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]

[1] Wellcome Sanger Institute, Wellcome Genome Campus, Cambridge CB10 1SD, UK
[2] YDEVS software development, Valencia, 46009, Spain.
[3] Theory of Condensed Matter Group, Cavendish Laboratory, University of Cambridge, JJ Thomson Ave, Cambridge CB3 0EH, UK

Corresponding authors

Correspondence to Roser Vento-Tormo rv4@sanger.ac.uk

Author's emails

me5@sanger.ac.uk
miquel@ydevs.com
st9@sanger.ac.uk
rv4@sanger.ac.uk

EDITORIAL SUMMARY CellPhoneDB combines an interactive database and a statistical framework for the exploration of ligand-receptor interactions inferred from single cell transcriptomics measurements.

Abstract

Cell-cell communication mediated by ligand-receptor complexes is crucial for coordinating diverse biological processes, such as development, differentiation and responses to infection. In order to understand 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, the 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 v2.0 is an updated version of our resource that incorporates additional functionalities to allow users to introduce new interacting molecules and reduces the time and resources needed to interrogate large datasets. CellPhoneDB v2.0 is publicly available at https://github.com/Teichlab/cellphonedb and as a user-friendly web interface at http://www.cellphonedb.org/ and can be used by both experts and researchers with little experience in computational genomics. In our protocol, we demonstrate how to reveal meaningful biological interactions with CellPhoneDB v2.0 using

published data sets. This protocol typically takes ~2 hours to complete, from installation to statistical analysis and visualisation, for a dataset of ~10GB, 10000 cells and 19 cell types using 5 threads.

Introduction

Complex extracellular responses start with the binding of a ligand to their cognate receptor and the activation of specific cell signalling pathways. Mapping these ligand-receptor interactions is fundamental to understanding cellular behaviour and response to neighbouring cells. With the exponential growth of single-cell RNA sequencing (scRNAseq)[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 scRNAseq requires both the annotation of the complex ligand-receptor relationships from the literature, and a statistical method that integrates the resource with scRNAseq 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 (Figure 1). This is crucial, as cell-cell communication relies on multi-subunit protein complexes that go beyond the binary representation used in most databases and studies[2]. In order 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 scRNAseq data. After uploading the scRNAseq data and performing subsampling using geometric sketching[3] (Figure 2a), cells with the same cluster annotation are pooled together as a cell state. We derive enriched ligand-receptor interactions between two cell states based on 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 is calculated (Figure 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 (Figure 2c and Figure 2d). This predicts molecular interactions between cell populations via specific protein complexes and generates potential cell–cell communication networks which can be visualised using intuitive tables and plots (Figure 2e). Specificity of the ligand-receptor interaction is important, as some of the ligand-receptor pairs are ubiquitously expressed by the cells in a tissue, and therefore 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 (larger than 10GB). Compared to the original CellPhoneDB platform, our updated version CellPhoneDB v2.0 has incorporated new features, such as

subsampling of the original dataset to enable the fast querying of large datasets (geometric sketching [2]) or the visualisation of the results using intuitive tables, plots and network files that can be directly uploaded into Cytoscape (https://cytoscape.org/). In addition, we now offer the user the possibility to use 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[4]. Briefly, our analysis revealed new immunoregulatory mechanisms and cytokine signalling networks existing between the cells in the maternal-fetal interface, which guarantee the coexistence of both the mother and developing fetus (Figure 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[5]. In the context of the kidney, cell-cell interaction analysis helped to reveal epithelium-immune crosstalk that coordinates recruitment of antibacterial macrophages and neutrophils to regions in the kidney most vulnerable to infections[6]. 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 molecules VCAM1, ITGB1 and ITGA4, all of them known to be important in haematopoiesis[7].

Furthermore, even though CellPhoneDB is created using human-specific ligand-receptor interactions, it can be easily applied on mouse datasets by mapping human genes onto 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 where multiple immunosuppressive mechanisms coexist within a heterogeneous stromal compartment[8].

Comparison with other approaches

There are now several other published methods 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 based on the expression level of the ligand and receptor. In these methods, only the interaction pairs that pass a certain threshold of cells expressing the specific interactors in the respective cell populations are selected for the downstream analysis[9-14]. For example, in addition to filtering based on

expression level, Cohen *et al.*[15] used hierarchical clustering with Spearman correlation to identify ligand-receptor modules and construct an interaction graph. Others, such as Kumar *et al*[16], 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.*[17] 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.*[18] 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.*[19] 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 compared with the numbers in the background model.

A major strength of CellPhoneDB compared to most other databases is that it takes into account the structural composition of ligands and receptors, which is important as ligand-receptor interactions often involve multiple subunits. This is particularly clear for protein families like many of the cytokine families, where receptors share structural subunits, and the affinity of the ligand is determined by the specific combination of the receptor subunits (Figure 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 which involve heteromers, and 163 of them comprise cytokines.

Limitations of the protocol

Our database, while 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 into a downstream signalling event. 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. Additionally, we use permutations to generate a null hypothesis, and this can be time-consuming and resource-intensive with large datasets (for example datasets with millions of cells). To address this, we introduced a subsampling approach, which preserves the heterogeneity of the dataset and reduces speed and memory requirements (1 hour vs 1.5 hours for a dataset of 10000 cells). Finally, our tool infers potential interactions using transcriptomics data without considering 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. [20-23]).

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. In order 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” (Figure 4).

“gene_input”

Mandatory fields: “gene_name”; “uniprot”; “hgnc_symbol” and “ensembl”

This file is crucial 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”); iiii) HUGO nomenclature committee symbol (HGNC) (“hgnc_symbol”) and iv) gene ensembl identifier (ENSG) (“ensembl”). In order 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 towards different UniProt identifiers.
- UniProt and Ensembl lists are also merged by their UniProt identifier but this information is only used when the UniProt or Ensembl identifier is missing in the original list merged by gene name.
- If the same gene name points towards 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: “uniprot”; “protein_name”
Optional fields: “transmembrane”; “peripheral”; “secreted”; “secreted_desc”; “secreted_highlight”; “receptor”; “receptor_desc” ; “integrin”; “pfam”; “other”; “other_desc”; “tags”; “tags_ description”; “tags_reason”; “pfam”

Two types of input are needed to create this file: i) systematic input using UniProt annotation, and ii) manual input using curated annotation both from 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 below the “Systematic input from other databases” section for interaction_list). Importantly, if a protein id is present in both the curated and systematic inputs, the curated information always has priority over the systematic one.

Optional fields are organised in the categories described below:

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 literature information. "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. Please 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 as they are not directly involved in the recognition of the ligand (eg. Co-receptors) or they require more specialised 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 add a brief description of the excluded protein.

Protein family Information about the family of the protein is downloaded from https://pfam.xfam.org/ [24] 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_ description" and "tags_reason".

There are three options for the "tags" field: (a) 'N/A': protein matches with UniProt description; (b) 'To_add': addition of secreted and/or plasma membrane protein annotation; and (c) '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: (a) 'extracellular_add': manual annotation of the protein as plasma membrane; (b) 'peripheral_add': manual annotation of the protein as peripheral; (c) 'secreted_add': manual annotation of the protein as secreted; (d) 'secreted_high': manual annotation of the protein as cytokine, hormone, growth factors or other immune-related protein (secreted_highlight); (e) '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: "complex_name"; "uniprot1, 2, etc."
Optional fields: "transmembrane"; "peripheral"; "secreted"; "secreted_desc"; "secreted_highlight"; "receptor"; "receptor_desc" ; "integrin"; "other"; "other_desc"; "pdb_id"; "pdb_structure" ; "stoichiometry"; "comments_complex"

Literature and UniProt descriptions were reviewed to annotate heteromeric proteins, which were defined as cases when 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",etc.) 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 description 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: "partner_a"; "partner_b"; "annotation_strategy"; "source"
Optional fields: "protein_name_a"; "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, yet 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: (a) IUPHAR (http://www.guidetopharmacology.org/): binary interactions only, (b) InnateDB (https://www.innatedb.com/): interactions involving cytokines, hormones and growth factors interactions, and (c) 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[25]. The IMEx[26], IntAct [27], InnateDB [28], UCL-BHF (https://www.ucl.ac.uk/cardiovascular/research/pre-clinical-and-fundamental-science/functional-gene-annotation/manual-curation/protein), MatrixDB [29], MINT [30], I2D [31], UniProt, 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: a) interactions where one of the components is part of a complex (see "complex_input" list in the above section); b) interactions which 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: a) uniprot_1: name of the interacting partner A that is going to be excluded; b) uniprot_2: name of the interacting partner B that is going to be excluded; c) name.1: name of the protein to be excluded corresponding to uniprot_1; d) name.2: name of the protein to be excluded corresponding to uniprot_2; e) comments: information about the exclusion of the protein.

