Spatial Transcriptomics of the Preoptic Region in Mouse Hypothalamus: Methodological Approach

Ibrahim Fangary

August 31st, 2024

Introduction

- Using spatial transcriptomics data from Moffitt et al. (2018).
- The data was generated using multiplexed error-robust fluorescence in situ hybridization (MERFISH).
- Investigate spatial gene expression in the hypothalamic preoptic region of mice.
- Analyze excitatory, inhibitory, and hybrid neural subpopulations.

Methods

- Using scanpy package in python for preprocessing, clustering, and visualizing single-cell gene expression data.

- Squidpy to spatial analysis of the dataset including neighborhood enrichment analysis and visualization of spatial data

- The analysis consists of 4 parts
  - Data preparation
  - Preprocessing and clustering
  - Cell type identification
  - Spatial distribution of neural subpopulation

Data Preparation

```
# extracting gene count matrix which begins from column 8
gene_counts_df = data.iloc[:,8:]
# extrating meta data which ends at column 8
cell_meta_data_df = data.iloc[:,:8]
# creating anndata object
adata = sc.AnnData(X=gene_counts_df.values,
                   obs=cell_meta_data_df, var=gene_metadata_df)
```

- utilizing jupyter notebook with Python kernel.
- Using pandas package to handle and prepare the data.
- Building Anndata dataframe for further analysis with scanpy.

Data Preparation

- Subsetting the required Bregma section in animal 1.
- Removing ‘Ambiguous’ cell class as identified in Moffitt et al. (2018).
- Removing empty cells and unexpressed genes.
- Final dataset: 28,317 cells across five Bregma sections (-0.04 mm, -0.09 mm, -0.14 mm, -0.19 mm, and -0.24 mm) and 155 genes.

preprocessing and clustering

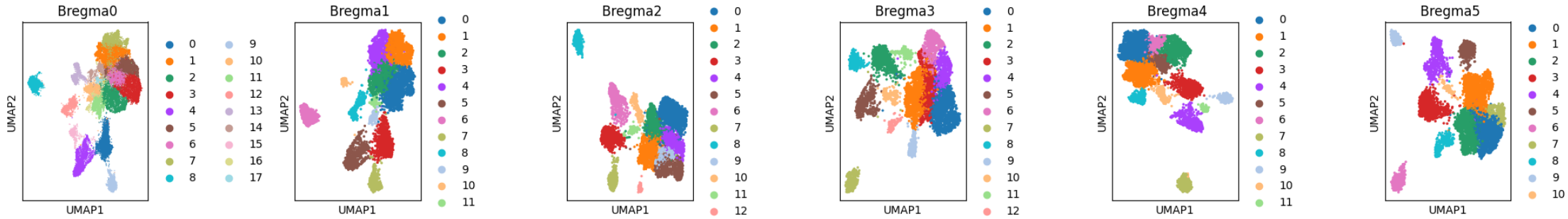
```python
def BasicScanpyPreprocessing(adata, resolution=None, n_comps=50):
    sc.pp.normalize_total(adata, inplace=True)
    # Normalize counts per cell
    sc.pp.log1p(adata) # Logarithmize the data matrix
#     sc.pp.scale(adata) # we dont need to sscale the data
    sc.pp.pca(adata, n_comps=n_comps)
    sc.pp.neighbors(adata)
    sc.tl.umap(adata)
    sc.tl.leiden(adata, key_added="Leiden",
                 resolution=resolution)
    sc.pp.calculate_qc_metrics(adata, percent_top=None,
                               log1p=False, inplace=True)
    return adata
```

- Building a Simple Preprocessing and Clustering Function
- The data was normalized to total count and log transformed
- Calculating PCA, neighbors and Leiden for dimensionally reduction and clustering
- Calculate UMAP to project the high-dimensional data

