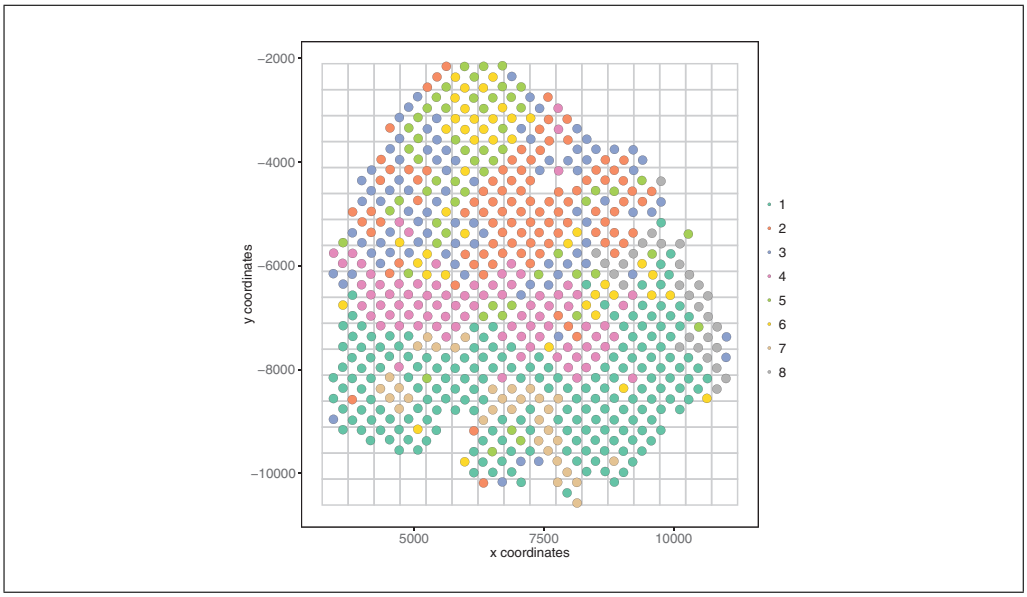

Figure 16 A spatial grid representing the average gene expression per box.

1. Spatial grid.

   To view a rough representation of the data, we will create and plot a spatial grid (Fig. 16).

```
my_giotto_object <- createSpatialGrid(gobject = my_giotto_object,
                                      sdimx_stepsize = 500,
                                      sdimy_stepsize = 500,
                                      minimum_padding = 0)
# Visualize spatial grid
spatPlot(gobject = my_giotto_object,
         cell_color = 'leiden_clus',
         point_size = 4,
         show_grid = T,
         grid_color = 'grey',
         spatial_grid_name = 'spatial_grid')
```

2. Spatial network.

   Prior to running spatial tissue composition and spatial gene expression analyses, as well as an HMRF analysis, we must create a spatial network, which provides spot resolution. In this example, we will use a Delaunay network. To assess network statistics for a Delaunay network, we can create the following plot (Fig. 17).

```
plotStatDelaunayNetwork(gobject = my_giotto_object, maximum_distance = 400)
```

   We can use the above plots to inform our parameter selection during network creation.

```
my_giotto_object <- createSpatialNetwork(gobject = my_giotto_object,
                                         minimum_k = 6,
                                         maximum_distance_delaunay = 350)
```

   After creating the network, we will visualize our network (Fig. 18A), with each spot labeled by cluster.

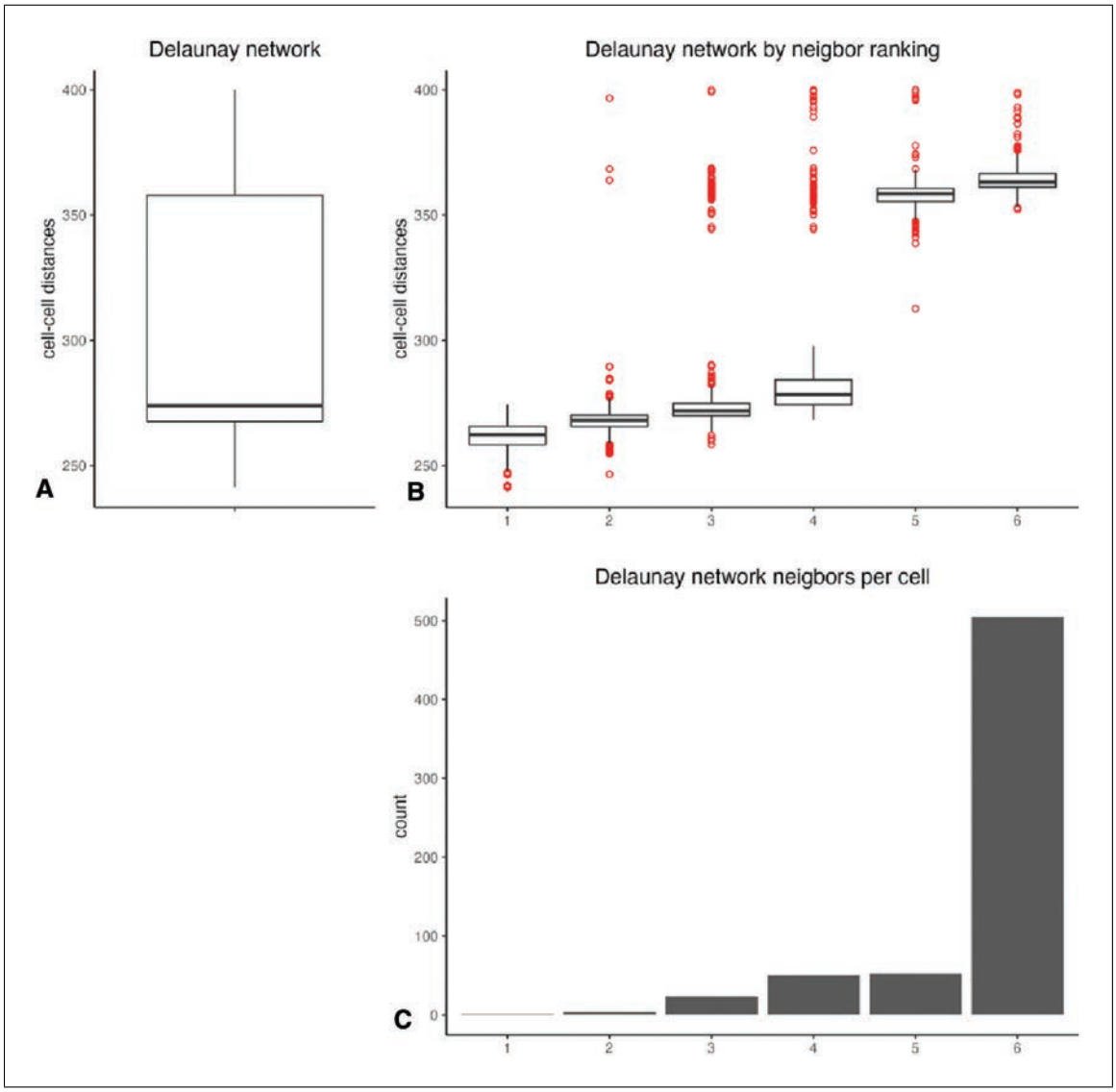

Figure 17 An overview of the network statistics for a Delaunay network. (A) The distribution of cell-cell distances. (B) The distribution of cell-cell distances based on neighbor ranking. (C) The count of neighbors per cell.

```
spatPlot(gobject = my_giotto_object,
         show_network = T,
         point_shape = "no_border",
         network_color = 'black',
         spatial_network_name = 'Delaunay_network',
         point_size = 3,
         cell_color = "leiden_clus",
         coord_fix_ratio = 1)
```

The results can be also overlaid with image information (Fig. 18B).

3. Spatial gene expression patterns.

   In the following example, we will use binSpect to analyze spatially coherent gene patterns. This analysis requires that a spatial network be created beforehand. The following function creates a matrix that displays each gene and its score as well as statistical significance (*p*-value).

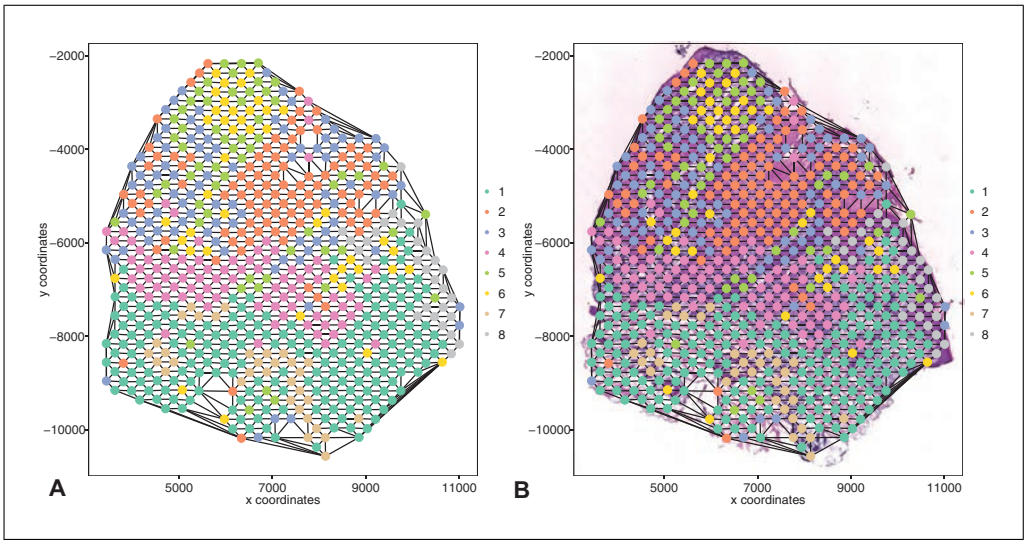

Figure 18 Visualization of the spatial network. (A) A visual representation of a Delaunay network connecting each spot of our data. (B) Delaunay network overlaid on an H&E-stained image.

