# EXPERIMENTAL PART

## Objectives

In this study, we analyze and compare four computational methods designed to detect cell-cell interactions from single-cell and spatial transcriptomics data: CellChatv2, SpaTalk, NATMI and CellPhoneDB. The objective is to assess their performance and suitability for CCI analysis across different datasets and provide insights into their practical and biological outputs.

The comparison is based on two main aspects: technical performance and biological relevance. Technical features include practical considerations, such as installation and setup requirements, computational efficiency–runtime and memory usage. Biological relevance is evaluated through metrics such as the number of detected inferred interactions, ligands and receptors. Moreover, the intersection of predicted interactions across methods applied to the same datasets is also examined.

These methods are applied to four distinct datasets with the aim of generalizing and highlighting the respective strengths and limitations of each method in different experimental contexts. The goal of this comparison is to provide practical recommendations for researchers selecting CCC inference tools, as well as to contribute to a broader understanding of how methodological choices influence downstream biological interpretation.

## Datasets

Four spatial transcriptomics datasets derived from the mouse brain were used for the analysis. Each dataset was preprocessed and annotated to ensure consistency in downstream comparisons.

This dataset was obtained from a MERFISH experiment on mouse brain tissue. A specific brain section (*C57BL6J-2.030*) was selected using the Seurat function subset(x = seu, subset = brain\_section\_label == "C57BL6J-2.030"). The data included pre-integrated layers from CZ CellXGene and the brain atlas. Gene annotation was enriched using SingleR, and in Python, Ensembl IDs were converted to gene symbols. Two additional layers, based on CellXGene and the brain atlas, were also integrated.

The Xenium dataset was downloaded from the official Xenium website. The data required cell type annotation, which was performed using SingleR in R. The input for the analysis was an .rds file, and the final annotated experiment was saved as adata\_xenium\_final\_annotated.h5ad for downstream analysis in Python.

The Visium HD dataset, processed in Seurat (R), was also sourced as an .rds file. SingleR was used to perform cell type annotation. Due to its high resolution and large size, this dataset posed particular computational challenges. The final annotated experiment was exported as adata\_visiumhd\_annotated.h5ad for analysis in Python.

The CosMx dataset represents a single tissue section from a larger CosMx experiment, processed in Seurat. A specific slide (*slide\_ID\_numeric == 2*) was selected using subset(x = cosmx, subset = slide\_ID\_numeric == 2). The dataset was downloaded from the CosMx website and annotated with SingleR. The processed and annotated data were saved as cosmx\_final\_annotated.h5ad for Python-based analysis.

## Methods

We decided on a threshold of 16 hours per launch.

*CellChat*

The CellChat analysis was performed in R (v4.2.2) using RStudio. Required packages included Seurat, devtools, and CellChat, along with Bioconductor dependencies (BiocManager, BiocGenerics, BiocNeighbors, and Biobase). Installation was completed iteratively to resolve dependencies.

For each dataset, a Seurat object was loaded (RDS file), extracting assay data, cell identities and spatial coordinates. Next, two spatial parameters were defined: the *conversion.factor* is set to 1 ensuring that the spatial data is on the correct scale for analysis. The parameter *spot.size* defining the size of the spots in the spatial transcriptomics data is set to 10. Furthermore, we create a metadata dataframe with labels, group information, sample names, data type, coordinates and spatial factors.

The CellChat workflow consists of creating a CellChat object using the prepared data and metadata. Afterwards, subsetting the database to include only relevant ligand-receptor interactions. Finally, a function identifies over-expressed genes and interactions within dataset. The *computeCommunProb* function is used to compute the communication probability, which is a crucial step in identifying significant cell-cell interactions. Key parameters are summarized in **Tab 3.1**. The only parameter that was varying in different dataset analysis is the *scale.distance* which scales the spatial distances to ensure they are in a suitable range for analysis. Lastly, the interaction data is extracted and filtered to include only cell-cell contact interactions. This data is then saved to an output *csv* file for further analysis.

