

Package ‘SpLin’

Mar 2, 2025

Type Package

Title An R Package for Morphometric Transformation and Visualization of Spatiotemporal Omics Data.

Version 1.0.6

Author Wen Luo

Maintainer Wen Luo <luokmhd@foxmail.com>

Description SpatioLinear (SpLin) is an R package designed to process spatiotemporal omics data, particularly from spatial transcriptomic chips. It accurately linearizes and horizontally maps the physical coordinates of specific curved-shaped, continuous local tissue structures (e.g., neural tubes, intestinal tissues) to the x-axis. By algorithmically transforming these complex tissue architectures into a standardized 1D coordinate system, SpatioLinear preserves topological integrity while enabling more intuitive display and analysis of the spatial distribution features of these tissue structures.

Depends R (>= 4.2.0)

License GPL (>= 2)

Imports sp, sf, spdep, gstat, mgcv, Seurat, jsonlite, ggplot2, data.table, patchwork, reshape2, RColorBrewer, raster, dplyr, WGCNA, concaveman, circlize, stringr, ggpibr, ggsignif, gghalves, rlang, ggsci, ggquiver, ggforce, pheatmap, fuzzyjoin, DNAcopy, dbSCAN, nls2, ggdensity, akima

SystemRequirements: Python (>=3.8), opencv-python (>=4.10.0.84), rasterio (>=1.3.11), numpy (>=1.23.5), matplotlib (>=3.7.1), pillow (>=10.4.0), joblib (>=1.4.2), scipy (>=1.10.1)

Encoding UTF-8

LazyData true

URL <https://github.com/scholarLW/SpLin>

BugReports <https://github.com/scholarLW/SpLin/issues>

Date/Publication 2025-03-12 14:43:10

I. Only the straightening of the signal matrix

1. Generate the dot plot of the signal matrix

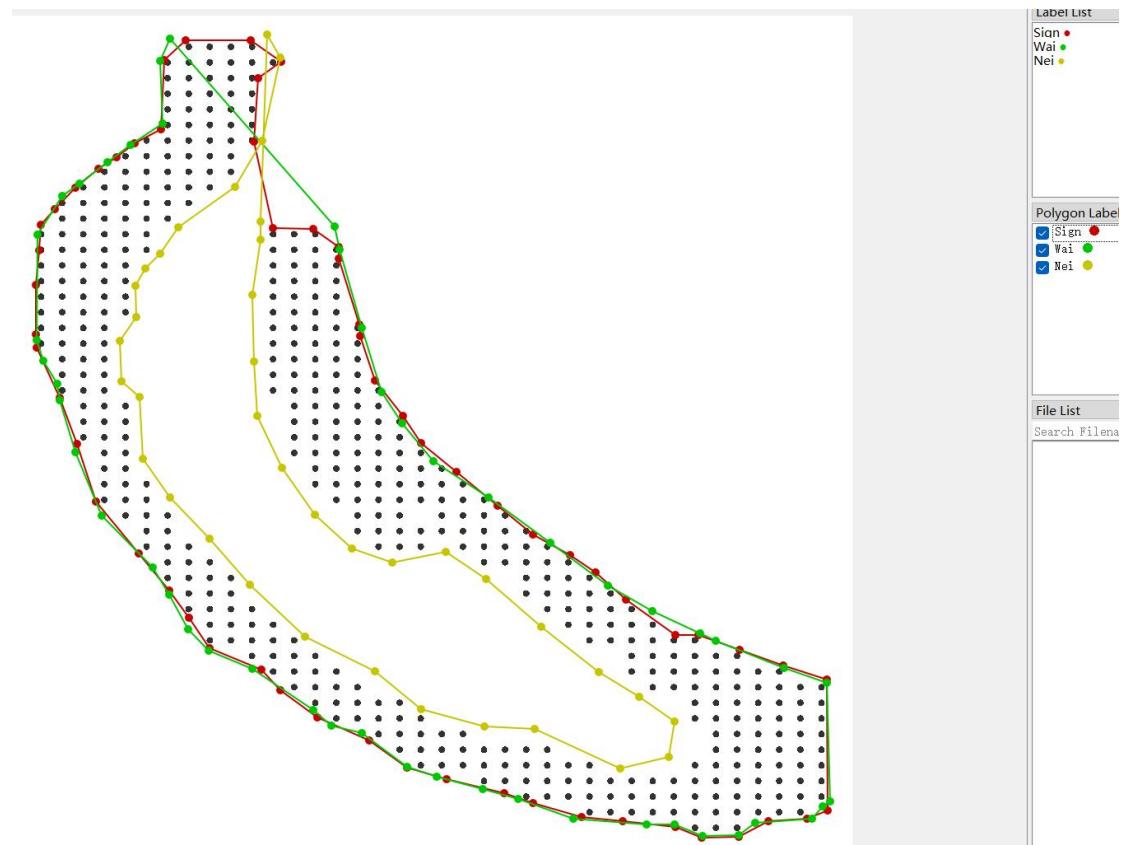


Run the code:

```
rds = 'modified_data.rds'  
coIN = NULL  
library(SpLin)  
signalToFig(rds, coIN = coIN)
```

2. Create Polygons

Perform Labelme (GitHub - wkentaro/labelme) annotation on the dot plot, select regions of interest, use the “Create Polygons” method for labeling, and check the JSON file



1) Signal Matrix Region (for Registration):

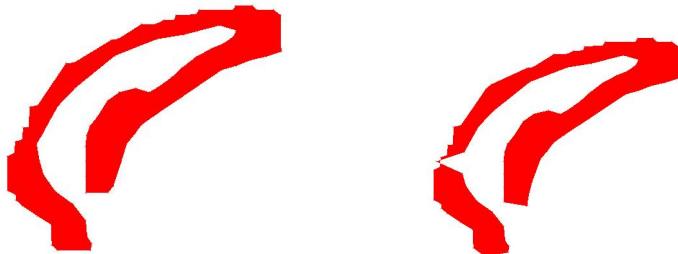
- Fixed label field for the outer contour: **Sign**.
- Requires precise edge tracing, with as many points as possible to ensure the accuracy of the contour.

2) Target Region (for Straightening):

- Fixed label field for the outer side: **Wai**.
 - ✓ Requires precise edge tracing.
- Fixed label field for the inner side: **Nei**.
 - ✓ Requires only that the target region is included.
 - ✓ For adjacent points (excluding the start and end points), the lines connecting Wai and Nei must not intersect.
 - ✓ Once a closed loop is formed, the annotated points should not be dragged or adjusted.

JSON File Verification:

After completing the annotation with Labelme, it is necessary to verify the generated JSON file to ensure that the annotation content meets the requirements.



[Correct Image vs. Image with Dragging/Non-Endpoint Intersections](#)

- **Correct Image:** The image where the annotations are precise, with no intersections between lines of adjacent points (except at endpoints), and the shapes are properly closed without any dragging of points after forming a closed loop.
- **Image with Dragging/Non-Endpoint Intersections:** The image where annotations have been dragged or adjusted after forming a closed loop, or where lines connecting adjacent points intersect (other than at the start and end points).

Run the code:

```
library(SpLin)  
JsonCheck('signalMatrix.json')
```

3. Registration (Alignment of Expression Matrix with Outer Contour Image)

1) Image Information Acquisition

```
total 184  
-rw-r--r-- 1 luow research 3140 Mar 20 16:32 JsonPolygonCheck.png  
-rw-r--r-- 1 luow research 164 Mar 20 16:23 image_dimensions.json  
-rw-r--r-- 1 luow research 105404 Mar 20 16:22 signalMatrix.json  
-rw-r--r-- 1 luow research 70802 Mar 20 14:51 signalMatrix.png
```

Run the code:

```
inputfile = 'SignalMatrix/signalMatrix.png'
```

```
library(SpLin)
reSizeIMG(inputfile)
```

2) Update Coordinate Information

```
-rw-r--r-- 1 luow research 1.5K Mar 20 16:37 markingPoints.txt
-rw-r--r-- 1 luow research 9.0K Mar 20 16:37 markingPoints.pdf
-rw-r--r-- 1 luow research 110K Mar 20 16:37 Adjusted_signalMatrix.json
-rw-r--r-- 1 luow research 151 Mar 20 16:37 image_dimensions.json
-rw-r--r-- 1 luow research 4.9K Mar 20 16:37 cellPoints.txt
-rw-r--r-- 1 luow research 26K Mar 20 16:37 cellPoints.pdf
-rw-r--r-- 1 luow research 3.1K Mar 20 16:32 JsonPolygonCheck.png
-rw-r--r-- 1 luow research 103K Mar 20 16:22 signalMatrix.json
-rw-r--r-- 1 luow research 70K Mar 20 14:51 signalMatrix.png
```



From left to right: Signal Matrix, Peripheral Contour “Sign” of the Signal Matrix.

Run the code:

```
rds = 'modified_data.rds'
image = 'SignalMatrix/signalMatrix.png'
json = 'SignalMatrix/signalMatrix.json'
idjson = 'SignalMatrix/image_dimensions.json'
library(SpLin)
getPolygonPionts(rds, image, json, idjson, Signal = TRUE)
```

3) Generate Mask Image and Select Initial Registration Points

A) Mask Image Generation



From left to right: Signal matrix mask image, and the mask image of the signal matrix outer contour “Sign” after correction.

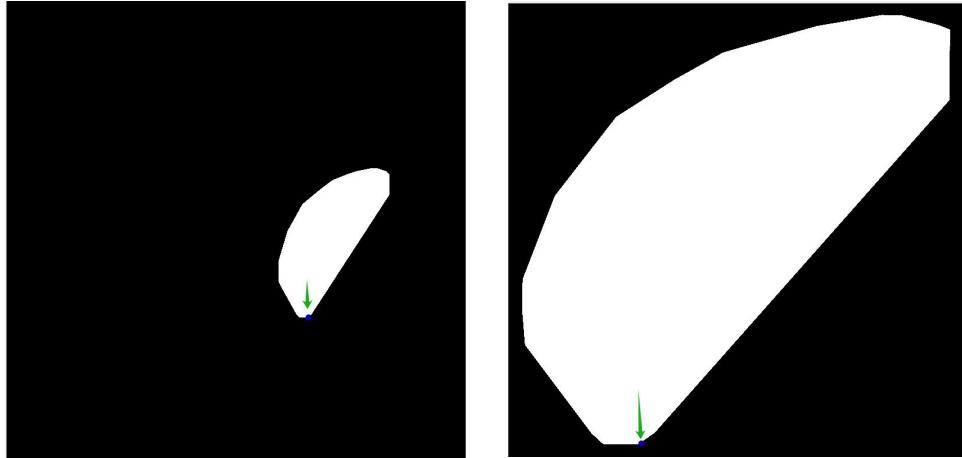
Run the code:

```

json = 'SignalMatrix/Adjusted_signalMatrix.json'
cellPointsFile = 'SignalMatrix/cellPoints.txt'
idjson = 'SignalMatrix/image_dimensions.json'
library(SpLin)
preAutoRegister(json, cellPointsFile, idjson, Signal = TRUE)

```

B) Selection of Registration Reference Points (using Labelme software to select appropriate points)



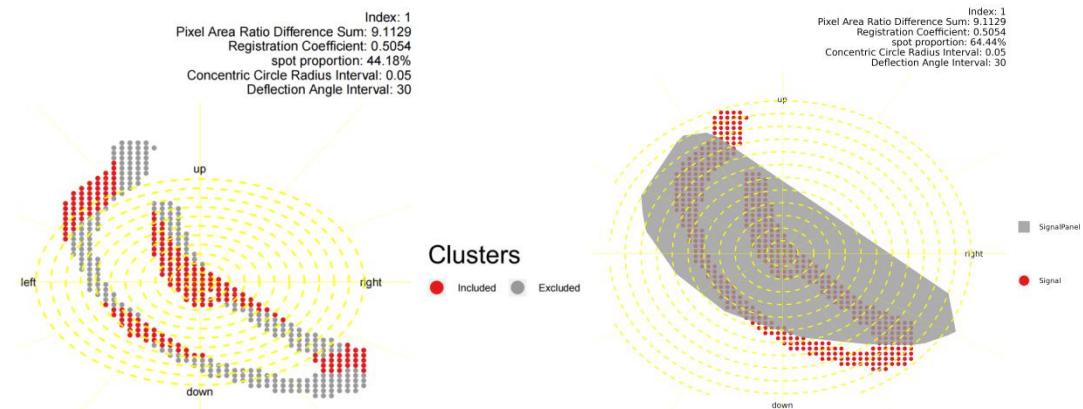
From left to right: Signal matrix mask image, and the mask image of the signal matrix outer contour "Sign" after correction.

Annotation Rules:

- For a single point, use the "Create Point" method.
- For multiple points, use the "Create Polygons" method, and the order of annotation must be consistent and correspond one-to-one.

4) Registration and Fine-tuning

A) Initial Registration:



Run the code:

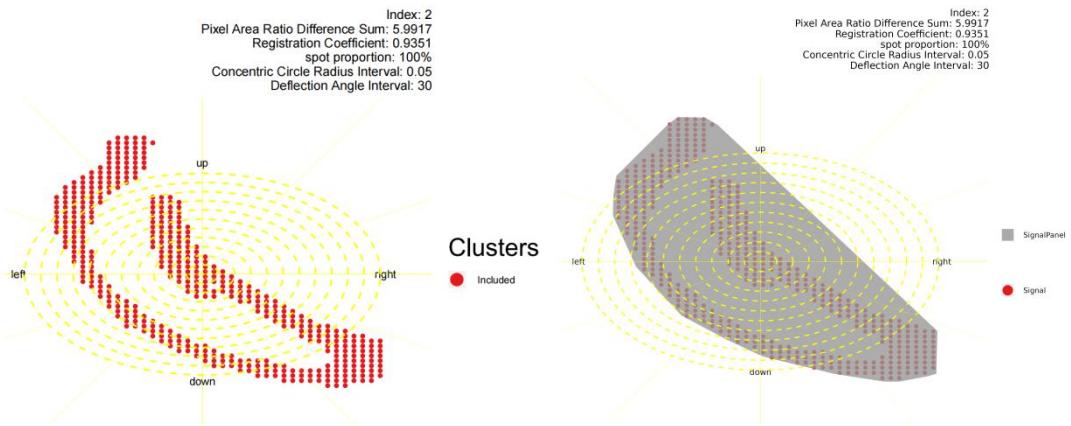
```
json = 'SignalMatrix/Adjusted_signalMatrix.json'
```

```

cellPointsFile = 'SignalMatrix/cellPoints.txt'
markingPointsImage = 'SignalMatrix/marketingPoints_MASK.png'
markingPoints_MASKJSON = 'SignalMatrix/marketingPoints_MASK.json'
cellPointsImage = 'SignalMatrix/cellPoints_MASK.png'
cellPoints_MASKJSON = 'SignalMatrix/cellPoints_MASK.json'
idjson = 'SignalMatrix/image_dimensions.json'
transform = NULL
kpixel = 10
epsilon = 0.001
interval = 0.05
intervalAngle = 30
up = 0
down = 0
left = 0
right = 0
theta = 0
spatialpointsize = 2
library(SpLin)
RapidRegister(json,      cellPointsFile,      markingPointsImage,      markingPoints_MASKJSON,
cellPointsImage, cellPoints_MASKJSON, idjson, transform = transform, kpixel = kpixel, epsilon =
epsilon, interval = interval, intervalAngle = intervalAngle, up = up, down = down, left = left, right =
right, theta = theta, spatialpointsize = spatialpointsize, Signal = TRUE)

```

B) Fine-tuning Registration:



Run the code:

```

json = 'SignalMatrix/Adjusted_signalMatrix.json'
cellPointsFile = 'SignalMatrix/cellPoints.txt'
markingPointsImage = 'SignalMatrix/marketingPoints_MASK.png'
markingPoints_MASKJSON = 'SignalMatrix/marketingPoints_MASK.json'

```

```

cellPointsImage = 'SignalMatrix/cellPoints_MASK.png'
cellPoints_MASKJSON = 'SignalMatrix/cellPoints_MASK.json'
idjson = 'SignalMatrix/image_dimensions.json'
transform = NULL
kpixel = 10
epsilon = 0.001
interval = 0.05
intervalAngle = 30
up = 0.1
down = 0.1
left = -0.15
right = -0.15
theta = -10
spatialpointsize = 2
library(SpLin)
RapidRegister(json,      cellPointsFile,      markingPointsImage,      markingPoints_MASKJSON,
cellPointsImage, cellPoints_MASKJSON, idjson, transform = transform, kpixel = kpixel, epsilon =
epsilon, interval = interval, intervalAngle = intervalAngle, up = up, down = down, left = left, right =
right, theta = theta, spatialpointsize = spatialpointsize, Signal = TRUE)

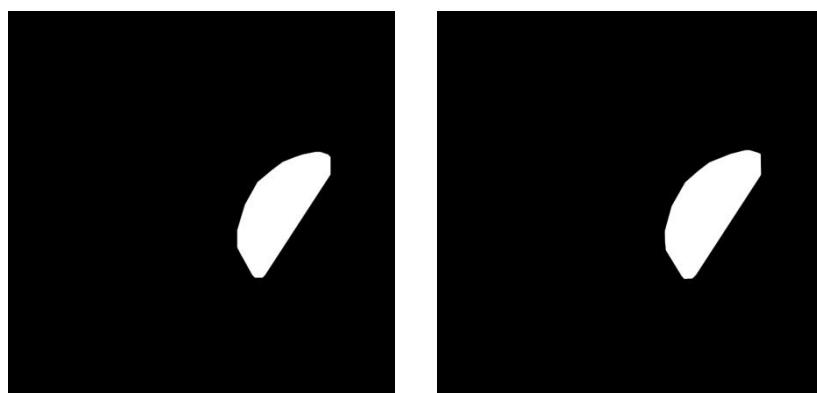
```

C) Extract Registration Information:

```

-rw-r--r-- 1 luow research 5.9M Mar 21 21:06 Update_Manual_modified_data.rds
-rw-r--r-- 1 luow research 273K Mar 21 21:06 markingPoints_MASK_update.png
-rw-r--r-- 1 luow research 1.3M Mar 21 21:06 Adjusted_Output_Manual.json
-rw-r--r-- 1 luow research 64K Mar 21 17:31 Multiple_AutoRegister_Sign_plot.par
-rw-r--r-- 1 luow research 2.5M Mar 21 17:31 AllRegistrationSchemes.json
drwxr-xr-x 4 luow research 36 Mar 21 17:14 Scheme
-rw-r--r-- 1 luow research 338K Mar 21 15:18 cellPoints_MASK.json
-rw-r--r-- 1 luow research 427K Mar 21 15:18 markingPoints_MASK.json
-rw-r--r-- 1 luow research 314K Mar 21 08:56 markingPoints_MASK.png
-rw-r--r-- 1 luow research 273K Mar 21 08:56 cellPoints_MASK.png
-rw-r--r-- 1 luow research 860 Mar 21 08:56 markingPoints.txt
-rw-r--r-- 1 luow research 7.4K Mar 21 08:56 markingPoints.pdf
-rw-r--r-- 1 luow research 1.3M Mar 21 08:56 Adjusted_signalMatrix.json
-rw-r--r-- 1 luow research 153 Mar 21 08:56 image_dimensions.json
-rw-r--r-- 1 luow research 4.9K Mar 21 08:56 cellPoints.txt
-rw-r--r-- 1 luow research 26K Mar 21 08:56 cellPoints.pdf
-rw-r--r-- 1 luow research 3.3K Mar 20 17:27 JsonPolygonCheck.png
-rw-r--r-- 1 luow research 1.2M Mar 20 17:26 signalMatrix.json
-rw-r--r-- 1 luow research 915K Mar 20 17:21 signalMatrix.png

```



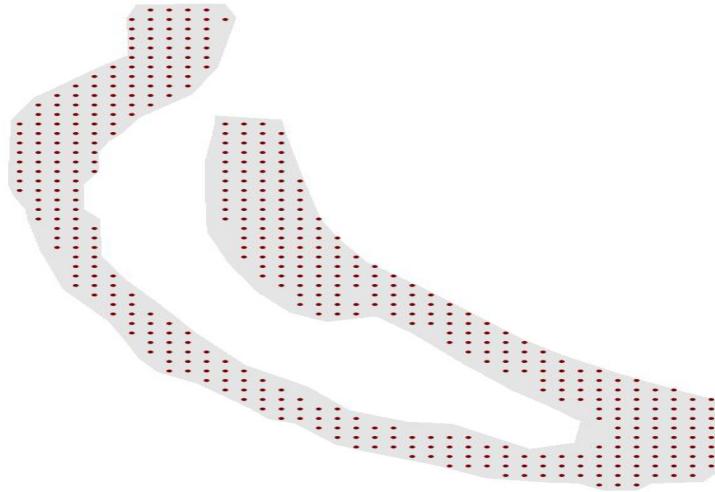
From left to right: Signal matrix mask image, and the mask image of the signal matrix outer contour "Sign" after correction.

