our repository, provide procedures for inferring cell–cell communication networks from single-cell RNA sequencing data and present a practical step-by-step guide to help implement the protocol. CellPhoneDB v.2.0 is an updated version of our resource that incorporates additional functionalities to enable users to introduce new interacting molecules and reduces the time and resources needed to interrogate large datasets. CellPhoneDB v.2.0 is publicly available, both as code and as a user-friendly web interface; it can be used by both experts and researchers with little experience in computational genomics. In our protocol, we demonstrate how to evaluate meaningful biological interactions with CellPhoneDB v.2.0 using published datasets. This protocol typically takes ~2 h to complete, from installation to statistical analysis and visualization, for a dataset of ~10 GB, 10,000 cells and 19 cell types, and using five threads.

Introduction

Complex extracellular responses start with the binding of a ligand to its cognate receptor and the activation of specific cell signaling pathways. Mapping these ligand–receptor interactions is fundamental to understanding cellular behavior and response to neighboring cells. With the exponential growth of single-cell RNA sequencing (scRNA-seq)[1], it is now possible to measure the expression of ligands and receptors in multiple cell types and systematically decode intercellular communication networks that will ultimately explain tissue function in homeostasis and their alterations in disease. Identifying ligand–receptor interactions from scRNA-seq requires both the annotation of complex ligand–receptor relationships from the literature and a statistical method that integrates the resource with scRNA-seq data and selects relevant interactions from the dataset.

Overview of the protocol

We developed CellPhoneDB, a public repository of ligands, receptors and their interactions to enable a comprehensive, systematic analysis of cell–cell communication molecules. Our repository relies on the use of public resources to annotate receptors and ligands, as well as manual curation of specific families of proteins involved in cell–cell communication. We include subunit architecture for both ligands and receptors to represent heteromeric complexes accurately (Fig. 1). This is critical, because cell–cell communication relies on multi-subunit protein complexes that go beyond the binary representation used in most databases and studies[2]. To integrate all the information in a flexible, distributable and amendable environment, we developed an SQLite relational database.

Our repository is integrated with a computational approach to identify biologically relevant interacting ligand–receptor partners from scRNA-seq data. After uploading the scRNA-seq data and performing subsampling using geometric sketching[3] (Fig. 2a), cells with the same cluster annotation

[1]Wellcome Sanger Institute, Wellcome Genome Campus, Cambridge, UK. [2]YDEVS Software Development, Valencia, Spain. [3]Theory of Condensed Matter Group, Cavendish Laboratory, University of Cambridge, Cambridge, UK. *e-mail: rv4@sanger.ac.uk

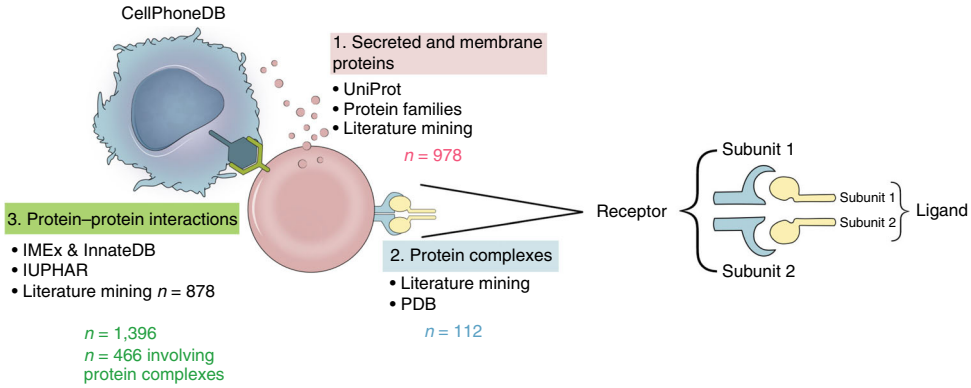


Fig. 1 | Overview of the database. (1) Secreted and membrane proteins stored in protein_input; (2) protein complexes stored in complex_input; and (3) protein–protein interactions stored in interaction_input. Information aggregated within www.CellPhoneDB.org. CellPhoneDB stores a total of 978 proteins: 501 are secreted proteins and 585 are membrane proteins. These proteins are involved in 1,396 interactions; out of all proteins stored in CellPhoneDB, 466 are heteromers. There are 474 interactions that involve secreted proteins and 490 interactions that involve only membrane proteins. There are a total of 250 interactions that involve integrins. Adapted with permission from CellPhoneDB.org.

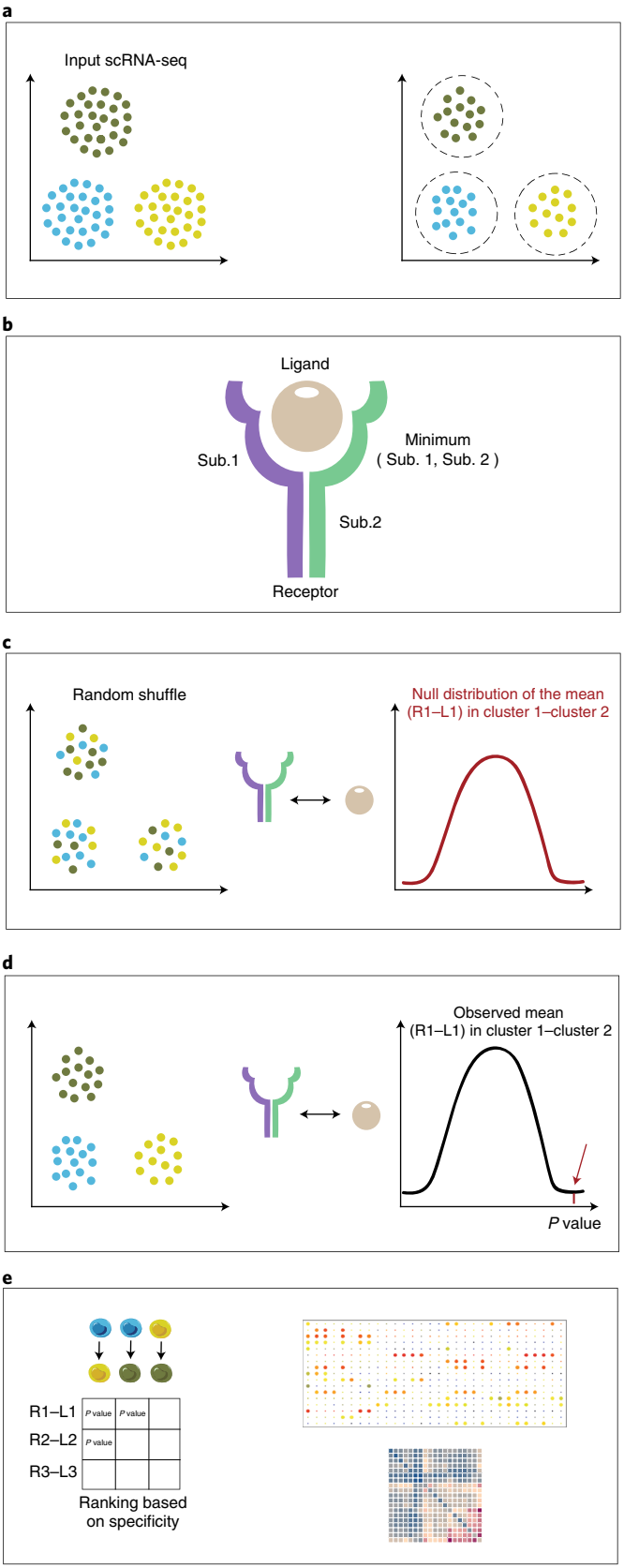
are pooled together as a cell state. We derive enriched ligand–receptor interactions between two cell states on the basis of expression of a receptor by one cell state and a ligand by another cell state. For each gene in the cluster, the percentage of cells expressing the gene and the gene expression mean are calculated (Fig. 2b). We consider the expression levels of ligands and receptors within each cell state and use empirical shuffling to calculate which ligand–receptor pairs display significant cell-state specificity (Fig. 2c,d). This predicts molecular interactions between cell populations via specific protein complexes and generates potential cell–cell communication networks, which can be visualized using intuitive tables and plots (Fig. 2e). Specificity of the ligand–receptor interaction is important, because some of the ligand–receptor pairs are ubiquitously expressed by the cells in a tissue and therefore are not informative regarding specific communication between particular cell states.

The computational code is available in GitHub (https://github.com/Teichlab/cellphonedb) and a user-friendly web interface is available at www.CellPhoneDB.org. The first option is recommended for large datasets (>10 GB). Compared to the original CellPhoneDB platform, our updated version, CellPhoneDB v.2.0, has incorporated new features, such as subsampling of the original dataset to enable the fast querying of large datasets (geometric sketching)[3] or the visualization of results using intuitive tables, plots and network files that can be directly uploaded into Cytoscape (https://cytoscape.org/)[4]. In addition, we now offer users the possibility of using their own list of ligand–receptor interactions through our easy-to-use Python GitHub package.

Applications of the protocol

We originally applied this computational framework to study maternal–fetal communication at the decidual–placental interface during early pregnancy[5]. Briefly, our analysis revealed new immunoregulatory mechanisms and cytokine signaling networks existing between the cells in the maternal–fetal interface that guarantee the coexistence of the mother and the developing fetus (Fig. 3). In the present protocol, we describe and discuss in detail how this analysis can be carried out, using our maternal–fetal study as an illustration.