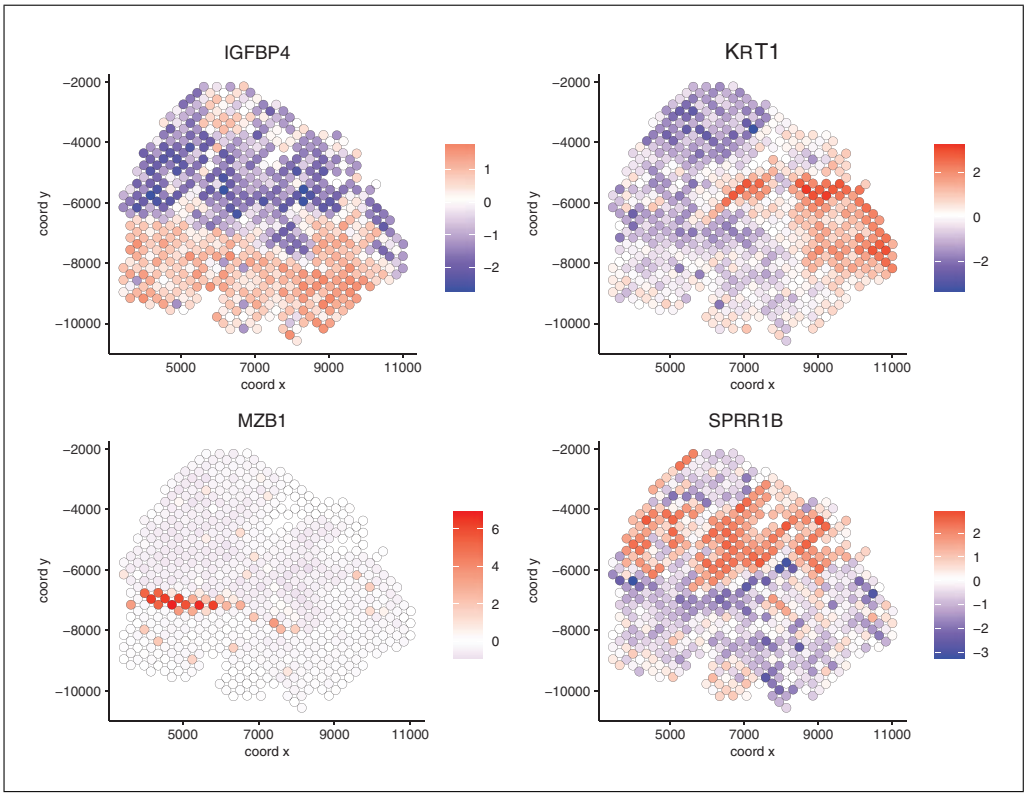

Figure 19 Spatial gene plots of 4 selected genes with different patterns.

```
binspect_k <- binSpect(gobject    my_giotto_object,
                       bin_method    "kmeans",
                       expression_values    "normalized",
                       spatial_network_name    "Delaunay_network")
```

We can now visualize our results. We have chosen 4 interesting genes that have different patterns to visualize (Fig. 19):

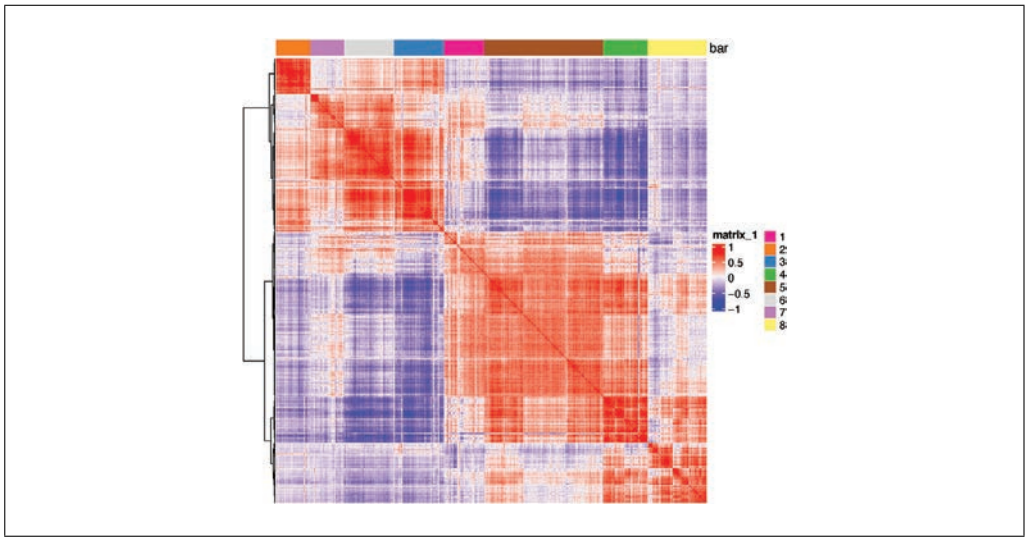

Figure 20 Heatmap representing spatial correlation genes.

```
interesting_genes = c("IGFBP4", "KRT1", "MZB1", "SPRR1B")
spatGenePlot(gobject = my_giotto_object,
             expression_values = "scaled",
             genes = interesting_genes,
             cow_n_col = 2,
             point_size = 2.5)
```

4. Spatial gene co-expression modules.

   First, we will detect spatial correlation genes and calculate spatial correlation scores.

```
ext_spatial_genes = binspect_k[1:400]$genes
spat_cor_netw_DT = detectSpatialCorGenes(my_giotto_object,
                                         method = 'network',
                                         spatial_network_name = 'Delaunay_network',
                                         subset_genes = ext_spatial_genes)
```

   Now, we can cluster and visualize the correlation scores using a heatmap (Fig. 20).

```
spat_cor_netw_DT = clusterSpatialCorGenes(spat_cor_netw_DT,
                                          name = 'spat_netw_clus',
                                          k = 8)
heatmSpatialCorGenes(gobject = my_giotto_object,
                     spatCorObject = spat_cor_netw_DT,
                     use_clus_name = 'spat_netw_clus')
```

   Now we can create metagenes from the cluster modules created in the previous step. Following analysis, we will plot metagenes per cluster (Fig. 21).

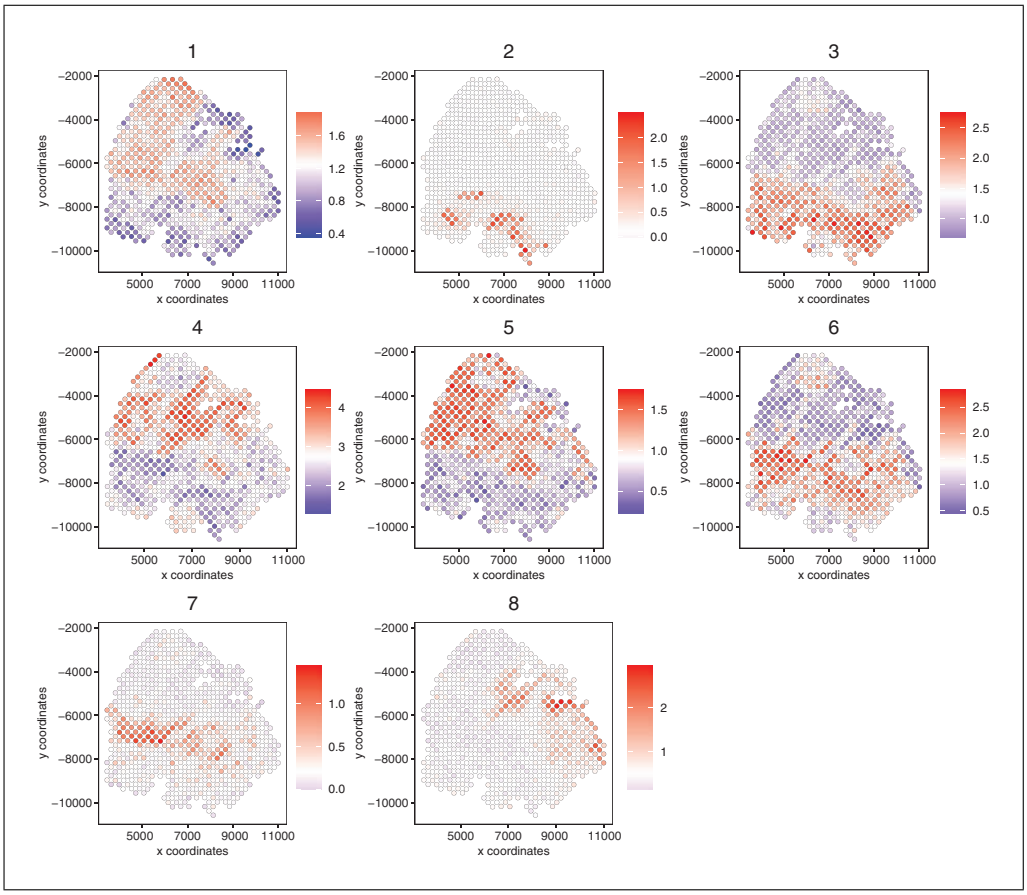

Figure 21 Spatial plot representing metagenes created from cluster modules.

```
cluster_genes_DT    showSpatialCorGenes(spat_cor_netw_DT,
                                        use_clus_name    'spat_netw_clus',
                                        show_top_genes    3)
top40_per_module    cluster_genes_DT[, head(.SD, 40), by    clus]
# 3. create metagenes from cluster modules and visualize
cluster_genes    top40_per_module$clus; names(cluster_genes)    top40_per_module$gene_ID
my_giotto_object    createMetagenes(my_giotto_object,
                                    gene_clusters    cluster_genes,
                                    name    'cluster_metagene')
spatCellPlot(my_giotto_object,
             spat_enr_names    'cluster_metagene',
             cell_annotation_values    as.character(c(1:8)),
             point_size    1.5,
             cow_n_col    3)
```