Run the code:

```
rds = 'modified_data.rds'  
json = 'SignalMatrix/AllRegistrationSchemes.json'  
idjson = 'SignalMatrix/image_dimensions.json'  
pixelssDNAFile = NULL  
pixelHEFile = NULL  
index = 2  
library(SpLin)  
RapidRegisterOUT(rds, json, idjson, pixelssDNAFile = pixelssDNAFile, pixelHEFile = pixelHEFile,  
index = index)
```

4. Verify whether the regions annotated by Labelme are aligned with the regions of interest

```
-rw-r--r-- 1 luow research 2.7K Mar 22 10:18 signalLabelmeRegion.png  
-rw-r--r-- 1 luow research 5.9M Mar 21 21:06 update_manual_modified_data.rds  
-rw-r--r-- 1 luow research 273K Mar 21 21:06 markingPoints_MASK_update.png  
-rw-r--r-- 1 luow research 1.3M Mar 21 21:06 Adjusted_Output_Manual.json  
-rw-r--r-- 1 luow research 64K Mar 21 17:31 Multiple_AutoRegister_ST_plot.pdf  
-rw-r--r-- 1 luow research 2.5M Mar 21 17:31 AllRegistrationSchemes.json  
drwxr-xr-x 4 luow research 36 Mar 21 17:14 Scheme
```



Run the code:

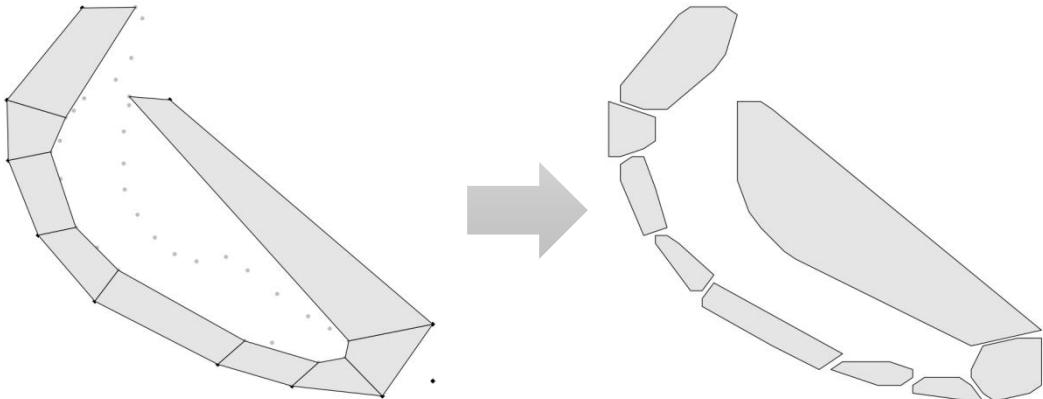
```
rds = 'modified_data.rds'  
json = 'SignalMatrix/Adjusted_Output_Manual.json'  
library(SpLin)  
checkLabelmeRegion(rds, json)
```

5. Straighten the spatial coordinates of the regions of interest

```

total 25M
-rw-r--r-- 1 luow research 5.9M Mar 22 12:31 SpLin_EXP_output.rds
-rw-r--r-- 1 luow research 26K Mar 22 12:31 metaDtaPointsUpdate.txt
-rw-r--r-- 1 luow research 20K Mar 22 12:31 metaDtaPoints.txt
-rw-r--r-- 1 luow research 2.0K Mar 22 12:31 BinIPoints.txt
-rw-r--r-- 1 luow research 4.6K Mar 22 12:31 BinRegionData.pdf
-rw-r--r-- 1 luow research 1.7K Mar 22 12:31 xyToBin.RData
-rw-r--r-- 1 luow research 7.1K Mar 22 12:31 BinRegion.pdf
-rw-r--r-- 1 luow research 42K Mar 22 12:19 signalLabelMeRegion.png
-rw-r--r-- 1 luow research 5.9M Mar 21 21:06 Update_Manual_modified_data.rds
-rw-r--r-- 1 luow research 273K Mar 21 21:06 markingPoints_MASK_update.png
-rw-r--r-- 1 luow research 1.3M Mar 21 21:06 Adjusted_Output_Manual.json
-rw-r--r-- 1 luow research 64K Mar 21 17:31 Multiple_AutoRegister_ST_plot.pdf
-rw-r--r-- 1 luow research 2.5M Mar 21 17:31 AllRegistrationSchemes.json
drwxr-xr-x 4 luow research 36 Mar 21 17:14 Scheme
-rw-r--r-- 1 luow research 338K Mar 21 15:18 cellPoints_MASK.json
-rw-r--r-- 1 luow research 427K Mar 21 15:18 markingPoints_MASK.json
-rw-r--r-- 1 luow research 314K Mar 21 08:56 markingPoints_MASK.png
-rw-r--r-- 1 luow research 273K Mar 21 08:56 cellPoints_MASK.png
-rw-r--r-- 1 luow research 860 Mar 21 08:56 markingPoints.txt
-rw-r--r-- 1 luow research 7.4K Mar 21 08:56 markingPoints.pdf
-rw-r--r-- 1 luow research 1.3M Mar 21 08:56 Adjusted_signalMatrix.json
-rw-r--r-- 1 luow research 153 Mar 21 08:56 image_dimensions.json
-rw-r--r-- 1 luow research 4.9K Mar 21 08:56 cellPoints.txt
-rw-r--r-- 1 luow research 26K Mar 21 08:56 cellPoints.pdf
-rw-r--r-- 1 luow research 3.3K Mar 20 17:27 JsonPolygonCheck.png
-rw-r--r-- 1 luow research 1.2M Mar 20 17:26 signalMatrix.json
-rw-r--r-- 1 luow research 915K Mar 20 17:21 signalMatrix.png

```



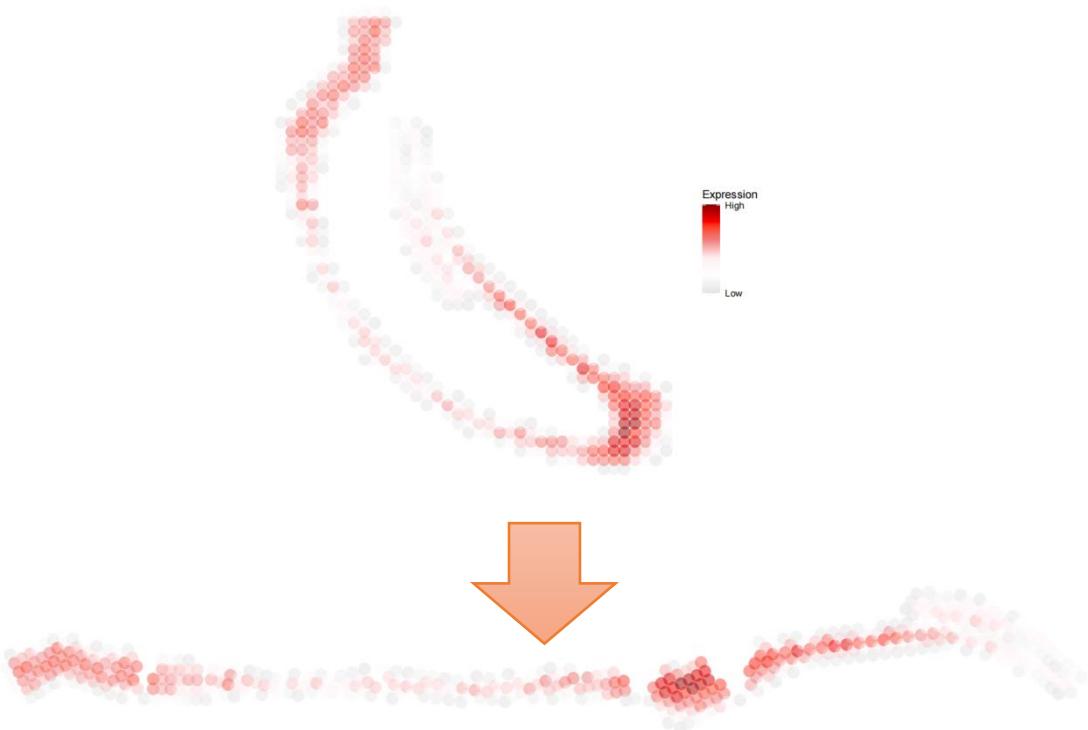
Run the code:

```

rds = 'SignalMatrix/Update_Manual_modified_data.rds'
json = 'SignalMatrix/Adjusted_Output_Manual.json'
nlim = 0
drop = TRUE
triangleMerge = TRUE
triangle.probs = 0.2
library(SpLin)
coordLinearEXP(rds, json, nlim = nlim, drop = drop, triangleMerge = triangleMerge,
triangle.probs = triangle.probs)

```

Visualization Results:

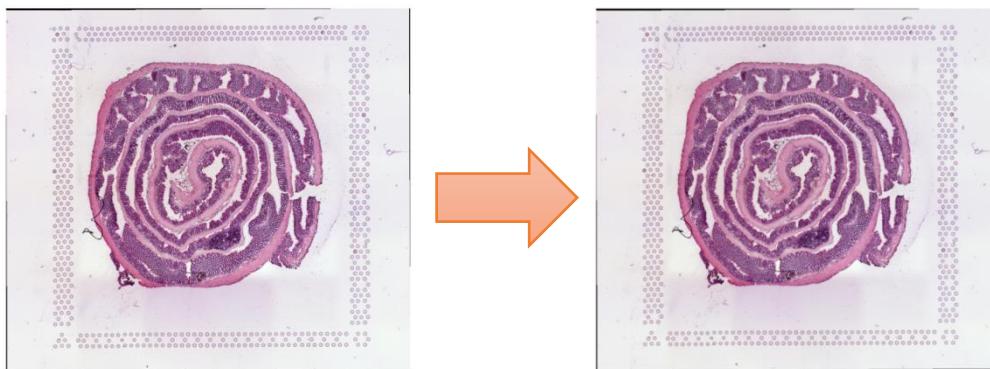


Run the code:

```
rds = 'SignalMatrix/Splin_EXP_output.rds'
library(Splin)
markPlot(rds, markerGenes = c('Test020564', 'Test009309', 'Test003615'),
         SpatialFeaturePlotpointsize = 5)
```

II. Integrated Straightening of Signal Matrix and Stained Images (H&E/ssDNA)

1. Rescaling of Chip Staining Images (Example for 10X Genomics)



From left to right: tif format image, png image with reduced resolution

Note: Images scanned by a microscope are usually in tif/tiff format. Due to their large size and multiple layers, these images cannot be directly annotated using Labelme. Therefore, format conversion is necessary, typically to the png file format.

Run the code:

```

inputfile = 'GSM5213483_V19S23-097_A1_S1_Region1_colon_d0_rotated.tif'
outputfile = 'HE/GSM5213483_V19S23-097_A1_S1.png'
dpi = 1500
imageType = 'HE'
library(SpLin)
reSizeIMG(inputfile,outputfile,dpi=dpi,it=imageType)

```

2. Annotate the staining image using Labelme

For the inner and outer sides of the tissue, use the "Create Polygons" method for annotation and check the JSON file.



Target Area (for straightening):

- Outer label fixed field: Wai, which requires precise edge tracing.
- Inner label fixed field: Nei, which should fully cover the target area. For adjacent points (excluding the start and end points), the lines connecting Wai and Nei must not intersect. After forming a closed loop, the annotated points should not be dragged.

Verification of JSON Files:



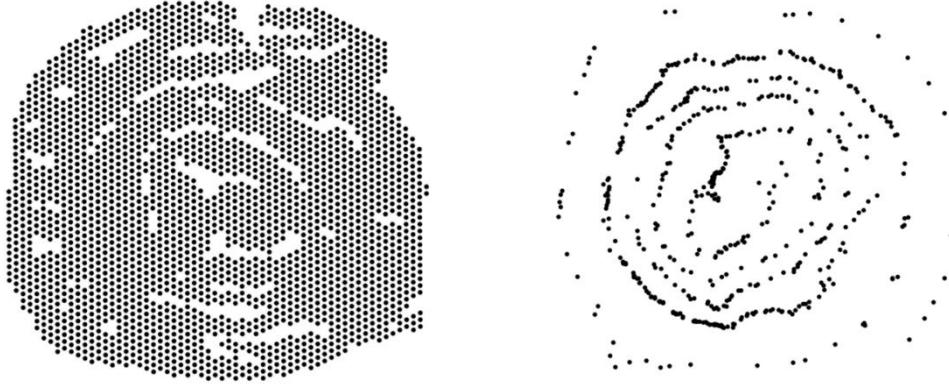
The correct image

Run the code:

```
library(SpLin)  
JsonCheck('GSM5213483_V19S23-097_A1_S1.json')
```

3. Registration (Alignment of Expression Matrix with Staining Images)

1) Update coordinate information



From left to right: Signal Matrix, Chip Tissue Region "Wai" & "Nei"

Run the code:

```
rds = 'GSM5213483_V19S23-097_A1_S1-NEW.rds'  
image = 'HE/GSM5213483_V19S23-097_A1_S1.png'  
json = 'HE/GSM5213483_V19S23-097_A1_S1.json'  
idjson = 'HE/image_dimensions.json'  
library(SpLin)  
getPolygonPionts(rds, image, json, idjson)
```

2) Generate mask images and select initial registration points

A) Mask Image Generation



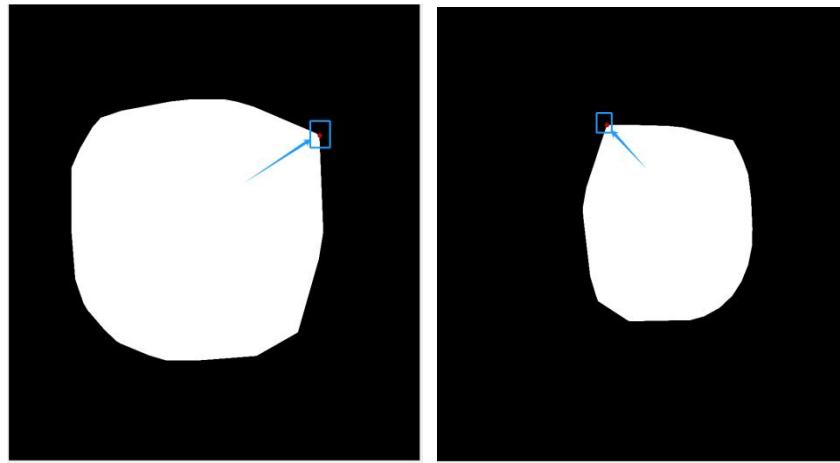
From left to right: Signal Matrix Mask Image, and the Signal Matrix Region of Interest "Wai" & "Nei" Mask Image

Run the code:

```
json = 'HE/Adjusted_GSM5213483_V19S23-097_A1_S1.json'  
cellPointsFile = 'HE/cellPoints.txt'  
idjson = 'HE/image_dimensions.json'
```

```
library(SpLin)
preAutoRegister(json, cellPointsFile, idjson)
```

B) Selection of Registration Reference Points (using Labelme software to select appropriate points)



From left to right: Signal Matrix Mask Image, and the Signal Matrix Region of Interest "Wai" & "Nei" Mask Image

Annotation Rules:

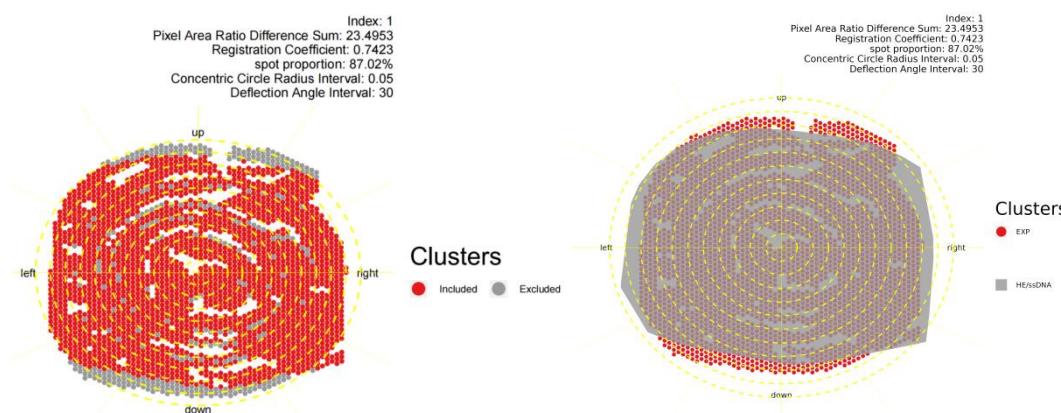
- For a single point, use the "Create Point" method.
- For multiple points, use the "Create Polygons" method, and the order of annotation must be consistent and correspond one-to-one.