The protocol is generalizable to any other scRNA-seq dataset containing potentially interacting cell populations and has been recently used in several single-cell atlases. For example, CellPhoneDB helped us identify a shift in the cellular communication from a network that was dominated by mesenchymal–epithelial interactions in healthy airways to a Th2 cell–dominated interactome in asthmatic airways[6]. In the context of the kidney, cell–cell interaction analysis helped to reveal epithelium–immune system crosstalk that coordinates recruitment of antibacterial macrophages and neutrophils to regions in the kidney most vulnerable to infections[7]. In a recent single-cell atlas of hematopoietic progenitors in the liver during the first trimester of development, we identified interactions between erythroblasts and erythroblastic island (EI) macrophages through interactions involving the molecules VCAM1, ITGB1 and ITGA4, all of them known to be important in hematopoiesis[8].


a
Input scRNA-seq
b
Ligand
Sub.1
Minimum
(Sub. 1, Sub. 2)
Sub.2
Receptor
c
Random shuffle
Null distribution of the mean
(R1–L1) in cluster 1–cluster 2
d
Observed mean
(R1–L1) in cluster 1–cluster 2
P value
e
R1–L1
R2–L2
R3–L3
P value
Ranking based
on specificity

◀ **Fig. 2 | Overview of the statistical method framework used to infer ligand–receptor complexes specific to two cell types from single-cell transcriptomics data. a**, CellPhoneDB input data consist of a scRNA-seq counts file and cell-type annotation. Large datasets can be subsampled using geometric sketching[3]. **b**, Enriched receptor–ligand interactions between two cell types are derived on the basis of expression of a receptor by one cell type and a ligand by another cell type. The member of the complex with the minimum average expression is considered for the subsequent statistical analysis. **c**, We generate a null distribution of the mean of the average ligand and receptor expression in the interacting clusters by randomly permuting the cluster labels of all cells. **d**, The *P* value for the likelihood of cell-type specificity of a given receptor–ligand complex is calculated on the basis of the proportion of the means that are as high as or higher than the actual mean. **e**, Ligand–receptor pairs are ranked on the basis of their total number of significant *P* values across the cell populations. Visualization of the results using intuitive tables and plots is provided via the web interface. L1, example ligand L1; R1, example receptor R1; Sub., subunit. Adapted from ref. [5], Macmillan Publishers Limited.

Furthermore, even though CellPhoneDB was created using human-specific ligand–receptor interactions, it can be easily applied to mouse datasets by mapping human genes to their mouse orthologs. In a recent example, we applied our cell–cell communication framework to demonstrate the complex interplay among diverse cells in the evolving tumor microenvironment of a murine melanoma model, in which multiple immunosuppressive mechanisms coexist within a heterogeneous stromal compartment[9].

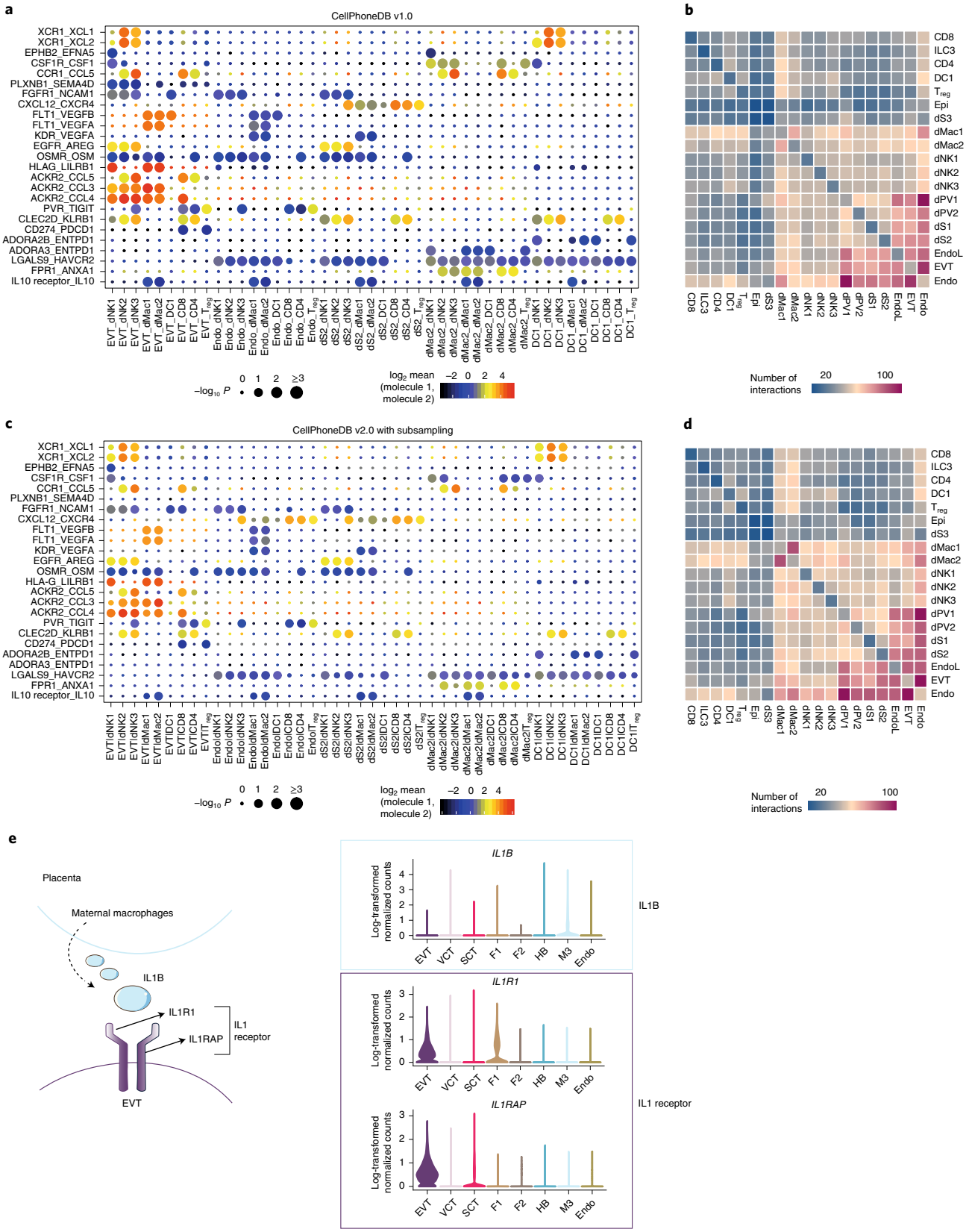
Comparison with other approaches

There are now several other published methods that can be used to infer potentially relevant interactions between two cell populations from scRNA-seq. The majority of these methods use lists of binary ligand–receptor pairs to assign communication between cells, without considering multimeric receptors. Relevant interactions are inferred by filtering on the basis of the expression levels of the ligand and receptor. In these methods, only the interaction pairs that pass a certain threshold for expression level or number of cells expressing the specific interactors in the respective cell populations are selected for the downstream analysis[10–15]. For example, in addition to filtering on the basis of expression level, Cohen et al.[16] used hierarchical clustering with Spearman correlation to identify ligand–receptor modules and construct an interaction graph. Others, such as Kumar et al.[17], scored interactions by calculating the product of average receptor and average ligand expression in the corresponding cell types and used a one-sided Wilcoxon rank-sum test to assess the statistical significance of each interaction score. Halpern et al.[18] computed a *z*-score of the mean of each interacting molecule in each cluster to calculate the enrichment of each ligand and receptor in each cluster. To test for enrichment of the number of ligand–receptor pairs between two cell populations, Joost et al.[19] performed random sampling of receptors and ligands and compared this number with the observed number of ligand–receptor pairs. In a similar way, Boisset et al.[20] applied cluster label permutations to create a null distribution of the number of random interactions between cell populations and then compared this to the actual number of interactions to identify enriched or depleted interactions as compared with the numbers in the background model.

A major strength of CellPhoneDB, as compared with most other databases, is that it takes into account the structural composition of ligands and receptors, which is important because ligand–receptor interactions often involve multiple subunits. This is particularly clear for protein families such as many of the cytokine families, in which receptors share structural subunits and the affinity of the ligand is determined by the specific combination of the receptor subunits (Fig. 3e). Roughly one-third of the ligand–receptor complexes in our database have a multi-subunit stoichiometry greater than binary one-to-one interactions. Specifically, there are 466 interactions in our repository that involve heteromers, and 163 of them involve cytokines.

Limitations of the protocol

Our database, although comprehensive, is not a complete list of all possible ligand–receptor interactions, and this should be taken into consideration when interpreting cell–cell communication networks, especially the total number of interactions between cell types. As more and more interactions are curated and added, both the analysis and interpretation of the results will improve. Furthermore, our statistical method prioritizes cell-type-enriched and potentially biologically important interactions that would result in downstream signaling events. Therefore, a non-significant *P* value does not indicate that the interaction is not present, only that it is not highly specific between two cell types. For a more permissive analysis, we also offer a simpler filtering method based on a threshold of cells expressing ligand–receptor complexes in the corresponding clusters. In addition, we



use permutations to generate a null hypothesis, and this can be time consuming and resource intensive with large datasets (e.g., datasets with millions of cells). To address this, we introduced a subsampling approach that preserves the heterogeneity of the dataset and reduces speed and memory

◀ **Fig. 3 | Example dataset run with CellPhoneDB and CellPhoneDB v.2.0. a**, Overview of selected ligand–receptor interactions using CellPhoneDB on the decidua dataset from ref. [5]; *P* values are indicated by circle size; scale is shown below the plot. The means of the average expression level of interacting molecule 1 in cluster 1 and interacting molecule 2 in cluster 2 are indicated by color. **b**, Heatmap showing the total number of interactions between cell types in the decidua dataset obtained with CellPhoneDB. **c**, Overview of selected ligand–receptor interactions using the CellPhoneDB v.2.0 with subsampling on the decidua dataset. *P* values are indicated by circle size; scale below the plot. The means of the average expression level of interacting molecule 1 in cluster 1 and interacting molecule 2 in cluster 2 are indicated by color. One-third of the dataset was subsampled. **d**, Heatmap showing the total number of interactions between cell types in the decidua dataset obtained with CellPhoneDB v.2.0 with subsampling. One-third of the dataset was subsampled. **e**, An example of significant interactions involving complexes identified by CellPhoneDB in the placenta dataset[5]. Violin plots show log-transformed, normalized expression levels of the components of the interleukin 1 receptor–interleukin 1 (IL1RN–IL1) complex in placental cells. IL1RN expression is enriched in the maternal macrophage cluster and the two subunits of the IL1 receptor (IL1R, IL1RAP) are co-expressed in the extravillous trophoblasts (EVTs). Endo, endothelial cell; EndoL, lymphatic endothelial cell; F, fibroblast; HB, Hofbauer cell; M, macrophage; SCT, syncytiotrophoblast; T_{reg}, regulatory T cells; VCT, villous cytotrophoblast. **a** adapted from ref. [5], Macmillan Publishers Limited.

requirements (1 h versus 1.5 h for a dataset of 10,000 cells). Finally, our tool infers potential interactions using transcriptomics data without considering the spatial proximity of the cells. We anticipate that the information in CellPhoneDB will have the potential to provide a more comprehensive view of cellular communication when combined with the spatial location of the cells as quantified using highly multiplexed spatial methods (e.g., refs. [21–24]).