Homomeric complexes - proteins interacting with themselves - are excluded from the systematic analysis. Importantly, in cases where 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 on membrane receptors were used to annotate ligand–receptor interactions and the International Union of Pharmacology annotation[32] 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 from 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 v2.0 allows users to create their own lists of genes, curated proteins, complexes and interactions. In order to do so, the format of the users' lists must be compatible with the input files. Users can run the analysis using 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 allow analysis on potentially large count matrices to be performed in parallel. This requires an efficient database design, including optimisation for query times, indices and related strategies. All application code is open source and uploaded to github and www.cellphonedb.org .

The database consists of 6 main tables: gene_table; protein_table; multidata_table; interaction_table; complex_table; complex_composition_table (Supplementary Figure 1).

All tables have an incremental numeric unique identifier with the structure id_{table_name} and one or more foreign keys, with 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 interactions 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 shared information 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 if 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* corresponds 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*, *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* corresponds 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*.

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 relations, 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 number of complex components) for internal optimization purposes. Supplementary Figure 2 represents an example of two *complex_input* rows with two and four protein components, respectively.

interaction_table

This table stores the interactions data from *interaction_input* file. The following columns to represent the data are used: *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 key *multidata_1_id* and *multidata_2_id* respectively with 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, i.e. 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 based on expression of a receptor by one cell type and a ligand by another cell type. To identify biologically relevant interactions, we look for the 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 which are as high as or higher than the actual mean. Based on 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 the multi-subunit heteromeric complexes, we require that all subunits of the complex are 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 of the number of cells obtained from scRNA-seq experiments[1]. Large-scale datasets can profile hundreds of thousands cells, which presents a challenge for the existing analysis methods in terms of both computer memory usage and runtime. In order 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:

- META file: The annotation file is generated by the users after they have annotated each cluster identified by scRNA-seq data (for example by using packages such as Seurat[33], SCANPY[34]). The file contains two columns: "Cell" indicating the name of the cell, and "cell_type" indicating the name of the cluster considered. Formats accepted: .csv, .txt, .tsv, .tab, pickle.
- COUNTS file: scRNA-seq count data containing gene expression values where rows are genes presented with gene names identifiers (Ensembl IDs, gene names or hgnc_symbol annotation) and columns are cells. We recommend using normalised count data. Importantly, the user needs to specify whether the data was log-transformed when using the subsampling option. Format accepted: .csv or .txt, .tsv, .tab, pickle.

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

Software:

- Python 3.5 or higher
- SQLAlchemy
- SQLite
- Preprocessing of the raw expression data to generate the input files can be done using packages such as Seurat[33], SCANPY[34], or any other pipeline that the user prefers.

Hardware

- Linux or MAC OS

Equipment Setup

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

Some of the most standard packages for scRNA-seq analysis include Seurat[33] and SCANPY[34]. 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 v2.0. We recommend using normalised count data as input.

For example, using the R package Seurat[33], the count input file can be obtained by taking the raw expression data from the Seurat object and applying the normalisation manually. The user can also normalise using their preferred method for normalisation.

```
# take raw data and normalise 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 meta 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[34] data object:

```
import pandas as pd
import scanpy.api as sc

# data after filtering and normalising
adata = sc.read(adata_filepath)
# we recommend using the normalised 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 meta 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 larger than 10GB).

Procedure

Installation
Timing: 5-10 min

CRITICAL: Steps 1-15 describe the Python implementation of CellPhoneDB v2.0, while 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 by 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 CellPhone DB v2.0

```
pip install cellphonedb
```

Running with statistical analysis
Timing: 1,5 hours for dataset of ~10GB, 10000 cells, threads=5

4. Activate the virtual environment if you have not activated it in Step 2.

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

5. Run CellPhoneDB v2.0 in statistical analysis mode using the input file names (including full path to the files) for metadata and counts (see Equipment Setup)

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

?TROUBLESHOOTING

Optional parameters:

```
--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: % of cells expressing the specific ligand or
receptor
--result-precision: Number of decimal digits in results
[default: 3]
--counts-data: [ensembl | gene_name | hgnc_symbol] Type of
gene identifiers in the counts data
--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 pvalues result file
[default: pvalues.txt]
--debug-seed: Debug random seed -1. To disable it please 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
yourcountsfile.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
yourcountsfile.txt --output-path=custom_folder
```

Running with subsampling and statistical analysis
Timing: 1 hour for dataset of ~10GB, 10000 cells subsampled to 5000, 19 cell types, threads=5

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

6. Run CellPhoneDB v2.0 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
yourcountsfile.txt --subsampling --subsampling-log true
```

The parameters are same as described in Step 5, in addition to the following subsampling specific parameters:

```
--subsampling-log: Enable 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: ~5min for dataset of ~10GB, 10000 cells, 19 cell_types

7. Run CellPhoneDB v2.0 in normal mode using the input files for metadata and counts and 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

Visualisation
Timing: seconds to minutes

CRITICAL The users can visualise the results from the analysis using dot plots and heatmaps.

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

```
cellphonedb plot dot_plot
```

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, e.g. 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:

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

Example content of rows.txt file:

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

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

```
cellphonedb plot heatmap_plot meta_data
```

Heatmap plot specific parameters:

```
--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 official available data. This data will be downloaded using the `--database` parameter.

10. CellPhoneDB v2.0 databases can be updated from a remote repository. Available versions of the database can be listed and downloaded to be used. This is relevant as users may have used one 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’:

    ```
    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:

```
--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, eg: 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, please use the --database parameter when executing the “cellphonedb method” command. E.g:

```
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: Comma separated file (use mandatory columns!) with interactions to add/correct.

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

  Result: 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: Comma separated file (use mandatory columns!) with interactions to use.

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

  Result: New database file with only user custom interactions.

- To correct any protein data:
  Input: your_custom_protein_file.csv: Comma separated file (use mandatory columns!) with proteins to overwrite.

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

  Result: New database file with CellPhoneDB interactions and user 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: Comma separated file (use mandatory columns!) with interactions to add/correct.

    your_custom_protein_file.csv: Comma separated file (use mandatory columns!) with proteins to overwrite.

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

  Result: New database file with CellPhoneDB interactions and user custom interactions. For duplicated interactions or proteins, user list overwrites CellPhoneDB original data.

- To update remote sources (UniProt, IMEx, ensembl, etc.)

Input:

- your_custom_interaction_file.csv: Comma separated file (use mandatory columns!) with interactions to add/correct.

- your_custom_protein_file.csv: Comma separated file (use mandatory columns!) with proteins to overwrite.

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

Some lists can be downloaded from original sources while creating the database, e.g. uniprot, or 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 "generate" command.

Result: New database file with the CellPhoneDB interactions and user 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 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 hour for dataset of ~10GB, 10000 cells, however this 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 for the user to define analysis parameters before submission. Downstream calculations are performed on the application's servers,

rendering the information of ligand and receptor expression, and visualisation diagrams once analysis is complete (Figure 5).

16. Go to the tab “Exploring your scRNAseq” and input your meta and count input files (Please see section **Input data files in Equipment**).
17. Provide an email address if you would like to get an update when the process finishes (Figure 5a).
18. The “significant_means” results table will appear as in Figure 5c (please see the next section – Anticipated results: formats of files). You can change the current view by clicking on the ”Data Shown” button (Figure 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 (Figure 5c).
19. Go to the tab “Plots” and pick the type of plot you would like to produce. For plotting dot plots, please select the columns and rows you need (Figure 5d).

The online results viewer allows you to select which columns you wish to display in each table. This option is quite useful as an aid to visualize the results.

Anticipated results

We originally applied CellPhoneDB to study the maternal-fetal communication at the decidual-placental interface during early pregnancy[4]. The results obtained with our new CellPhoneDB v2.0 using subsampling were consistent with our original conclusions (Figure 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 “desconvoluted.csv” are generated. 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 as 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 containing 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.