Results

```
sc.pl.umap(adata, color=["Leiden"], wspace=0.4)
```

- Ploting umap of combined tissue once and each tissue separately



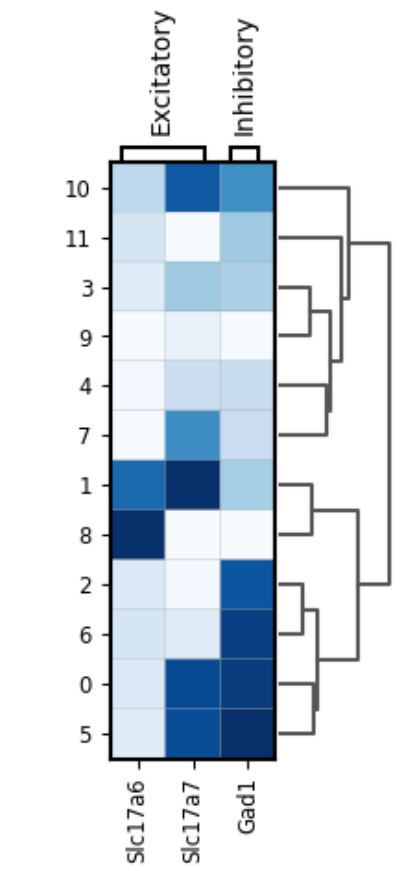
- Visualization of all clusters identified in each approach (PP and clustering on combined tissue and each tissue separately) to see if there any cluster specific for certain tissue.

Cell types identification

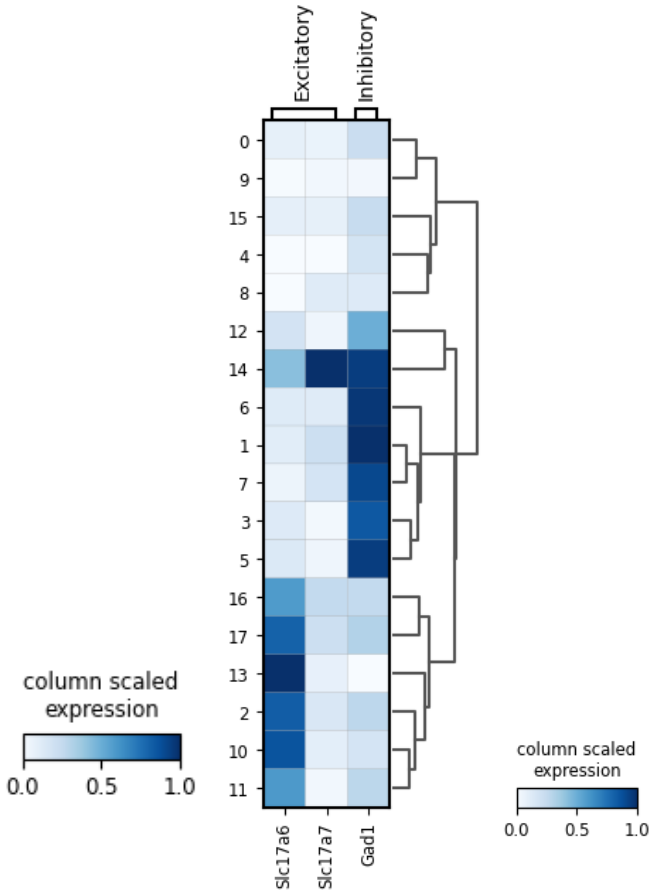
- Differential gene expression analysis across clusters identified marker genes for each clusters
- Those marker genes was used to assign each cluster cell types
- database PanglaoDB was used to identify cell types based on highly Differential expression genes.

Cell types identification

- Ploting heatmap with dendogram to identify excitatory, inhibitory and hybrid clusters based on expression of Slc17a6, Slc17a7 and Gad1
- After assigning cell types the Oligodendrocytes (OD) were not identified in tissue sections -0.19mm