3) Registration and Fine-Tuning

A) Initial Registration:

```
-rw-r--r-- 1 luow research 316K Feb 12 10:28 Multiple_AutoRegister_ST_plot.pdf
-rw-r--r-- 1 luow research 159M Feb 12 10:26 AllRegistrationSchemes.json
drwxr-xr-x 4 luow research 36 Feb 12 09:43 Scheme

$ ll Scheme/1/
total 1360
-rw-r--r-- 1 luow research 1390467 Feb 12 10:28 Multiple_AutoRegister_ST_plot.png
```



Run the code:

```
json = 'HE/Adjusted_GSM5213483_V19S23-097_A1_S1.json'
cellPointsFile = 'HE/cellPoints.txt'
```

```

markingPointsImage = 'HE/marketingPoints_MASK.png'
markingPoints_MASKJSON = 'HE/marketingPoints_MASK.json'
cellPointsImage = 'HE/cellPoints_MASK.png'
cellPoints_MASKJSON = 'HE/cellPoints_MASK.json'
idjson = 'HE/image_dimensions.json'
transform = NULL
kpixel = 10
epsilon = 0.001
interval = 0.05
intervalAngle = 30
up = 0
down = 0
left = 0
right = 0
theta = 0
spatialpointsize = 2
library(SpLin)
RapidRegister(json,      cellPointsFile,      markingPointsImage,      markingPoints_MASKJSON,
cellPointsImage, cellPoints_MASKJSON, idjson, transform = transform, kpixel = kpixel, epsilon =
epsilon, interval = interval, intervalAngle = intervalAngle, up = up, down = down, left = left, right =
right, theta = theta, spatialpointsize = spatialpointsize)

```

B) Fine-Tuning Registration:

```

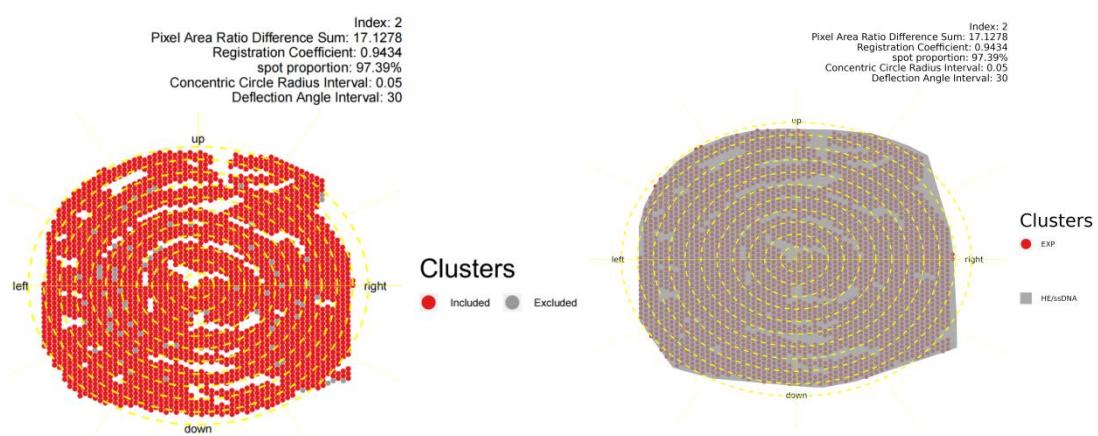
-rw-r--r-- 1 luow research 316K Feb 12 10:28 Multiple_AutoRegister_ST_plot.pdf
-rw-r--r-- 1 luow research 159M Feb 12 10:26 AllRegistrationSchemes.json
drwxr-xr-x 4 luow research   36 Feb 12 09:43 Scheme

```

```

$ ll Scheme/2/
total 1312
-rw-r--r-- 1 luow research 1342568 Feb 12 10:28 Multiple_AutoRegister_ST_plot.png

```



Run the code:

```

json = 'HE/Adjusted_GSM5213483_V19S23-097_A1_S1.json'
cellPointsFile = 'HE/cellPoints.txt'
markingPointsImage = 'HE/marketingPoints_MASK.png'
markingPoints_MASKJSON = 'HE/marketingPoints_MASK.json'
cellPointsImage = 'HE/cellPoints_MASK.png'
cellPoints_MASKJSON = 'HE/cellPoints_MASK.json'
idjson = 'HE/image_dimensions.json'
transform = NULL
kpixel = 10
epsilon = 0.001
interval = 0.05
intervalAngle = 30
up = 0.116
down = 0.116
left = -0.085
right = -0.07
theta = 6.5
spatialpointsize = 2
library(SpLin)
RapidRegister(json,      cellPointsFile,      markingPointsImage,      markingPoints_MASKJSON,
cellPointsImage, cellPoints_MASKJSON, idjson, transform = transform, kpixel = kpixel, epsilon =
epsilon, interval = interval, intervalAngle = intervalAngle, up = up, down = down, left = left, right =
right, theta = theta, spatialpointsize = spatialpointsize)

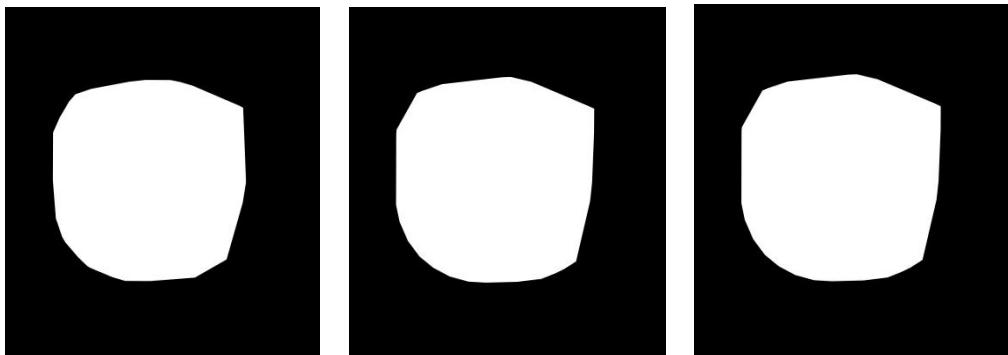
```

C) Extract the registration information:

```

-rw-r--r-- 1 luow research 45K Feb 12 11:53 pixel_markingPointsMatch.png
-rw-r--r-- 1 luow research 51M Feb 12 11:50 Update_Manual_GSM5213483_V19S23-097_A1_S1-NEW.rds
-rw-r--r-- 1 luow research 302M Feb 12 11:48 pixel_markingPoints_XY.txt
-rw-r--r-- 1 luow research 126K Feb 12 11:48 HE_pixel_MASK_pixel_Update.png
-rw-r--r-- 1 luow research 126K Feb 12 11:31 HE_pixel_MASK_pixel.png
-rw-r--r-- 1 luow research 126K Feb 12 11:30 marketingPoints_MASK_update.png
-rw-r--r-- 1 luow research 80M Feb 12 11:30 Adjusted_Output_Manual.json

```

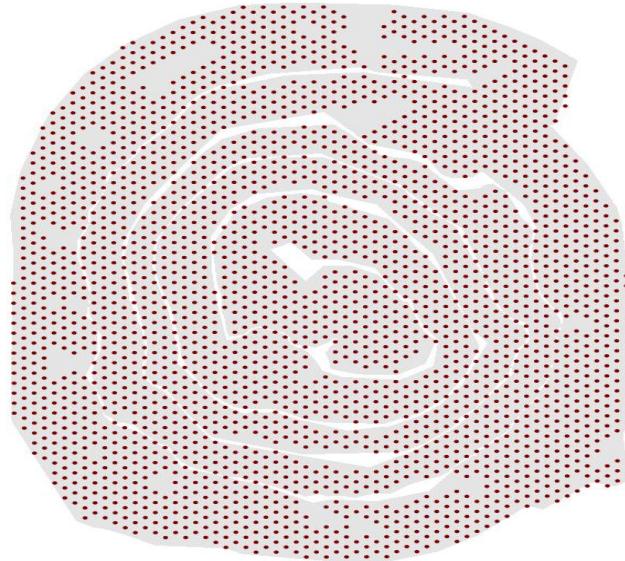


From left to right: Signal matrix mask image, staining base image mask after correction of "Wai" & "Nei," and staining base image pixel region mask.

Run the code:

```
rds = 'GSM5213483_V19S23-097_A1_S1-NEW.rds'  
json = 'HE/AllRegistrationSchemes.json'  
idjson = 'HE/image_dimensions.json'  
pixelssDNAFile = NULL  
pixelHEFile = 'HE/Adjusted_output_coordinates_and_colors.txt'  
index = 2  
library(SpLin)  
RapidRegisterOUT(rds, json, idjson, pixelssDNAFile = pixelssDNAFile, pixelHEFile = pixelHEFile,  
index = index)
```

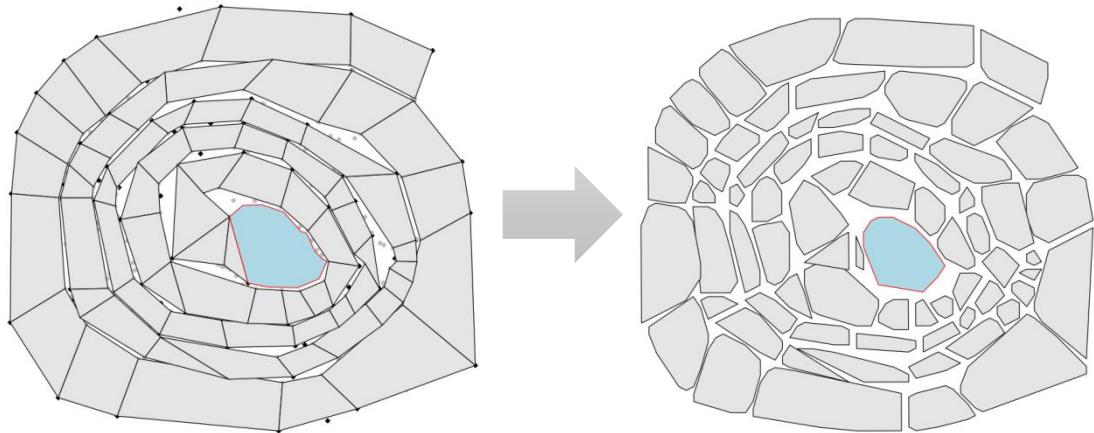
4. Check whether the regions annotated by Labelme are registered with the tissue regions



Run the code:

```
rds = 'GSM5213483_V19S23-097_A1_S1-NEW.rds'  
json = 'HE/Adjusted_Output_Manual.json'  
library(SpLin)  
checkLabelmeRegion(rds, json)
```

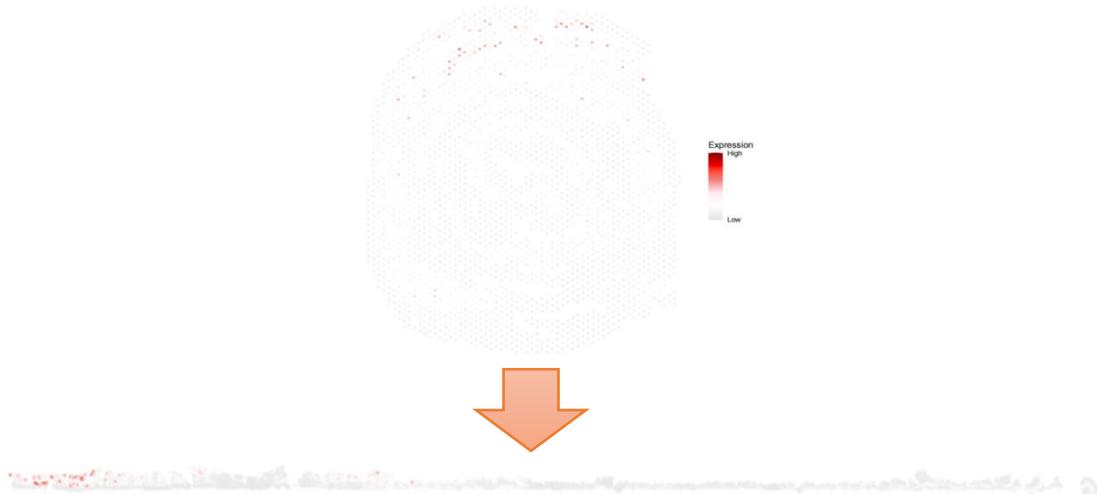
5. Straightening the Spatial Coordinates of Tissue Regions



Run the code:

```
rds = 'HE/Update_Manual_GSM5213483_V19S23-097_A1_S1-NEW.rds'
json = 'HE/Adjusted_Output_Manual.json'
nlim = 0
drop = TRUE
triangleMerge = TRUE
triangle.probs = 0.2
library(SpLin)
coordLinearEXP(rds, json, nlim = nlim, drop = drop, triangleMerge = triangleMerge,
triangle.probs = triangle.probs)
```

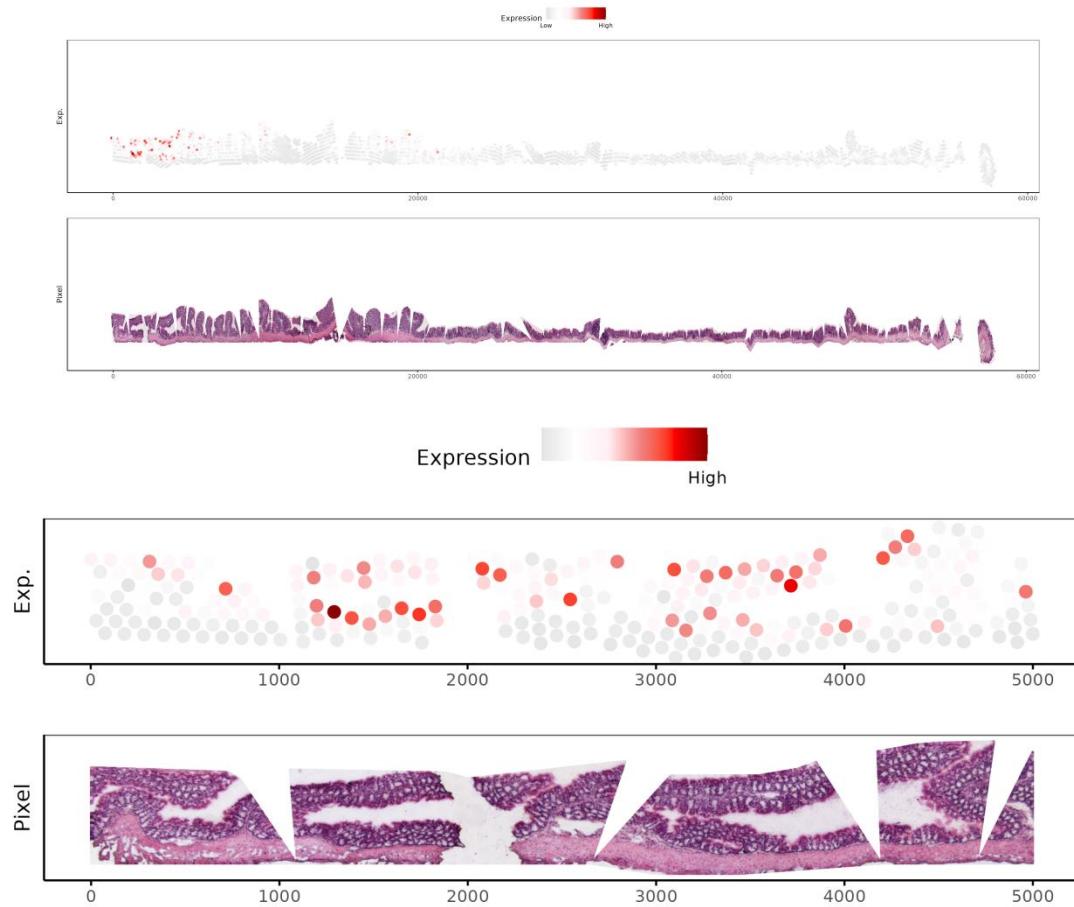
Visualization Results:



Run the code:

```
rds = 'HE/SpLin_EXP_output.rds'
library(SpLin)
markPlot(rds, markerGenes = c('Igkc', 'Muc2', 'Lypd8', 'Igha', 'Actb', 'Zg16', 'Saa1', 'S100a6'),
SpatialFeaturePlotpointsize = 1.5)
```

6. Visualization of Straightened Staining Images



Run the code:

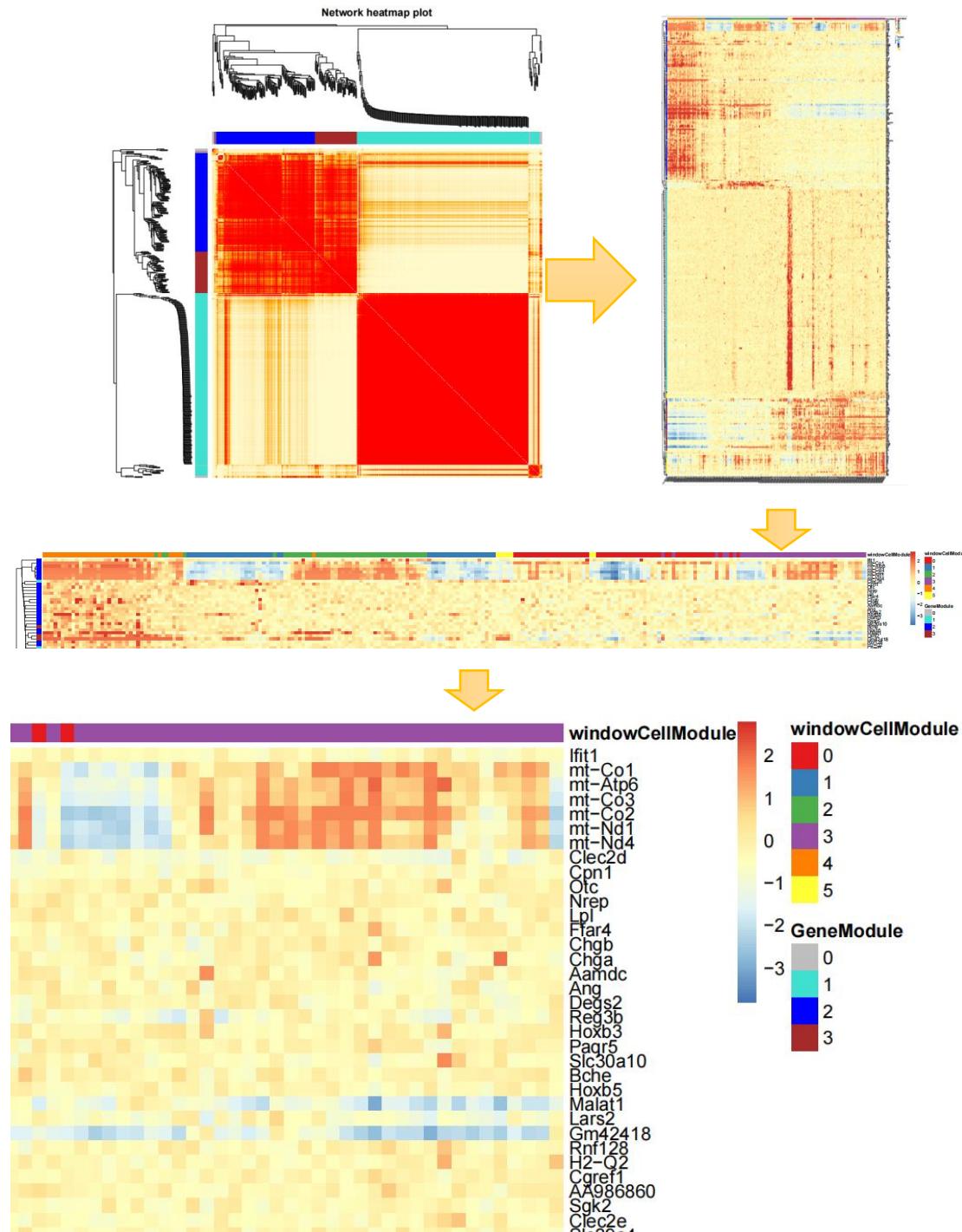
```
rds = 'HE/SpLin_EXP_output.rds'  
pixelPointsFile = 'HE/pixelPointsUpdate.txt'  
markerGenes = c('Muc2')  
xmin = NULL  
xmax = NULL  
SpatialFeaturePlotstriptextsize = 20  
SpatialFeaturePlotpointsize = 1.5  
SpatialFeaturePlotpointalpha = 1  
library(SpLin)  
pixelEXPPlot(rds, pixelPointsFile, markerGenes = markerGenes, xmin = xmin, xmax = xmax,  
SpatialFeaturePlotstriptextsize = SpatialFeaturePlotstriptextsize, SpatialFeaturePlotpointsize =  
SpatialFeaturePlotpointsize, SpatialFeaturePlotpointalpha = SpatialFeaturePlotpointalpha)  
SpatialFeaturePlotstriptextsize = 20  
SpatialFeaturePlotpointsize = 3  
SpatialFeaturePlotpointalpha = 1  
xmin = 0
```

```

xmax = 5000
pixelEXPPlot(rds, pixelPointsFile, markerGenes = markerGenes, xmin = xmin, xmax = xmax,
SpatialFeaturePlotstriptextsize = SpatialFeaturePlotstriptextsize, SpatialFeaturePlotpointsizes =
SpatialFeaturePlotpointsizes, SpatialFeaturePlotpointalpha = SpatialFeaturePlotpointalpha)

```

7. Identifying Regions with Characteristic Gene Expression Patterns



Run the code:

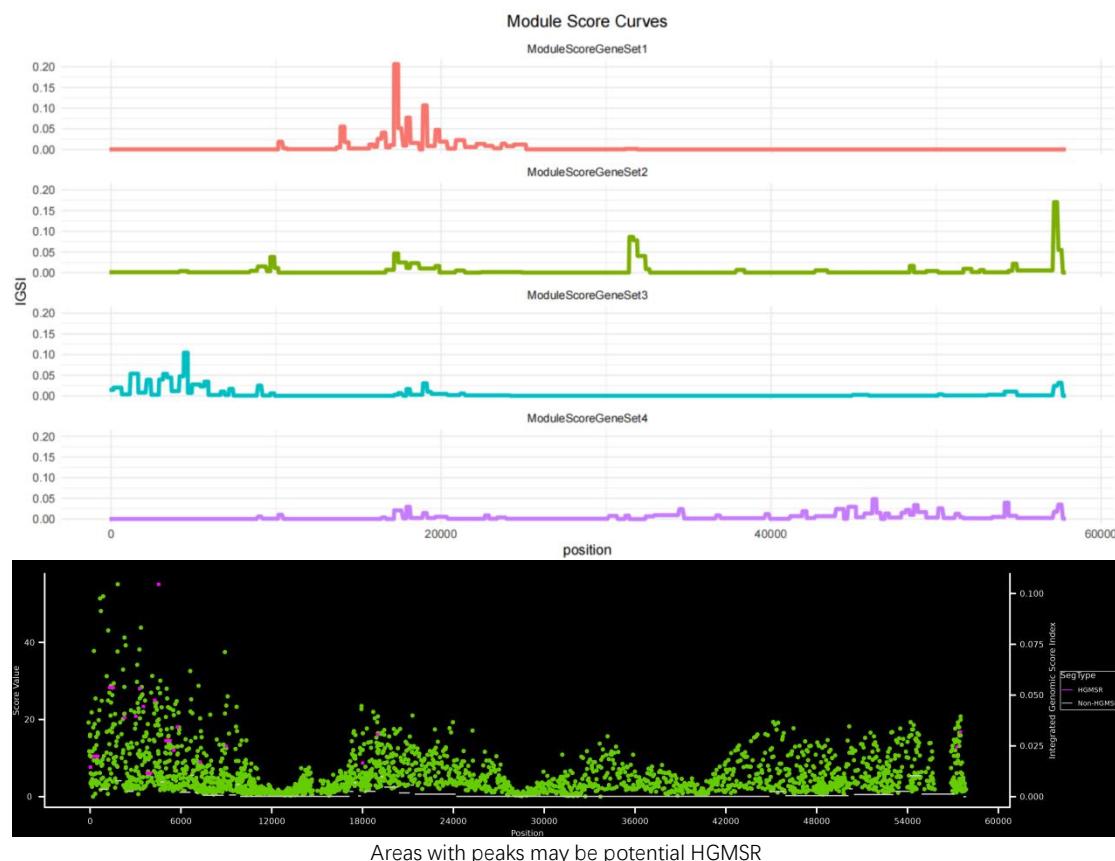
```
rds = 'HE/Spln_EXP_output.rds'
```

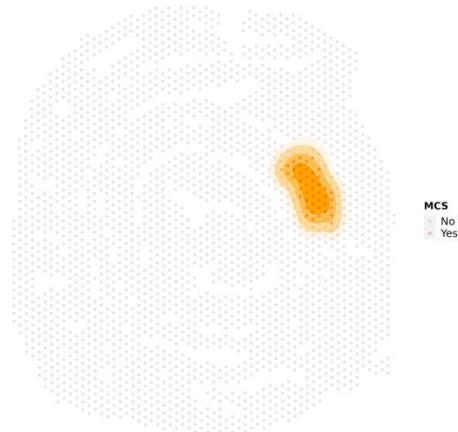
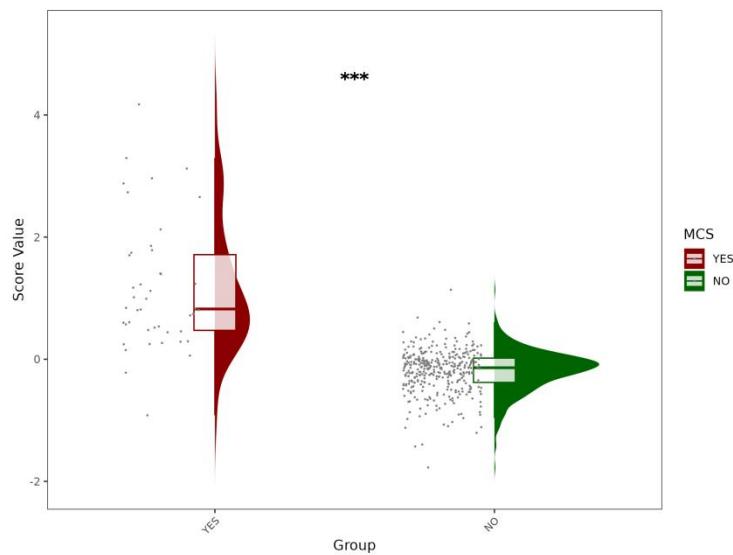
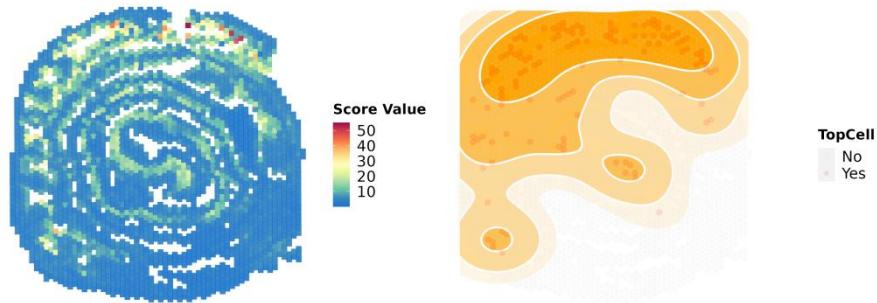
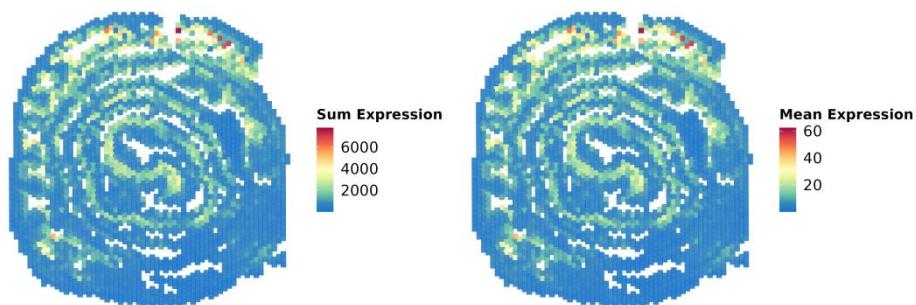
```

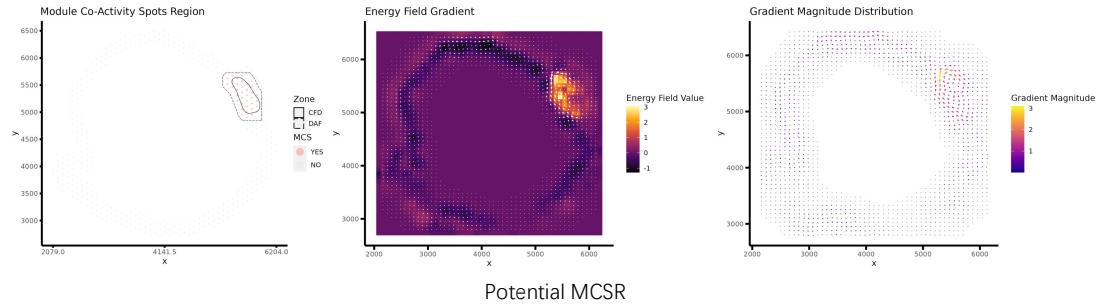
windowSV = 250
library(SpLin)
wcellDue(rds, windowSV = windowSV)
rds = 'HE/IGSI/winCell/windowCell.RDS'
dims <- 10
resolution <- 0.8
strict <- TRUE
show_rownames = TRUE
show_colnames = TRUE
angle_col = "45"
width = 28
height = 42
moduleGene(rds, dims = dims, resolution = resolution, strict = strict, show_rownames =
show_rownames, show_colnames = show_colnames, angle_col = angle_col, width = width, height =
height)

```

8. Identifying High Gene Module Score Region (HGMSR) and Module Co-Activity Spots (MCS)





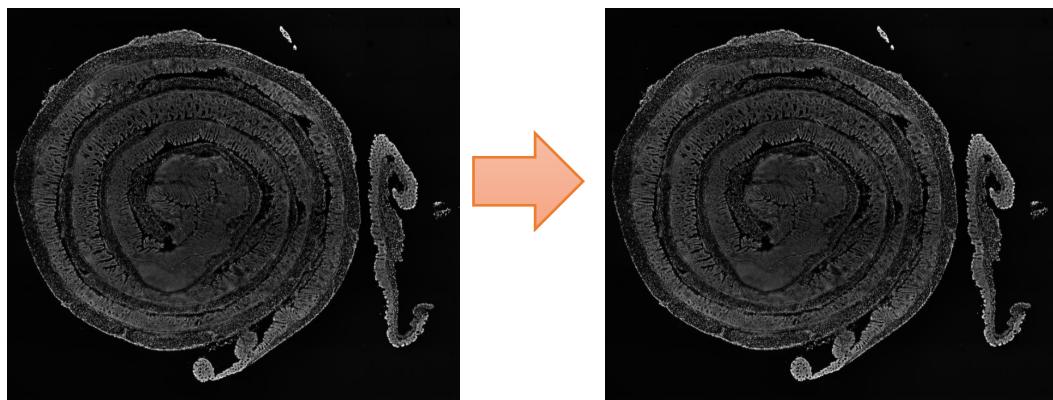


Run the code:

```
rds = 'HE/IGSI/winCell/SpLin_EXP_output_withWindow.RDS'
GMF = 'HE/IGSI/winCell/Modules/Gene/moduleLabel.txt'
library(SpLin)
IGSI(rds, GMF, topSegFreq = 0.4, piontsize = 2.5)
```

III. Straightening Signal Matrix, ssDNA Image, and Adjacent H&E Sections Together

1. Staining Image Rescaling (Example for Stereo-Seq)



From left to right: tif format image, png image with reduced resolution



From left to right: tif format image, png image with reduced resolution

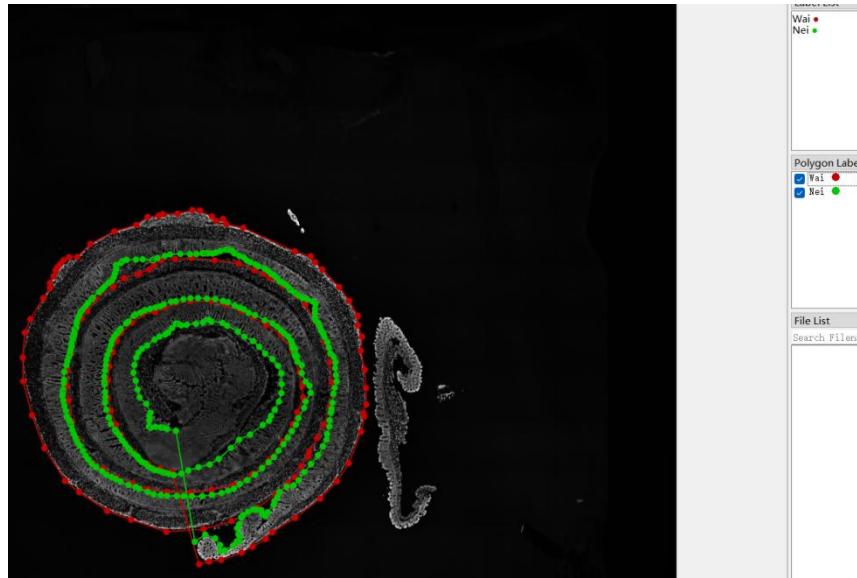
Note: Images scanned by a microscope are usually in tif/tiff format. Due to their large size and multiple layers, these images cannot be directly annotated using Labelme. Therefore, format conversion is necessary, typically to the png file format.

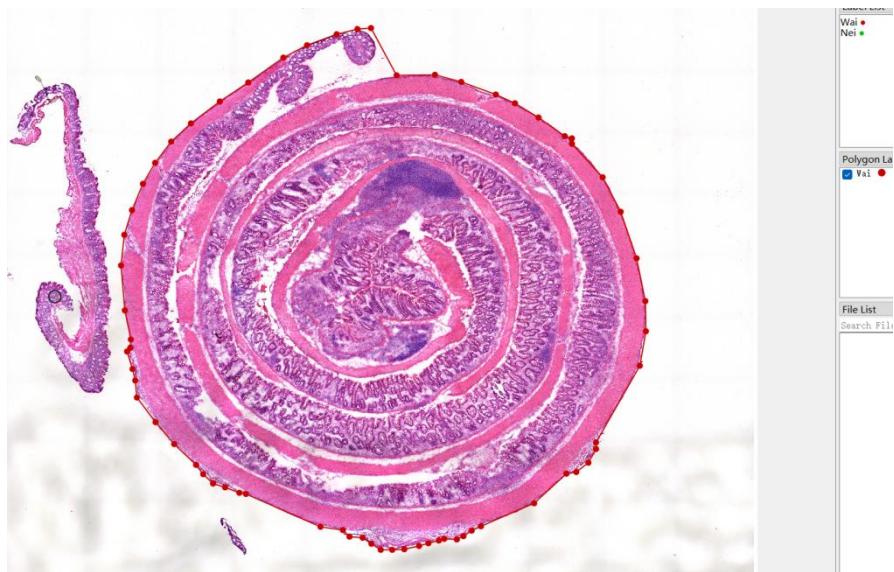
Run the code:

```
library(SpLin)  
  
inputfile = 'DSS12_ssDNA.tif'  
  
outputfile = 'ssDNA/DSS12_ssDNA.png'  
  
dpi = 300  
  
imageType = 'ssDNA'  
  
reSizeIMG(inputfile,outputfile,dpi=dpi,it=imageType)  
  
  
inputfile = 'DSS12_HE.tif'  
  
outputfile = 'HE/DSS12_HE.png'  
  
dpi = 2000  
  
imageType = 'HE'  
  
reSizeIMG(inputfile,outputfile,dpi=dpi,it=imageType)
```

2. Perform Labelme annotation on the staining image

For the inner and outer sides of the tissue, use the "Create Polygons" method for annotation and check the JSON file.