Timing

Python package, Steps 1-15, ~2 hours

Step 1-3, Installation, 5 - 10 min

Step 5, Running with statistical method, 1,5 hours for dataset of ~10GB, 10000 cells, threads=5
Step 6, Subsampling and statistical method, 1 hour for dataset of ~10GB, 10000 cells subsampled to 5000, 19 cell_types, threads=5
Step 7, Analysis without the statistical method, ~5min for dataset of ~10GB, 10000 cells, 19 cell_types
Step 8-9, Visualisation, seconds to minutes
Step 10-13, Using different database versions, seconds to minutes
Step 14, Generating user-specific database, ~10 min
Webserver, Step 16 - 19, ~1 hour for dataset of ~10GB, 10000 cells, however this depends on how many jobs are running in parallel and the resources available at the moment.

Code availability

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

Data availability

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

Acknowledgments

We thank Kerstin Meyer and Mike Stubbington for scientific discussions, Pablo Porras for advice on querying the IMEx database, Luz Garcia-Alonso and Krzysztof Polanski for carefully reading the manuscript, Gavin J Wright, Laura Wood and Gerard Graham for advice on protein-protein interactions and Jana Eliasova and Ania Hupalowska for help with the illustrations. We are grateful to Adria Lopez and YDEVS members for their help with the webserver and the implementation of the code in github, and all the Teichmann lab and Vento-Tormo lab members for their fruitful advice. The project was supported by Wellcome Sanger core funding (no. WT206194) and a Wellcome Strategic Support Science award (no 211276/Z/18/Z).

Author contribution

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 web server and github and contributed to writing the manuscript.

Competing interests

The authors declare no competing interests.

Figures

Figure 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". **a**, 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 1396 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 is a total of 250 interactions that involve integrins.

Figure 2. Overview of the statistical method framework used to infer ligand–receptor complex specific to two cell types from single-cell transcriptomics data. **a,** CellPhoneDB input data consist of 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 based on 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 permute 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 based on the proportion of the means which are as or higher than the actual mean. **e,** Ligand-receptor pairs are ranked based on their total number of significant p-values across the cell populations. Visualisation of the results using intuitive tables and plots is provided in the web interface. R1, example receptor R1; L1, example ligand L1.

Figure 3. Example dataset run with CellPhoneDB and CellPhoneDB v2.0. **a,** Overview of selected ligand–receptor interactions using CellPhoneDB on the decidua dataset from [3]; 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 colour. **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 v2.0 with subsampling on the decidua dataset. P values indicated by circle size, scale on right. The means of the average expression level of interacting molecule 1 in cluster 1 and interacting molecule 2 in cluster 2 are indicated by colour. ⅓ of the dataset was subsampled. **d**, Heatmap showing the total number of interactions between cell types in the decidua dataset obtained with CellPhoneDB v2.0 with subsampling. ⅓ of the dataset was subsampled. **e**. An example of significant interactions involving complexes identified by CellPhoneDB in the placenta dataset [3]. 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 macrophages cluster and the two subunits of the IL1 receptors (IL1R1 and IL1RAP) are co-expressed in the extravillous trophoblasts (EVT). SCT,