SPATIAL DOMAIN DETECTION BY USING A HIDDEN MARKOV RANDOM FIELD MODEL

BASIC PROTOCOL 6

Whereas the spatial patterns of individual genes can be identified by using the previous protocol, additional methods are needed to characterize the spatial organization of cell states defined by the combinatorial pattern of all genes. To this end, Giotto implements a hidden Markov random field (HMRF) (Zhu, Shah, Dries, Cai, & Yuan, 2018) to identify spatial coherent domains. A spatial domain may be formed by a cluster of cells from the same cell type, but more generally consists of a mixture of cell types that share similar expression patterns of spatial genes.

The protocols described above are standard in the sense that they require minimal information (gene expression matrix and spatial coordinates) and address the most common tasks. However, further analyses are needed to gain additional insights or to incorporate additional data information. In the following section, we present two support protocols as concrete examples.

Necessary Resources

See Basic Protocol 1

1. Implementation.

   In addition to examining the spatial pattern of individual genes, Giotto also allows users to identify the spatial coherence at the cell state level. This is achieved through the use of a hidden Markov random field (HMRF), a method that assigns clustering by considering both the expression of the group of cells and that of its spatial neighbors. Unlike the previous methods of clustering which are performed purely on the expression space and then mapped onto spatial data, HMRF requires the use of a spatial network (either kNN or Delaunay).

   We will be assuming that the reader is continuing from after the generation of the Delaunay network shown in the previous section. Results can be viewed using `viewHMRFresults2D()`, and the annotations can be added to the Giotto object using `addHMRF()`. The following code generates new files, so we will first create a new directory and then run our HMRF analysis.

```
hmrf_folder <- paste0(path.expand(save_directory),'//','11_HMRF/')
if(!file.exists(hmrf_folder)) dir.create(hmrf_folder, recursive = T)
# input is the top 40 genes per co-expression module
HMRF_spat_genes = doHMRF(gobject = my_giotto_object,
                         expression_values = "normalized",
                         spatial_genes = names(cluster_genes),
                         spatial_network_name = "Delaunay_network",
                         zscore = "none",
                         k = 8,
                         betas = c(0,5,8),
                         output_folder = paste0(hmrf_folder, '/', 'HMRF_output2'))
## add HMRF of interest to giotto object
my_giotto_object = addHMRF(gobject = my_giotto_object,
                           HMRFoutput = HMRF_spat_genes,
                           k = 8, betas_to_add = c(0, 10, 15, 20),
                           hmrf_name = 'HMRF')
```

   We can visualize our results over a spatial plot (Fig. 22).

```
spatPlot2D(my_giotto_object,
           cell_color = 'HMRF_k8_b.0',
           show_image = TRUE,
           point_size = 4.75,
           coord_fix_ratio = 1)
spatPlot2D(my_giotto_object,
           cell_color = 'HMRF_k8_b.20',
           show_image = TRUE,
           point_size = 4.75,
           coord_fix_ratio = 1)
```

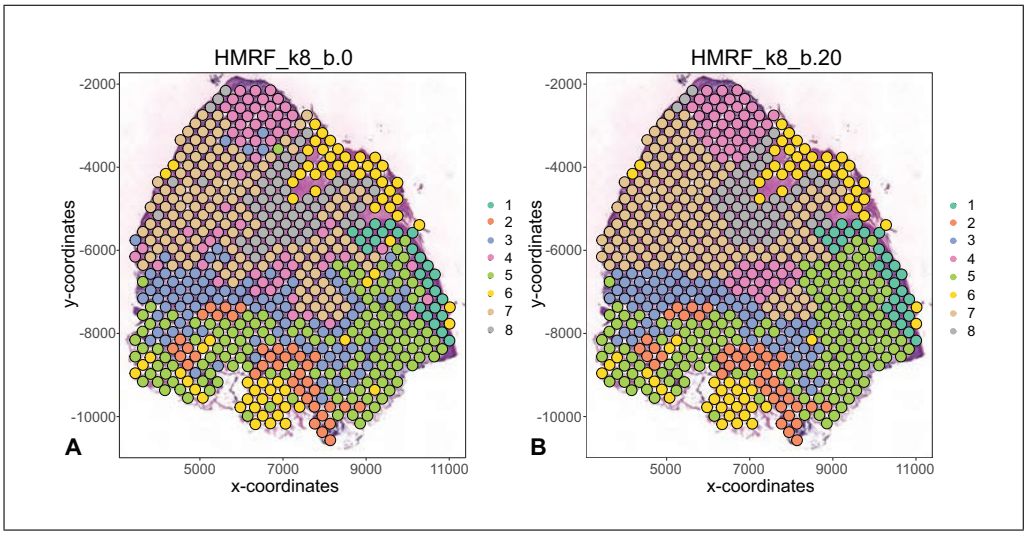

Figure 22 HMRF results with (A) beta 0 and (B) beta 20.

SPATIAL PROXIMITY–ASSOCIATED CELL-CELL INTERACTIONS

SUPPORT PROTOCOL 1

In this support protocol, we will first assess spatial organization at the cellular level. Giotto can use the spatial network obtained in Basic Protocol 5 to calculate how frequently cell types, or other spatial annotations, are found in close proximity to each other. Visualization of these results is provided through barplot, network, or heatmap representations. Next, we will demonstrate how the unique spatial domain organization of a tissue, as discussed in Basic Protocol 6, might (in)directly affect gene expression and signaling pathways Giotto introduces a number of complementary methods. First, Giotto identifies interaction-changed genes (ICG) by examining which genes are differentially expressed when a pair of cell types interact. In other words, these genes are up- or down-regulated in one cell type when in close spatial proximity to another cell type. Second, the results of individual ICGs can be combined to explore, in an unbiased manner, where a pair of genes are differentially expressed in two neighboring cell types. In this latter analysis, any identified gene pair does not necessarily need to be linked at the molecular level, and the expression levels of both genes can be up-regulated, down-regulated, or unchanged. Finally, Giotto provides a method to identify ligand-receptor interactions between neighboring cells. More specifically, it uses known ligand-receptor pairs as proxies to examine how a pair of cells communicate when they are in close physical proximity, such that one cell expresses increased levels of the ligand and the other cell of the cognate receptor.

Necessary Resources

See Basic Protocol 1

1. Cell proximity enrichment.

   We will run an analysis to assess how frequently two cell types are found in close spatial proximity by comparing the observed and expected cell-to-cell interactions within the created spatial network. Here we use the leiden clustering results as proxies for cell types for each ST spot.

```
cell_proximities    cellProximityEnrichment(gobject    my_giotto_object,
                                            cluster_column    'leiden_clus',
                                            spatial_network_name    'Delaunay_network',
                                            adjust_method    'fdr',
                                            number_of_simulations    1000)
```

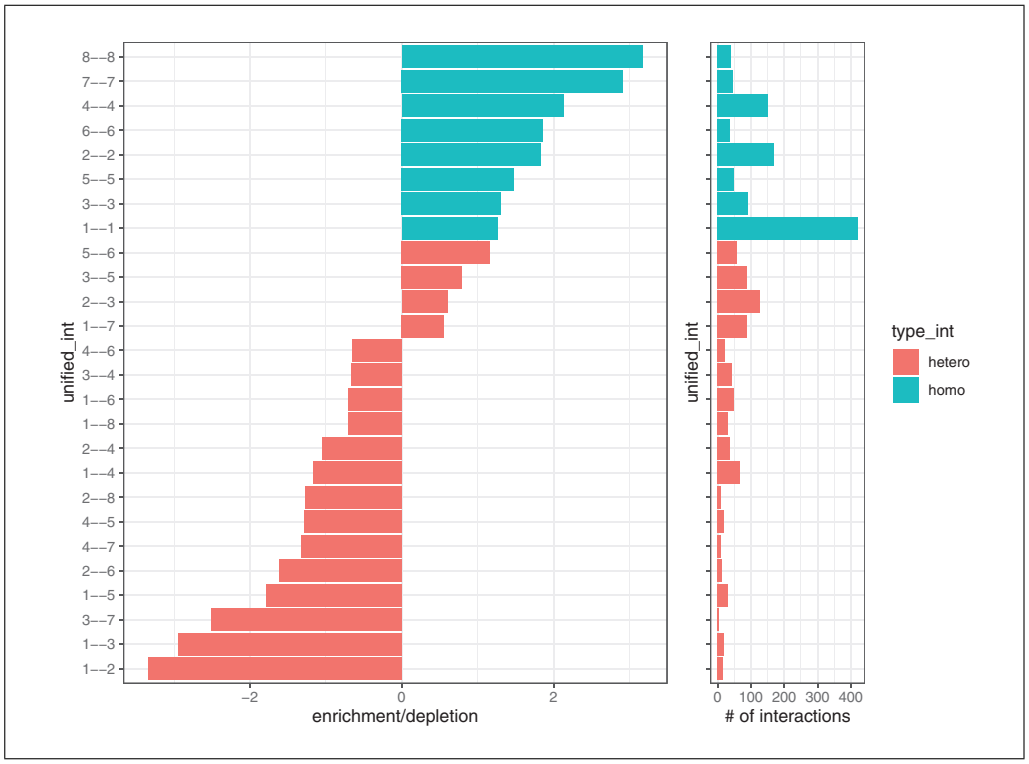

Figure 23 Barplot depicting the ratio of observed over expected frequencies of pairwise interacting clusters (left). Barplot showing the total number of interactions between each pair (right).