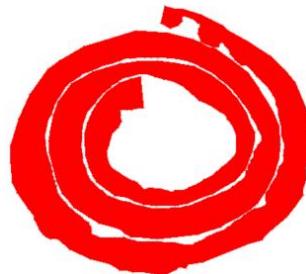




Target Area (for straightening):

- Outer label fixed field: Wai, which requires precise edge tracing.
- Inner label fixed field: Nei, which should fully cover the target area. For adjacent points (excluding the start and end points), the lines connecting Wai and Nei must not intersect. After forming a closed loop, the annotated points should not be dragged.
- For the HE-stained image of the adjacent section: Only the outer edge needs to be precisely traced.

Verification of the JSON file for the ssDNA staining image:



The correct image

Run the code:

```
library(SpLin)
JsonCheck('ssDNA/DSS12_ssDNA.json')
```

3. Registration (Alignment of Expression Matrix with Staining Images)

1) Update coordinate information



From left to right: Signal Matrix, Chip Tissue Region "Wai" & "Nei", and Adjacent Section Tissue Region "Wai" on the Chip

Run the code:

```
library(SpLin)  
rds = 'SpLin_input.rds'  
image = 'ssDNA/DSS12_ssDNA.png'  
json = 'ssDNA/DSS12_ssDNA.json'  
idjson = 'ssDNA/image_dimensions.json'  
getPolygonPionts(rds, image, json, idjson)  
  
image = 'HE/DSS12_HE.png'  
json = 'HE/DSS12_HE.json'  
idjson = 'HE/image_dimensions.json'  
getPolygonPionts(rds, image, json, idjson)
```

2) Registration between the H&E-stained histological image of the adjacent section on the chip and the ssDNA staining image on the chip

A) Mask Image Generation (Results Directory: HE/)

```
-rw-r--r-- 1 luow research 1.2M Apr  8 16:33 HE_pixel_MASK.png  
-rw-r--r-- 1 luow research 1.2M Apr  8 16:28 ssDNA_pixel_MASK.png
```

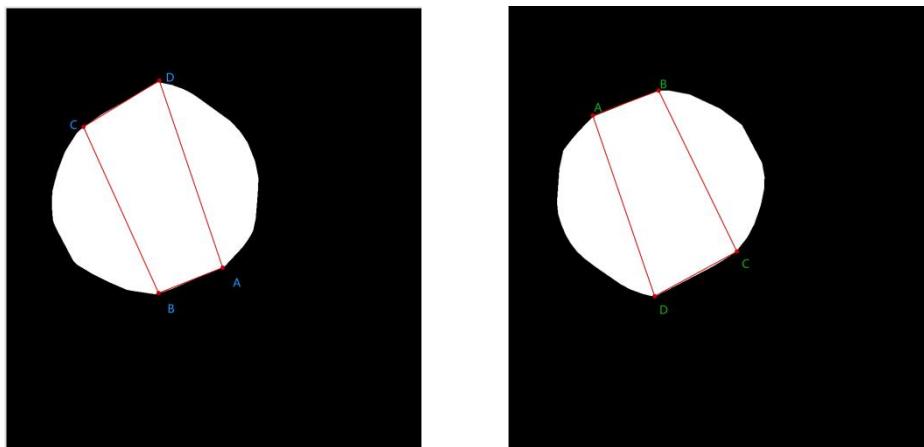


From left to right: Mask image of the ssDNA staining image, and the Mask image of the H&E-stained histological image of the adjacent section

Run the code:

```
markingPointsFile = 'HE/Adjusted_output_coordinates_and_colors.txt'  
cellPointsFile = 'ssDNA/Adjusted_output_coordinates_and_colors.txt'  
idjson = 'ssDNA/image_dimensions.json'  
library(SpLin)  
preAutoRegisterHE(marketingPointsFile, cellPointsFile, idjson)
```

B) Selection of Registration Reference Points (using Labelme software to select appropriate points)



From left to right: Mask image of the ssDNA staining image, and the Mask image of the H&E-stained histological image of the adjacent section

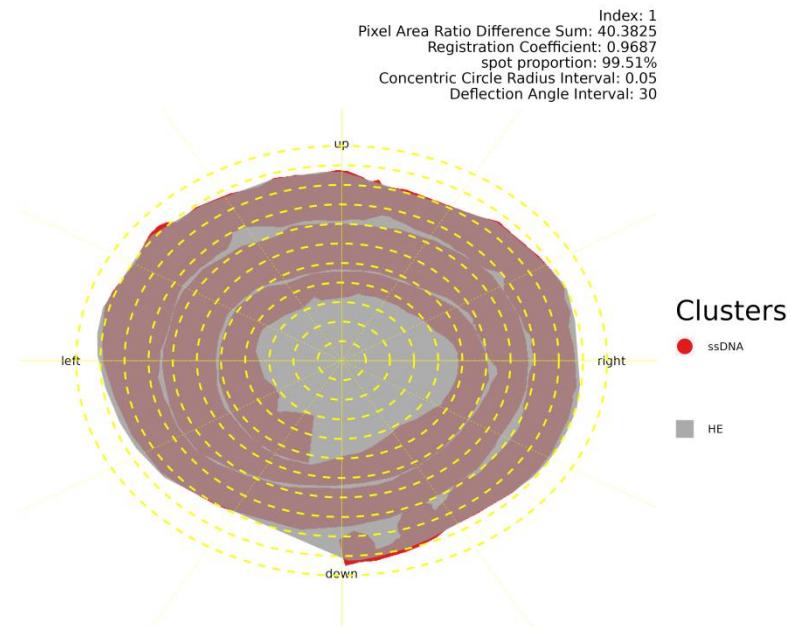
Annotation Rules:

- For a single point, use the "Create Point" method.
- For multiple points, use the "Create Polygons" method, and the order of annotation must be consistent and correspond one-to-one.

C) Registration and Fine-Tuning

① Initial Registration:

```
-rw-r--r-- 1 luow research 2.5M Apr  8 17:45 AllRegistrationSchemes.json
drwxr-xr-x 3 luow research    23 Apr  8 17:12 Scheme
```



```

$ ll -th Scheme
total 0
drwxr-xr-x 2 luow research 194 Apr  8 17:47 1
luow@fat01 09:28:20 ~ /HE
$ ll -th Scheme/1/
total 834M
-rw-r--r-- 1 luow research 577K Apr  8 17:47 Multiple_AutoRegister_ST_plot.png
-rw-r--r-- 1 luow research 831M Apr  8 17:45 ssDNAHE_Adjusted_output_coordinates_and_colors.txt
-rw-r--r-- 1 luow research 1.2M Apr  8 17:44 HE_pixel_MASK_pixel_Update.png
-rw-r--r-- 1 luow research 1.2M Apr  8 17:16 HE_pixel_MASK_pixel.png
luow@fat01 09:28:25 ~ /HE

```

The file indicated by the blue arrow can be used for subsequent operations

Run the code:

```

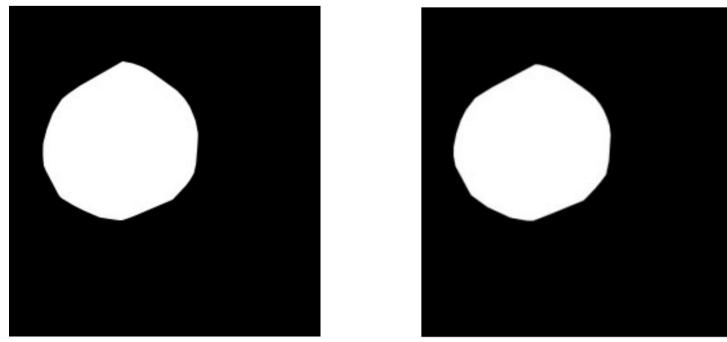
pixelssDNAFile = 'ssDNA/Adjusted_output_coordinates_and_colors.txt'
pixelHEFile = 'HE/Adjusted_output_coordinates_and_colors.txt'
markingPointsImage = 'HE/HE_pixel_MASK.png'
markingPoints_MASKJSON = 'HE/HE_pixel_MASK.json'
cellPointsImage = 'HE/ssDNA_pixel_MASK.png'
cellPoints_MASKJSON = 'HE/ssDNA_pixel_MASK.json'
idjson = 'ssDNA/image_dimensions.json'
transform = NULL
kpixel = 10
epsilon = 0.001
interval = 0.05
intervalAngle = 30
up = 0
down = 0
left = 0
right = 0
theta = 0
library(SpLin)
RapidRegisterHE(pixelssDNAFile, pixelHEFile, markingPointsImage, markingPoints_MASKJSON,
cellPointsImage, cellPoints_MASKJSON, idjson, transform = transform, kpixel = kpixel, epsilon =
epsilon, interval = interval, intervalAngle = intervalAngle, up = up, down = down, left = left, right =
right, theta = theta)

```

② Fine-Tuning Registration:

Note: If the initial registration meets the expected requirements, you can decide whether to proceed with fine-tuning the registration based on your needs. In this case, since the initial registration already satisfies the requirements, no fine-tuning is performed here. If fine-tuning had been performed, the result file used for subsequent analysis would be: HE/Scheme/2/ssDNAHE_Adjusted_output_coordinates_and_colors.txt.

③ Extract the registration information:

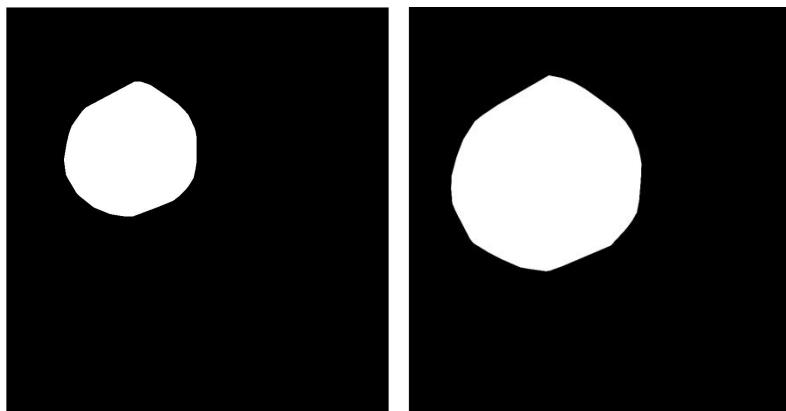


From left to right: the mask image of the ssDNA staining image, and the corrected mask image of the H&E-stained histological image of the adjacent section on the chip.

3) Registration between the Chip Signal Matrix and the Chip ssDNA Staining Image

A) Mask Image Generation (Results Directory: ssDNA/)

```
total 4.3G
-rw-r--r-- 1 luow research 1.2M Apr 10 09:53 markingPoints_MASK.png
-rw-r--r-- 1 luow research 1.1M Apr 10 09:53 cellPoints_MASK.png
```

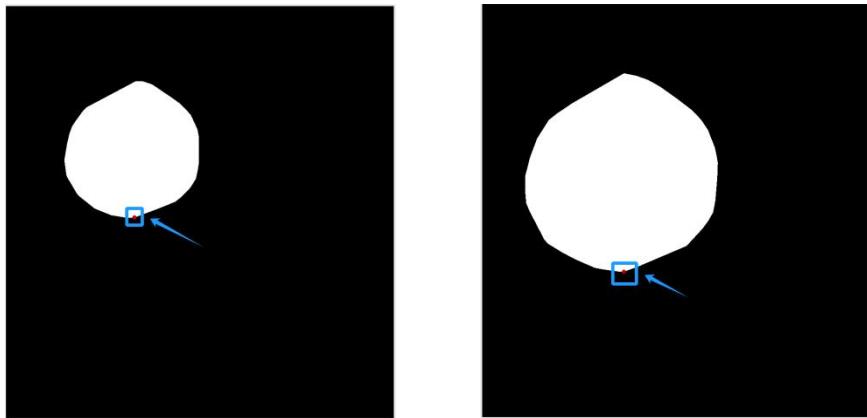


From left to right: Signal Matrix Mask Image, and the Signal Matrix Region of Interest "Wai" & "Nei" Mask Image

Run the code:

```
json = 'ssDNA/Adjusted_DSS12_ssDNA.json'
cellPointsFile = 'ssDNA/cellPoints.txt'
idjson = 'ssDNA/image_dimensions.json'
library(SpLin)
preAutoRegister(json, cellPointsFile, idjson)
```

B) Selection of Registration Reference Points (using Labelme software to select appropriate points)



From left to right: Signal Matrix Mask Image, and the Signal Matrix Region of Interest "Wai" & "Nei" Mask Image

Annotation Rules:

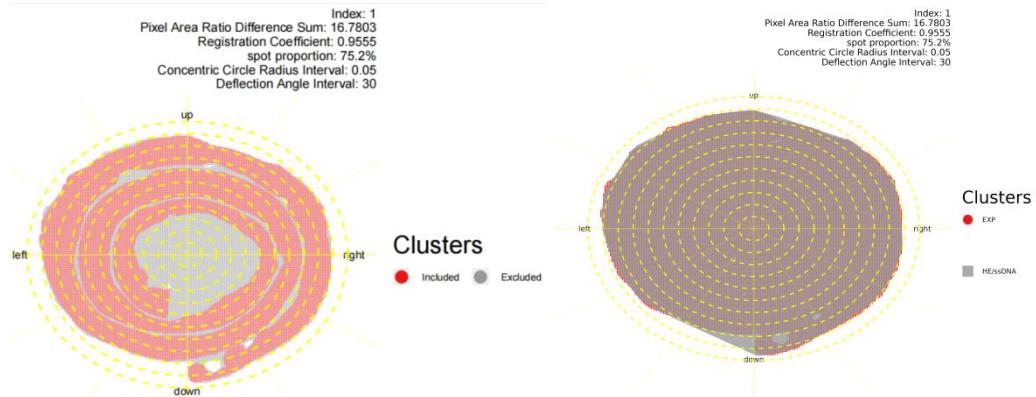
- For a single point, use the "Create Point" method.
- For multiple points, use the "Create Polygons" method, and the order of annotation must be consistent and correspond one-to-one.

C) Registration and Fine-Tuning

① Initial Registration:

```
drwxr-xr-x 3 luow research 23 Apr 11 14:00 Scheme
-rw-r--r-- 1 luow research 1.7M Apr 11 14:00 Multiple_AutoRegister_ST_plot.pdf
-rw-r--r-- 1 luow research 973K Apr 11 14:00 AllRegistrationSchemes.json
```

```
$ l Scheme/1/
total 1572
-rw-r--r-- 1 luow research 1606922 Apr 11 14:00 Multiple_AutoRegister_ST_plot.png
```



Run the code:

```
pixelssDNAFile = 'ssDNA/Adjusted_output_coordinates_and_colors.txt'
pixelHEFile = 'HE/Adjusted_output_coordinates_and_colors.txt'
markingPointsImage = 'HE/HE_pixel_MASK.png'
markingPoints_MASKJSON = 'HE/HE_pixel_MASK.json'
cellPointsImage = 'HE/ssDNA_pixel_MASK.png'
cellPoints_MASKJSON = 'HE/ssDNA_pixel_MASK.json'
idjson = 'ssDNA/image_dimensions.json'
```

```

json = 'ssDNA/Adjusted_DSS12_ssDNA.json'
cellPointsFile = 'ssDNA/cellPoints.txt'
markingPointsImage = 'ssDNA/marketingPoints_MASK.png'
markingPoints_MASKJSON = 'ssDNA/marketingPoints_MASK.json'
cellPointsImage = 'ssDNA/cellPoints_MASK.png'
cellPoints_MASKJSON = 'ssDNA/cellPoints_MASK.json'
idjson = 'ssDNA/image_dimensions.json'
transform = NULL
kpixel = 10
epsilon = 0.001
interval = 0.05
intervalAngle = 30
up = 0
down = 0
left = 0
right = 0
theta = 0
library(SpLin)
RapidRegister(json,      cellPointsFile,      markingPointsImage,      markingPoints_MASKJSON,
cellPointsImage, cellPoints_MASKJSON, idjson, transform = transform, kpixel = kpixel, epsilon =
epsilon, interval = interval, intervalAngle = intervalAngle, up = up, down = down, left = left, right =
right, theta = theta)

```

② Fine-Tuning Registration:

Note: If the initial registration meets the expected requirements, you can decide whether to proceed with fine-tuning the registration based on your needs. In this case, since the initial registration already satisfies the requirements, no fine-tuning is performed here.