Database input files

CellPhoneDB stores ligand–receptor interactions, as well as other properties of the interacting partners, including their subunit architecture and gene and protein identifiers. To create the content of the database, four main .csv data files are required: gene_input.csv, protein_input.csv, complex_input.csv and interaction_input.csv (Fig. 4).

gene_input

Mandatory fields are ‘gene_name’, ‘uniprot’, ‘hgnc_symbol’ and ‘ensembl’.

This file is critical for establishing the link between the scRNA-seq data and the interaction pairs stored at the protein level. It includes the following gene and protein identifiers: (i) gene name (‘gene_name’), (ii) UniProt identifier (‘uniprot’), (iii) HUGO Nomenclature Committee (HGNC) symbol (‘hgnc_symbol’) and (iv) gene Ensembl identifier (ENSG) (‘ensembl’). To create this file, lists of linked proteins and gene identifiers are downloaded from UniProt and merged using gene names. Several rules need to be considered when merging the files

- UniProt annotation prevails over the gene Ensembl annotation when the same gene Ensembl identifier points toward different UniProt identifiers.
- UniProt and Ensembl lists are also merged by their UniProt identifier, but this information is used only when the UniProt or Ensembl identifier is missing in the original list merged by gene name.
- If the same gene name points toward different HGNC symbols, only the HGNC symbol matching the gene name annotation is considered.
- Only one HLA isoform is considered in our interaction analysis, and it is stored in a manually HLA-curated list of genes, named HLA_curated.

protein_input

Mandatory fields are ‘uniprot’ and ‘protein_name’.

Optional fields are ‘transmembrane’, ‘peripheral’, ‘secreted’, ‘secreted_desc’, ‘secreted_highlight’, ‘receptor’, ‘receptor_desc’, ‘integrin’, ‘pfam’, ‘other’, ‘other_desc’, ‘tags’, ‘tags_ description’, ‘tags_-reason’ and ‘pfam’.

Two types of input are needed to create this file: (i) systematic input using UniProt annotation and (ii) manual input using curated annotation, from both developers of CellPhoneDB (‘proteins_curated’) and users. For the systematic input, the UniProt identifier (‘uniprot’) and the name of the protein (‘protein_name’) are downloaded from UniProt. For the curated input, developers and users can introduce additional fields relevant to the future systematic assignment of ligand–receptor interactions (see the ‘Systematic input from other databases’ section below under ‘interaction_list’). Importantly, if a protein ID is present in both the curated and systematic inputs, the curated information always takes priority over the systematic one.

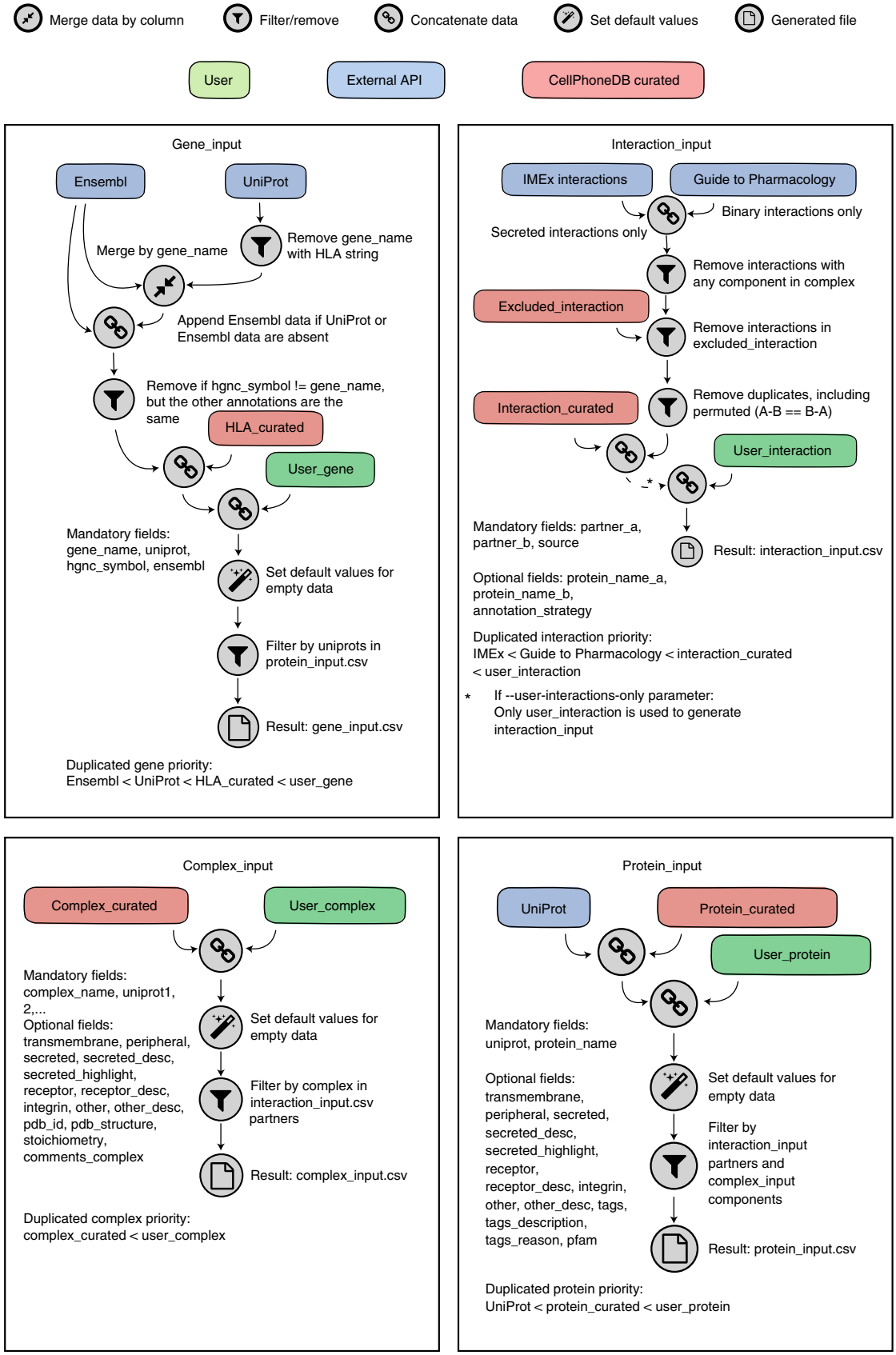


Fig. 4 | Diagram showing how lists are generated. Basic steps in the generation of lists to populate the tables in CellPhoneDB. API, application progamming interface.

Optional fields are organized into the following categories:

- *Location of the protein in the cell*. There are four non-exclusive options: transmembrane ('transmembrane'), peripheral ('peripheral') and secreted ('secreted', 'secreted_desc' and 'secreted_highlight').

  We downloaded plasma membrane proteins from UniProt using the keyword KW-1003 (cell membrane) and annotated them as peripheral proteins using the keyword SL-9903 or as transmembrane proteins (remaining plasma membrane proteins). A systematic manual curation of proteins with transmembrane and immunoglobulin-like domains was performed to improve the lists of plasma transmembrane proteins.

  We downloaded secreted proteins from UniProt using the keyword KW-0964 (secreted) and further annotated them as cytokines (KW-0202), hormones (KW-0372), growth factors (KW-0339) and immune-related proteins, using UniProt keywords and manual annotation based on the literature. 'secreted_highlight' includes cytokines, hormones, growth factors and other immune-related proteins, and 'secreted_desc' indicates a description of the protein function.

  All the manually annotated information is carefully tagged and can be identified. See the 'Curation tags' section below.
- *Receptors and integrins*. Three fields are allocated to annotate receptors or integrins: 'receptor', 'receptor_desc' and 'integrin'.

  Receptors were defined by the UniProt keyword KW-0675 and by a revision of UniProt descriptions and bibliography. For some of the receptors, a short description is included in 'receptor_desc'.

  'Integrin' is a manual curation field that indicates the protein is part of the integrin family. All the annotated information is carefully tagged and can be identified. For details, see the 'Curation tags' section below.
- *Others*. We created another column named 'others' that consists of membrane and secreted proteins that are excluded from our cell–cell communication analysis because they are not directly involved in the recognition of the ligand (e.g., co-receptors) or they require more specialized annotation (e.g., nerve-specific receptors such as those related to ear binding, olfactory receptors, taste receptors and salivary receptors). In addition, we excluded small-molecule receptors, immunoglobulin chains and viral and retroviral proteins, pseudogenes, cancer antigens and photoreceptors. We also added 'others_desc' to allow a brief description of the excluded protein.
- *Protein family*. Information about the family of the protein is downloaded from https://pfam.xfam.org/ (ref. [25]) and stored in 'pfam'. This information may be useful for the annotation of ligand–receptor interactions.
- *Curation tags*. Three fields indicate whether the protein has been manually curated: 'tags', 'tags_reason' and 'tags_ description'.

There are three options for the 'tags' field: (i) 'N/A': protein matches UniProt description; (ii) 'To_add': addition of secreted and/or plasma membrane protein annotation; and (iii) 'To_comment': manual addition of a specific property of the protein, for example, annotation of a protein as a receptor.

There are five options for the 'tags_reason' field: (i) 'extracellular_add': manual annotation of the protein as plasma membrane; (ii) 'peripheral_add': manual annotation of the protein as peripheral; (iii) 'secreted_add': manual annotation of the protein as secreted; (iv) 'secreted_high': manual annotation of the protein as cytokine, hormone, growth factor or other immune-related protein (secreted_high); (v) 'receptor_add': manual annotation of a receptor.

Finally, the 'tags_description' field is a short description of the manually curated protein.

complex_input

Mandatory fields are 'complex_name' and 'uniprot1', 'uniprot2' and so on.