**Table 3.1**: Key parameters for CellChat analysis in *computeCommunProb* function.

|  |  |  |
| --- | --- | --- |
| Parameter | Value | Purpose |
| method | truncatedMean | Reduces the impact of outliers when computing communication probabilities. |
| trim | 0.1 | Indicates the proportion of extreme values to be trimmed from each end of the distribution before computing the mean. |
| interaction.range | 250 µm | Maximum distance within which interactions are considered. |
| contact.range | 10 µm | Maximum distance for contact-dependent interactions ensuring that only close-range interactions are considered. |

This run\_cellchat function was applied to four spatial transcriptomics datasets, including VisiumHD, Xenium, CosMx and Merfish.

**Table 3.2**: Dataset-specific settings for CellChat analysis,

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset | Assay Name | scale.distance | Notes |
| VisiumHD | Spatial.008um | 5.4 | Intersection of cell types and spatial coordinates. |
| Xenium | Xenium | 5.4 |  |
| CosMx | RNA | 40 | Higher scale.distance needed due to small spatial distances. |
| MERFISH | RNA | 5.4 | Subsetted to specific brain sections. |

As seen in **Tab 3.2**, the VisiumHD dataset featured a relatively high spatial resolution (8 µm layer thickness), but the spots covered multiple cells rather than true single-cell resolution. This mismatch necessitated extra preprocessing steps, specifically intersecting cell and spot names to ensure alignment between the Seurat object and spatial metadata.

On the other hand, CosMx data required specific adjustments. Initial attempts using a *scale.distance* of 5.4 µm did not yield meaningful interactions in CellChat. This likely stemmed from CosMx’s ultra-high resolution and tightly packed spatial coordinates, which led us to increase the scale.distance to 40 µm.

Both the Xenium and MERFISH datasets allowed straightforward extraction of cell type and positional data, without major preprocessing hurdles.

*SpaTalk*

Further, we used the SpaTalk R package to infer CCC interactions from the same four spatial transcriptomics datasets. The workflow began with the loading of Seurat objects, from which we extracted key metadata, including x-y spatial coordinates and cell type annotations necessary for spatially resolved CCC analysis.

Next, expression matrices were prepared from the assay data of each dataset to serve as input for SpaTalk. The *createSpaTalk* function was used to build SpaTalk objects, with parameters specifying the species ("mouse") and providing cell type information.

The core analytical step was performed using the *find\_lr\_path* function, which identifies ligand–receptor pairs and their associated signaling pathways. This function requires a predefined list of ligand-receptor pairs and pathway definitions, which were supplied accordingly. Once potential interactions were identified, we applied the *dec\_cci\_all* function, which decodes and infers global patterns of CCC across the tissue based on the detected ligand–receptor interactions.

Finally, the results were prepared for downstream analysis and visualization by saving the output in RDS containing all relevant interaction data and facilitated easy integration into subsequent comparative and visualization workflows.

*NATMI*

The NATMI method is implemented using the liana Python package to infer CCC interactions across the individual datasets. For each dataset, gene expression data and associated metadata is loaded from an .h5ad file as an AnnData object. Depending on the characteristics of the datasets, the raw expression data were normalized and log-transformed. Normalization scales total counts to a fixed target and Log Transformation applies log1p to the normalized data. The raw data is copied to a layer called *raw* and transformed data is stored in a new layer *lognorm*.

The NATMI method requires a resource of ligand-receptor pairs, which is selected from the *liana.resource* as mouseconsensus. A filtering step ensures that only ligand-receptor pairs where both genes are present in the dataset are retained. NATMI is then run on the preprocessed data with specific settings such as grouping column, gene layer, expression proportion and resource. The results of the NATMI analysis are saved as CSV files.