③ Extract the registration information:

Results Directory: [ssDNA/](#)

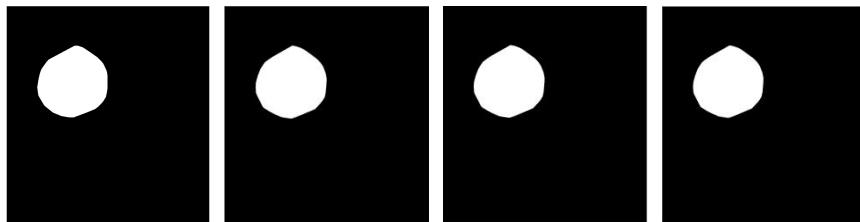
```

-rw-r--r-- 1 luow research 48K Apr 11 14:58 pixel_markingPointsMatch.png
-rw-r--r-- 1 luow research 2.2G Apr 11 14:58 Update_Manual_SCIL_input.rds
-rw-r--r-- 1 luow research 5.7M Apr 11 14:50 pixel_markingPoints_XY.txt
-rw-r--r-- 1 luow research 1.1M Apr 11 14:50 ssDNA_pixel_MASK_pixel_Update.png
-rw-r--r-- 1 luow research 1.1M Apr 11 14:19 markingPoints_MASK_update.png
-rw-r--r-- 1 luow research 954K Apr 11 14:19 Adjusted_Output_Manual.json

```

Results Directory: [HE/](#)

```
$ ll -th Scheme/1/
total 1.7G
-rw-r--r-- 1 luow research 37K Apr 11 15:07 pixel_markingPointsMatch.png
-rw-r--r-- 1 luow research 827M Apr 11 14:50 pixel_markingPoints_XY.txt
-rw-r--r-- 1 luow research 1.1M Apr 11 14:49 HE_pixel_MASK_pixel_Update.png
-rw-r--r-- 1 luow research 1.2M Apr 11 14:24 HE_pixel_MASK_pixel.png
```



The images from left to right are:

- Signal matrix mask image
- Mask image of the signal matrix region of interest after correction of "Wai" & "Nei"
- Mask image of the ssDNA staining image after correction
- Mask image of the HE-stained tissue of the adjacent section after correction

Run the code:

```
rds = 'SpLin_input.rds'
json = 'ssDNA/AllRegistrationSchemes.json'
idjson = 'ssDNA/image_dimensions.json'
pixelssDNAFile = 'ssDNA/Adjusted_output_coordinates_and_colors.txt'
pixelHEFile = 'HE/Scheme/1/ssDNAHE_Adjusted_output_coordinates_and_colors.txt'
index = 1
library(SpLin)
RapidRegisterOUT(rds, json, idjson, pixelssDNAFile = pixelssDNAFile, pixelHEFile = pixelHEFile,
index = index)
```

- ④ Check whether the regions annotated by Labelme are registered with the regions of interest

```
total 4.3G
-rw-r--r-- 1 luow research 58K Apr 11 15:24 signalLabelmeRegion.png
```

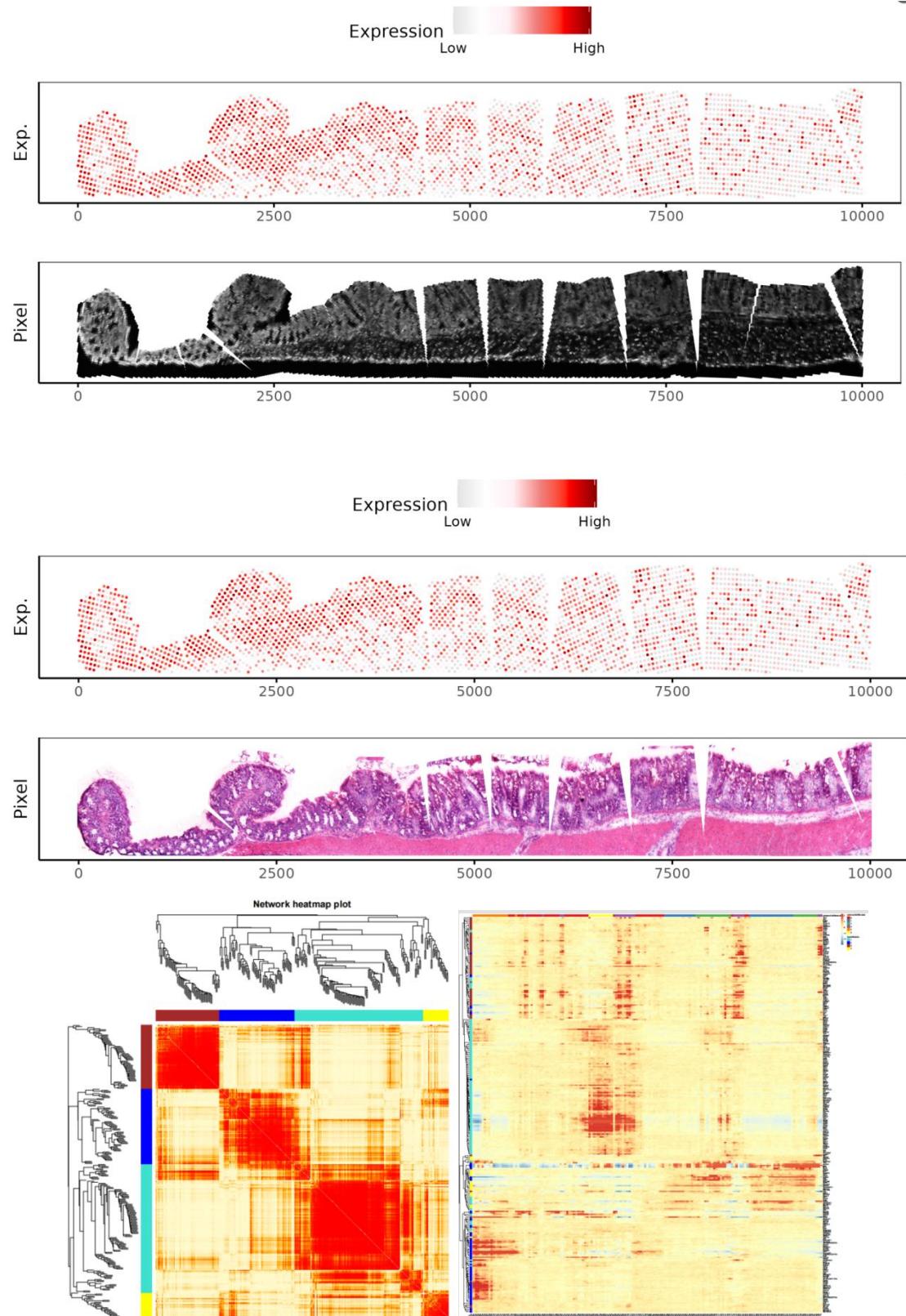


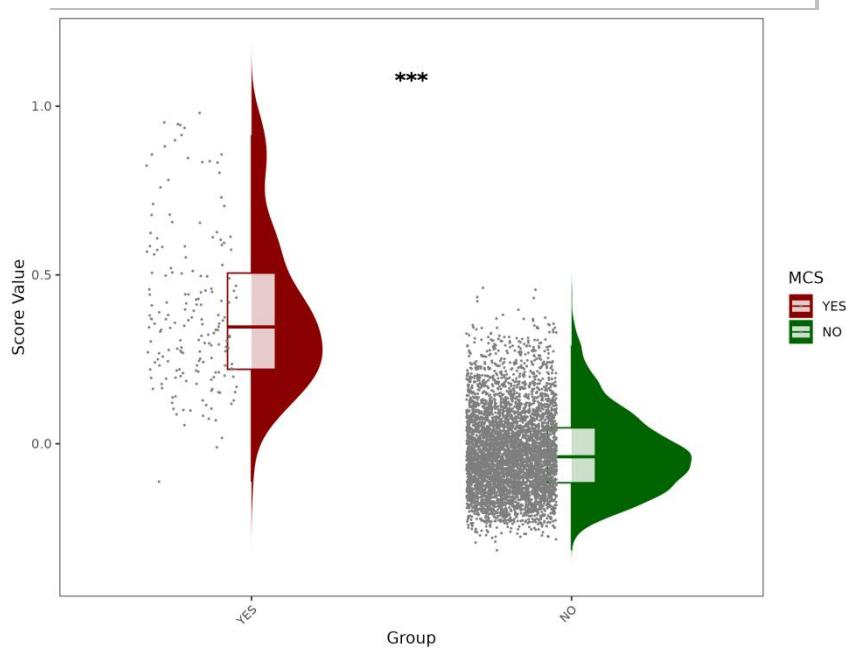
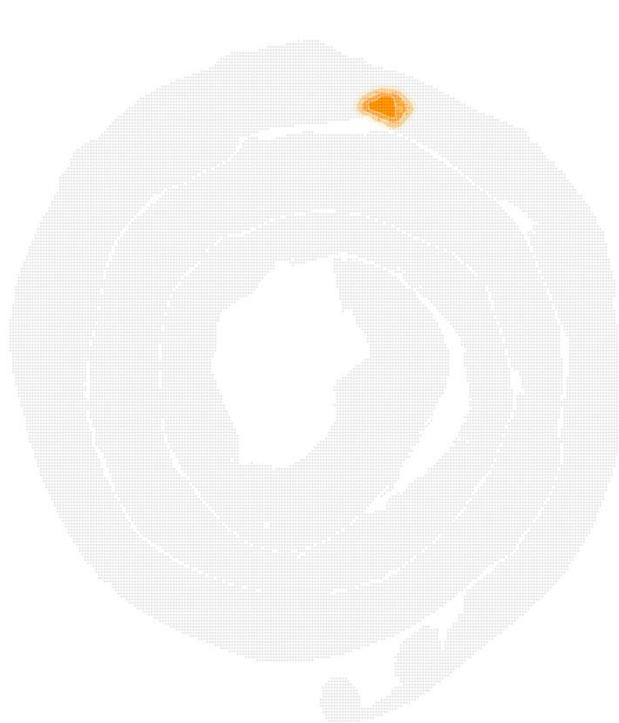
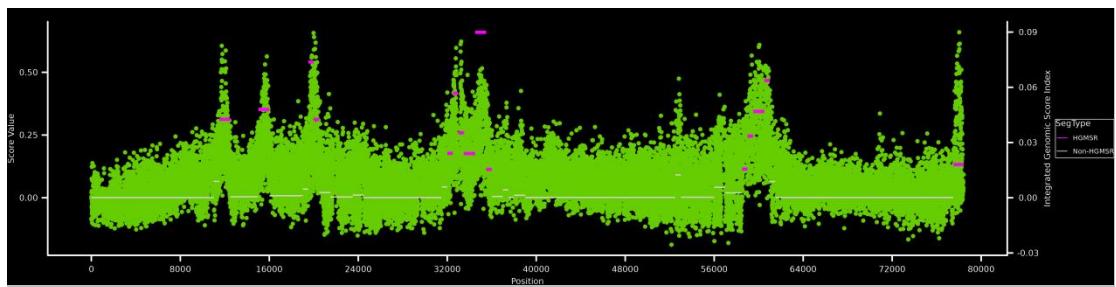
Run the code:

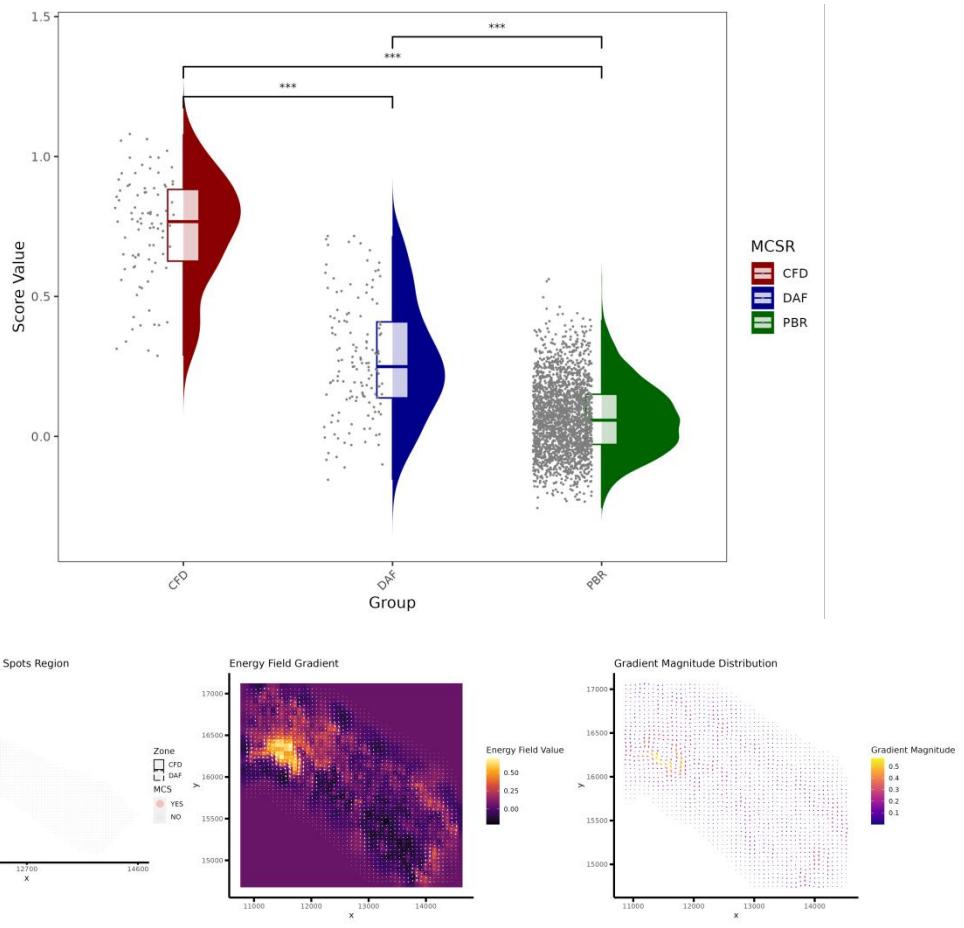
```
rds = 'SpLin_input.rds'
json = 'ssDNA/Adjusted_Output_Manual.json'
library(SpLin)
```

```
checkLabelmeRegion(rds, json)
```