Optional fields are 'transmembrane', 'peripheral', 'secreted', 'secreted_desc', 'secreted_highlight', 'receptor', 'receptor_desc', 'integrin', 'other', 'other_desc', 'pdb_id', 'pdb_structure', 'stoichiometry' and 'comments_complex'

Literature and UniProt descriptions were reviewed to annotate heteromeric proteins, which were defined as cases in which the functional receptor or ligand required more than one gene product, and a careful annotation was performed for cytokine complexes, TGF family complexes and integrin complexes.

These lists contain the UniProt identifiers for each of the heteromeric ligands and receptors ('uniprot1', 'uniprot2' and so on) and a name given to the complex ('complex_name'). These entries have common fields with 'protein_input' that are described in the previous section. These are 'transmembrane', 'peripheral', 'secreted', 'secreted_desc', 'secreted_highlight', 'receptor', 'receptor_desc', 'integrin', 'other', 'other_desc' (see descriptions in the above 'protein_input' section for clarification). We also include additional optional information that may be relevant for the stoichiometry of the heterodimers. Structural information is included in 'pdb_structure', 'pdb_id' and 'stoichiometry', if heteromers are defined in the RCSB Protein Data Bank (http://www.rcsb.org/). An additional field, 'comments_complex', was created to add a short description of the heteromer.

interaction_input

Mandatory fields are 'partner_a', 'partner_b', 'annotation_strategy' and 'source'.

Optional fields are 'protein_name_a' and 'protein_name_b'.

Interactions stored in CellPhoneDB are annotated using their UniProt identifier (binary interactions) or the name of the complex (interactions involving heteromers: 'partner_a' and 'partner_b'). The name of the protein is also included but is not mandatory ('protein_name_a' and 'protein_name_b'). Protein names are not stored in the database.

There are two main inputs of interactions: (i) a systematic input querying other databases and (ii) a manual input using curated information from CellPhoneDB developers ('interactions_curated') and users. The method used to assign the interaction is indicated in the 'annotation_strategy' column.

Each interaction stored has a CellPhoneDB unique identifier ('id_cp_interaction') generated automatically by the internal pipeline.

Systematic input from other databases. Three sources of interacting partners were considered: (i) IUPHAR (http://www.guidetopharmacology.org/): binary interactions only; (ii) InnateDB (https://www.innatedb.com/): interactions involving cytokines, hormones and growth factor interactions; and (iii) IMEx Consortium (https://www.imexconsortium.org/): interactions involving cytokines, hormones and growth factors interactions.

Binary interactions from IUPHAR are directly downloaded from http://www.guidetopharmacology.org/DATA/interactions.csv and 'guidetopharmachology.org' is indicated in the 'annotation_strategy' field. For the IMEx Consortium, all protein–protein interactions are downloaded using the PSICQUIC REST APIs[26]. The IMEx[27], IntAct[28], InnateDB[29], UCL-BHF (https://www.ucl.ac.uk/cardiovascular/research/pre-clinical-and-fundamental-science/functional-gene-annotation/manual-curation/protein), MatrixDB[30], MINT[31], I2D[32], UniProt and MBInfo (https://www.mechanobio.info/) registries are used. Interacting partners are defined as follows:

- Interacting partner A has to be a transmembrane receptor and cannot be classified as 'others' (see the 'protein_input' section for more information).
- Interacting partner B has to be 'secreted_highlight'. This group of proteins includes cytokines, hormones, growth factors and other immune-related proteins (see the 'protein_input' section for more information).

Some interactions in the systematic approach are excluded: (i) interactions in which one of the components is part of a complex (see 'complex_input' list in the above section) and (ii) interactions that are not involved in cell–cell communication or are wrongly annotated by our systematic method. These are stored in a curated list of proteins named excluded_interaction. The excluded_interaction file contains five fields: (i) 'uniprot_1': name of the interacting partner A that is going to be excluded; (ii) 'uniprot_2': name of the interacting partner B that is going to be excluded; (iii) 'name.1': name of the protein to be excluded corresponding to uniprot_1; (iv) 'name.2': name of the protein to be excluded corresponding to uniprot_2; and (v) comments: information about the exclusion of the protein.

Homomeric complexes (proteins interacting with themselves) are excluded from the systematic analysis. Importantly, in cases in which both the systematic and the curated input detect the interactions, the curated input always prevails over the systematic information.

Curated approach. UniProt descriptions and PubMed information about membrane receptors were used to annotate ligand–receptor interactions, and the International Union of Pharmacology annotation[33] was used to annotate cytokine and chemokine interactions. The interactions of other groups of cell-surface proteins, including the TGF family, integrins, lymphocyte receptors, semaphorins, ephrins, Notch and TNF receptors, were manually reviewed using a bibliography. The bibliography used to annotate the interaction is stored in 'source'. 'Uniprot' indicates that the interaction has been annotated using UniProt descriptions.

User-defined ligand–receptor datasets

CellPhoneDB v.2.0 enables users to create their own lists of genes, curated proteins, complexes and interactions. To do so, the format of the users' lists must be compatible with the input files. Users can run the analysis with their sets of interactions using the Python package version of CellPhoneDB. User's lists can either be merged with the information already stored in CellPhoneDB or considered on their own. In addition, users can send the interaction lists via email, the cellphonedb.org form, or a pull request to the CellPhoneDB data repository (https://github.com/Teichlab/cellphonedb-data) to be considered in the new versions of CellPhoneDB.

Database structure

Information is stored in an SQLite relational database (https://www.sqlite.org/). SQLAlchemy (www.sqlalchemy.org) and Python 3 were used to build the database structure and the query logic. The application is designed to enable analyses of potentially large count matrices to be performed in parallel. This requires an efficient database design, including optimization of query times, indices and related strategies. All application code is open source and uploaded to GitHub and www.cellphonedb.org.

The database consists of six main tables: gene_table, protein_table, multidata_table, interaction_table, complex_table and complex_composition_table (Supplementary Fig. 1).

All tables have an incremental numeric unique identifier with the structure id_[table_name], and one or more foreign keys, with the structure [foreign_table_name]_id, to connect all tables.

gene_table

This table stores all the information generated in the gene_input database input file. This includes the gene name ('gene_name'), the HUGO nomenclature committee symbol (HGNC) ('hgnc_symbol') and the ensembl identifier ('ensembl'). Importantly, only the gene and protein information of the interaction participants from 'interactions_list' is stored in our database.

The gene table is related to the protein table via the 'protein_id' - 'id_protein' (one to many) foreign key.

multidata_table

This table stores the information shared between the protein_table and the complex_table.

All the information required in this table is obtained from the protein_input and complex_input input files. It stores the following fields: (i) 'name', corresponding to uniprot if the specific entry (row) represents a protein, or 'complex_name' if the entry represents a complex; (ii) 'transmembrane', (iii) 'peripheral', (iv) 'secreted', (v) 'secreted_desc', (vi) 'secreted_highlight', (vii) 'receptor', (viii) 'receptor_desc', (ix) 'integrin', (x) 'other' and (xi) 'other_desc'. In addition, an 'is_complex' column is added for internal optimization and indicates whether the entry (row) is a complex.

protein_table

This table stores the information obtained from the database input file, protein_input. It contains the name of the protein ('protein_name'), 'tags', 'tags_reason', 'tags_description' and 'pfam'. The table is related to multidata_Table (1..0 - 1 relation, meaning that one or zero elements of protein_table correspond to one element of multidata_table) through the 'protein_multidata_id' foreign key.

complex_table

This table stores complex information from the database input file complex_input and stores the following fields: 'pdb_id', 'pdb_structure', 'stoichiometry' and 'comments_complex'. The table is related to multidata_table (this is a 1..0 - 1 relation, meaning that one or zero elements of complex_table correspond to one element of multidata_table) through the 'complex_multidata_id' foreign key.

All information about the complex components is stored in the complex_composition_table file.

complex_composition_table

This table stores the proteins ('uniprot_1'–'uniprot_4') that compose a complex. It is connected to multidata_table through 'complex_multidata_id' and 'protein_multidata_id' (this is a 1..* - 1 relation, meaning that multiple proteins and/or complexes with IDs stored in multidata_table can participate in one complex_composition and can be included in the complex_composition_table). We also created an additional column called 'total_protein' (with a number of complex components) for

internal optimization purposes. Supplementary Fig. 2 represents an example of two complex_input rows with two and four protein components, respectively.

interaction_table

This table stores the interaction data from the interaction_input file. The following columns are used to represent the data: 'id_cp_interaction', 'annotation_strategy' and 'source'. To identify the interaction partners ('partner_a' and 'partner_b' in interaction_input), the table is connected to multidata_table through the foreign keys 'multidata_1_id' and 'multidata_2_id', respectively, with a 1 - 1..* relation, meaning that one multidata_id can be present multiple times in the interaction_table. multidata_table stores both protein and complex data. Importantly, only genes and proteins participating in cell–cell communication are stored in our database; that is, not all the proteins present in the input files are stored in our database (see the 'interaction_input' section).

Analysis methods

Statistical inference of ligand–receptor specificity

To assess cellular crosstalk between different cell types, we use our repository in a statistical framework for inferring cell–cell communication networks from scRNA-seq data. We predict enriched receptor–ligand interactions between two cell types on the basis of expression of a receptor by one cell type and a ligand by another cell type. To identify biologically relevant interactions, we look for cell-type-enriched ligand–receptor interactions. Only receptors and ligands expressed in more than a user-specified threshold percentage of the cells in the specific cluster are considered for the analysis (default is 10%).

We then perform pairwise comparisons between all cell types in the dataset. First, we randomly permute the cluster labels of all cells (1,000 times by default) and determine the mean of the average ligand expression level in a cluster and the average receptor expression level in the interacting cluster. In this way we generate a null distribution for each ligand–receptor pair in each pairwise comparison between two cell types. We obtain a *P* value for the likelihood of cell-type enrichment of each ligand–receptor complex by calculating the proportion of the means that are as high as or higher than the actual mean. On the basis of the number of significant pairs, we then prioritize interactions that are highly specific between cell types, so that the user can manually select biologically relevant ones. For multi-subunit heteromeric complexes, we require that all subunits of the complex be expressed (using a user-specified threshold), and we use the member of the complex with the minimum average expression for random shuffling.