syncytiotrophoblast; VCT, villous cytotrophoblast; F, fibroblasts; HB, Hofbauer cells; M, macrophages, Endo, endothelial cells.

Figure 4. Diagram showing how lists are generated. Basic steps in the generation of lists to populate the tables in CellPhoneDB.

Figure 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 visualisation page.

Supplementary Figure Legends:

Supplementary Figure 1. Diagram of the database structure. **a)** database schema, **b)** protein_input/complex_input storage in the CellPhoneDB database tables. The multidata entity stores fields common to complex_input and protein_input. This makes it easier and faster for the user to perform interaction queries because interaction_table is only related to multidata_table. All non-common fields are stored in either protein_table or complex_table. Complex fields are stored in complex_composition_table. The is_complex and total_protein field are created for optimization purposes.

Supplementary figure 2. Example of complex_input components stored in CellPhoneDB. An example of two complex_input rows with two and four components

Tables

Table 1. Troubleshooting table.
Table 2. Description of the output files means.csv, pvalues.csv and significant_means.csv.
Table 3. Description of the output file deconvoluted.csv.

Related links
Key references using this protocol
Vento-Tormo R. et al. Nature 563(7731), 347–353 (2018): https://doi.org/10.1038/s41586-018-0698-6
Stewart, B. et al. Science 365(6460), 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

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. *Nature Communications* 6, (2015).
3. Hie, B., Cho, H., DeMeo, B., Bryson, B. & Berger, B. Geometric Sketching Compactly Summarizes the Single-Cell Transcriptomic Landscape: Supplementary Information. doi:10.1101/536730
4. Vento-Tormo, R. *et al.* Single-cell reconstruction of the early maternal-fetal interface in humans. *Nature* **563**, 347–353 (2018).
5. Braga, F. A. V. *et al.* A cellular census of human lungs identifies novel cell states in health and in asthma. *Nature Medicine* (2019). doi:10.1038/s41591-019-0468-5
6. Stewart, B. J. *et al*. Spatiotemporal immune zonation of the human kidney. *Science* **365**, 1461-1466 (2019).
7. Popescu, D.-M. *et al.* Decoding the development of the blood and immune systems during human fetal liver haematopoiesis. *bioRxiv* 654210 (2019). doi:10.1101/654210
8. Davidson, S. *et al.* Single-cell RNA sequencing reveals a dynamic stromal niche within the evolving tumour microenvironment.*bioRxiv* doi:10.1101/467225
9. Skelly, D. A. *et al.* Single-Cell Transcriptional Profiling Reveals Cellular Diversity and Intercommunication in the Mouse Heart. *Cell Rep.* **22**, 600–610 (2018).
10. Camp, J. G. *et al.* Multilineage communication regulates human liver bud development from pluripotency. *Nature* **546**, 533–538 (2017).
11. 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).
12. 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).
13. Suryawanshi, H. *et al.* A single-cell survey of the human first-trimester placenta and decidua. *Sci Adv* **4**, eaau4788 (2018).

14. 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).
15. Cohen, M. *et al.* Lung Single-Cell Signaling Interaction Map Reveals Basophil Role in Macrophage Imprinting. *Cell* **175**, 1031–1044.e18 (2018).