tissue sections -0.19mm

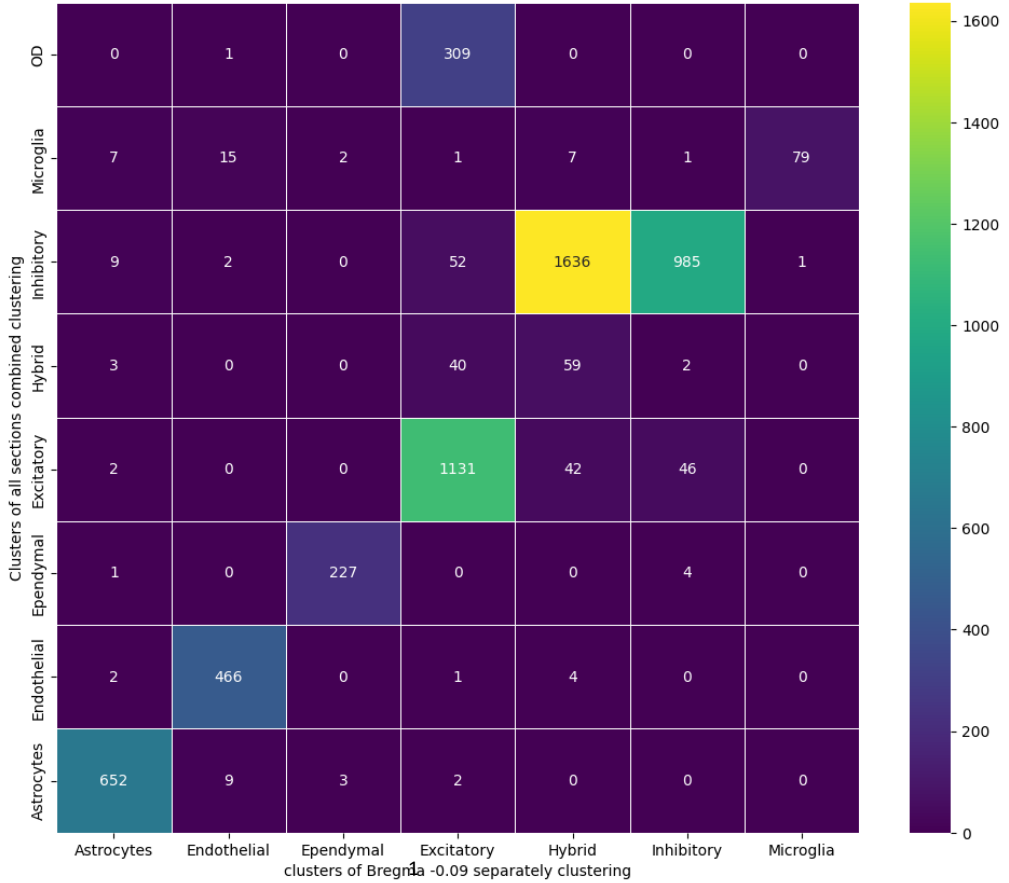


Combined tissue sections

Results

- Utilizing seaborn package to plot heatmap
- similar trends in cell type assignment. Except Oligodendrocytes (*OD*) that were not identified in separate tissue
- clusters were similar in Astrocyte, Endothelial, Ependymal, Inhibitory and Microglia
- large portion of cells was identified as inhibitory in the case of combined tissue sections but identified as hybrid in the case of separate tissue.
- The clusters of case of combined tissue sections stated cells heterogeneity more effectively and explored more cell types than separate tissue

```
seaborn.heatmap(sorted_table, annot=True,
fmt='d', cmap='viridis', linewidths=0.5)
```



Spatial distribution of neural subpopulation

- Subsetting the inhibitory, excitatory and hybrid clusters and re-cluster them
- Re-assigning the new identified clusters based on expression of Slc17a6, Slc17a7 and Gad1
- Using the Squidpy package in python to Perform neighborhood enrichment analysis
- Using squidpy to compare the spatial distribution of these genes with the newly identified clusters