Cell subsampling for accelerated analyses

Technological developments and protocol improvements have enabled an exponential growth in the number of cells obtained from scRNA-seq experiments[1]. Large-scale datasets can profile hundreds of thousands of cells, which presents a challenge for the existing analysis methods in terms of both computer memory usage and runtime. To improve the speed and efficiency of our protocol and facilitate its broad accessibility, we integrated subsampling as described in Hie et al.[2]. This 'geometric sketching' approach aims to maintain the transcriptomic heterogeneity within a dataset with a smaller subset of cells. It projects high-dimensional data into a low-dimensional space and divides that low-dimensional space into a predefined number of equal subspaces. The subsampling is then performed by sampling an equal number of data points from each subspace. The subsampling step is optional, enabling users to perform the analysis either on all cells or with other subsampling methods of their choice.

Materials

Equipment

Input data files

- *Metadata file*. The annotation file is generated by the users after they have annotated each cluster identified by scRNA-seq data (e.g., by using packages such as Seurat[34] and SCANPY[35]). The file contains two columns: 'Cell', indicating the name of the cell; and 'cell_type', indicating the name of the cluster considered. Formats accepted are .csv, .txt, .tsv, .tab and .pickle. See Equipment setup.
- *Counts file*. scRNA-seq count data containing gene expression values in which rows are genes presented with gene names identifiers (Ensembl IDs, gene names or hgnc_symbol annotation) and columns are cells. We recommend using normalized count data. Importantly, the user needs to specify whether the data was log-transformed when using the subsampling option. Formats accepted are .csv, .txt, .tsv, .tab and .pickle. See Equipment setup.

Hardware
- Linux or MAC OS

Software
- Python v.3.5 or higher (https://www.python.org/downloads/)
- SQLAlchemy (https://www.sqlalchemy.org/)
- SQLite (https://www.sqlite.org/index.html)
- Preprocessing of the raw expression data to generate the input files can be done using packages such as Seurat[34], SCANPY[35], or any other pipeline that the user prefers.

Equipment setup

Example input data

Example input data can be downloaded from our webserver at https://www.cellphonedb.org/explore-sc-rna-seq or by running the following from the command line:

```
curl  https://raw.githubusercontent.com/Teichlab/cellphonedb/master/
in/example_data/test_counts.txt --output test_counts.txt
curl  https://raw.githubusercontent.com/Teichlab/cellphonedb/master/
in/example_data/test_meta.txt --output test_meta.txt
```

Pre-processing of raw data and generation of input files for the protocol

Some of the most standard packages for scRNA-seq analysis include Seurat[34] and SCANPY[35]. Therefore, we include instructions for how to use these packages to pre-process the raw expression data to generate the input files necessary for CellPhoneDB v.2.0. We recommend using normalized count data as input.

For example, using the R package Seurat[34], the count input file can be obtained by taking the raw expression data from the Seurat object and applying the normalization manually. Users can also normalize using their preferred method for normalization.

```
# Take raw data and normalize it.
count_raw <- data_object@raw.data[,data_object@cell.names]
count_norm <- apply(count_raw, 2, function(x) (x/sum(x))*10000)
write.table(count_norm, 'cellphonedb_count.txt', sep='\t', quote=F)
# Generating metadata file.
meta_data <- cbind(rownames(data_object@meta.data), data_object@meta.
data[,'cluster', drop=F])
# Cluster is the user's corresponding cluster column.
write.table(meta_data,  'cellphonedb_meta.txt',  sep='\t',  quote=F,
row.names=F)
```

The input files can also be extracted from a SCANPY[35] data object:

```
import pandas as pd
import scanpy.api as sc
# Data after filtering and normalizing.
adata = sc.read(adata_filepath)
# We recommend using the normalized non-log-transformed data; you can save
it in adata.norm, for example.
df_expr_matrix = adata.norm
df_expr_matrix = df_expr_matrix.T
df_expr_matrix = pd.DataFrame(df_expr_matrix.toarray())
# Set cell IDs as columns.
df_expr_matrix.columns = adata.obs.index
# Genes should be either Ensembl IDs or gene names.
df_expr_matrix.set_index(adata.raw.var.index, inplace=True)
df_expr_matrix.to_csv(savepath_counts,sep='\t')
# Generating metadata file:
```

```
df_meta = pd.DataFrame(data={'Cell':list(adata.obs[cell_ids]), 'cell_type'
:list(adata.obs[annotation_name])})
df_meta.set_index('Cell',inplace=True)
df_meta.to_csv(savepath_meta, sep='\t')
```

▲ **CRITICAL** CellPhoneDB can be used either through the interactive website (cellphonedb.org), which executes calculations in our private cloud, or as a Python package using the user's computer/cloud/farm. The Python package is recommended for large datasets (datasets >10 GB).

Procedure

Installation ● Timing 5–10 min

▲ **CRITICAL** Steps 1–15 describe the Python implementation of CellPhoneDB v.2.0, whereas Steps 16–19 describe using the webserver.

▲ **CRITICAL** If the default Python interpreter is for Python v2.x (can be checked with the command `python --version`), calls to python/pip must be substituted with python3/pip3.

▲ **CRITICAL** We highly recommend using a virtual environment (Steps 1 and 2), but this can be omitted.

1 Create a Python virtual environment.

```
python -m venv cpdb-venv
```

2 Activate the virtual environment.

```
source cpdb-venv/bin/activate
```

3 Install CellPhoneDB v.2.0.

```
pip install cellphonedb
```

Running with statistical analysis ● Timing 1.5 h for a dataset of ~10 GB, 10,000 cells, five threads

4 Activate the virtual environment if you did not activate it in Step 2.

```
source cpdb-venv/bin/activate
```

5 Run CellPhoneDB v.2.0 in statistical analysis mode, using the input file names (including full path to the files) for metadata and counts (Equipment setup).

```
cellphonedb method statistical_analysis test_meta.txt test_counts.txt
```

Optional parameters are as follows:
`--project-name` Name of the project. A subfolder with this name is created in the output folder [default: `./out`].
`--iterations` Number of iterations for the statistical analysis [default: `1000`]
`--threshold`: Percentage of cells expressing the specific ligand or receptor
`--result-precision`: Number of decimal digits in results [default: `3`]
`--counts-data` Type of gene identifiers in the counts data `[ensembl | gene_name | hgnc_symbol]`
`--output-path` Directory where the results will be allocated (the directory must exist) [default: `./out`]
`--output-format` Output format of the results files (extension will be added to filename if not present) [default: `txt`]
`--means-result-name` Name of the means result file [default: `means.txt`]
`--significant-mean-result-name` Name of the significant means result file [default: `significant_means.txt`]
`--deconvoluted-result-name` Name of the deconvoluted result file [default: `deconvoluted.txt`]
`--verbose/--quiet` Print or hide CellPhoneDB logs [`verbose`]

`--pvalues-result-name` Name of the *P* value results file [default: `pvalues.txt`]
`--debug-seed` Debug random seed −1. To disable it, use a value ≥0 [default: `-1`]
`--threads` Number of threads to use; ≥1 [default: `4`]
Below we present three usage examples.
• Set number of iterations and threads:

```
cellphonedb method statistical_analysis yourmetafile.txt yourcounts-
file.txt --iterations=10 --threads=2
```

• Set project subfolder:

```
cellphonedb method analysis yourmetafile.txt yourcountsfile.txt
--project-name=new_project
```

• Set output path:

```
mkdir custom_folder
cellphonedb method statistical_analysis yourmetafile.txt yourcounts-
file.txt --output-path=custom_folder
```

? TROUBLESHOOTING

Running with subsampling and statistical analysis ● Timing 1 h for dataset of ~10 GB, 10,000 cells subsampled to 5,000, 19 cell types, five threads

▲ **CRITICAL** This step can be used instead of Step 5 with large datasets to increase speed and reduce memory requirements.

6 Run CellPhoneDB v.2.0 in statistical analysis mode, using the input files for metadata and counts and add subsampling and other subsampling-specific parameters

```
cellphonedb method statistical_analysis yourmetafile.txt yourcounts-
file.txt --subsampling --subsampling-log true
```

The parameters are same as described in Step 5, with the addition of the following subsampling-specific parameters:
`--subsampling-log` Enables log transformation for non-log-transformed data inputs (mandatory parameter)
`--subsampling-num-pc` Subsampling `NumPC` argument
`--subsampling-num-cells` Number of cells to subsample to [default: 1/3 of the cells)

? TROUBLESHOOTING

Running without statistical analysis ● Timing ~5 min for dataset of ~10 GB, 10,000 cells, 19 cell types

7 Run CellPhoneDB v.2.0 in normal mode, using the input files for metadata and counts and the specified -threshold parameter. The parameters are same as described in Step 5. The parameters `--pvalues-result-name`, `--threads` and `--debug-seed` should be omitted.

```
cellphonedb method analysis test_meta.txt test_counts.txt
```

? TROUBLESHOOTING

Visualization ● Timing seconds to minutes

▲ **CRITICAL** Users can visualize the results of the analysis using dot plots and heatmaps.

8 Run the dot plot visualization command in either statistical analysis mode (Steps 4 and 5 or Step 6) or normal mode (Step 7), using the means.csv and pvalues.csv output files.

```
cellphonedb plot dot_plot
```

The following are dot plot–specific parameters:

`--means-path` The means output file [default: `./out/means.txt`]
`--pvalues-path` The pvalues output file [default: `./out/pvalues.txt`]
`--output-path` Output folder [default: `./out`]
`--output-name` Name of the output plot [default: `plot.pdf`]; available output formats are those supported by R's ggplot2 package, for example, .pdf, .png, .jpeg
`-rows` File with a list of rows to plot, one per line
`--columns` File with a list of columns to plot, one per line
`--verbose/--quiet` Print or hide CellPhoneDB logs [`verbose`].