*CellPhoneDB*

Next, we applied the CellPhoneDB framework using the Python package. The analytical workflow consisted of the following main steps. Firstly, raw expression matrices were loaded as AnnData objects using *scanpy*. The datasets were normalized to a fixed total count and log-transformed to prepare the expression values for downstream analysis. Since CellPhoneDB operates on human ligand–receptor pairs, we used the *mousipy* library to map mouse gene symbols to their human orthologs. Genes without clear orthologs were excluded to ensure compatibility with the CellPhoneDB database. Cell-level metadata (cell type annotations and IDs) are extracted into a meta file (meta.txt) and exported the expression matrix in counts format (HDF5), both required inputs for CellPhoneDB. The CellPhoneDB statisticalanalysis is run with a result of identification of statistically significant ligand-receptor interactions across cell-type pairs, based on random permutation testing. The results including interaction matrices, mean expression values, and p-values—were saved to dataset-specific output directories for further interpretation.

*Misty* didn’t work.

*NicheNet* was not compatible with our research. The output of the NicheNet analysis is a table that doesn’t show the interaction’s source and target cluster. We could have chosen specific clusters and run the analysis uniformly, but we wouldn’t have been able to compare this method with the rest.

# RESULTS AND DISCUSSION

Todo:

* Comparing interactions ligand-receptor / celltype-celltype – in SpaTalk there are same sender and receiver celltype but with different ligand receptors

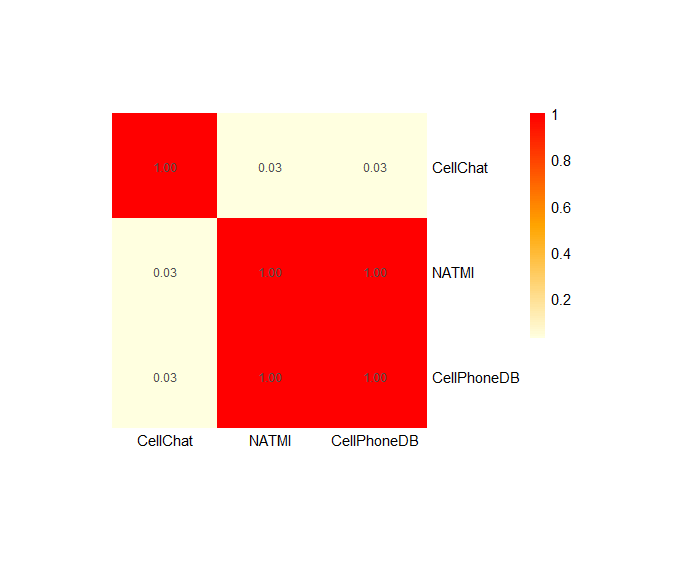
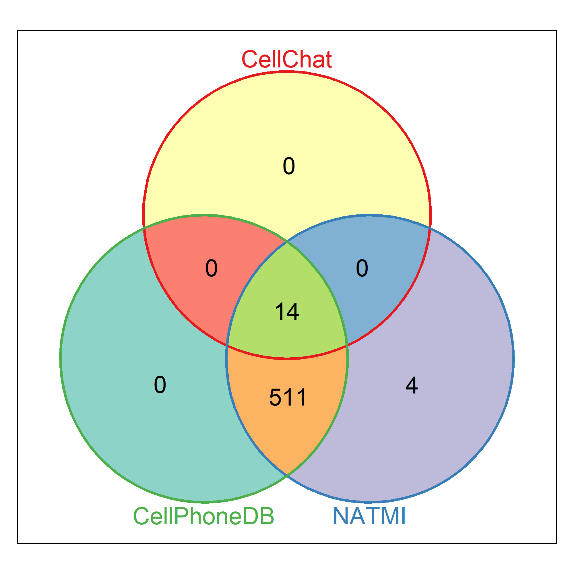
## Dataset comparison

*VisiumHD*

Technical:

* CellChat and CellPhoneDB took a lot longer than any other method. NATMI was very quick.
* SpaTalk allocated 55.9GiB and CellChat 3.5GiB, the other methods didn’t have a warning about memory usage.