4. Display of Partial Visualization Results After Straightening

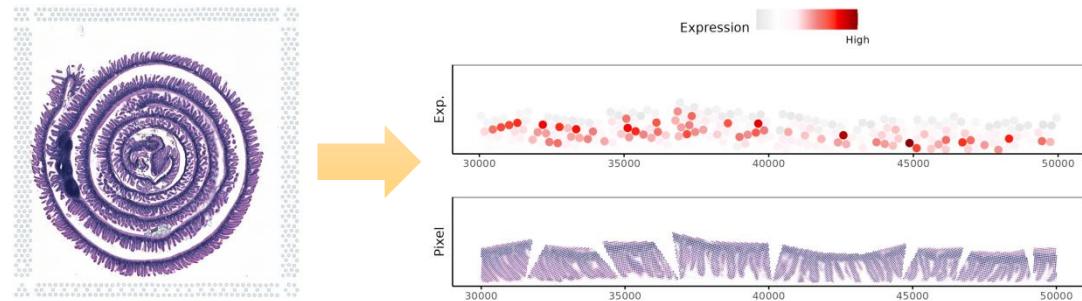


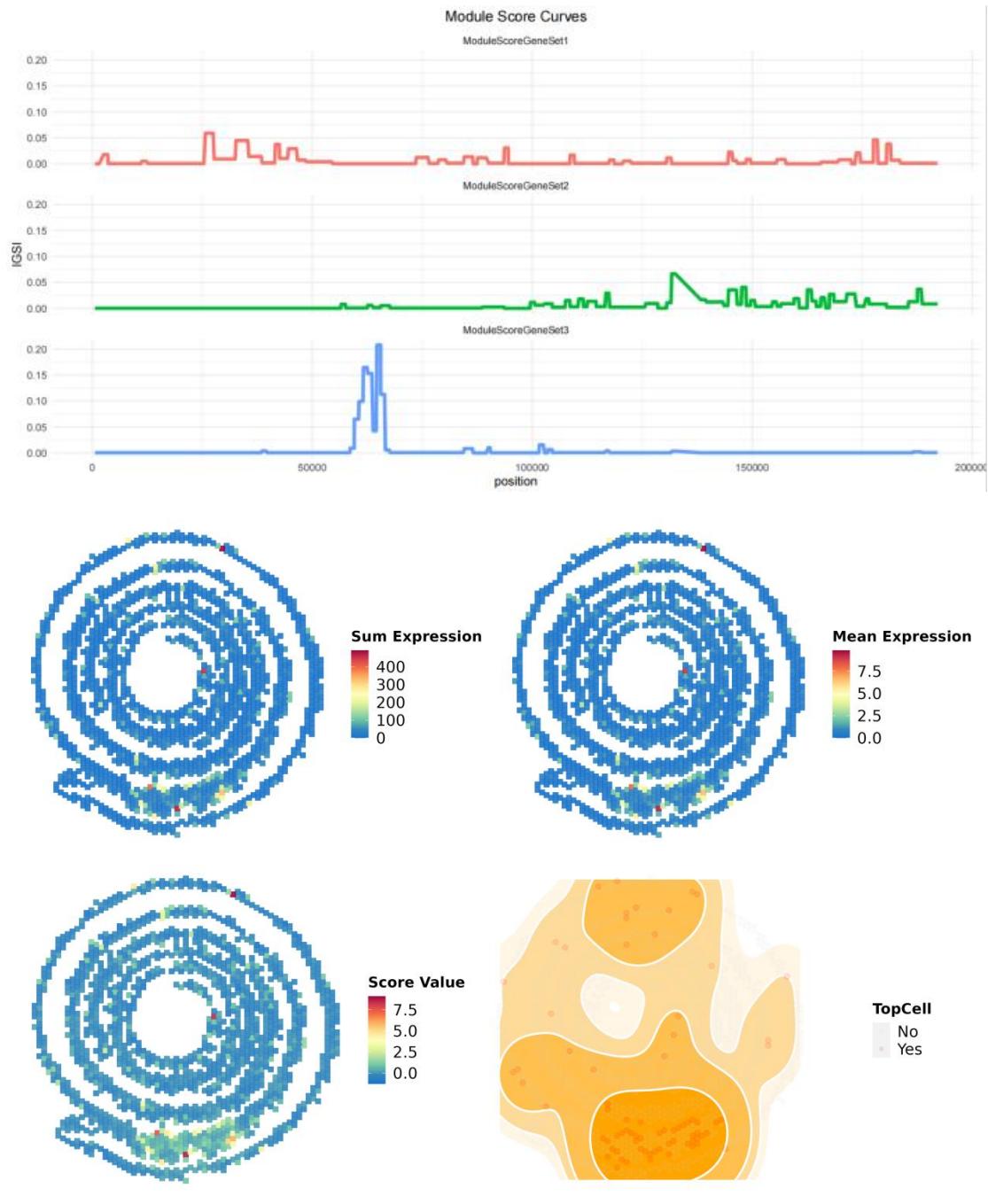


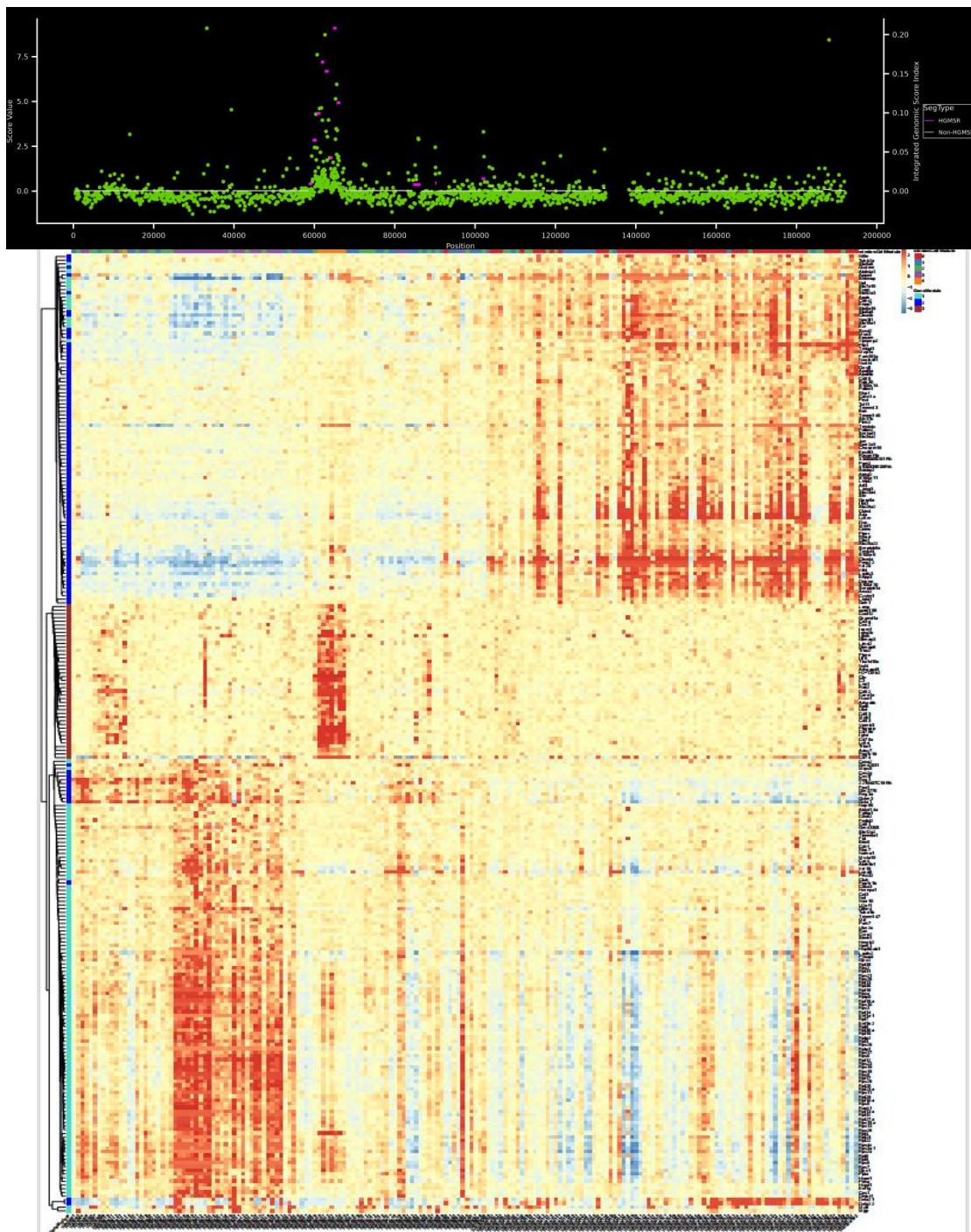


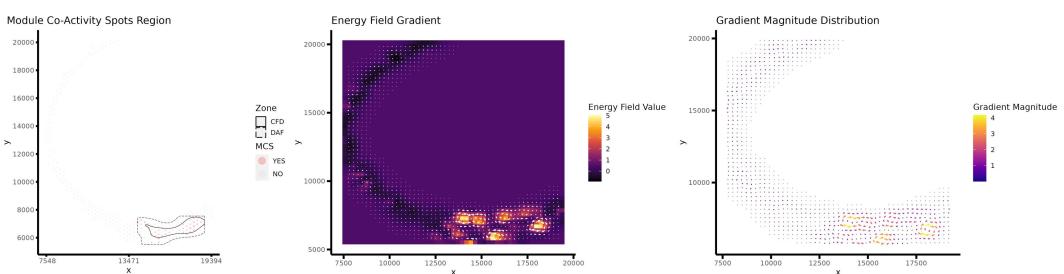
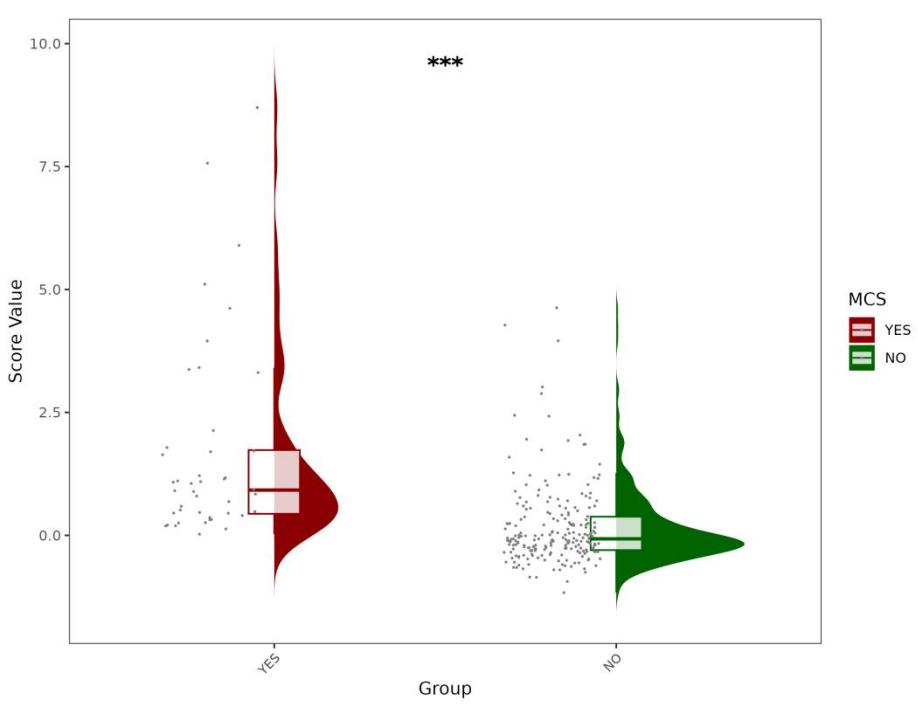
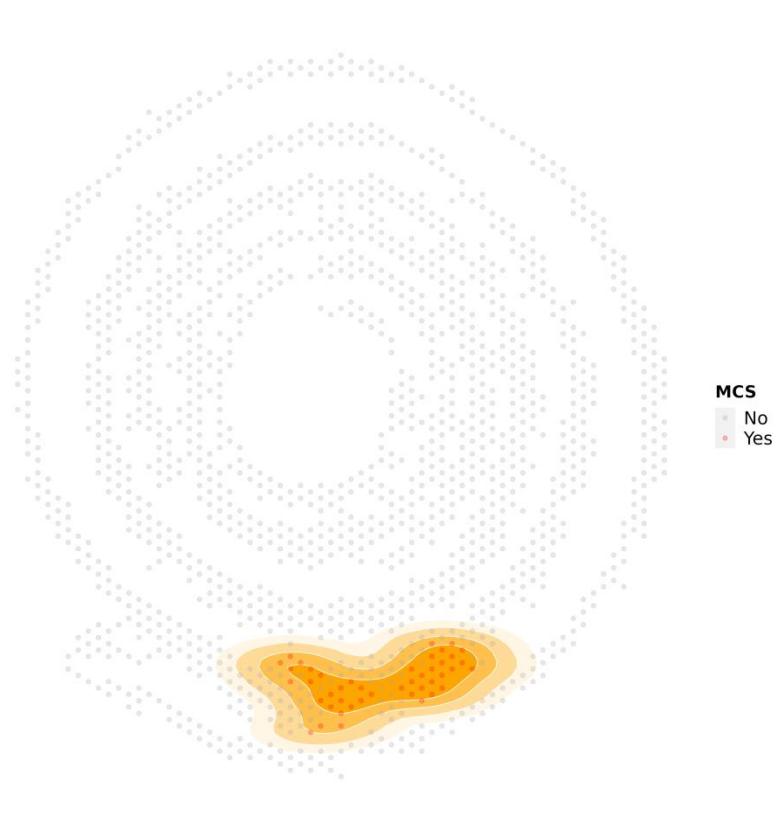
5. Display of Partial Results from More Cases

The sample [GSM7840112_V11Y11-298_B](#) from the 10X Genomics dataset [GSE245316](#)





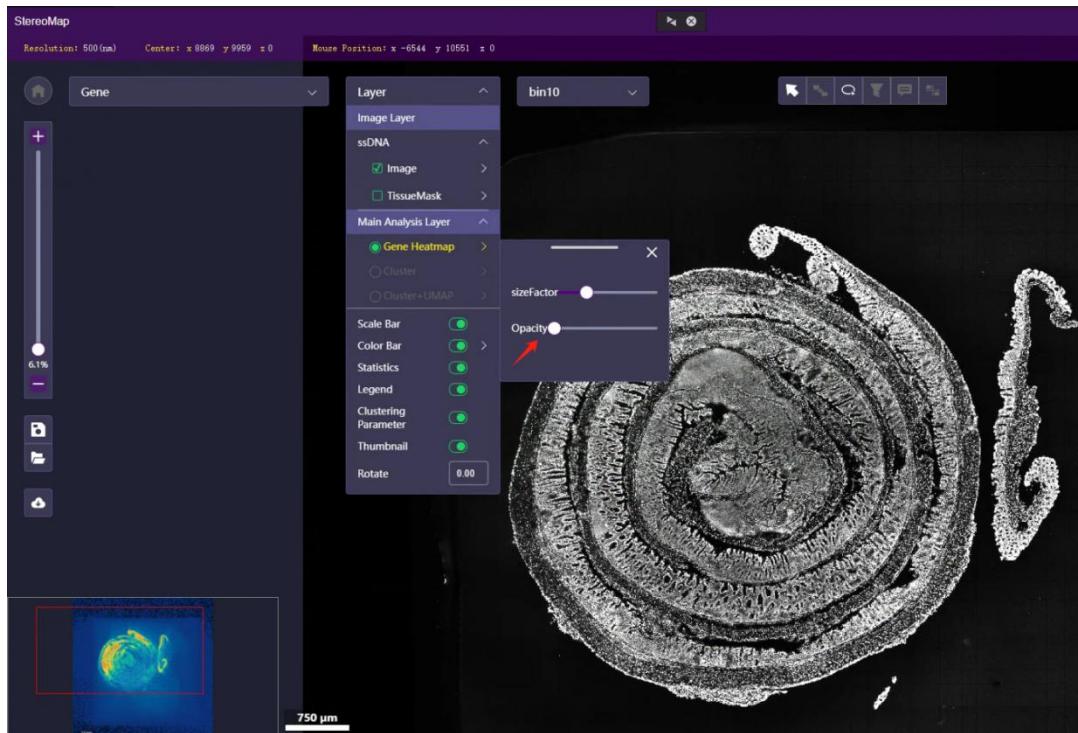




IV. Precautions

1. For Stereo-seq data, it is recommended to perform Lasso operations on the signal matrix to remove noise signal points, especially when aligning with stained film images

The following provides an example of how to use Lasso (see screenshot):



Note: In StereoMap, set the transparency of the expression matrix to the highest level (i.e., slide the bar all the way to the left). Then, perform precise Lasso selection along the tissue edges for the region that needs to be straightened. Note that non-physically continuous tissue parts should not be included in the Lasso area, as this will prevent straightening (e.g., the small segment of intestinal tissue on the right side in the figure above).

The results after Lasso are shown in the screenshot below:



2. It is recommended to use Labelme or the annotation software independently

developed by KMHD for contour registration and annotation

Annotation Method: Dual-label or multi-label mode is used for annotation. For biological structures, the inner and outer sides of the same tissue must not intersect. Precise edge tracing is required for the outer contour.

Annotation Rules:

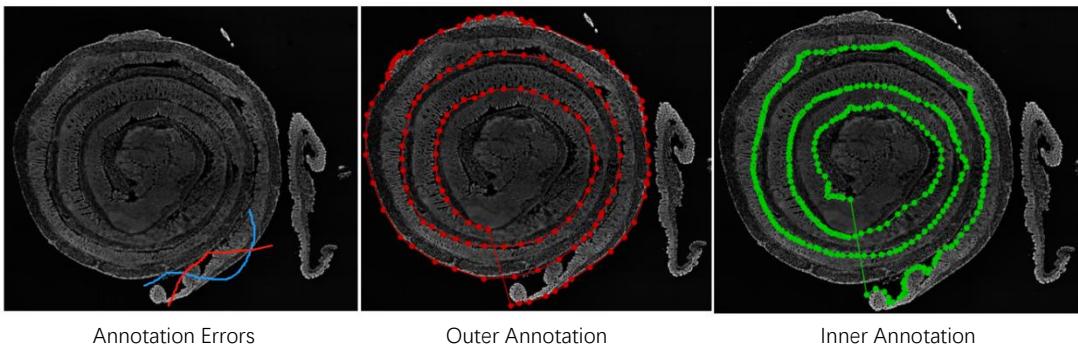
1) Signal Matrix Area (for registration): The fixed field for the outer contour label is “Sign,” which requires precise edge tracing and as many annotation points as possible.

2) Target Area (for straightening):

A. The fixed field for the outer label is “Wai,” which requires precise edge tracing.

B. The fixed field for the inner label is “Nei,” which should fully cover the target area. For adjacent points (excluding the start and end points), the lines connecting Wai and Nei must not intersect. After forming a closed loop, the annotated points should not be dragged.

Example of Legend:



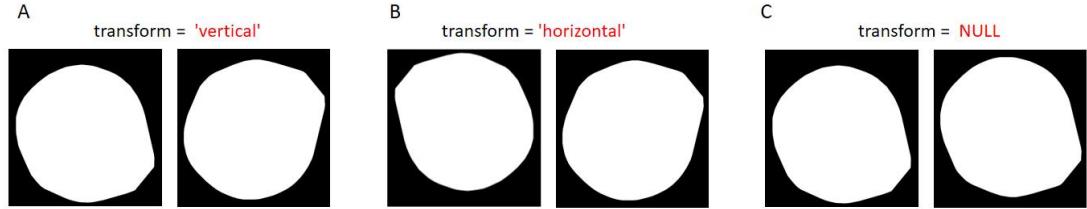
3. Mask Image Registration and Reference Point Selection

Theoretically, the method used in this study is capable of registering mask images with similar shapes, whether they are regular or irregular, and is not limited to data with spiral roll shapes. However, the manual annotation of tissue contour edges may introduce errors at the detail level for the following reasons:

- 1) Manual operations in StereoMap for Lasso selection and Labelme annotation are prone to detail errors;
- 2) White noise introduced during sample collection, experimental procedures, and sequencing may cause some tissue distribution areas to have very weak signal expression or even fail to be captured. Even if StereoMap is used to perform Lasso processing along the tissue contour, the resulting expression distribution contour may not match the Lasso contour;
- 3) The choice of Bin size is another influencing factor. Its distribution is in the form of scattered points, which lacks the continuity of tissue boundaries.

Therefore, the method used in this study can only ensure the overall trend of registration and is not capable of achieving precise contour matching. For highly regular shapes (such as circles), it is necessary to combine the registration results and repeatedly attempt to find suitable matching points to complete the registration.

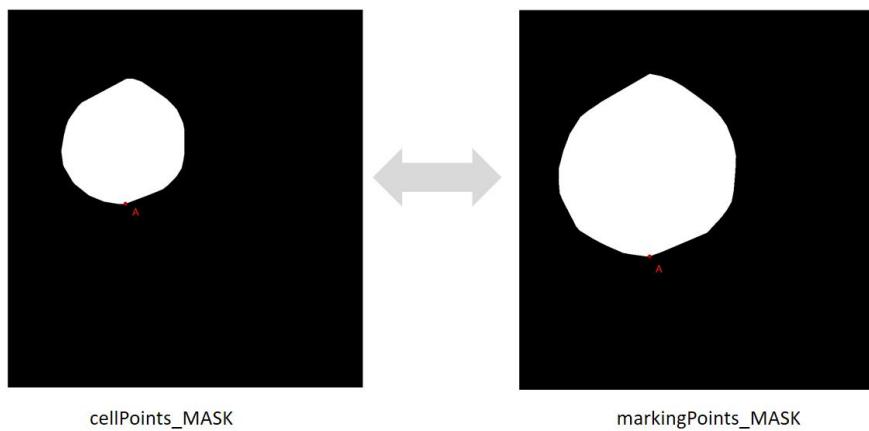
Mask Image Registration Parameters: The registration parameters for mask images need to be determined based on the specific conditions of the mask images, as follows:



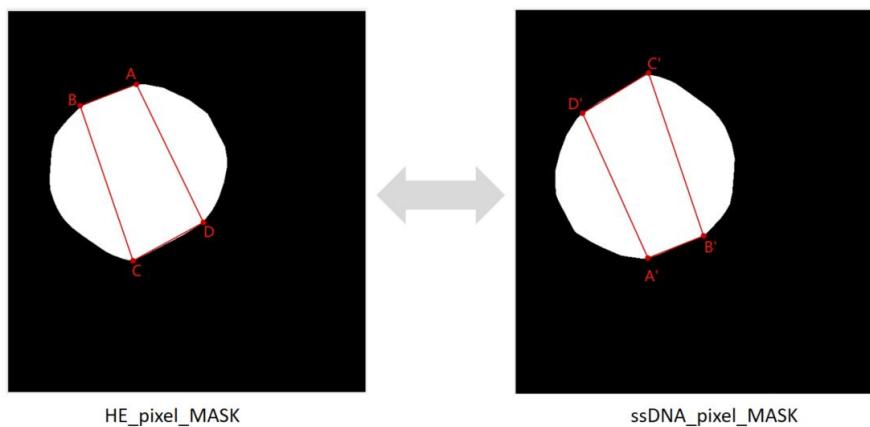
In the figure, A and B have undergone a mirror transformation, while Group C has not.

Reference Points: Construct ordered registration reference points based on the contour features of the mask images (at least one pair). For non-regular mask images, it is recommended to accurately annotate one pair of reference points; for regular mask images, it is recommended to accurately annotate two pairs of reference points. These reference points form the initial values for fine-tuning. Ordered registration reference points refer to selecting points on the contours of the two mask images to be registered, which can serve as the basis for alignment. If there are more than one pair of reference points, these points must correspond to each other one-to-one, meaning they should be annotated in the same order.

For example, if the signal matrix mask image has sequentially annotated a single point A, then the mask image for the region of interest in Labelme must also annotate a single point, and the order must be A', as shown in the screenshot below:



For example, if the HE mask image has sequentially annotated four points A, B, C, and D, then the ssDNA mask image must also annotate four points, and the order must be A', B', C', and D', as shown in the screenshot below:



Note: If you need to adjust the reference points, please first delete the local mask images (PNG files) and their corresponding JSON files, and then re-download them from the server to proceed with the annotation.

Annotation Rules (using Labelme software as an example):

- ① For a single point, use the "Create Point" method.
- ② For multiple points, use the "Create Polygons" method.