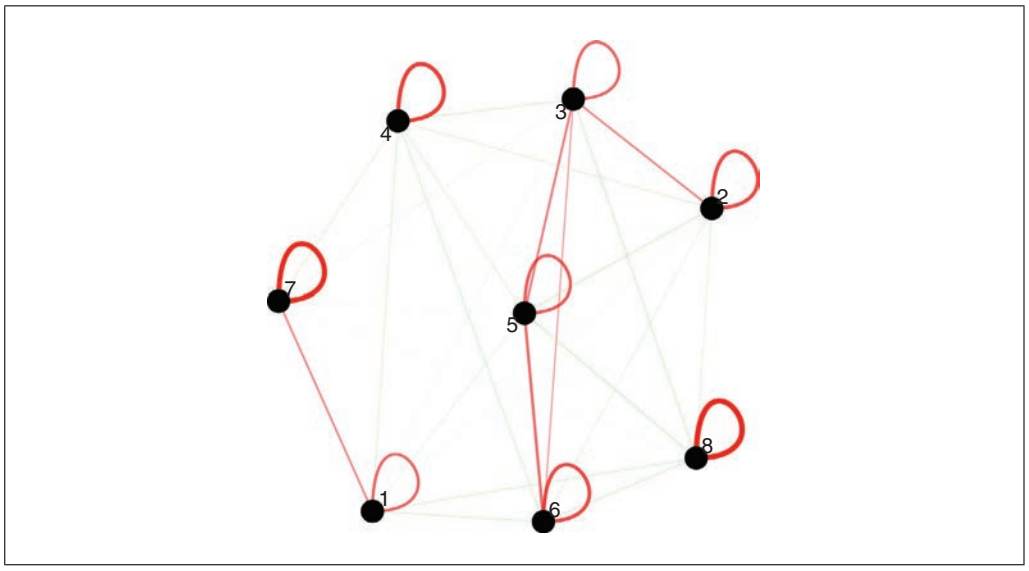

Figure 24 Visualization of the cell proximity network. Enriched or depleted interactions are depicted in red and green, respectively. Width of the edges indicates the strength of enrichment or depletion.

Now we can visualize our cell-to-cell proximity results in a variety of ways, such as a barplot (Fig. 23).

```
cellProximityBarplot(gobject    my_giotto_object,
                     CPscore    cell_proximities,
                     min_orig_ints    3,
                     min_sim_ints    1)
```

We can also show a cell-to-cell proximity network, including self-edges (Fig. 24).

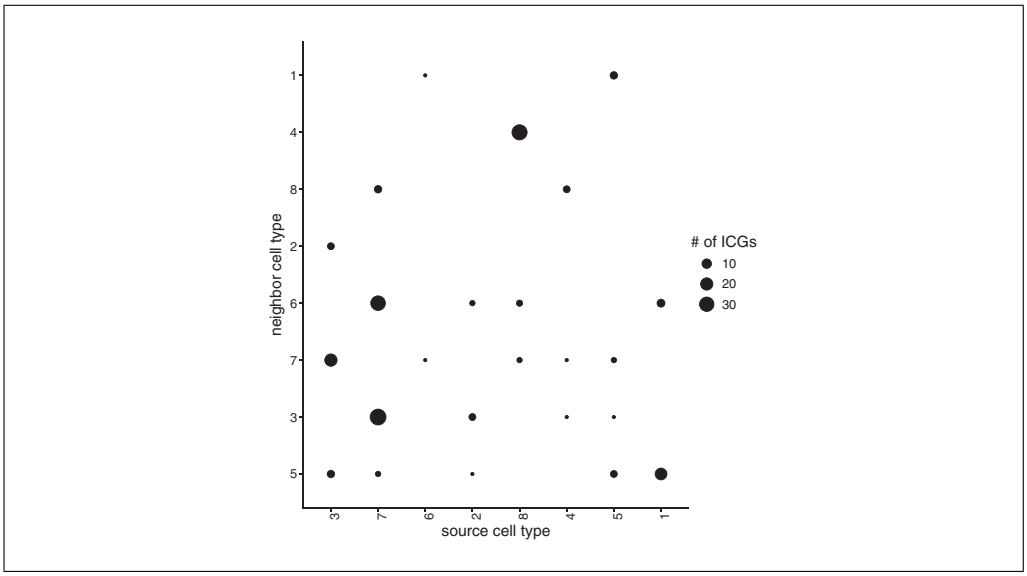

Figure 25 The total number of interaction changed genes (ICG). The corresponding spatial context is shown in the *x*- and *y*-axis.

```
cellProximityNetwork(gobject = my_giotto_object,
                     CPscore = cell_proximities,
                     remove_self_edges = F,
                     self_loop_strength = 0.3,
                     only_show_enrichment_edges = F,
                     rescale_edge_weights = T,
                     node_size = 8,
                     edge_weight_range_depletion = c(1,2),
                     edge_weight_range_enrichment = c(2,10))
```

2. Interaction-changed genes (ICGs).

   Next, we will assess the interaction-changed genes (ICGs) that are found to be differentially expressed when 2 spatial domains, identified in Basic Protocol 6, are in proximity. First, we find our ICGs and visualize how many ICGs are shared between neighbor cells (Fig. 25).

```
## select top 25th highest expressing genes
gene_metadata = fDataDT(my_giotto_object)
high_expressed_genes = gene_metadata[mean_expr_det > quan-
tile(gene_metadata$mean_expr_det)[4]]$gene_ID
## identify genes that are associated with proximity to other cell types
ICGscoresHighGenes = findInteractionChangedGenes(gobject = my_giotto_object,
                                    selected_genes = high_expressed_genes,
                                    spatial_network_name = 'Delaunay_network',
                                    cluster_column = 'HMRF_k8_b.20',
                                    diff_test = 'permutation',
                                    adjust_method = 'fdr',
                                    nr_permutations = 2000,
                                    do_parallel = TRUE, cores = 4)
## visualize
plotCellProximityGenes(my_giotto_object,
                       cpgObject = ICGscoresHighGenes,
                       method = 'dotplot')
```

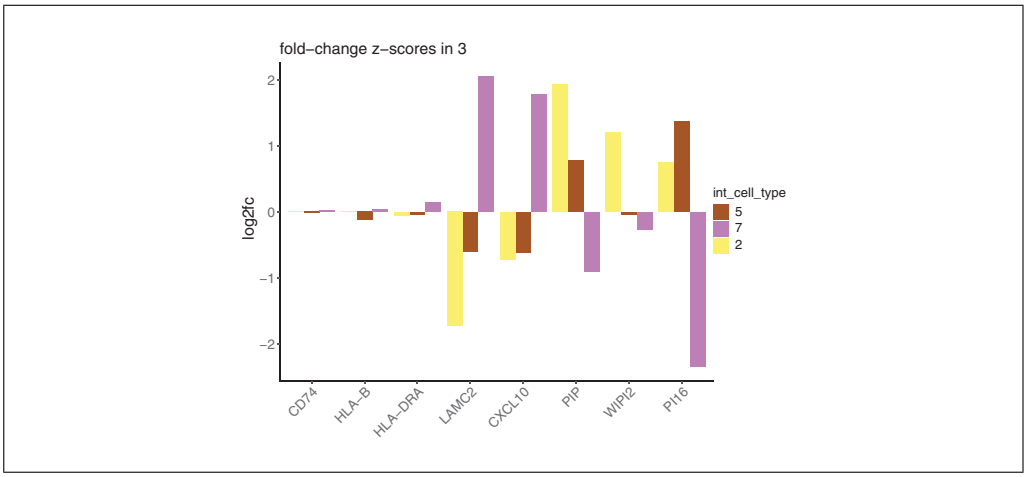

Figure 26 Examples of interaction changed genes identified by Giotto.

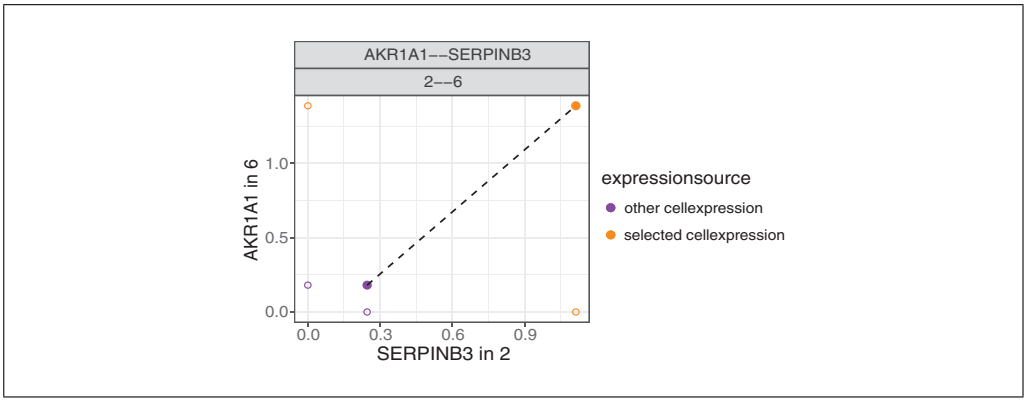

Figure 27 Depiction of a representative pair of identified ICGs from one-way analysis.