To plot only desired rows/columns, use the following command:

```
cellphonedb plot dot_plot --rows in/rows.txt --columns in/columns.txt
```

The following is example content of rows.txt file:

```
TNFRSF11B_TNFSF11
PlexinA3_complex1_SEMA3A
TTR_NGFR
NGF_NGFR
PTHLH_PTH1R
EFNB2_EPHB3
```

9 Run the heatmap visualization command in either statistical analysis mode or normal mode, using the pvalues.csv output file.

```
cellphonedb plot heatmap_plot meta_data
```

Heatmap plot specific parameters are as follows:

`--pvalues-path` The pvalues output file [default: `./out/pvalues.txt`]
`-output-path` Output folder [default: `./out`]
`--count-name` Filename of the output plot [default: `heatmap_count.pdf`]
`--log-name` Filename of the output plot using log-count of interactions [default: `heatmap_log_count.pdf`]
`--count-network-name` Filename of the output network file [default: `network.txt`]
`--interaction-count-name` Filename of the output interactions-count file [default: `interaction_count.txt`]
`--verbose/--quiet`: Print or hide cellphonedb logs [`verbose`]

Using different versions of the database ● Timing seconds to minutes

▲ **CRITICAL** 'Local repository' refers to CellPhoneDB data available locally on the user's computer. 'Remote repository' corresponds to the CellPhoneDB available official data. These data will be downloaded using the `--database` parameter.

10 CellPhoneDB v.2.0 databases can be updated from a remote repository. Available versions of the database can be listed and downloaded to be used. This is relevant because users may have used a specific version of the databases for their analysis and may want to continue with this version for consistency and reproducibility of their analysis.
To use one of those versions, a user must provide the parameter `--database <version_or_file>` to the command `cellphonedb method` as follows:

```
cellphonedb method statistical_analysis in/example_data/test_meta.txt in/example_data/test_counts.txt --database=v0.0.2
```

If the `--database <version_or_file>` parameter is a readable database file, it will be used as it is. Otherwise, a database version matching the specified parameter will be used.

If the selected database version does not exist in the user's local environment, it will be downloaded from the remote repository (see below).
If the `--database` argument is not specified in the command for running the analysis, the latest local database version available will be used. Downloaded versions of the database will be stored in a user folder under ~/.cpdb/releases.

11 To list available database versions from the remote repository, execute the code below:

```
cellphonedb database list_remote
```

12 To list available versions from the local repository, execute the code below:

```
cellphonedb database list_local
```

Downloading different versions of the database ● Timing seconds to minutes

13 To download a version from the remote repository, type:

```
cellphonedb database download
```
or
```
cellphonedb database download --version <version_spec|latest>
```

`version_spec` must be one of the database versions listed in the database. The list of database versions can be obtained using the `list_remote` command. If no version is specified or `latest` is used as a `version_spec`, the newest available version will be downloaded.

Generating a user-specific database ● Timing ~10 min

14 To generate such a database with user-specific input files, type:

```
cellphonedb database generate
```

Specific parameters for the `database generate` command are as follows:
`--user-protein` Protein input file
`--user-gene` Gene input file
`--user-complex` Complex input file
`--user-interactions` Interactions input file
`--fetch` Some lists can be downloaded from original sources while creating the database, for example, Uniprot, Ensembl. By default, the input tables included in the CellPhoneDB package will be used; to enable downloading an updated copy from the remote servers `--fetch` must be appended to the command.

`--result-path` Output folder
`--log-file` Log file

The resulting database file will be generated in the folder 'out' with cellphonedb_user_[datetime].db. The user-defined input tables will be merged with the current CellPhoneDB input tables. To use this database, use the `--database` parameter when executing the `cellphonedb method` command, for example:

```
cellphonedb method statistical_analysis in/example_data/test_meta.
txt in/example_data/test_counts.txt --database out/cellphonedb_
user_2019-05-10-11_10.db
```

Below we describe the input and results of several examples of user-specific custom databases

- To add or correct some interactions, input your_custom_interaction_file.csv (a comma-separated file; make sure to use the mandatory columns) with interactions to add/correct.

```
cellphonedb database generate --user-interactions your_custom_
interaction_file.csv
```

Result. A new database file with CellPhoneDB interactions and user custom interactions. For duplicated interactions, user lists overwrite the CellPhoneDB original data.

- To use only user-specific interactions, input: your_custom_interaction_file.csv (make sure to use the mandatory columns), specifying the interactions to use:

```
cellphonedb database generate --user-interactions your_custom_interaction_file.csv --user-interactions-only
```

Result. A new database file with only the user's custom interactions.

- To correct any protein data, input your_custom_protein_file.csv (make sure to use the mandatory columns), specifying the proteins to overwrite:

```
cellphonedb database generate --user-protein your_custom_protein_file.csv
```

Result. A new database file with CellPhoneDB interactions and the user's custom interactions. For duplicated interactions or proteins, the user list overwrites CellPhoneDB original data.

To add some interactions and correct any protein data, input your_custom_interaction_file.csv (make sure to use mandatory the columns), specifying interactions to add/correct, and your_custom_protein_file.csv (make sure to use the mandatory columns), specifying proteins to overwrite.

```
cellphonedb database generate --user-interactions your_custom_interaction_file.csv --user-protein your_custom_protein_file.csv
```

Result. A new database file with CellPhoneDB interactions and the user's custom interactions. For duplicated interactions or proteins, the user list overwrites CellPhoneDB original data.

- To update remote sources (e.g., UniProt, IMEx, Ensembl), input:
  - your_custom_interaction_file.csv: (make sure to use the mandatory columns) with interactions to add/correct.
  - your_custom_protein_file.csv (make sure to use the mandatory columns) with proteins to overwrite.

```
cellphonedb database generate --fetch
```

Some lists can be downloaded from original sources, for example, UniProt or Ensembl, while creating the database. By default, the input tables included in the CellPhoneDB package will be used; to enable downloading an updated copy from the remote servers `--fetch` must be appended to the `generate` command.

Result. A new database file with the CellPhoneDB interactions and the user's custom interactions. For duplicated interactions or proteins, user lists overwrite the CellPhoneDB original data.

▲ **CRITICAL STEP** This command uses external resources allocated in external servers. The command may not end correctly if external servers are not available. The timing of this step depends on external servers and the user's Internet connection and can take longer to finish.

Getting descriptions of mandatory and optional parameters ● Timing seconds

15 Obtain a detailed description of the mandatory and optional parameters, using the 'help' option:

```
cellphonedb method statistical_analysis yourmetafile.txt yourcountsfile.txt --help
```

Interactive web portal ● Timing ~1 h for a dataset of ~10 GB, 10,000 cells; depends on how many jobs are running in parallel and the computing resources available at the time of analysis

▲ **CRITICAL** The web interface includes form inputs with which the user can define analysis parameters before submission. Downstream calculations are performed on the application's servers, rendering the

Fig. 5 | Screenshot of the web portal. a, Screenshot showing how to input the user's email in order to get a notification when the analysis is finished. **b**, Screenshot showing the significant_means results table. The user can click on a selected id_cp_interaction field to get more detailed information for the specific interaction pair. **c**, Screenshot showing detailed information for the specific interaction pair that appears when the user clicks on a specific id_cp_interaction field. **d**, Screenshot showing the dot plot visualization page. ▶

information on ligand and receptor expression, as well as visualization diagrams once analysis is complete (Fig. 5).

16 Go to the 'Exploring your scRNAseq' tab and input your meta and count input files (see 'Input data files' section in 'Equipment').

17 Provide an email address if you would like to get an update when the process finishes (Fig. 5a).

18 The 'significant_means' results table will appear as in Fig. 5c. You can change the current view by clicking on the 'Data Shown' button (Fig. 5b) and can download the results as well. Click on any field from the 'id_cp_interaction' column to display detailed information for the specific interaction pair (Fig. 5c).

19 Go to the 'Plots' tab and pick the type of plot you would like to produce. For plotting dot plots, select the columns and rows you need (Fig. 5d).
The online results viewer enables you to select which columns you wish to display in each table. This option is quite useful as an aid in visualizing the results.

Troubleshooting

Troubleshooting advice can be found in Table 1.

Table 1 | Troubleshooting table

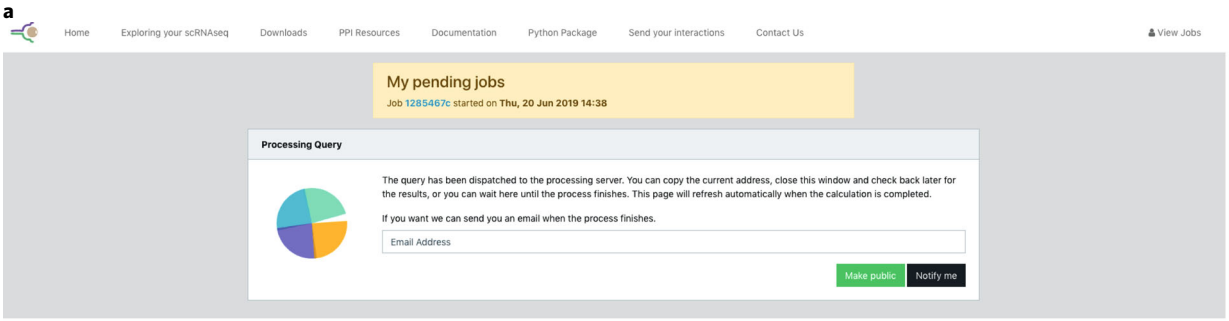
Step	Problem	Possible reason	Solution
5,6,7	[ERROR] Invalid Counts data	The order of the input count and metadata might be switched or the genes are neither Ensembl IDs nor gene names	Use the metadata as first and the count data as second input parameters and provide a count table with genes presented as either Ensembl IDs or gene names
	Some cell IDs in the meta file do not exist in counts columns or the input file is in a format that is not compatible with CellPhoneDB v.2.0.	The cell IDs in the columns of the counts data do not match the cell IDs in the 'cell_type' column of the metadata	Make sure that you have the same cell IDs in the columns of the counts data and the 'cell_type' column of the metadata
6	[ERROR] In order to perform subsampling you need to specify whether to log1p input counts or not: to do this, specify it in your command as `--subsampling-log [true|false]`	`--subsampling-log` needs to be specified (`true` or `false`)	Provide BOOLEAN value to the `--subsampling-log` input parameter