16. 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).
17. Halpern, K. B. *et al.* Paired-cell sequencing enables spatial gene expression mapping of liver endothelial cells. *Nat. Biotechnol.* **36**, 962–970 (2018).
18. 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).
19. Boisset, J.-C. *et al.* Mapping the physical network of cellular interactions. *Nat. Methods* **15**, 547–553 (2018).
20. Lubeck, E., Coskun, A. F., Zhiyentayev, T., Ahmad, M. & Cai, L. Single-cell in situ RNA profiling by sequential hybridization. *Nature methods* **11**, 360–361 (2014).
21. Ståhl, P. L. *et al.* Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science* **353**, 78–82 (2016).
22. Rodriques, S. G. *et al.* Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science* **363**, 1463–1467 (2019).
23. Svensson, V. A method for transcriptome-wide gene expression quantification in intact tissues. *Immunol. Cell Biol.* **97**, 439–441 (2019).
24. Finn, R. D. *et al.* Pfam: the protein families database. *Nucleic Acids Res.* **42**, D222 (2014).
25. Proteomics Standards Initiative Common Query Interface. *Encyclopedia of Systems Biology* 1798–1798 (2013). doi:10.1007/978-1-4419-9863-7_101243
26. Orchard, S. *et al.* Protein interaction data curation: the International Molecular Exchange

(IMEx) consortium. *Nat. Methods* **9**, 345–350 (2012).

27. Orchard, S. *et al.* The MIntAct project--IntAct as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res.* **42**, D358–63 (2014).
28. Breuer, K. *et al.* InnateDB: systems biology of innate immunity and beyond--recent updates and continuing curation. *Nucleic Acids Res.* **41**, D1228–33 (2013).
29. Clerc, O. *et al.* MatrixDB: integration of new data with a focus on glycosaminoglycan interactions. *Nucleic Acids Res.* **47**, D376–D381 (2019).
30. Licata, L. *et al.* MINT, the molecular interaction database: 2012 update. *Nucleic Acids Res.* **40**, D857–61 (2012).
31. Brown, K. R. & Jurisica, I. Unequal evolutionary conservation of human protein interactions in interologous networks. *Genome Biol.* **8**, R95 (2007).
32. Bachelerie, F. *et al.* International Union of Basic and Clinical Pharmacology. [corrected]. LXXXIX. Update on the extended family of chemokine receptors and introducing a new nomenclature for atypical chemokine receptors. *Pharmacol. Rev.* **66**, 1–79 (2014).
33. Satija, R. *et al*. Spatial reconstruction of the single-cell gene expression data. *Nat. Biotechnol.* **33**, 495-502 (2015).
34. Wolf, F.A. *et al.* SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol*. **19**, 15 (018).

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 meta data might be switched or the genes are neither Ensembl IDs nor	Please use the meta data as first and the count data as second input parameter and provide a count table

		gene names	with genes presented as either Ensembl IDs or gene names
5, 6, 7	[ERROR] Invalid Counts data: 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 v2.0.	The cell IDs in the columns of the count data do not match the cell IDs in the cell_type column of the meta data	Please make sure that you have the same cell IDs in the columns of the count data and the cell_type column of the meta data
6	[ERROR] In order to perform subsampling you need to specify whether to log1p input counts or not: to do this specify in your command as --subsampling-log [true|false]	--subsampling-log needs to be specified (True or False)	Please provide BOOLEAN value to the --subsampling-log input parameter

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	means.csv	0.53

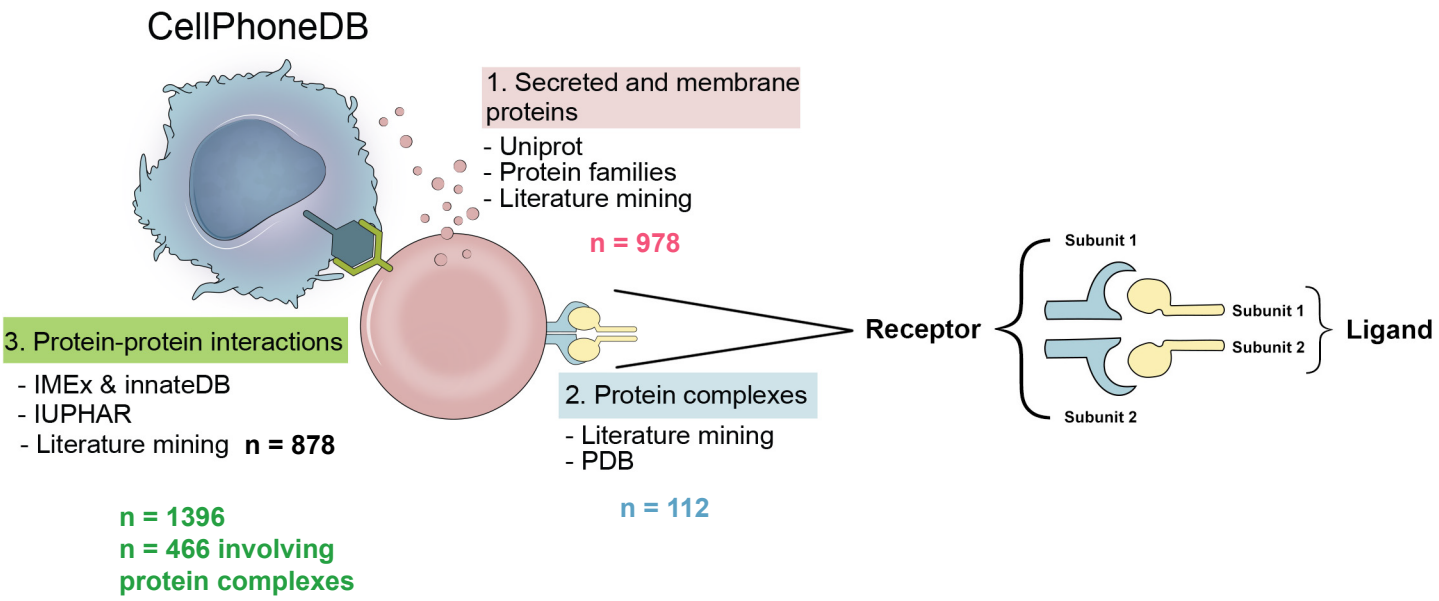
	the mean values is 0, then the total mean is set to 0.		