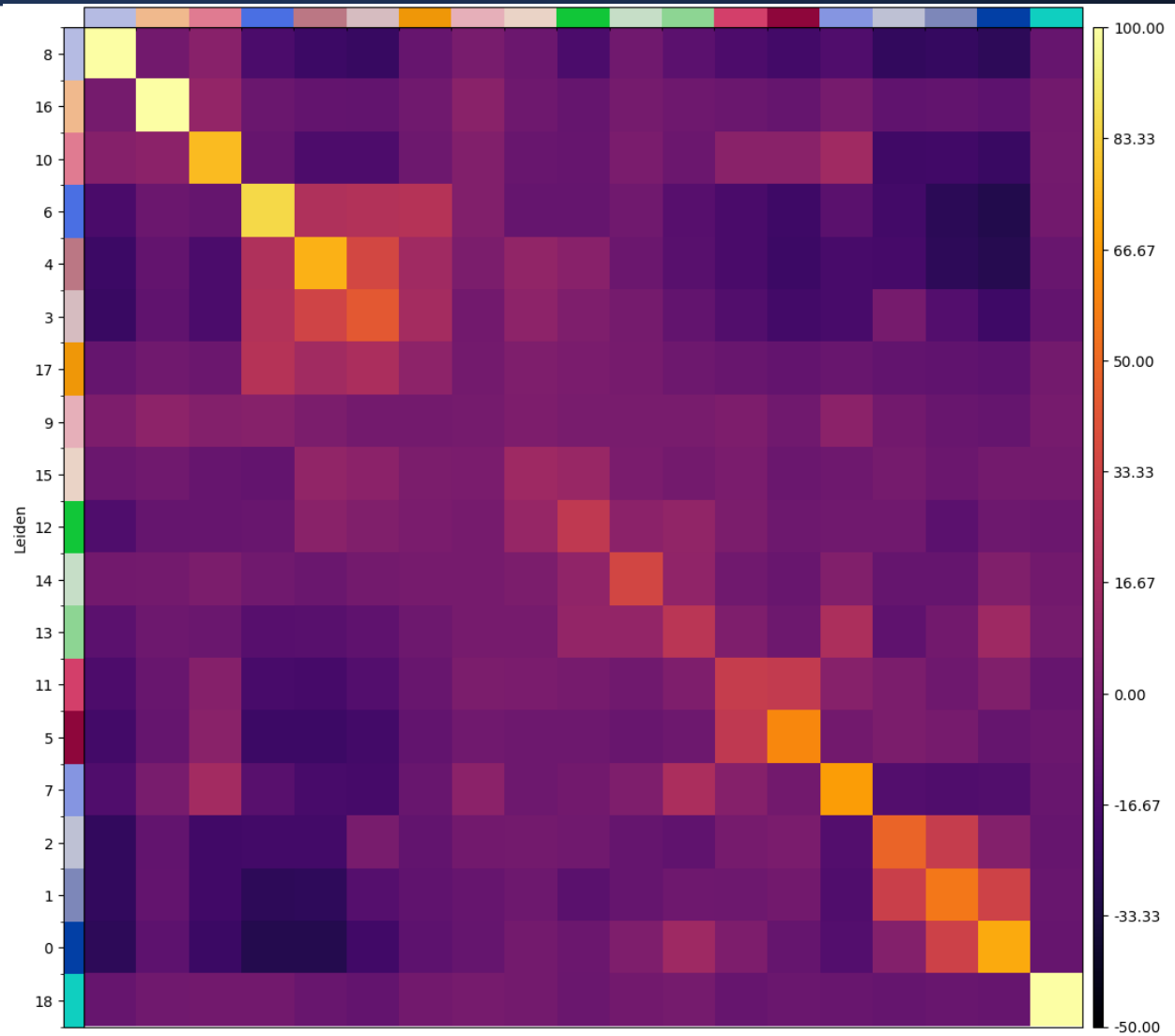
Spatial distribution of neural subpopulation

```
# Neighborhood Enrichment analysis
sq.gr.spatial_neighbors(IE_subset, coord_type="generic", spatial_key="spatial")
sq.gr.nhood_enrichment(IE_subset, cluster_key="Leiden")
#  hierarchically clustered heatmap which shows clusters of enriched neighborhoods
in our tissue
sq.pl.nhood_enrichment(
    IE_subset, cluster_key="Leiden", method="single", cmap="inferno", vmin=-50,
vmax=100
)
```

- Using squidpy for
  - Calculating neighborhood graph
  - Neighborhood Enrichment Calculation
  - Plotting the Neighborhood Enrichment Heatmap