Biological:

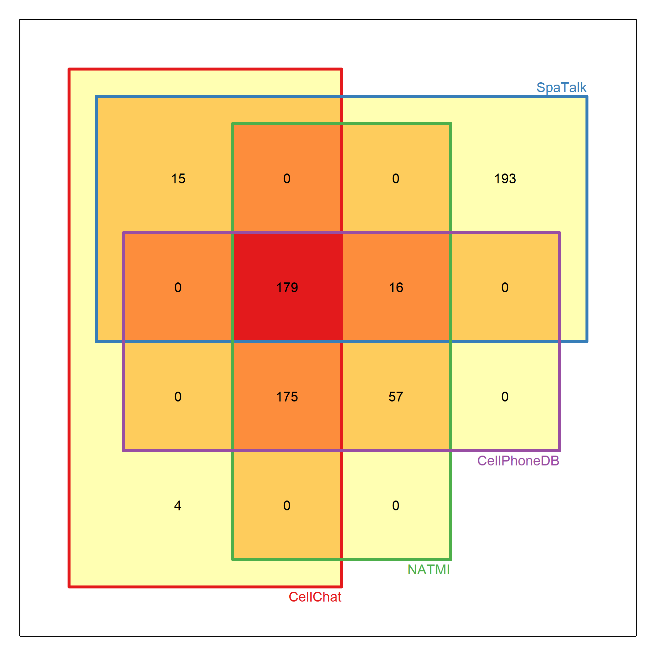


*Xenium*

Technical:

* Quite quick compared to Visium, CellChat still so long. NATMI and CellPhoneDB are comparable. No problems with memory usage.

Biological:

A screen shot of a cell phone chart

AI-generated content may be incorrect.

* SpaTalk 15212 (more LR pairs for one sender-receiver pair)
* NATMI 4024
* CPDB 913
* CellChat 9164

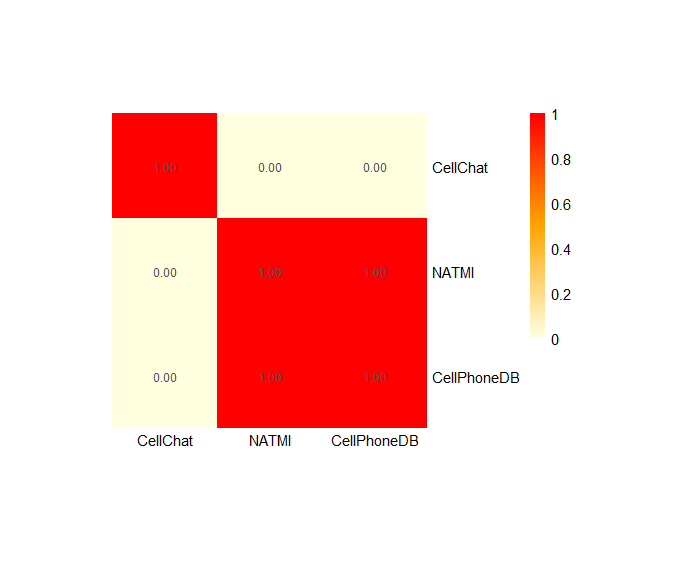
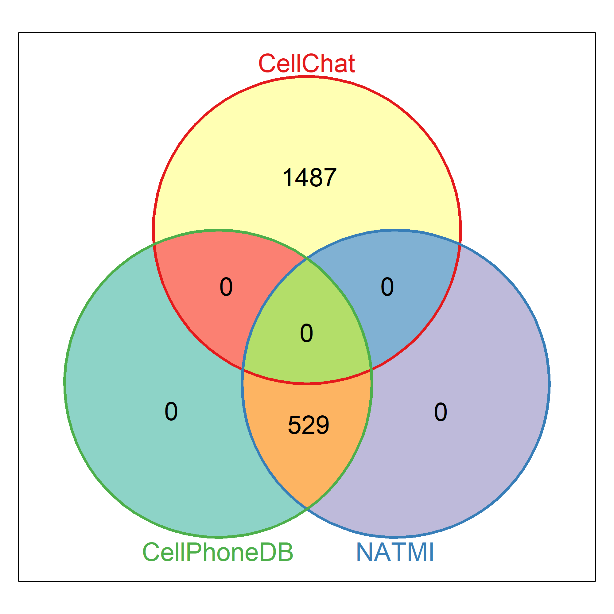
Vyznamnost 5 nejvyznamejsich s korelaci – se jmeny diagram,

*CosMx*

Technical:

* SpaTalk took a very long time. NATMI again in the range of seconds and CellPhoneDB now in the range of minutes.