p.values	p-values for all the interacting partners: p.value refers to the enrichment of the interacting ligand-receptor pair in each of the interacting pairs of cell types.	pvalues.csv	0.01
significant_mean	Significant mean calculation for all 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

1105 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 “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	deconvoluted.csv	JAG2_HUMAN

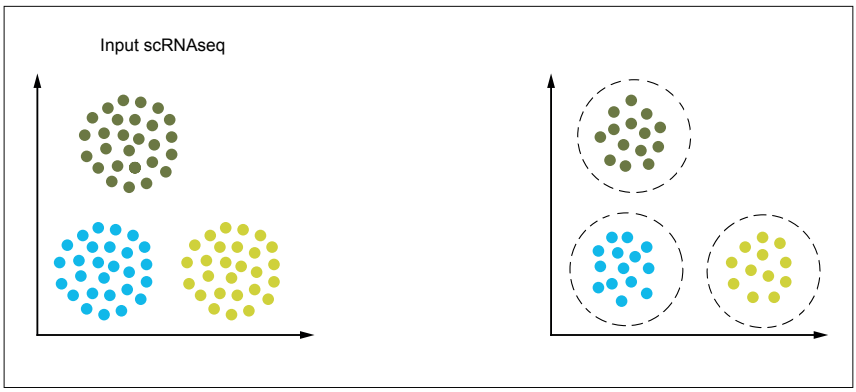
	one of the subunits that is participating in the interaction defined in “means.csv” file.		
complex_name	Complex name if the subunit is part of a complex. Empty if not.	deconvoluted.csv	a10b1 complex
id_cp_interaction	Unique CellPhoneDB identifier for each of the interactions stored in the database.	deconvoluted.csv	CPI-SS0DB3F5A37
mean	Mean expression of the corresponding gene in each cluster.	deconvoluted.csv	0.9

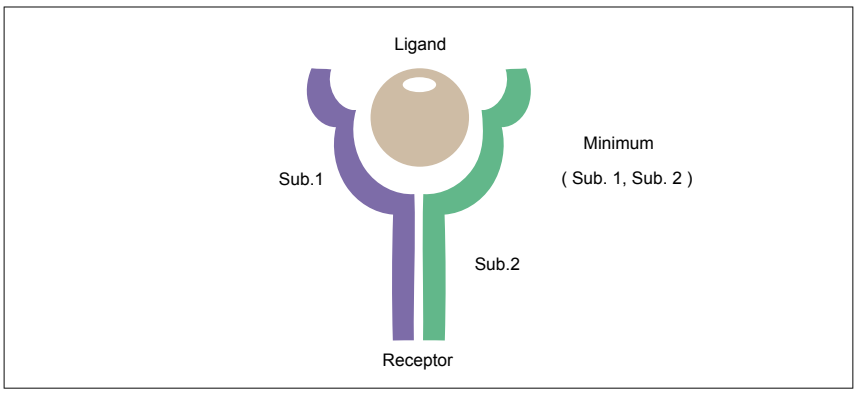
1110

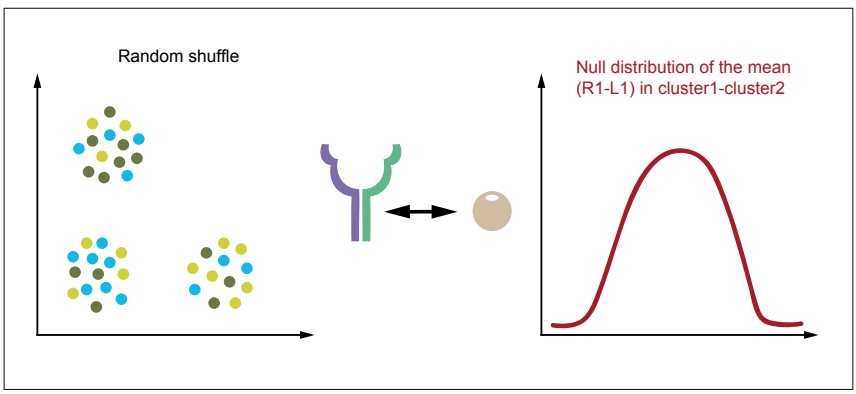


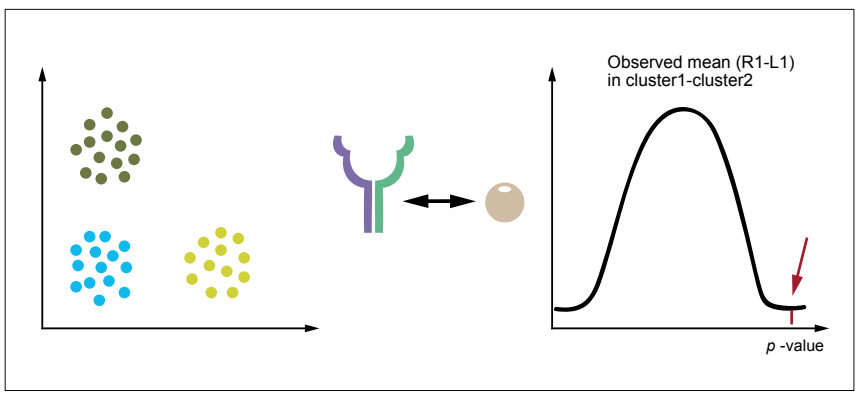
n = 1396
n = 466 involving protein complexes

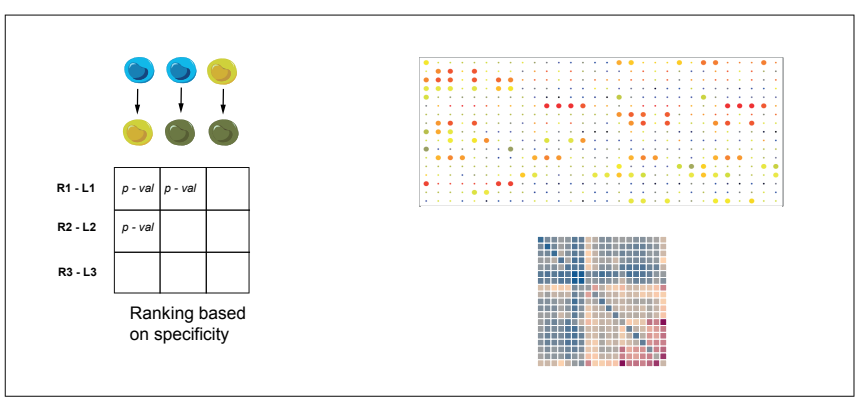
a

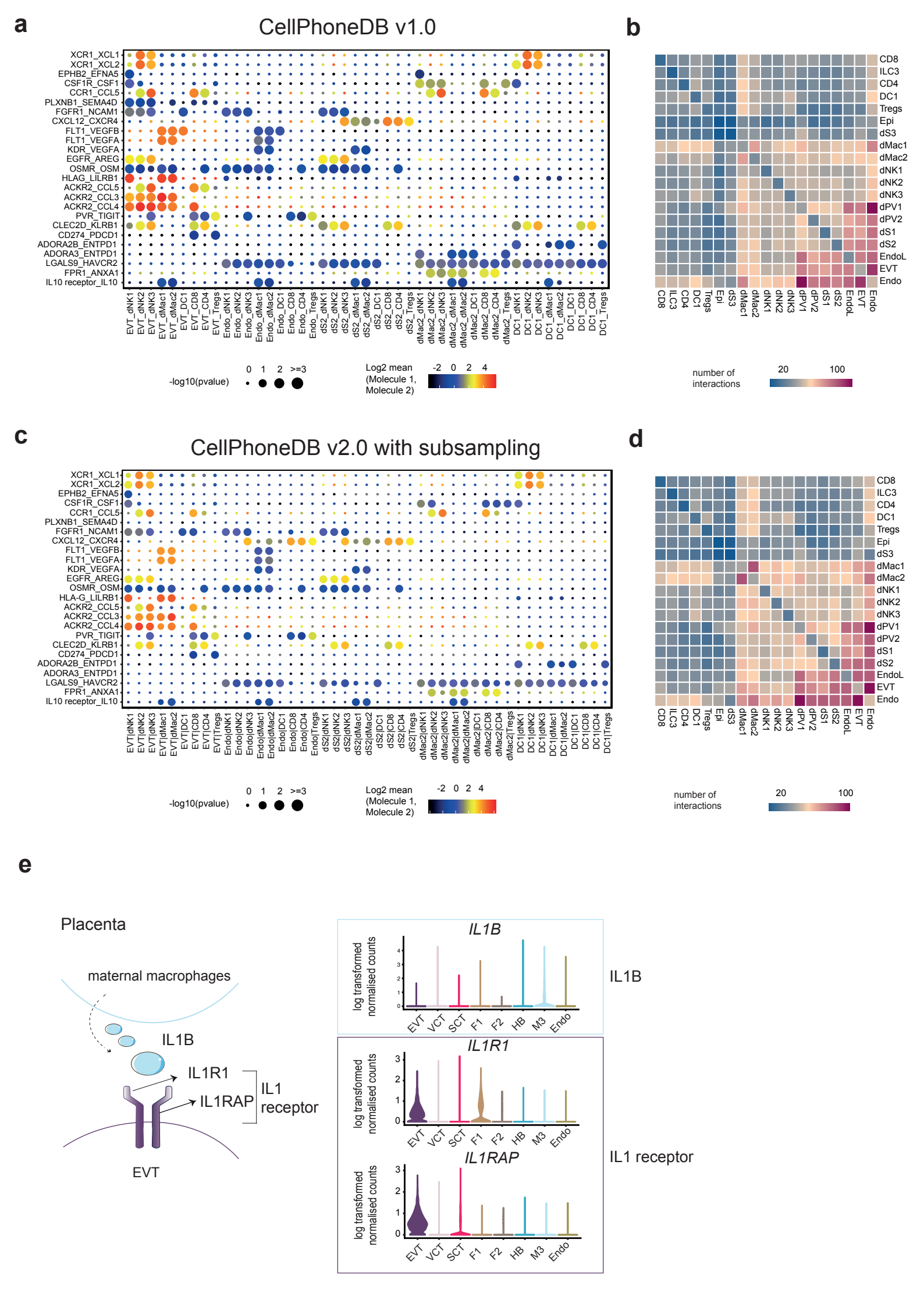

Input scRNAseq


Ligand
Sub.1
Sub.2
Receptor
Minimum
(Sub. 1, Sub. 2)

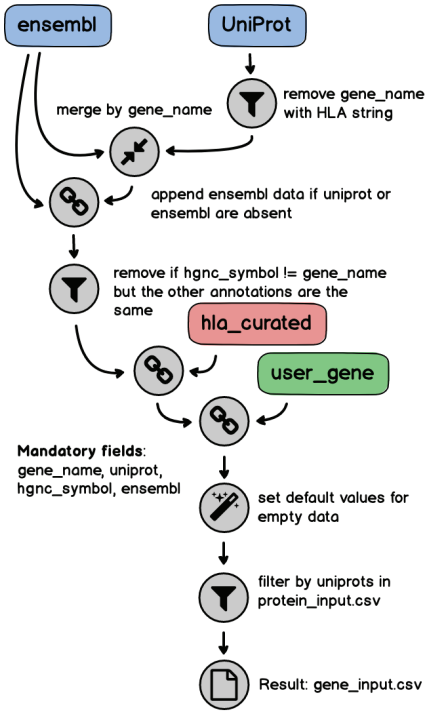

Random shuffle
Null distribution of the mean
(R1-L1) in cluster1-cluster2


Observed mean (R1-L1)
in cluster1-cluster2
p -value


R1 - L1
R2 - L2
R3 - L3
p - val
p - val
p - val
Ranking based
on specificity

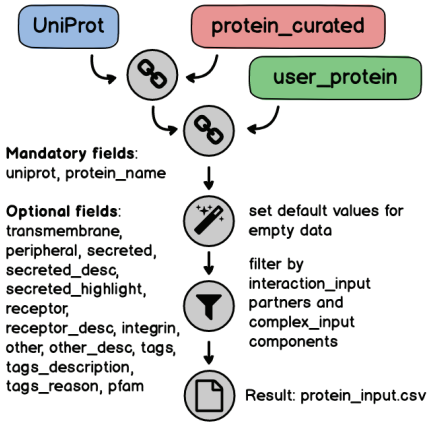

a
CellPhoneDB v1.0
XCR1_XCL1
XCR1_XCL2
EPHB2_EFNA5
CSF1R_CSF1
CCR1_CCL5
PLXNB1_SEMA4D
FGFR1_NCAM1
CXCL12_CXCR4
FLT1_VEGFB
FLT1_VEGFA
KDR_VEGFA
EGFR_AREG
OSMR_OSM
HLAG_LILRB1
ACKR2_CCL5
ACKR2_CCL3
ACKR2_CCL4
PVR_TIGIT
CLEC2D_KLRB1
CD274_PDCD1
ADORA2B_ENTPD1
ADORA3_ENTPD1
LGALS9_HAVCR2
FPR1_ANXA1
IL10 receptor_IL10
EVT_dNK1
EVT_dNK2
EVT_dNK3
EVT_dMac1
EVT_dMac2
EVT_DC1
EVT_CD8
EVT_CD4
EVT_Tregs
Endo_dNK1
Endo_dNK2
Endo_dNK3
Endo_dMac1
Endo_dMac2
Endo_DC1
Endo_CD8
Endo_CD4
Endo_Tregs
dS2_dNK1
dS2_dNK2
dS2_dNK3
dS2_dMac1
dS2_dMac2
dS2_DC1
dS2_CD8
dS2_CD4
dS2_Tregs
dMac2_dNK1
dMac2_dNK2
dMac2_dNK3
dMac2_dMac1
dMac2_dMac2
dMac2_DC1
dMac2_CD8
dMac2_CD4
dMac2_Tregs
DC1_dNK1
DC1_dNK2
DC1_dNK3
DC1_dMac1
DC1_dMac2
DC1_DC1
DC1_CD8
DC1_CD4
DC1_Tregs
-log10(pvalue)
0 1 2 >=3
Log2 mean (Molecule 1, Molecule 2)
-2 0 2 4
b
CD8
ILC3
CD4
DC1
Tregs
Epi
dS3
dMac1
dMac2
dNK1
dNK2
dNK3
dPV1
dPV2
dS1
dS2
EndoL
EVT
Endo
number of interactions
20 100
c
CellPhoneDB v2.0 with subsampling
XCR1_XCL1
XCR1_XCL2
EPHB2_EFNA5
CSF1R_CSF1
CCR1_CCL5
PLXNB1_SEMA4D
FGFR1_NCAM1
CXCL12_CXCR4
FLT1_VEGFB
FLT1_VEGFA
KDR_VEGFA
EGFR_AREG
OSMR_OSM
HLA-G_LILRB1
ACKR2_CCL5
ACKR2_CCL3
ACKR2_CCL4
PVR_TIGIT
CLEC2D_KLRB1
CD274_PDCD1
ADORA2B_ENTPD1
ADORA3_ENTPD1
LGALS9_HAVCR2
FPR1_ANXA1
IL10 receptor_IL10
EVT|dNK1
EVT|dNK2
EVT|dNK3
EVT|dMac1
EVT|dMac2
EVT|DC1
EVT|CD8
EVT|CD4
EVT|Tregs
Endo|dNK1
Endo|dNK2
Endo|dNK3
Endo|dMac1
Endo|dMac2
Endo|DC1
Endo|CD8
Endo|CD4
Endo|Tregs
dS2|dNK1
dS2|dNK2
dS2|dNK3
dS2|dMac1
dS2|dMac2
dS2|DC1
dS2|CD8
dS2|CD4
dS2|Tregs
dMac2|dNK1
dMac2|dNK2
dMac2|dNK3
dMac2|dMac1
dMac2|dMac2
dMac2|DC1
dMac2|CD8
dMac2|CD4
dMac2|Tregs
DC1|dNK1
DC1|dNK2
DC1|dNK3
DC1|dMac1
DC1|dMac2
DC1|DC1
DC1|CD8
DC1|CD4
DC1|Tregs
-log10(pvalue)
0 1 2 >=3
Log2 mean (Molecule 1, Molecule 2)
-2 0 2 4
d
CD8
ILC3
CD4
DC1
Tregs
Epi
dS3
dMac1
dMac2
dNK1
dNK2
dNK3
dPV1
dPV2
dS1
dS2
EndoL
EVT
Endo
number of interactions
20 100
e
Placenta
maternal macrophages
IL1B
IL1R1
IL1RAP
IL1 receptor
EVT
IL1B
log transformed normalised counts
EVT VCT SCT F1 F2 HB M3 Endo
IL1R1
log transformed normalised counts
EVT VCT SCT F1 F2 HB M3 Endo
IL1RAP
log transformed normalised counts