Now we can visualize a selected subset of our identified ICGs (Fig. 26).

```
## filter genes
ICGscoresFilt = filterInteractionChangedGenes(ICGscoresHighGenes)
## visualize subset of interaction changed genes (ICGs)
ICGscoresFilt$CPGscores[type_int == 'hetero'][cell_type == '3']
ICG_genes = c('LAMC2', 'CXCL10', 'PIP', 'WIPI2', 'PI16')
ICG_genes_types = c('7', '7', '2', '2', '5')
names(ICG_genes) = ICG_genes_types
plotInteractionChangedGenes(gobject = my_giotto_object,
                            cpgObject = ICGscoresHighGenes,
                            source_type = '3',
                            source_markers = c('CD74', 'HLA-B', 'HLA-DRA'),
                            ICG_genes = ICG_genes)
```

We can now also combine the ICGs and identify pairs of ICG in two interaction cell types. This provides a straightforward and unbiased manner in which to associate ICG in two directions. Here we highlight two examples. In the first example (Fig. 27), when HMRF domain 2 and 6 are in proximity the gene AKR1A1 is upregulated in domain 6 while SERPINB3 is upregulated in domain 2.

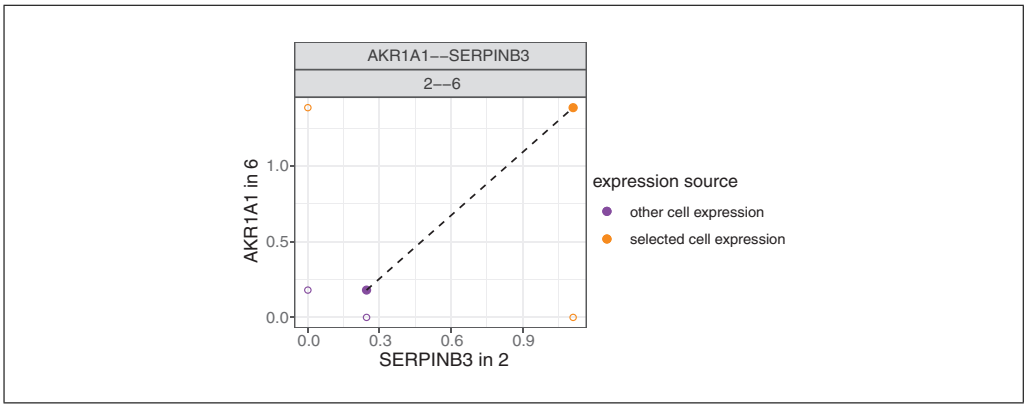

Figure 28 Depiction of a representative pair of identified ICGs from two-way analysis.

```
combo_ICGs  combineInteractionChangedGenes(cpgObject  ICGscoresHighGenes)
combo_ICGs$combCPGscores[type_int  'hetero' & direction  'both_up'][p.adj_1 <
0.01 & p.adj_2 < 0.01 & abs(log2fc_1) > 1 & abs(log2fc_2) > 1]
# visualize
plotCombineInteractionChangedGenes(gobject  my_giotto_object,
                        combCpgObject  combo_ICGs,
                        selected_interactions  '2-6',
                        selected_gene_to_gene  c('AKR1A1-SERPINB3'))
```

In a second example (Fig. 28) between interacting cells in HMRF domain 1 and 5, the gene changes are in opposite directions. Here OLA1 is upregulated in cells within domain 1 and MUCL1 is downregulated in cells within domain 5.

```
plotCombineInteractionChangedGenes(gobject  my_giotto_object,
                        combCpgObject  combo_ICGs,
                        selected_interactions  '1-5',
                        selected_gene_to_gene  c('MUCL1-OLA1'))
```

3. Ligand-receptor signaling.
   We can also analyze ligand-receptor signaling in the context of our spatial domain data, as identified in Basic Protocol 6. The ligand-receptor dataset has already been imported and can be loaded here.

```
LR_data  data.table::fread(file  paste0(data_directory, '/', 'PairsLigRec.txt'))
LR_data[, ligand_det :  ifelse(HPMR.Ligand %in% my_giotto_object@gene_ID, T, F)]
LR_data[, receptor_det :  ifelse(HPMR.Receptor %in% my_giotto_object@gene_ID, T, F)]
LR_data_det  LR_data[ligand_det  T & receptor_det  T & Pair.Source  'known']
select_ligands  LR_data_det$HPMR.Ligand
select_receptors  LR_data_det$HPMR.Receptor
spatial_all_scores  spatCellCellcom(my_giotto_object,
                                    spatial_network_name  'Delaunay_network',
                                    cluster_column  'HMRF_k8_b.20',
                                    random_iter  1000,
                                    gene_set_1  select_ligands,
                                    gene_set_2  select_receptors,
                                    adjust_method  'fdr',
                                    do_parallel  T,
                                    cores  4,
                                    verbose  'none')
```

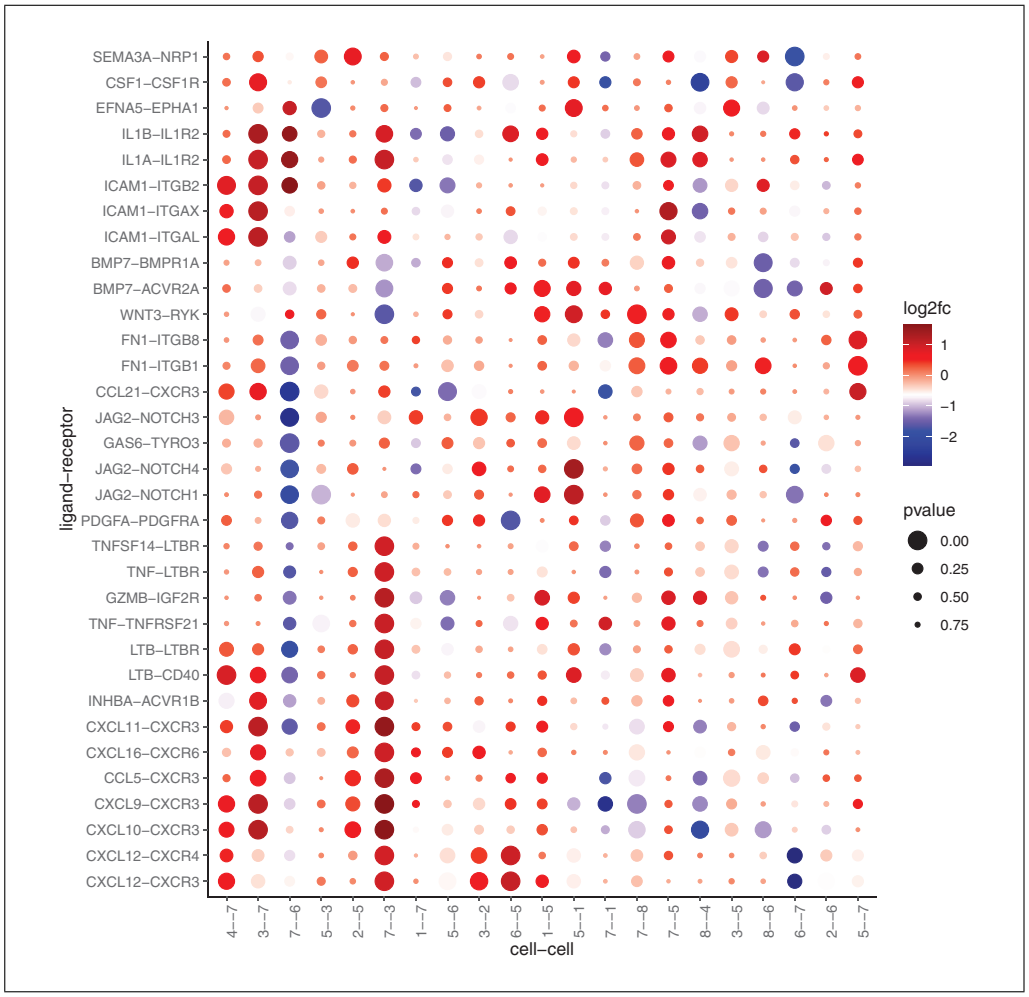

Figure 29 Ligand-receptor pairs (*y*-axis) that are detected by Giotto to exhibit differential cell-cell communication scores due to spatial cell-cell interactions (*x*-axis). The size of the dot is correlated with the adjusted *p*-value and the color indicates increased (red) or decreased (blue) activity.