Results

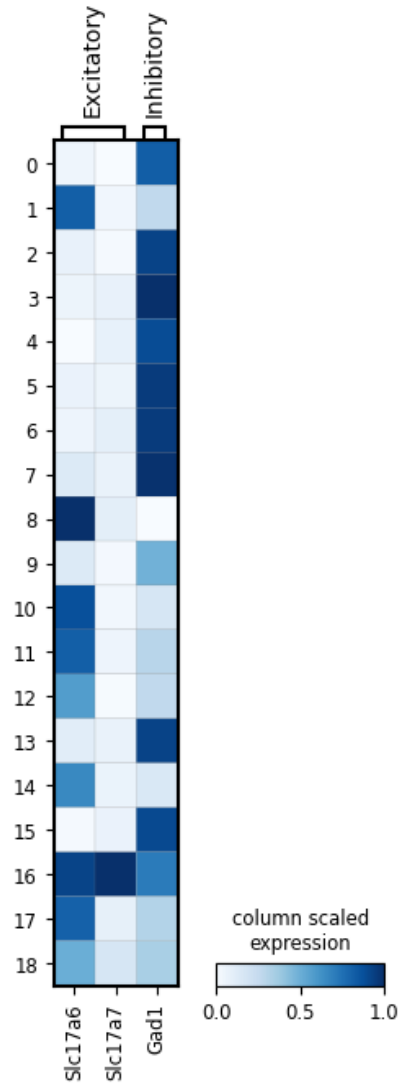
- most of clusters exhibited strong spatial neighborhood relations, except for clusters 9 and 17 showeing the lowest spatial scores



re-assigning the I, E and H clusters

- Each cluster was assigned to a cell type based on the expression of markers such as *Slc17a6*, *Slc17a7*, and *Gad1*

- identifying 10 inhibitory subpopulations, 8 excitatory subpopulations, and 1 hybrid subpopulation