Timing

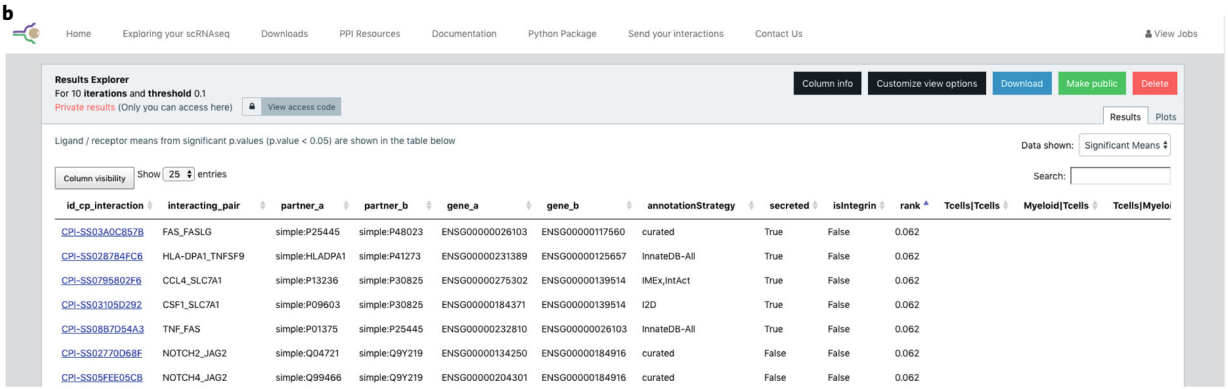
Python

Steps 1–3, installation of Python package: 5–10 min
Steps 4 and 5, running with statistical method: 1.5 h for dataset of ~10 GB, 10,000 cells, five threads
Step 6, subsampling and statistical method: 1 h for dataset of ~10 GB, 10,000 cells subsampled to 5,000, 19 cell types, five threads
Step 7, analysis without the statistical method: ~5 min for dataset of ~10 GB, 10,000 cells, 19 cell types
Steps 8 and 9, visualization: seconds to minutes
Steps 10–13, using different database versions: seconds to minutes
Step 14, generating a user-specific database: ~10 min
Step 15, descriptions of parameters: seconds

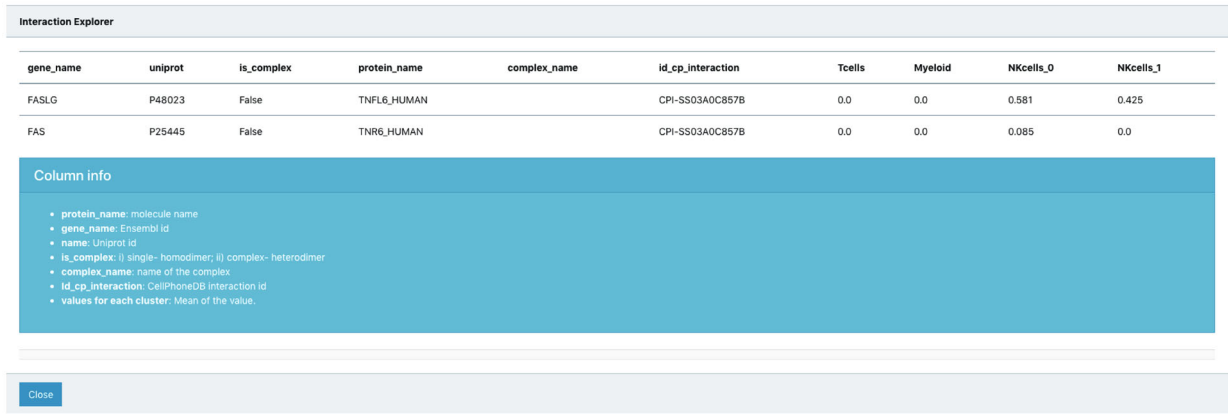
a


Home
Exploring your scRNAseq
Downloads
PPI Resources
Documentation
Python Package
Send your interactions
Contact Us
View Jobs
My pending jobs
Job 1285467c started on Thu, 20 Jun 2019 14:38
Processing Query
The query has been dispatched to the processing server. You can copy the current address, close this window and check back later for the results, or you can wait here until the process finishes. This page will refresh automatically when the calculation is completed.
If you want we can send you an email when the process finishes.
Email Address
Make public
Notify me

b


Home
Exploring your scRNAseq
Downloads
PPI Resources
Documentation
Python Package
Send your interactions
Contact Us
View Jobs
Results Explorer
For 10 iterations and threshold 0.1
Private results (Only you can access here)
View access code
Column info
Customize view options
Download
Make public
Delete
Results
Plots
Ligand / receptor means from significant p.values (p.value < 0.05) are shown in the table below
Data shown: Significant Means
Column visibility
Show 25 entries
Search:
id_cp_interaction | interacting_pair | partner_a | partner_b | gene_a | gene_b | annotationStrategy | secreted | isIntegrin | rank | Tcells|Tcells | Myeloid|Tcells | Tcells|Myeloi
CPI-SS03A0C857B | FAS_FASLG | simple:P25445 | simple:P48023 | ENSG00000026103 | ENSG00000117560 | curated | True | False | 0.062
CPI-SS028784FC6 | HLA-DPA1_TNFSF9 | simple:HLADPA1 | simple:P41273 | ENSG00000231389 | ENSG00000125657 | InnateDB-All | True | False | 0.062
CPI-SS0795802F6 | CCL4_SLC7A1 | simple:P13236 | simple:P30825 | ENSG00000275302 | ENSG00000139514 | IMEx,IntAct | True | False | 0.062
CPI-SS03105D292 | CSF1_SLC7A1 | simple:P09603 | simple:P30825 | ENSG00000184371 | ENSG00000139514 | I2D | True | False | 0.062
CPI-SS08B7D54A3 | TNF_FAS | simple:P01375 | simple:P25445 | ENSG00000232810 | ENSG00000026103 | InnateDB-All | True | False | 0.062
CPI-SS02770D68F | NOTCH2_JAG2 | simple:Q04721 | simple:Q9Y219 | ENSG00000134250 | ENSG00000184916 | curated | False | False | 0.062
CPI-SS05FEE05CB | NOTCH4_JAG2 | simple:Q99466 | simple:Q9Y219 | ENSG00000204301 | ENSG00000184916 | curated | False | False | 0.062

c


Interaction Explorer
gene_name | uniprot | is_complex | protein_name | complex_name | id_cp_interaction | Tcells | Myeloid | NKcells_0 | NKcells_1
FASLG | P48023 | False | TNFL6_HUMAN | | CPI-SS03A0C857B | 0.0 | 0.0 | 0.581 | 0.425
FAS | P25445 | False | TNR6_HUMAN | | CPI-SS03A0C857B | 0.0 | 0.0 | 0.085 | 0.0
Column info
protein_name: molecule name
gene_name: Ensembl id
name: Uniprot id
is_complex: i) single- homodimer; ii) complex- heterodimer
complex_name: name of the complex
Id_cp_interaction: CellPhoneDB interaction id
values for each cluster: Mean of the value.
Close

d


Results Explorer
For 10 iterations and threshold 0.1
Private results (Only you can access here)
View access code
Column info
Customize view options
Download
Make public
Delete
Results
Plots
Plot type
Dot plot
Description
Columns
Select one or more columns
Rows
Select one or more rows
Request plot
KIR2DL3_FAM3C
HLA-C_FAM3C
PVR_TNFSF9
PVR_TIGIT
SPP1_CD44
PVR_CD96
−log10 P
0
1
2
3
log2 mean (molecule 1, molecule 2)
1
0
−1
−2

Table 2 | Description of the output files means.csv, pvalues.csv and significant_means.csv

Identifier	Definition	Output file	Example
id_cp_interaction	Unique CellPhoneDB identifier for each interaction stored in the database	means.csv; pvalues.csv; significant_means.csv	CPI-SS096F3E0F2
interacting_pair	Name of the interacting pairs separated by '|'	means.csv; pvalues.csv; significant_means.csv	JAG2|NOTCH4
partner A or B	Identifier for the first interacting partner (A) or the second (B). It could be: UniProt (prefix 'simple:') or complex (prefix 'complex:')	means.csv; pvalues.csv; significant_means.csv	simple:Q9Y219
gene A or B	Gene identifier for the first interacting partner (A) or the second (B). The identifier will depend on the input user list	means.csv; pvalues.csv; significant_means.csv	ENSG00000184916
secreted	True if one of the partners is secreted	means.csv; pvalues.csv; significant_means.csv	FALSE
Receptor A or B	True if the first interacting partner (A) or the second (B) is annotated as a receptor in our database	means.csv; pvalues.csv; significant_means.csv	FALSE
annotation_strategy	Curated if the interaction was annotated by the CellPhoneDB developers. Otherwise, the name of the database where the interaction has been downloaded from	means.csv; pvalues.csv; significant_means.csv	curated
is_integrin	True if one of the partners is an integrin	means.csv; pvalues.csv; significant_means.csv	FALSE
rank	Total number of significant *P* values for each interaction divided by the number of cell type–cell type comparisons	significant_means.csv	0.25
means	Mean values for all the interacting partners: mean value refers to the total mean of the individual partner average expression values in the corresponding interacting pairs of cell types. If one of the mean values is 0, then the total mean is set to 0	means.csv	0.53
p.values	*P* values for all the interacting partners: p.value refers to the enrichment of the interacting ligand–receptor pair in each of the interacting pairs of cell types	pvalues.csv	0.01
significant_mean	Significant mean calculation for all the interacting partners. If p.value < 0.05, the value will be the mean. Alternatively, the value is set to 0	significant_means.csv	0.53

Webserver

Steps 16–19, using the webserver: ~1 h for dataset of ~10 GB, 10,000 cells; depends on how many jobs are running in parallel and the resources available at the moment

Anticipated results

We originally applied CellPhoneDB to study maternal–fetal communication at the decidual–placental interface during early pregnancy[5]. The results obtained with our new CellPhoneDB v.2.0 using subsampling were consistent with our original conclusions (Fig. 3). Here, we provide an explanation of the results generated in this example.