Now we can select our top ligand-receptor pairs and visualize the results in a dotplot (Fig. 29). This analysis illustrates how ligand-receptors pairs are used more or less frequently when cells from neighboring domains are found in proximity.

```
selected_spat    spatial_all_scores[p.adj <   0.1 & abs(log2fc) > 0.5 & lig_nr >   3 &
rec_nr >   3]
data.table::setorder(selected_spat, -PI)
top_LR_ints    unique(selected_spat[order(-abs(PI))]$LR_comb)[1:33]
top_LR_cell_ints    unique(selected_spat[order(-abs(PI))]$LR_cell_comb)[1:33]
plotCCcomDotplot(gobject    my_giotto_object,
                    comScores    spatial_all_scores,
                    selected_LR    top_LR_ints,
                    selected_cell_LR    top_LR_cell_ints,
                    cluster_on    'PI')
```

SUPPORT PROTOCOL 2 ASSEMBLY OF A REGISTERED 3D GIOTTO OBJECT FROM 2D SLICES

Many spatial transcriptomic methods can only generate 2D datasets, but it is possible to *z*-stack these 2D slices into a 3D dataset. In this support protocol, we will demonstrate how to combine 2D spatial transcriptomic data together into a single 3D dataset using

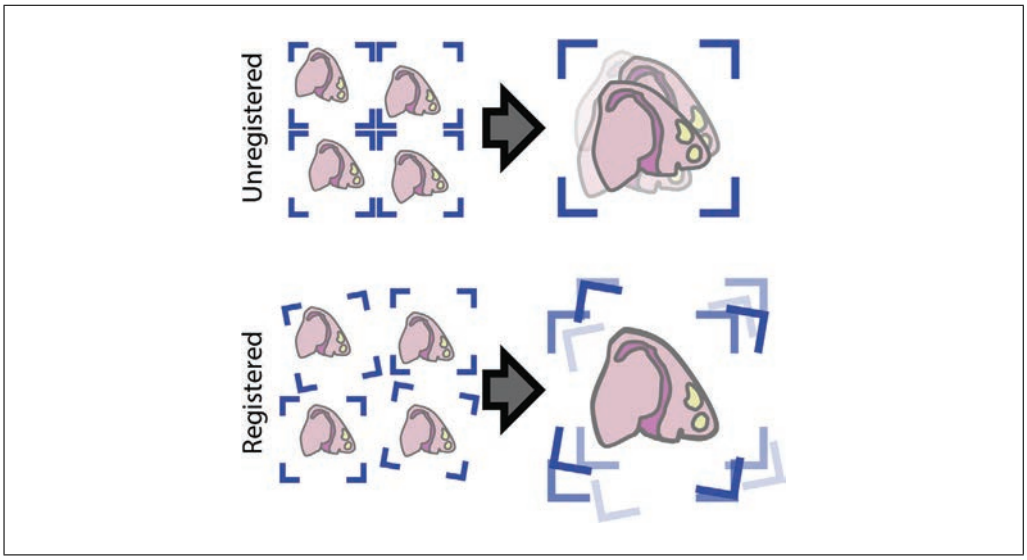

Figure 30 Stacking unregistered 2D datasets results in a 3D dataset where the tissue regions are out of sync with each other. Registering the data prior to combining resolves this issue.

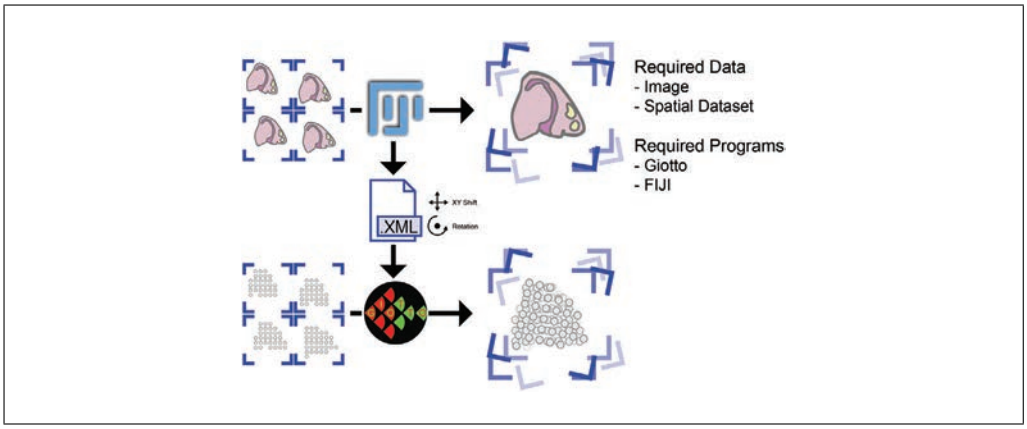

Figure 31 Overview of the image registration and 3D dataset assembly process.

Giotto and Fiji. 2D slices make spatial sense within their own planes, but when stacking them together, the data are often out of sync with each other due to differences in tissue placement on or across capture regions which, rather than the tissues, define the spatial frame of reference of spatial transcriptomic data. Thus, in order to faithfully represent the 3D information, it is necessary to first align all the 2D slices. Since it is difficult to use gene expression data alone as the ground truth for tissue alignment, image registration is performed based on the paired staining images (Fig. 30). Using Fiji, this protocol performs "rigid" registration, meaning that the only transformations allowed when the program seeks for a good alignment across images are rotations and x and y shifts. This method is fully technology-agnostic, and the resulting rotations and translations performed on the images can then be mapped back onto the spatial locations in Giotto to bring them into sync (Fig. 31).

This protocol will begin by using Fiji to perform image registration based on the 3 slices of spatial data from Patient 2 of the spatial transcriptomic dataset (Ji et al., 2020). The `.xml` transforms registration output will then be used in Giotto to create the 3D dataset.

Necessary Resources

See Basic Protocol 1

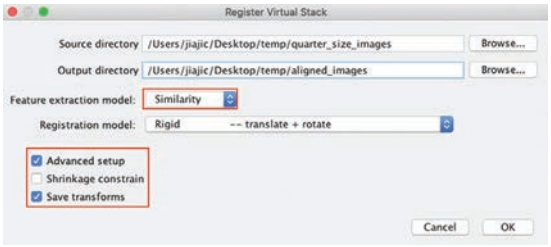

Figure 32 Image registration settings for the Register Virtual Stack Slices plugin in Fiji.

Rigid image registration

1. Image preparation:

   Image registration is done in Fiji (version 2.3.0 /1.53m) to find the necessary spatial transformations before moving to Giotto. The first step is to scale down the staining image, if necessary, in order to both speed up the registration process and make it less RAM intensive. Image dimensions scaled down to around 2000 × 2000px to 4000 × 4000px are recommended. If image scaling is desired, all images must be scaled in exactly the same way while keeping the aspect ratio the same. The scaling factor used must also be kept note of.

   To demonstrate functionality, in this example a scale factor of **0.25** was performed on all three images of this dataset and these can be found in the **quarter_size_images** subdirectory of the **imgReg** folder in the **working directory** after unzipping it.

   The outputs for image registration should be placed in their own folders. Empty folders for the registered images and the spatial transformation `.xml` files are **aligned_images** and **transforms,** respectively, also in the unzipped **imgReg** folder. There is an additional folder called **exampleXMLS** in the **transforms** folder that contains example .xml files for this dataset in case the reader would like to skip this portion of the protocol.

2. Image registration:

   In Fiji, go to Plugins → Registration → Register Virtual Stack Slices (Fig. 32). Find the directories for the source (quarter_size_images) and output (aligned_images) folders. Ensure that feature extraction is set to **Similarity** and that the registration model is set to **Rigid**. Also, ensure that **Save transforms** is checked.

   Upon pressing OK, a new window will appear asking for a directory in which to save the `.xml` transform files. Select the prepared folder (**transforms**) (Fig. 33).

   Next, a new window will appear asking for the target file to which all other images will be registered. Navigate to the image of a section that has tissue representative of most of the other images and is as close to the middle of the stack as possible and open it as the target file. In this example, all images will be aligned according to the image for the patient 2 rep2 sample (Fig. 34).

   Another window labeled **Feature extraction** will then open as part of the advanced options checked earlier. Click OK on these settings.

   Registration will then proceed, followed by the output directory being populated with the registered image and then updated once as the software performs a second pass to increase image bounds so that all images line up with each other afterwards. The image transformations performed will all be recorded in TrakEM2 format in `.xml` files that will appear in the transforms folder. If registration does not succeed, see the troubleshooting section.

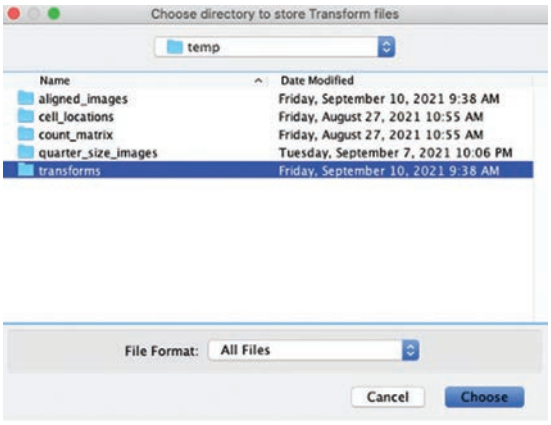

Figure 33 Register Virtual Stack Slices save directory prompt window.

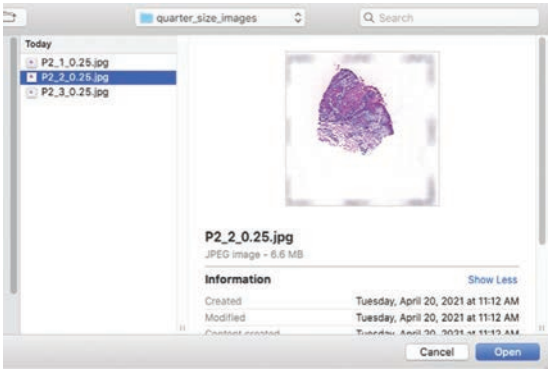

Figure 34 Register Virtual Stack Slices registration target selection prompt window.