Biological:



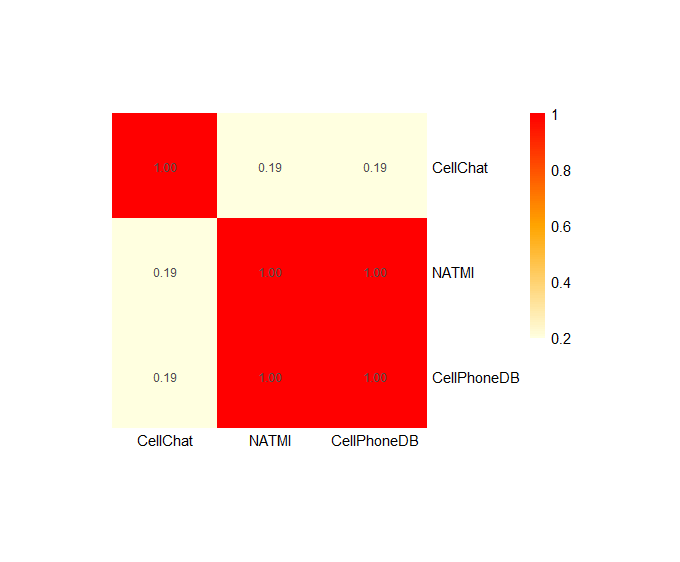
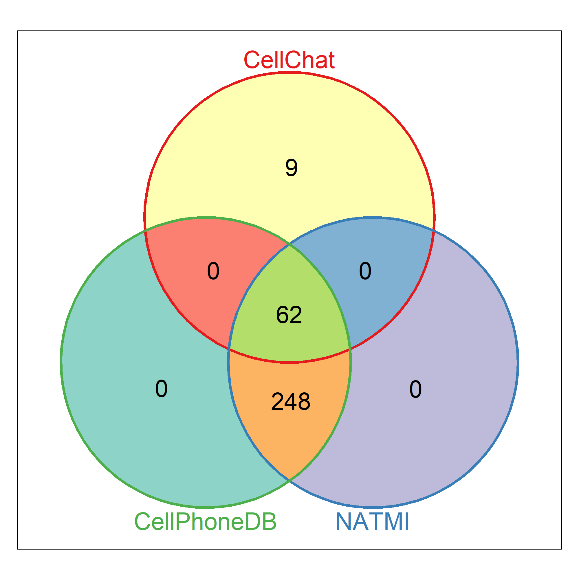
* CellChat 7830
* NATMI 101047
* CPDB 146

*MERFISH*

Technical:

* Every method took kind of the same time. In the range of minutes.

Biological:



* CellChat 90
* SpaTalk didn’t find anything
* NATMI 3273
* CPDB 99

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| VisiumHD | CellChat | SpaTalk | NATMI | CellPhoneDB |
| Speed | 1h 36m |  | 22s |  |
| Memory | 3,5 GiB | Allocating 55.9 GiB |  |  |
| Database | CellChatDB.mouse |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Xenium | CellChat | SpaTalk | NATMI | CellPhoneDB |
| Speed | 15m |  | 11s | 21s |
| Memory |  |  |  |  |
| Database | CellChatDB.mouse |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CosMx | CellChat | SpaTalk | NATMI | CellPhoneDB |
| Speed |  | 2h 28m | 7s | 3m 51s |
| Memory |  |  |  |  |
| Database | CellChatDB.mouse |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Merfish | CellChat | SpaTalk | NATMI | CellPhoneDB |
| Speed | 3m |  | 3s | 3m 17s |
| Memory |  |  |  |  |
| Database | CellChatDB.mouse |  |  |  |

# CONCLUSION

Shrnuje podstatné výsledky, podává přehled o celé práci, zdůrazňuje, co nového práce přináší, formuluje doporučení vyplývající ze závěrečné práce. **Nesmí kopírovat souhrn**, nemůže to být pouhé zopakování dosažených výsledků.