EVT VCT SCT F1 F2 HB M3 Endo
IL1B
IL1 receptor

gene_input



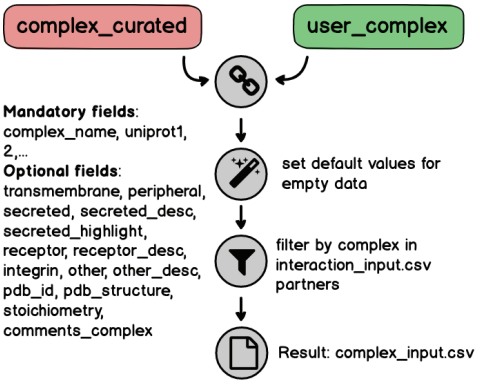
Duplicated genes priority:
ensembl < UniProt < hla_curated < user_gene

protein_input



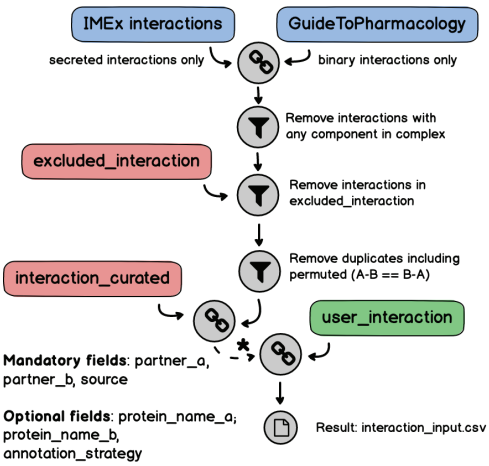
Duplicated proteins priority:
UniProt < protein_curated < user_protein

complex_input



Duplicated complex priority:
complex_curated < user_complex

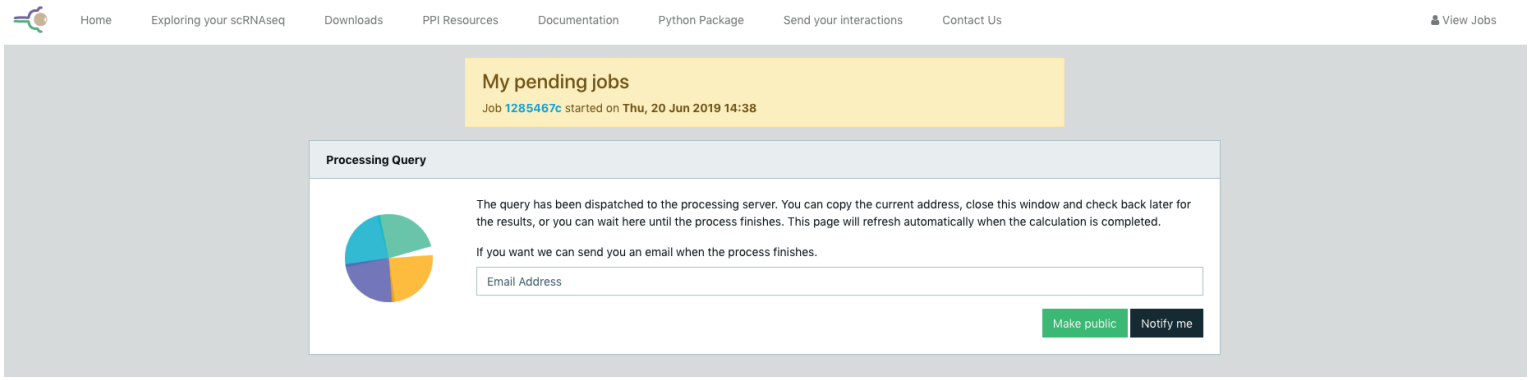
interaction_input



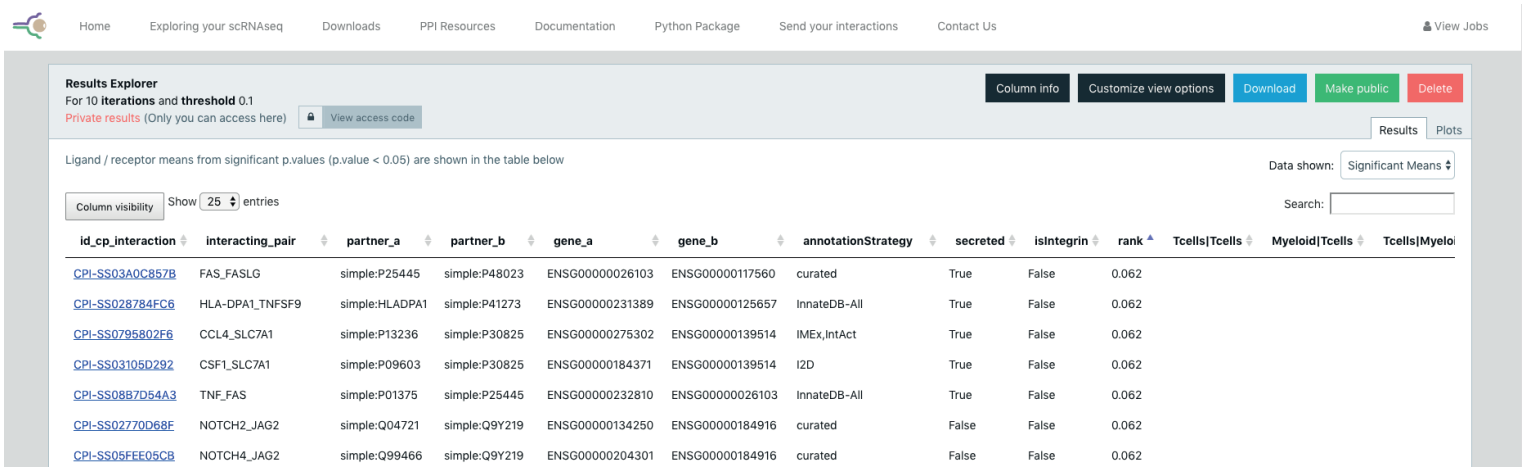
Duplicated interactions priority:
IMEx < GuideTorPharmacology < interaction_curated < user_interaction

* **if --user-interactions-only paramter**:
Only user_interaction is used to generate interaction_input

a

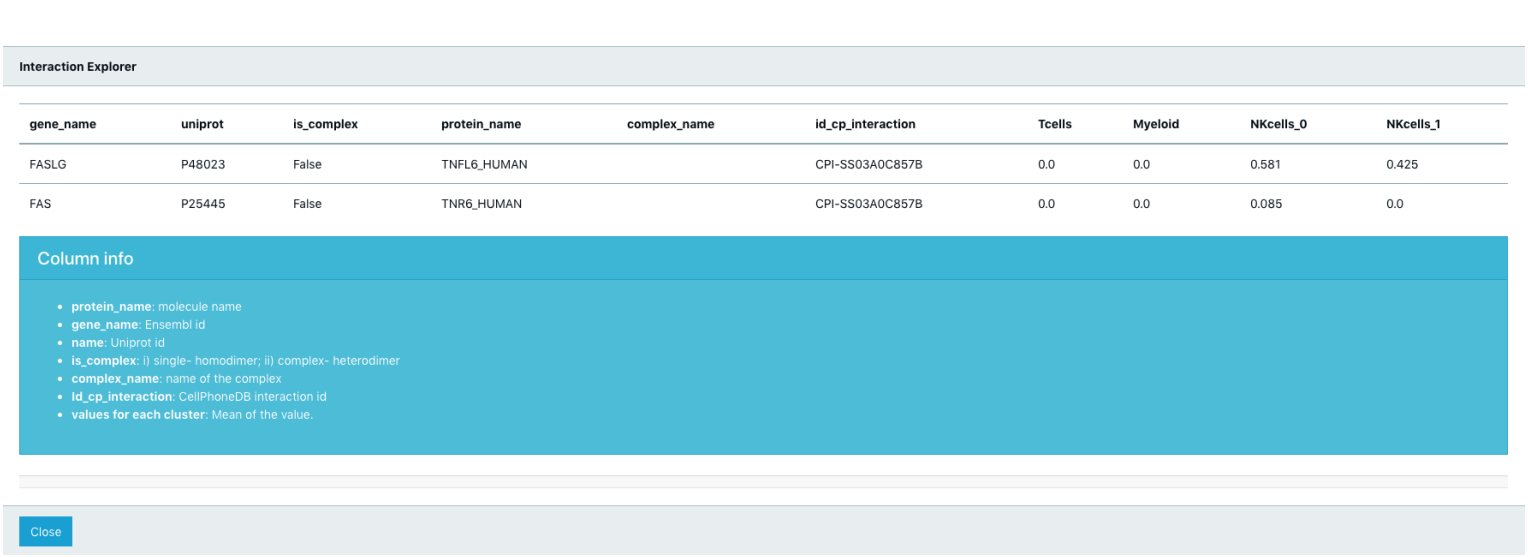


b



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			

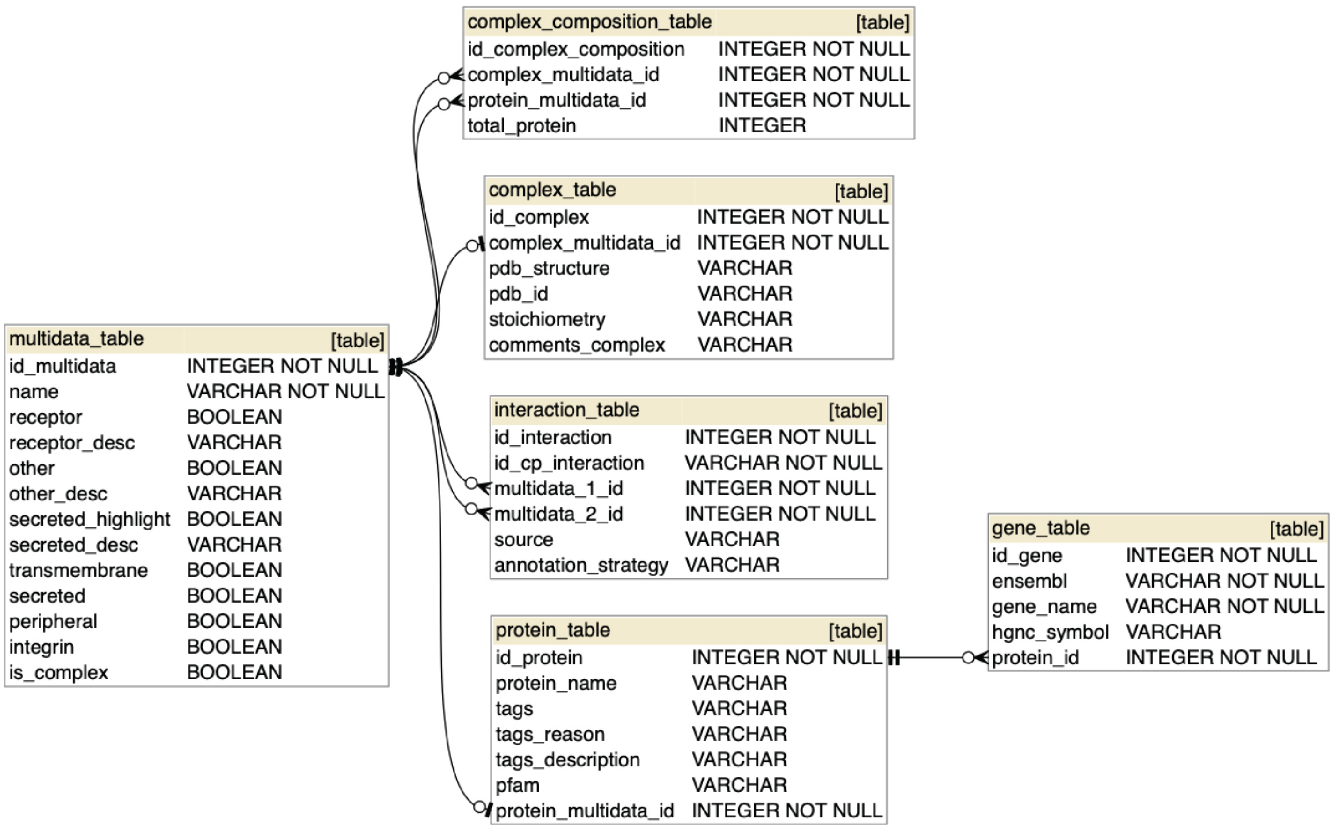
c



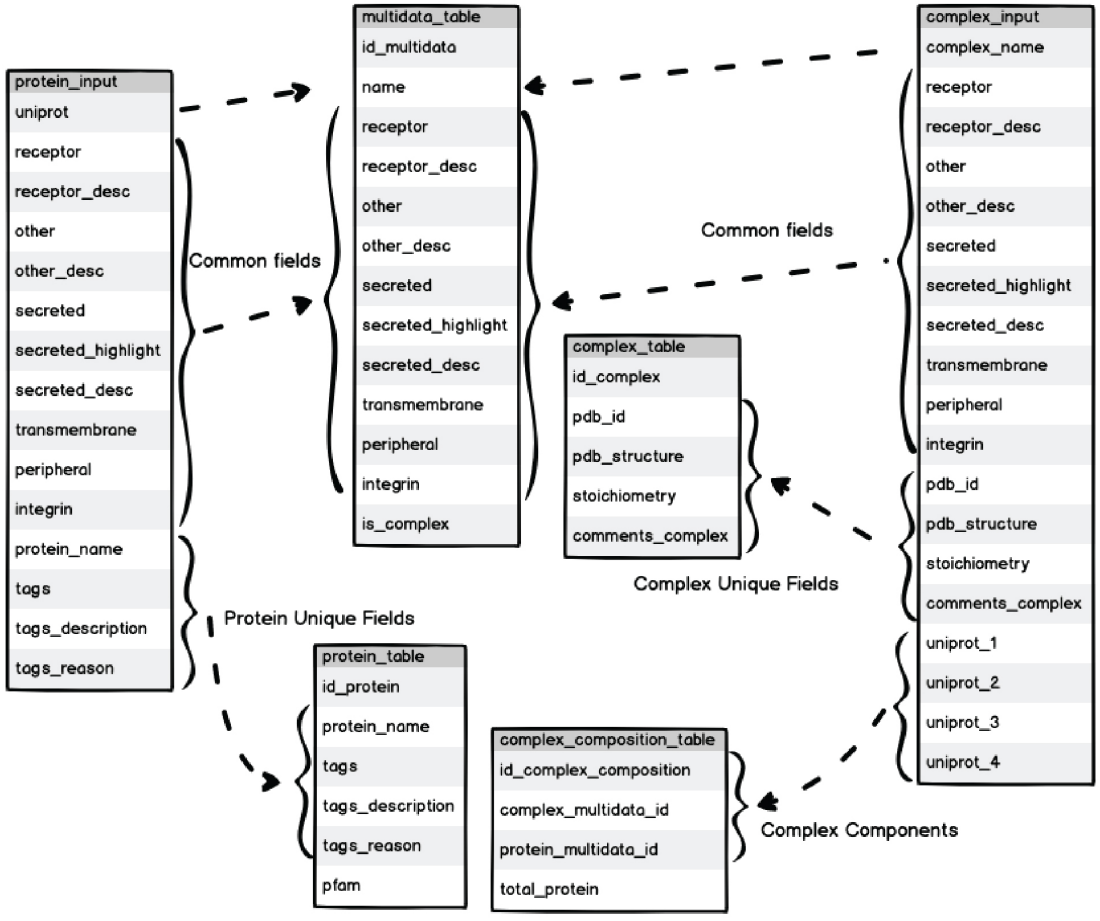
gene_name	uniprot	is_complex	protein_name	complex_name	id_cp_interaction	Tcells	Myeloid	NKcells_0	NKcells_1
FASLG	P48023	False	TNFL6_HUMAN		CPI-SS03A0C857B	0.0	0.0	0.581	0.425
FAS	P25445	False	TNR6_HUMAN		CPI-SS03A0C857B	0.0	0.0	0.085	0.0

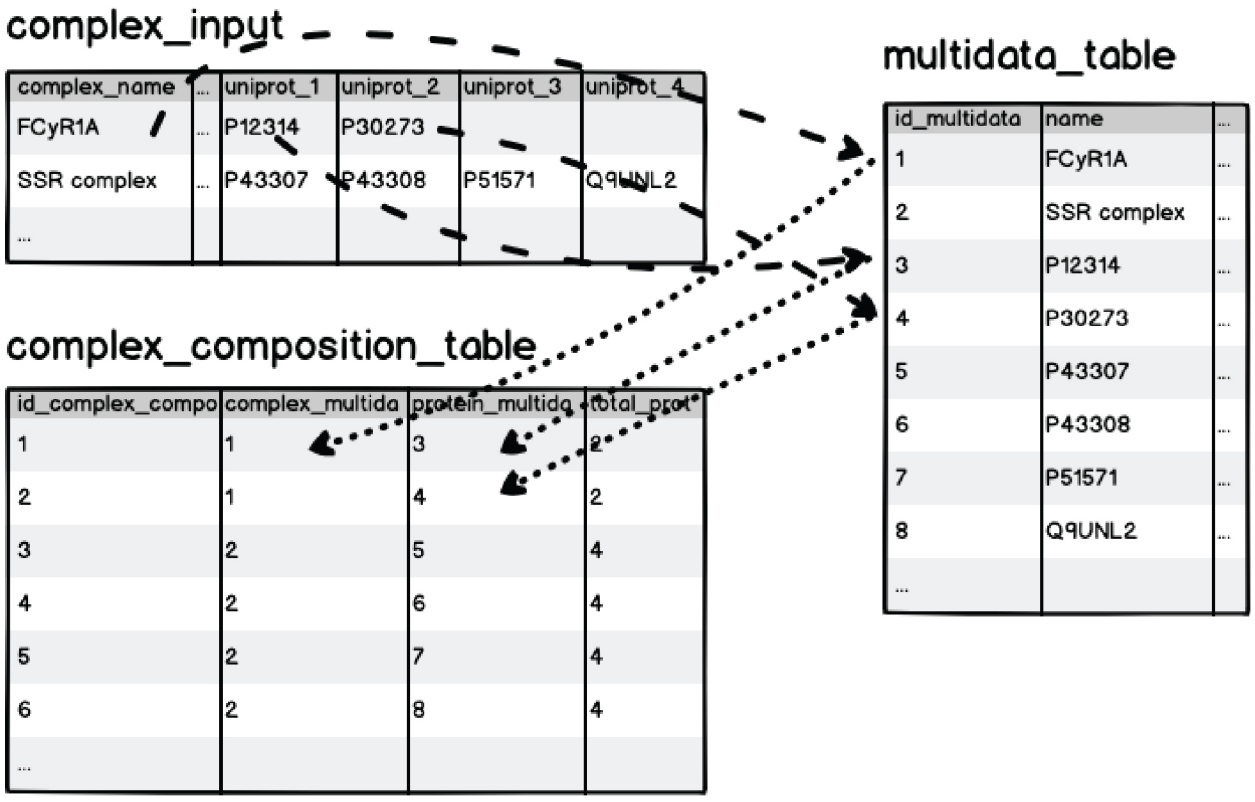
d

a


complex_composition_table [table]
id_complex_composition INTEGER NOT NULL
complex_multidata_id INTEGER NOT NULL
protein_multidata_id INTEGER NOT NULL
total_protein INTEGER
complex_table [table]
id_complex INTEGER NOT NULL
complex_multidata_id INTEGER NOT NULL
pdb_structure VARCHAR
pdb_id VARCHAR
stoichiometry VARCHAR
comments_complex VARCHAR
multidata_table [table]
id_multidata INTEGER NOT NULL
name VARCHAR NOT NULL
receptor BOOLEAN
receptor_desc VARCHAR
other BOOLEAN
other_desc VARCHAR
secreted_highlight BOOLEAN
secreted_desc VARCHAR
transmembrane BOOLEAN
secreted BOOLEAN
peripheral BOOLEAN
integrin BOOLEAN
is_complex BOOLEAN
interaction_table [table]
id_interaction INTEGER NOT NULL
id_cp_interaction VARCHAR NOT NULL
multidata_1_id INTEGER NOT NULL
multidata_2_id INTEGER NOT NULL
source VARCHAR
annotation_strategy VARCHAR
gene_table [table]
id_gene INTEGER NOT NULL
ensembl VARCHAR NOT NULL
gene_name VARCHAR NOT NULL
hgnc_symbol VARCHAR
protein_id INTEGER NOT NULL
protein_table [table]
id_protein INTEGER NOT NULL
protein_name VARCHAR
tags VARCHAR
tags_reason VARCHAR
tags_description VARCHAR
pfam VARCHAR
protein_multidata_id INTEGER NOT NULL

b


protein_input
uniprot
receptor
receptor_desc
other
other_desc
secreted
secreted_highlight
secreted_desc
transmembrane
peripheral
integrin
protein_name
tags
tags_description
tags_reason
Common fields
multidata_table
id_multidata
name
receptor
receptor_desc
other
other_desc
secreted
secreted_highlight
secreted_desc
transmembrane
peripheral
integrin
is_complex
Common fields
complex_input
complex_name
receptor
receptor_desc
other
other_desc
secreted
secreted_highlight
secreted_desc
transmembrane
peripheral
integrin
pdb_id
pdb_structure
stoichiometry
comments_complex
uniprot_1
uniprot_2
uniprot_3
uniprot_4
complex_table
id_complex
pdb_id
pdb_structure
stoichiometry
comments_complex
Complex Unique Fields
Protein Unique Fields
protein_table
id_protein
protein_name
tags
tags_description
tags_reason
pfam
complex_composition_table
id_complex_composition
complex_multidata_id
protein_multidata_id
total_protein
Complex Components


complex_input
complex_name
...
uniprot_1
uniprot_2
uniprot_3
uniprot_4
FCyR1A
...
P12314
P30273
SSR complex
...
P43307
P43308
P51571
Q9UNL2
...
multidata_table
id_multidata
name
...
1
FCyR1A
...
2
SSR complex
...
3
P12314
...
4
P30273
...
5
P43307
...
6
P43308
...
7
P51571
...
8
Q9UNL2
...
...
complex_composition_table
id_complex_compo
complex_multida
protein_multida
total_prot
1
1
3
2
2
1
4
2
3
2
5
4
4
2
6
4
5
2
7
4
6
2
8
4
...