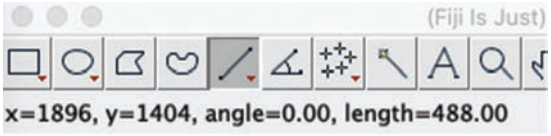

Figure 35 Fiji measuring tool can be used to measure pixel distances to determine a micron to pixel scaling factor. This is needed for accurate *z*-axis spacing for the 3D dataset.

3. Determine micron to pixel scaling factor.

   This process is different depending on the dataset due to differences in spatial transcriptomic methods and image magnifications, and currently can only be done manually by measuring pixel distances between landmarks with known real-world distances. This can be done using Fiji by drawing a line, after which the distance will appear in the status bar (Fig. 35).

   Pressing "M" on the keyboard will also save the values into a table.

4. Aligning 2D slices.

   The remaining steps return to R and Giotto.

5. Loading spatial expression and registration data:

   First, ensure that transform `.xml` files were outputted to the **transforms** folder. If the registration step was skipped, then please run the commented-out code to set **xmlpaths** by removing the preceding hash mark.

```
# Get filepaths
exprpaths <- list(paste0(data_directory, "/P2_1_expression.csv"),
                  paste0(data_directory, "/P2_2_expression.csv"),
                  paste0(data_directory, "/P2_3_expression.csv"))
spatpaths <- list(paste0(data_directory, "/P2_1_spatial_locs.csv"),
                  paste0(data_directory, "/P2_2_spatial_locs.csv"),
                  paste0(data_directory, "/P2_3_spatial_locs.csv"))
xmlpaths <- list(paste0(data_directory, "/imgReg/transforms/P2_1_0.25.xml"),
                  paste0(data_directory, "/imgReg/transforms/P2_2_0.25.xml"),
                  paste0(data_directory, "/imgReg/transforms/P2_3_0.25.xml"))
# Alternatively, use the given example .xml files by setting the following path:
# xmlpaths <-
list.files(paste0(data_directory, "/imgReg/transforms/exampleXMLs"), full.names   TRUE)
# Read in data
xmls <- lapply(xmlpaths, readChar, nchar   1000)
spatlocs <- lapply(spatpaths, read.csv)
exprCounts <- lapply(exprpaths, Giotto::readExprMatrix)
```

6. Setting z-axis spacing:

   For this dataset, each section was taken serially with a thickness of 10 microns. This value of 10 microns will then be multiplied by the ratio of microns to pixels which was calculated to be 1.8.

```
z_vals <- c(0,10,20)
z_vals <- z_vals * 1.8
```

7. Creating 3D Giotto object:

   A Giotto object with the spatial locations of all slices registered to each other is then generated. Required inputs, in order supplied in the example, are the list of expression count matrices, the list of unregistered spatial locations, the names of the columns in those spatial location matrices holding x and y coordinate values, the scaling factor of images used in image registration relative to spatial locations, the list of transformation `.xml` files from image registration, and a vector of z values to be used.

   Note that all data provided in lists and in the vector of z values must be in the same order by slice of origin.

```
my_giotto_object_3D <- createRegZStackGobject(expression_list   exprCounts,
                                              spatlocs_list   spatlocs,
                                              xvals   "pixel_x",
                                              yvals   "pixel_y",
                                              scalefactor   0.25,
                                              transformXML   xmls,
                                              z_vals   z_vals)
```

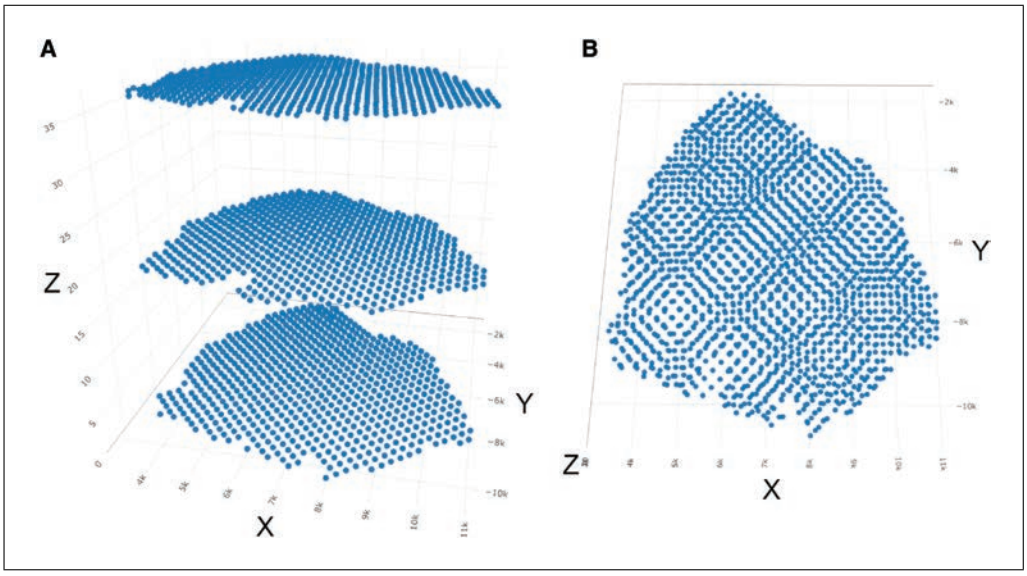

Figure 36 Visualization of the spatial locations of the final 3D Giotto object. (A) axis_scale "cube"; (B) axis_scale "real".

8. Visualizing 3D data:

   The 3D plot shows that all the slices' spatial locations are now aligned together. The **axis_scale** argument takes either **cube** or **real** as input. **cube** displays the spatial locations with all axes scaled to the same length, making it easy to see any issues with alignment. **real** plots all axes to actual scale (Fig. 36).

```
spatPlot3D(gobject = my_giotto_object_3D,
           axis_scale = "cube",
           point_size = 5)
spatPlot3D(gobject = my_giotto_object_3D,
           axis_scale = "real",
           point_size = 5)
```

   The 3D Giotto object can then be analyzed in the same ways as previously described. Loading in Leiden cluster annotations, (steps shown in the markdown) we can see that the clusters are aligned across 2D slices (Fig. 37).

```
# Visualize 3D data with leiden clusters
my_colors = c("#9442f2","#b6d326","#fa002a","#fa00d4","#013bb5",
              "#666666","#017fd6","#50cfff","#cf8cff","#01d671")
cellMetadata_3D <- readRDS(file = paste0(data_directory, "/imgReg/cellMetadata3D.rds"))
my_giotto_object_3D <- addCellMetadata(gobject = my_giotto_object_3D,
                                       new_metadata = cellMetadata_3D,
                                       by_column = "cell_ID")
spatPlot3D(gobject = my_giotto_object_3D,
           axis_scale = "cube",
           point_size = 5,
           cell_color_code = my_colors,
           cell_color = "leiden_clus")
spatPlot2D(gobject = my_giotto_object_3D,
           point_size = 3.5,
           coord_fix_ratio = 1,
           cell_color_code = my_colors,
           cell_color = "leiden_clus")
```

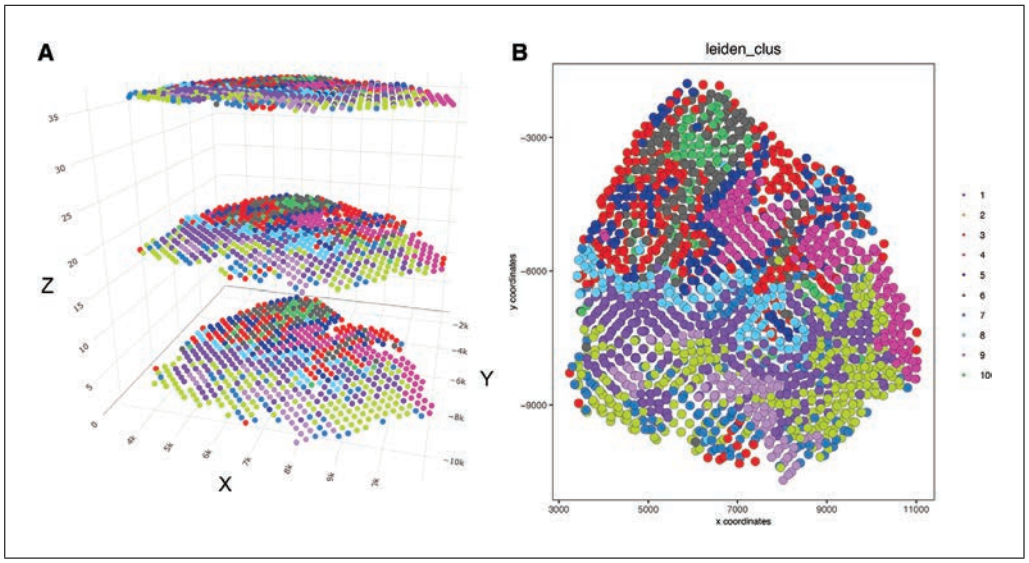

Figure 37 Giotto objects with 3D data can be analyzed in the same way that 2D datasets are. They can also be plotted in either (A) 3D or (B) 2D depending on whether the 2D or 3D plotting command is used.