Comprehensive Evaluation of Results:

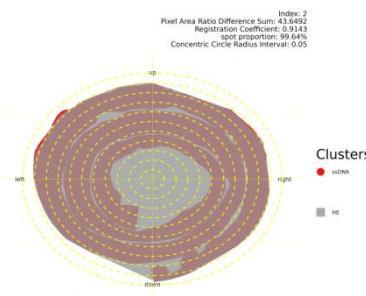
- ① Small difference in pixel area;
- ② High registration coefficient (recommended to be above 0.85);
- ③ High spot utilization rate;
- ④ The contour of the signal matrix matches the outer contour of the annotation box in appearance (visual inspection).

Here, the spot utilization rate refers to:

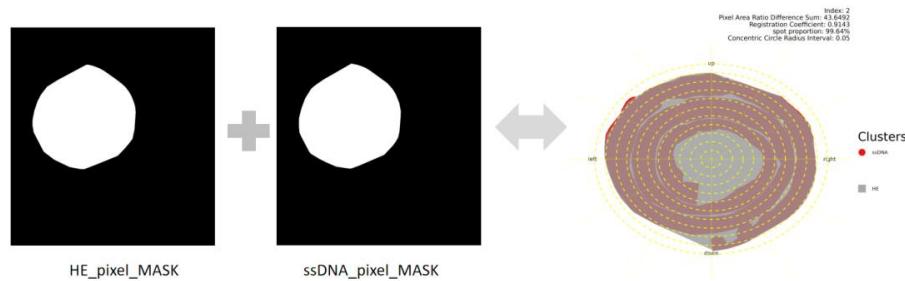
- 1) The proportion of pixels within the ssDNA area, which has been annotated using Labelme, that fall within the outer polygon annotated by Labelme in the HE-stained image of the adjacent section;
- 2) The proportion of signal points in the signal matrix that fall within the polygon annotated by Labelme in the ssDNA staining image.

For the registration between the tissue staining image of adjacent sections and the chip staining image:

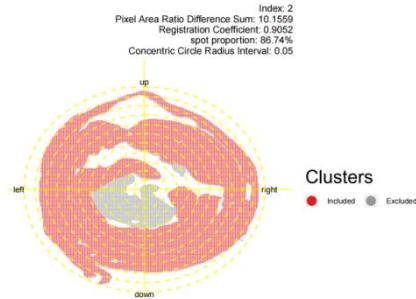
The stretching and scaling operations are applied to the red area representing the H&E-stained area, with the light gray area representing the ssDNA staining image serving as the control background, as shown in the screenshot below:



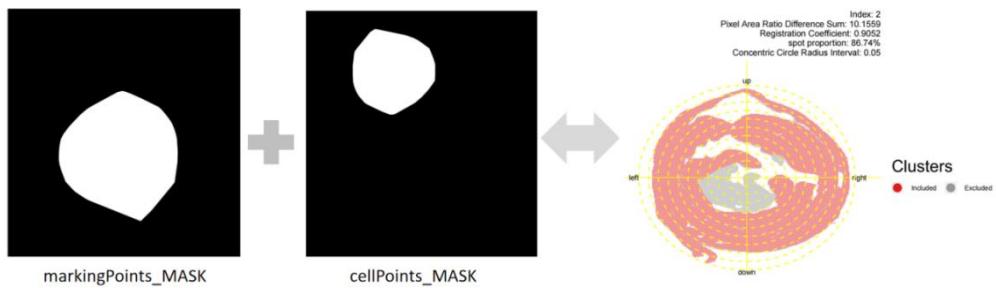
Registration Results Display:



For the registration between the signal matrix image and the chip staining image: The stretching and scaling operations are applied to the light red areas, with the gray areas serving as the control background, as shown in the screenshot below:

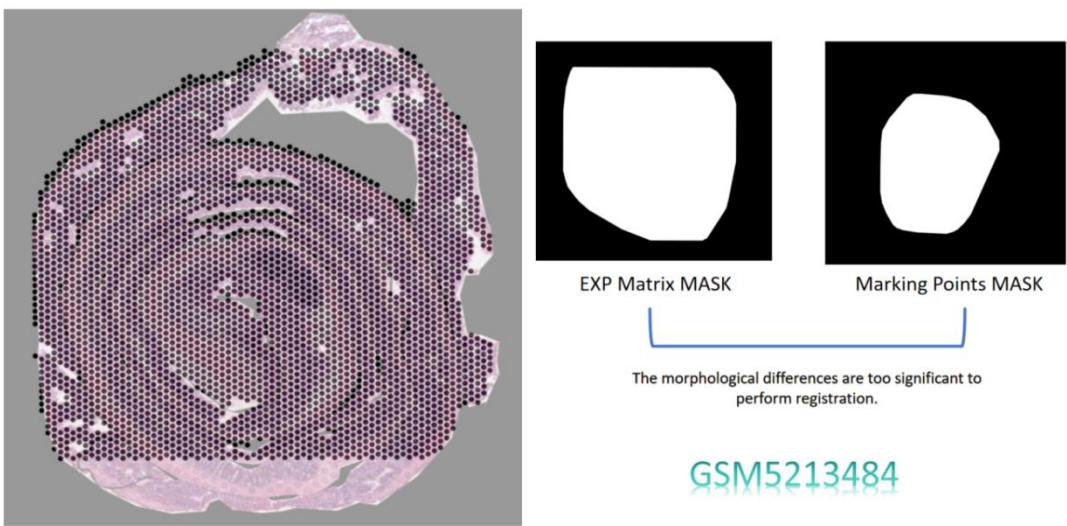


Registration Results Display:



Mismatch between the signal matrix image and the chip staining image: Signal loss caused by experimental, sequencing, or signal quality issues prevents the signal matrix image from aligning with the chip staining image. In such cases: a) Using Labelme is not feasible for aligning the signal matrix image with the chip staining image. However, the registration software developed by KMHD can achieve alignment and perform straightening operations (undisclosed). b) Straightening operations can be performed on the signal matrix independently.

The sample is shown in the screenshot below:

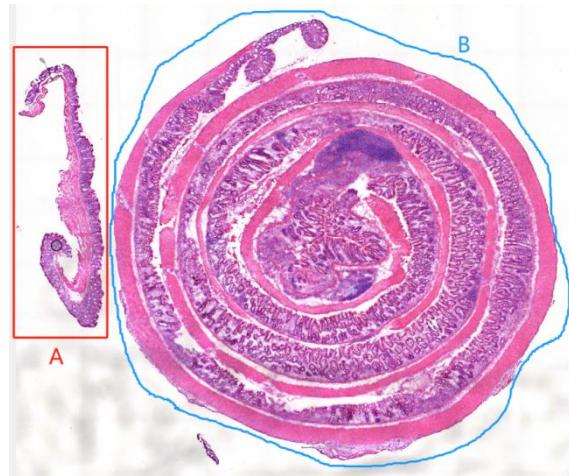


If the matrix expression is missing and the matrix contour edges do not match the film image, registration and straightening of the staining image are temporarily not supported. However, the signal matrix can be straightened separately.

10X Genomics sequencing type, the GSM5213484 sample in GSE169749

4. Non-physically contiguous areas should not be annotated together

For example, Area A should not be annotated together with Area B as shown in the figure below:



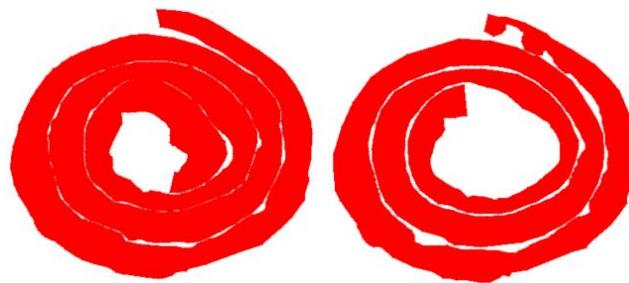
5. Verify the JSON file

Determining the Correctness of Polygons Formed by Wai and Nei Points:

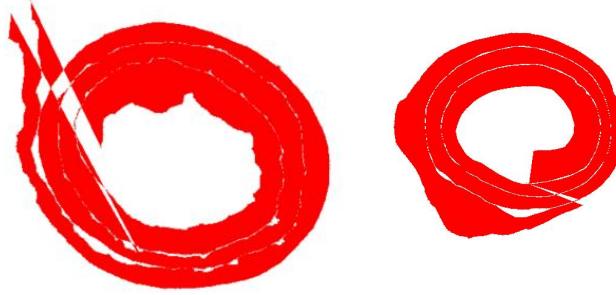
It is crucial to ensure that the polygons formed by the points of Wai and Nei are correct. If they are not, the points should be redrawn. The requirements are as follows:

- 1) Wai and Nei should not be too close to each other to avoid coordinate overlap due to precision issues during program execution.
- 2) The points should be orderly and form a closed loop. Once a closed loop is formed, dragging to modify the shape is not allowed. Dragging introduces new points that are automatically added to the end of the point set, disrupting the original order.

Qualified Polygons:

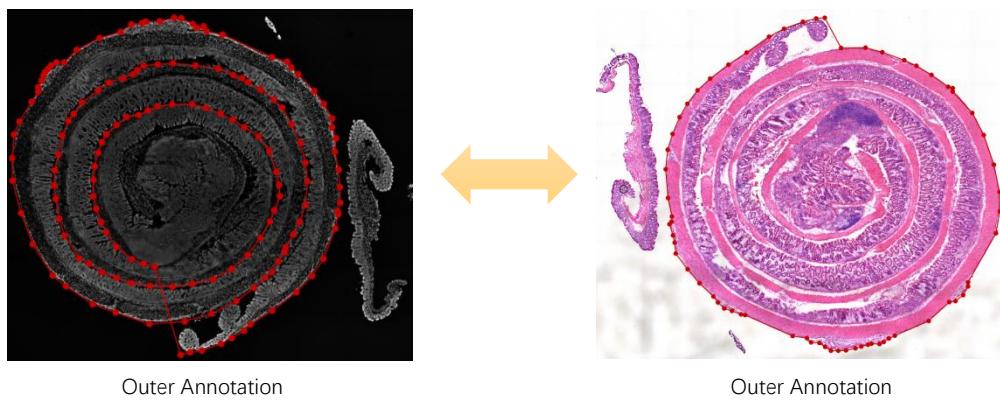


Unqualified Polygons (caused by intersections or dragging after forming a closed loop):



6. Annotations for HE-stained sections adjacent to the chip

Requirement: To maintain sufficient alignment with the outer annotated regions in the ssDNA staining image.



7. Integration of Multi-Chip Data: Steps for Identifying Regions with Characteristic Gene Expression Patterns

- 1) Conduct segmentation analysis on each chip independently

Note:

- ① If using straightened data, select an appropriate window size.
- ② If using non-straightened tissue microarray data, select an appropriate grid density.

The following code example is based on straightened data.

Run the code:

```
rds = 'ssDNA/SCIL_EXP_output.rds'
windowSV = 500
library(SpLin)
wcellDue(rds, windowSV = windowSV)
```

Output:

```
-rw-r--r-- 1 luow research 2801119387 Apr  8 17:15 SpLin_EXP_output_withWindow.RDS
-rw-r--r-- 1 luow research    26548184 Apr  8 17:07 windowCell.RDS
```

- 2) Merge the windowCell.RDS files from all chips

```
$ cat list
/home/luow/LazhiIn/test/jiangxuanting/A02882A2/ssDNA/STEDZ/winCell/windowCell.RDS
/home/luow/LazhiIn/test/jiangxuanting/A02882C6/ssDNA/STEDZ/winCell/windowCell.RDS
/home/luow/LazhiIn/test/jiangxuanting/A02883B5/ssDNA/STEDZ/winCell/windowCell.RDS
/home/luow/LazhiIn/test/jiangxuanting/A02885F3/ssDNA/STEDZ/winCell/windowCell.RDS
/home/luow/LazhiIn/test/jiangxuanting/A02989D4/ssDNA/STEDZ/winCell/windowCell.RDS
/home/luow/LazhiIn/test/jiangxuanting/A02996E5/ssDNA/STEDZ/winCell/windowCell.RDS
```

Run the code:

```
library(Seurat)
CTR20 <- readRDS('/home/luow/LazhiIn/test/jiangxuanting/A02882A2/ssDNA/IGSI/winCell/windowCell.RDS')
CTR95 <- readRDS('/home/luow/LazhiIn/test/jiangxuanting/A02882C6/ssDNA/IGSI/winCell/windowCell.RDS')
DSS12 <- readRDS('/home/luow/LazhiIn/test/jiangxuanting/A02883B5/ssDNA/IGSI/winCell/windowCell.RDS')
DSS37 <- readRDS('/home/luow/LazhiIn/test/jiangxuanting/A02885F3/ssDNA/IGSI/winCell/windowCell.RDS')
GH79 <- readRDS('/home/luow/LazhiIn/test/jiangxuanting/A02989D4/ssDNA/IGSI/winCell/windowCell.RDS')
GH82 <- readRDS('/home/luow/LazhiIn/test/jiangxuanting/A02996E5/ssDNA/IGSI/winCell/windowCell.RDS')
Em <- list()
Em[[1]] <- CTR20
Em[[2]] <- CTR95
Em[[3]] <- DSS12
Em[[4]] <- DSS37
Em[[5]] <- GH79
Em[[6]] <- GH82
saveRDS(Em, 'winCell/windowCell.RDS')
```

3) Identifying Regions with Characteristic Gene Expression Patterns

Run the code:

```
library(SpLin)
rds = 'winCell/windowCell.RDS'
dims <- 30
resolution <- 0.6
strict <- TRUE
show_rownames = TRUE
show_colnames = TRUE
angle_col = "45"
width = 30
height = 45
ColumnCluster = TRUE
rdsList = TRUE
moduleGene(rds, dims = dims, resolution = resolution, strict = strict, show_rownames = show_rownames,
show_colnames = show_colnames, angle_col = angle_col, width = width, height = height, ColumnCluster = ColumnCluster,
rdsList = rdsList)
```

4) Conduct MCS analysis on each chip independently

Run the code:

```
library(SpLin)
rds = 'winCell/SpLin_EXP_output_withWindow.RDS'
GMF =
'/home/luow/LazhiIn/test/jiangxuanting/windowMergeAna/win500/New/winCell/Modules/Gene/moduleLabel.txt'
```

```
IGSI(rds, GMF, topFreq = 0.05, nbin = 10, alpha = 0.001, SD = 3, topSegFreq = 0.4, TopCellFreq = 0.01, piontsize =  
0.5)
```

8. Notes on Processing Non-Straightened Data

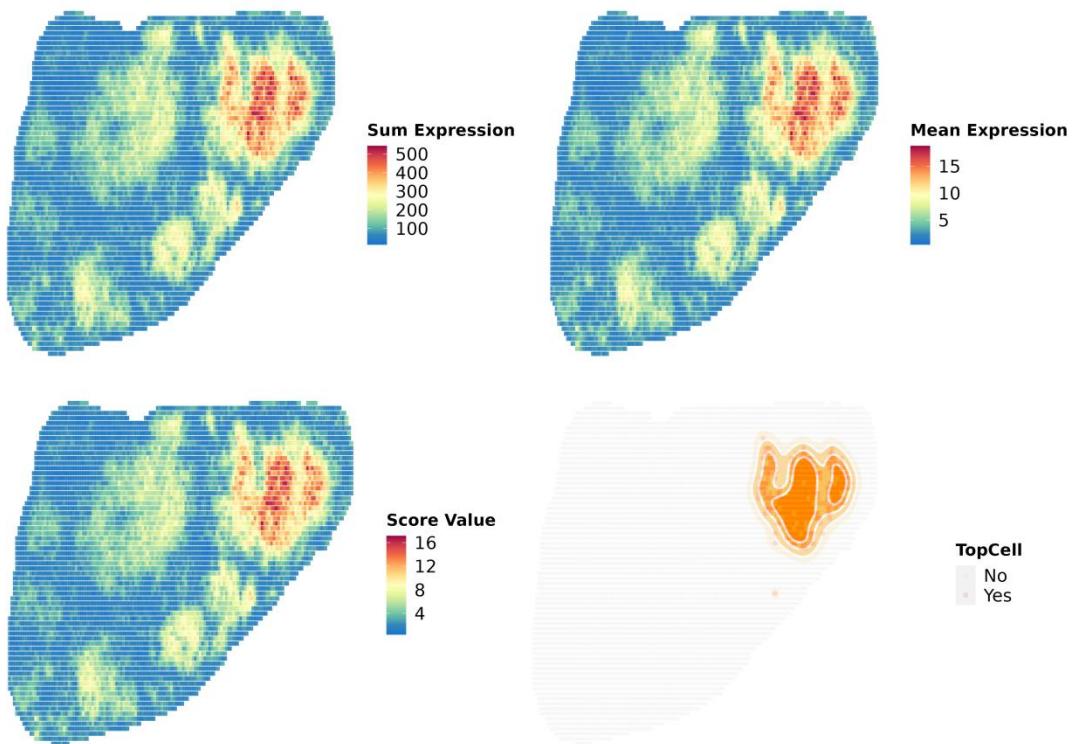
Case Study (Data Source: PMID: 37156441)

Run the code:

```
#Step01  
rds = 'ssDNA/GF.bin50.RDS'  
grid_density = 30  
library(SpLin)  
wcellDueGrid(rds, grid_density = grid_density)  
#Step02  
rds = 'ssDNA/IGSI/winCellGrid/wcellDueGrid.RDS'  
dims <- 10  
resolution <- 0.8  
strict <- TRUE  
ColumnCluster <- TRUE  
show_rownames = TRUE  
show_colnames = TRUE  
angle_col = "45"  
width = 28  
height = 32  
moduleGene(rds, dims = dims, resolution = resolution, strict = strict, ColumnCluster = ColumnCluster,  
show_rownames = show_rownames, show_colnames = show_colnames, angle_col = angle_col, width = width, height =  
height)  
#Step03  
rds = 'ssDNA/IGSI/winCellGrid/SpLin_EXP_output_withWindowGrid.RDS'  
GMF = 'ssDNA/IGSI/winCellGrid/Modules/Gene/moduleLabel.txt'  
IGSIGrid(rds, GMF, topFreq = 0.05, nbin = 10, topSegFreq = 0.4, TopCellFreq = 0.01, piontsize = 2.5)
```

Partial Results Display:

Spleen of germ-free (GF) mice



Spleen of specific pathogen-free (SPF) mice