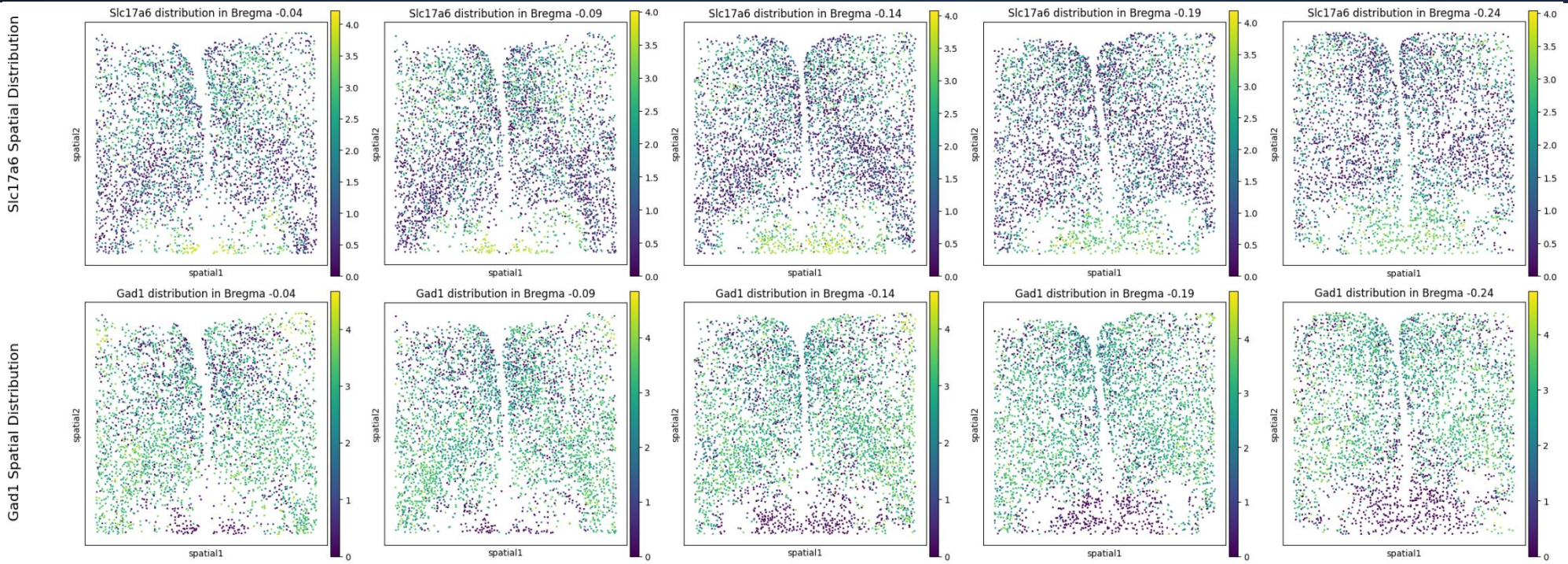


Spatial distribution of neural subpopulation

```
# Subset the AnnData object for each Bregma section
sq.gr.spatial_neighbors(adata_slice, coord_type="generic")
sq.gr.spatial_autocorr(adata_slice, mode="moran")
sq.pl.spatial_scatter(adata_slice, shape=None, color="Slc17a6")
sq.pl.spatial_scatter(adata_slice, shape=None, color="Gad1")
```

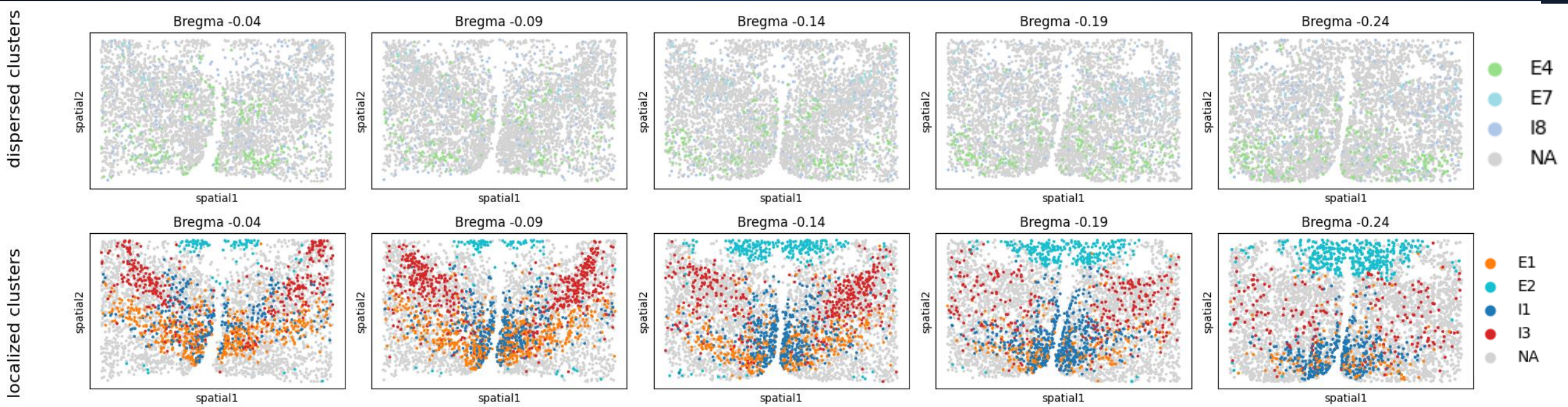
- Using squidpy for
  - Calculating Spatial Neighbors
  - Computing Spatial Autocorrelation
  - Plotting scatter plot of distribution of each gene
- Using matplotlib package in python for handling and visualization of plots

Results



- The plot is inverted
- The distribution of *Slc17a6* was predominantly localized in the PVT and FX regions
- *Gad1* expression was mainly concentrated in the BST region

Spatial distribution of neural subpopulation



- clusters E4, E7, and I8 were widely dispersed consistent with their lower spatial neighboring scores
- clusters E1, E2, I1, and I3 exhibited more localized patterns

Conclusion

- Combined tissue section clustering reveals greater heterogeneity.
- Identified additional neural subtypes not detected in individual sections
- Spatial context is crucial for understanding the organization of neural populations.

Thank you