Table 1 Troubleshooting

Issue	Fix
Error in inti (j, n x@Dim[2], dn[[2]], give.dn FALSE): invalid character indexing (when subsetting and reordering the expression matrix according to the spatial locations	Ensure that every spatial location has a corresponding set of expression data.
Error: cannot coerce class "structure ("dgCMatrix", package "Matrix")" to a data.frame	Load the Matrix library first by running `library (Matrix)`
Fiji returns: mpicbg.models.NotEnoughDataPointsException: 0 data points are not enough to estimate a 2d rigid model, at least 2 data points required… Not all images were registered and/or no .xml transform files were written	After making sure that the image is not larger than roughly 4000 × 4000px, in the **Feature extraction** window, try increasing the **feature descriptor size** in increments of 1.
Fiji returns: Could not open file after registration completes Registered images show up in output folder, but preview stack only shows a subset of images No `.xml` files were written	Run Fiji as administrator (need to have admin rights on the machine) **PC:** Close Fiji, then right click the Fiji icon → run as administrator **MacOS:** in terminal, run this line: `sudo /Applications/Fiji.app/Contents/MacOS/ImageJ-macosx` (or wherever else Fiji is installed) If still an issue, try the above fix of going to advanced options and increasing feature descriptor size in increments of 1.
Fiji does not display the registration settings screen and becomes unresponsive. May happen after another registration completes	Force quit then restart Fiji

COMMENTARY

Troubleshooting

Table 1 summarizes a few common issues and the recommended solutions.

Acknowledgments

The work was supported by the National Cancer Institute (UH3CA255134), the National Institute of Mental Health (RF1MH 128970), and the National Institute on Aging (R01AG066028).

Author Contributions

Natalie Del Rossi: data curation, formal analysis, investigation, software, visualization, writing original draft, writing review and editing; **Jiaji Chen:** data curation, formal analysis, investigation, methodology, software, visualization, writing original draft, writing review and editing; **Guo-Cheng Yuan:** conceptualization, investigation, methodology, project administration, writing original draft, writing review and editing; **Ruben Dries:** conceptualization, formal analysis, investigation, methodology, software, supervision, writing original draft, writing review and editing.

Conflict of Interest

The authors declare no conflicts of interests.

Data Availability Statement

The data that supports the findings of this study are openly available in Github at *https://github.com/RubD/spatial-datasets/tree/master/data/2020_ST_SCC*.

Literature Cited

Becht, E., McInnes, L., Healy, J., Dutertre, C.-A., Kwok, I. W. H., Ng, L. G., … Newell, E. W. (2018). Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*. doi: 10.1038/nbt.4314. Online ahead of print.

Bettcher, B. M., Tansey, M. G., Dorothée, G., & Heneka, M. T. (2021). Peripheral and central immune system crosstalk in Alzheimer disease—a research prospectus. *Nature Reviews Neurology*, *17*, 689–701. doi: 10.1038/s41582-021-00549-x.

Binnewies, M., Roberts, E. W., Kersten, K., Chan, V., Fearon, D. F., Merad, M. … Krummel, M. F. (2018). Understanding the tumor immune microenvironment (TIME) for effective therapy. *Nature Medicine*, *24*, 541–550. doi: 10.1038/s41591-018-0014-x.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*, P10008. doi: 10.1088/1742-5468/2008/10/p10008.

BRAIN Initiative Cell Census Network (BICCN) (2021). A multimodal cell census and atlas of the mammalian primary motor cortex. *Nature*, *598*, 86–102. doi: 10.1038/s41586-021-03950-0.

Buckley, C. D., Ospelt, C., Gay, S., & Midwood, K. S. (2021). Location, location, location: How the tissue microenvironment affects inflammation in RA. *Nature Reviews. Rheumatology*, *17*, 195–212. doi: 10.1038/s41584-020-00570-2.

Chung, N. C. (2020). Statistical significance of cluster membership for unsupervised evaluation of cell identities. *Bioinformatics*, *36*, 3107–3114. doi: 10.1093/bioinformatics/btaa087.

Codeluppi, S., Borm, L. E., Zeisel, A. G., van Lunteren, J. A., … Linnarsson, S. (2018). Spatial organization of the somatosensory cortex revealed by osmFISH. *Nature Methods*, *15*, 932–935. doi: 10.1038/s41592-018-0175-z.

Dong, R., & Yuan, G.-C. (2021). SpatialDWLS: Accurate deconvolution of spatial transcriptomic data. *Genome Biology*, *22*, 145. doi: 10.1186/s13059-021-02362-7.

Dries, R., Chen, J., Del Rossi, N., Khan, M. M., Sistig, A., & Yuan, G.-C. (2021a). Advances in spatial transcriptomic data analysis. *Genome Research*, *31*, 1706–1718. doi: 10.1101/gr.275224.121.

Dries, R., Zhu, Q., Dong, R., Eng, C.-H. L., Li, H., Liu, K., Fu, Y. Zhao, T., Sarkar, A., Bao, F. … Yuan G. C. (2021b). Giotto: A toolbox for integrative analysis and visualization of spatial expression data. *Genome Biology*, *22*, 78. doi: 10.1186/s13059-021-02286-2.

Edsgärd, D., Johnsson, P., & Sandberg, R. (2018). Identification of spatial expression trends in single-cell gene expression data. *Nature Methods*, *15*, 339–342. doi: 10.1038/nmeth.4634.

Finak, G., McDavid, A., Yajima, M., Deng, J., Gersuk, V., Shalek, A. K., … Gottargo, R. (2015). MAST: A flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA-seq data. *Bioinformatics*, *16*, 278. doi: 10.1186/s13059-015-0844-5.

Hanahan, D., & Weinberg, R. A. (2011). Hallmarks of cancer: The next generation. *Cell*, *144*, 646–674. doi: 10.1016/j.cell.2011.02.013.

HuBMAP Consortium (2019) The human body at cellular resolution: The NIH Human Biomolecular Atlas Program. *Nature*, *574*, 187–192. doi: 10.1038/s41586-019-1629-x.

Ji, A. L., Rubin, A. J., Thrane, K., Jiang, S., Reynolds, D. L., Meyers, R. M. … Khavari, P. A. (2020). Multimodal analysis of composition and spatial architecture in human squamous cell carcinoma. *Cell*, *182*, 497–514.e22. doi: 10.1016/j.cell.2020.05.039.

Jiang, L., Chen, H., Pinello, L., & Yuan, G.-C. (2016). GiniClust: Detecting rare cell types

from single-cell gene expression data with Gini index. *Genome Biology*, *17*, 144. doi: 10.1186/s13059-016-1010-4.

Kim, S.-Y., & Volsky, D. J. (2005). PAGE: Parametric analysis of gene set enrichment. *BMC Bioinformatics*, *6*, 144. doi: 10.1186/1471-2105-6-144.

Lewis, S. M., Asselin-Labat, M.-L., Nguyen, Q., Berthelet, J., Tan, X., Wimmer, V. C., … Naik, S. H. (2021). Spatial omics and multiplexed imaging to explore cancer biology. *Nature Methods*, *18*, 997–1012. doi: 10.1038/s41592-021-01203-6.

Longo, S. K., Guo, M. G., Ji, A. L., & Khavari, P. A. (2021). Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics. *Nature Reviews Genetics*, *22*, 627–644. doi: 10.1038/s41576-021-00370-8.

Lun, A. T. L., McCarthy, D. J., & Marioni, J. C. (2016). A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. *F1000Research*, *5*, 2122.

van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Marx, V. (2021). Method of the Year: Spatially resolved transcriptomics. *Nature Methods*, *18*, 9–14. doi: 10.1038/s41592-020-01033-y.

Rao, A., Barkley, D., França, G. S., & Yanai, I. (2021). Exploring tissue architecture using spatial transcriptomics. *Nature*, *596*, 211–220. doi: 10.1038/s41586-021-03634-9.

Regev, A., Teichmann, S. A., Lander, E. S., Amit, I., Benoist, C., Birney, E. … Human Cell Atlas Meeting Participants. (2017). The human cell atlas. *eLife*, *6*, doi: 10.7554/eLife.27041.

Rozenblatt-Rosen, O., Regev, A., Oberdoerffer, P., Nawy, T., Hupalowska, A., Rood, J. E. … Human Tumor Atlas Network (2020). The human tumor atlas network: charting tumor transitions across space and time at single-cell resolution. *Cell*, *181*, 236–249. doi: 10.1016/j.cell.2020.03.053.

Sun, S., Zhu, J., & Zhou, X. (2020). Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies. *Nature Methods*, *17*, 193–200. doi: 10.1038/s41592-019-0701-7.

Svensson, V., Teichmann, S. A., & Stegle, O. (2018). SpatialDE: Identification of spatially variable genes. *Nature Methods*, *15*, 343–346. doi: 10.1038/nmeth.4636.

Traag, V. A., Waltman, L., & van Eck, N. J. (2019). From Louvain to Leiden: Guaranteeing well-connected communities. *Scientific Reports*, *9*, 5233. doi: 10.1038/s41598-019-41695-z.

Tsoucas, D., Dong, R., Chen, H., Zhu, Q., Guo, G., & Yuan, G.-C. (2019). Accurate estimation of cell-type composition from gene expression data. *Nature Communications*, *10*, 1–9. doi: 10.1038/s41467-019-10802-z.

Zhu, Q., Shah, S., Dries, R., Cai, L., & Yuan, G.-C. (2018). Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence in situ hybridization data. *Nature Biotechnology*, doi: 10.1038/nbt.4260.