Without running statistical inference of ligand–receptor interactions, only means.csv and deconvoluted.csv are generated (see Tables 2 and 3 for descriptions of the output files). The means.csv file contains mean values for each ligand–receptor interaction. The deconvoluted.csv file gives additional information for each of the interacting partners. This is important because some of the interacting partners are heteromers. In other words, multiple molecules have to be expressed in the same cluster in order for the interacting partner to be functional. If the user uses the statistical inference approach, additional pvalues.csv and significant_means.csv files are generated that contain the values for the significant interactions.

Importantly, interactions are not symmetric. In other words, when testing a ligand–receptor pair A_B between clusters X_Y, the expression of partner A is considered within the first cluster (X), and the expression of partner B within the second cluster (Y). Therefore, X_Y and Y_X represent different comparisons and will have different *P* values and means.

Table 3 | Description of the output file deconvoluted.csv

Identifier	Definition	Output file	Example
gene_name	Gene identifier for one of the subunits that is participating in the interaction defined in the means.csv file. The identifier will depend on the input of the user list	deconvoluted.csv	JAG2
uniprot	UniProt identifier for one of the subunits that is participating in the interaction defined in the means.csv file	deconvoluted.csv	Q9Y219
is_complex	True if the subunit is part of a complex. Single if it is not, complex if it is	deconvoluted.csv	FALSE
protein_name	Protein name for one of the subunits that is participating in the interaction defined in means.csv file	deconvoluted.csv	JAG2_HUMAN
complex_name	Complex name if the subunit is part of a complex. Empty if not	deconvoluted.csv	a10b1 complex
id_cp_interaction	Unique CellPhoneDB identifier for each of the interactions stored in the database	deconvoluted.csv	CPI-SS0DB3F5A37
mean	Mean expression of the corresponding gene in each cluster	deconvoluted.csv	0.9

Reporting Summary

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

Data availability

The decidua and placenta datasets can be downloaded from ArrayExpress, with experiment code E-MTAB-6701.

Code availability

The CellPhoneDB code is available at https://github.com/Teichlab/cellphonedb. It can also be downloaded from https://cellphonedb.org/downloads. The code in this paper has been peer-reviewed.

References

1. Svensson, V., Vento-Tormo, R. & Teichmann, S. A. Exponential scaling of single-cell RNA-seq in the past decade. *Nat. Protoc.* **13**, 599–604 (2018).
2. Ramilowski, J. A. et al. A draft network of ligand-receptor-mediated multicellular signalling in human. *Nat. Commun.* **6**, 7866 (2015).
3. Hie, B., Cho, H., DeMeo, B., Bryson, B. & Berger, B. Geometric sketching compactly summarizes the single-cell transcriptomic landscape. *Cell Syst.* **8**, 483–493.e7 (2018).
4. Shannon, P. et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**, 2498–2504 (2003).
5. Vento-Tormo, R. et al. Single-cell reconstruction of the early maternal-fetal interface in humans. *Nature* **563**, 347–353 (2018).
6. Braga, F. A. V. et al. A cellular census of human lungs identifies novel cell states in health and in asthma. *Nat. Med.* **25**, 1153–1163 (2019).
7. Stewart, B. J. et al. Spatiotemporal immune zonation of the human kidney. *Science* **365**, 1461–1466 (2019).
8. Popescu, D.-M. et al. Decoding the development of the blood and immune systems during human fetal liver haematopoiesis. *Nature* **574**, 365–371 (2019).
9. Davidson, S. et al. Single-cell RNA sequencing reveals a dynamic stromal niche within the evolving tumour microenvironment. Preprint at *bioRxiv*: https://doi.org/10.1101/467225 (2018)
10. Skelly, D. A. et al. Single-cell transcriptional profiling reveals cellular diversity and intercommunication in the mouse heart. *Cell Rep.* **22**, 600–610 (2018).
11. Camp, J. G. et al. Multilineage communication regulates human liver bud development from pluripotency. *Nature* **546**, 533–538 (2017).
12. Pavličev, M. et al. Single-cell transcriptomics of the human placenta: inferring the cell communication network of the maternal-fetal interface. *Genome Res.* **27**, 349–361 (2017).
13. Puram, S. V. et al. Single-cell transcriptomic analysis of primary and metastatic tumor ecosystems in head and neck cancer. *Cell* **171**, 1611–1624.e24 (2017).

14. Suryawanshi, H. et al. A single-cell survey of the human first-trimester placenta and decidua. *Sci. Adv.* **4**, eaau4788 (2018).
15. Zhou, J. X., Taramelli, R., Pedrini, E., Knijnenburg, T. & Huang, S. Author correction: Extracting intercellular signaling network of cancer tissues using ligand-receptor expression patterns from whole-tumor and single-cell transcriptomes. *Sci. Rep.* **8**, 17903 (2018).
16. Cohen, M. et al. Lung single-cell signaling interaction map reveals basophil role in macrophage imprinting. *Cell* **175**, 1031–1044.e18 (2018).
17. Kumar, M. P. et al. Analysis of single-cell rna-seq identifies cell-cell communication associated with tumor characteristics. *Cell Rep.* **25**, 1458–1468.e4 (2018).
18. Halpern, K. B. et al. Paired-cell sequencing enables spatial gene expression mapping of liver endothelial cells. *Nat. Biotechnol.* **36**, 962–970 (2018).
19. Joost, S. et al. Single-cell transcriptomics of traced epidermal and hair follicle stem cells reveals rapid adaptations during wound healing. *Cell Rep.* **25**, 585–597.e7 (2018).
20. Boisset, J.-C. et al. Mapping the physical network of cellular interactions. *Nat. Methods* **15**, 547–553 (2018).
21. Lubeck, E., Coskun, A. F., Zhiyentayev, T., Ahmad, M. & Cai, L. Single-cell in situ RNA profiling by sequential hybridization. *Nat. Methods* **11**, 360–361 (2014).
22. Ståhl, P. L. et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science* **353**, 78–82 (2016).
23. Rodriques, S. G. et al. Slide-seq: a scalable technology for measuring genome-wide expression at high spatial resolution. *Science* **363**, 1463–1467 (2019).
24. Svensson, V. A method for transcriptome-wide gene expression quantification in intact tissues. *Immunol. Cell Biol.* **97**, 439–441 (2019).
25. Finn, R. D. et al. Pfam: the protein families database. *Nucleic Acids Res.* **42**, D222–D230 (2014).
26. Proteomics Standards Initiative. Proteomics Standards Initiative common query interface in *Encyclopedia of Systems Biology* (eds Dubitzky, W., Wolkenhauer, O., Cho, K.H. & Yokota, H.) 1798–1798 (Springer, 2013): https://doi.org/10.1007/978-1-4419-9863-7_101243
27. Protein interaction data curation: the International Molecular Exchange (IMEx) consortium. *Nat. Methods* **9**, 345–350 (2012).
28. Orchard, S. et al. The MIntAct project-IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res.* **42**, D358–D363 (2014).
29. Breuer, K. et al. InnateDB: systems biology of innate immunity and beyond-recent updates and continuing curation. *Nucleic Acids Res.* **41**, D1228–D1233 (2013).
30. Clerc, O. et al. MatrixDB: integration of new data with a focus on glycosaminoglycan interactions. *Nucleic Acids Res.* **47**, D376–D381 (2019).
31. Licata, L. et al. MINT, the molecular interaction database: 2012 update. *Nucleic Acids Res.* **40**, D857–D861 (2012).
32. Brown, K. R. & Jurisica, I. Unequal evolutionary conservation of human protein interactions in interologous networks. *Genome Biol.* **8**, R95 (2007).
33. Bachelerie, F. et al. International Union of Pharmacology. LXXXIX. Update on the extended family of chemokine receptors and introducing a new nomenclature for atypical chemokine receptors. *Pharmacol. Rev.* **66**, 1–79 (2014). erratum **66**, 467 (2014).
34. Satija, R. et al. Spatial reconstruction of the single-cell gene expression data. *Nat. Biotechnol.* **33**, 495–502 (2015).
35. Wolf, F. A. et al. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).

Acknowledgements

We thank K. Meyer and M. Stubbington for scientific discussions; P. Porras for advice on querying the IMEx database; L. Garcia-Alonso and K. Polanski for carefully reading the manuscript; G.J. Wright, L. Wood and G. Graham for advice on protein–protein interactions; and J. Eliasova and A. Hupalowska for help with the illustrations. We are grateful to A. Lopez and YDEVS members for their help with the webserver and the implementation of the code in GitHub, as well as to all the Teichmann lab and Vento-Tormo lab members for their fruitful advice. The project was supported by Wellcome Sanger core funding (WT206194) and a Wellcome Strategic Support Science award (211276/Z/18/Z).

Author contributions

M.E., S.A.T. and R.V.-T. conceived and developed the protocol and wrote the manuscript. M.V.-T. developed the database, implemented the code in the webserver and GitHub and contributed to writing the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41596-020-0292-x.

Correspondence and requests for materials should be addressed to R.V-T.

Peer review information *Nature Protocols* thanks Evangelia Petsalaki 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.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 24 June 2019; Accepted: 31 December 2019;
Published online: 26 February 2020

Related links
Key references using this protocol
Vento-Tormo, R. et al. *Nature* **563**, 347–353 (2018): https://doi.org/10.1038/s41586-018-0698-6
Stewart, B. et al. *Science* **365**, 1461–1466 (2019): https://doi.org/10.1126/science.aat5031
Popescu, D. et al. *Nature* **574**, 365–371 (2019): https://doi.org/10.1038/s41586-019-1652-y

nature research

Corresponding author(s): Roser Vento-Tormo

Last updated by author(s): 25.06.2019

Reporting Summary

Nature Research 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 Research policies, see Authors & Referees 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	Software used include: R, Python 3, SQLAlchemy, PostgreSQL, 10X Genomics' Cell Ranger, Seurat, Illustrator.
Data analysis	Software used include: R, Python 3, SQLAlchemy, PostgreSQL, 10X Genomics' Cell Ranger, Seurat, Illustrator.

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/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research 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 list of figures that have associated raw data
- A description of any restrictions on data availability

Data is available in ArrayExpress:
Experiment: E-MTAB-6701

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

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
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data

Methods

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