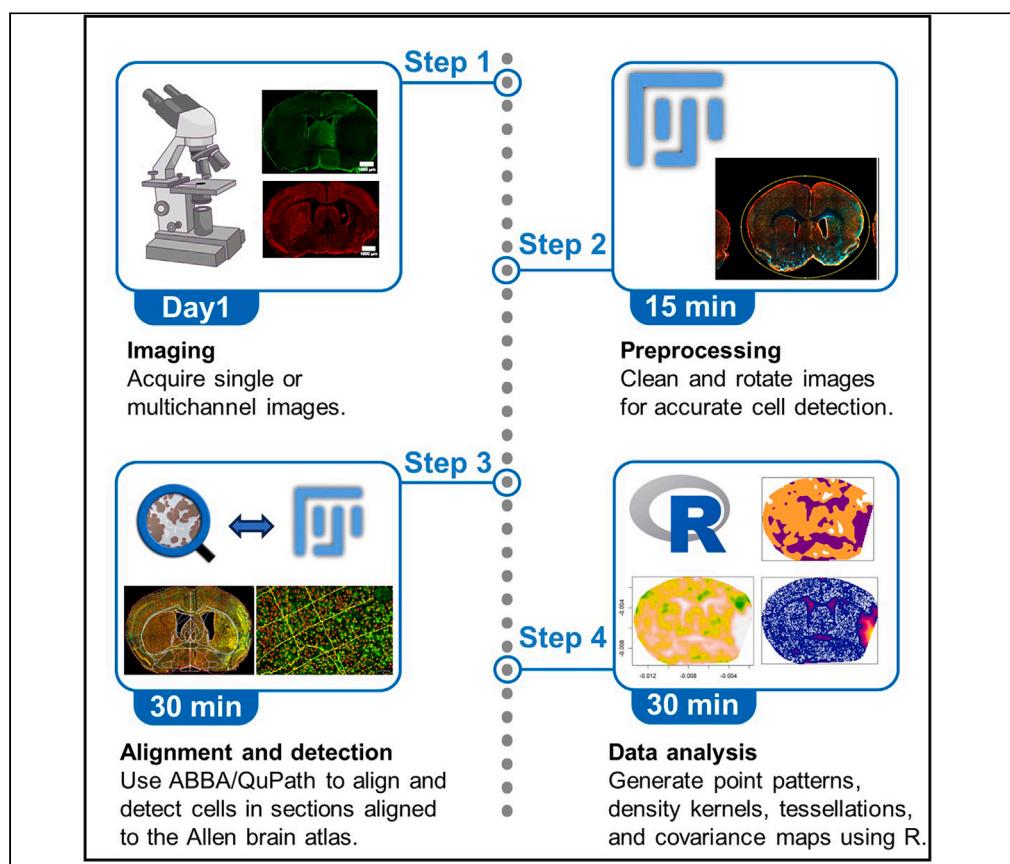


Protocol

Unbiased quantification of the spatial distribution of murine cells using point pattern analysis



CNS injuries are associated with profound changes in cell organization. This protocol presents a stepwise approach to quantitatively describe the spatiotemporal changes in glial cell rearrangement in the injured murine brain, which is applicable to other biological contexts. Herein, we apply common immunolabeling of neurons and glial cells and wide-field microscopy imaging. Then, we employ computational tools for alignment to the Allen Brain Atlas, unbiased/automatic detection of cells, generation of point patterns, and data analysis.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Daniel Manrique-Castano, Ayman ElAli

damac36@ulaval.ca (D.M.-C.)
ayman.el-ali@crchudequebec.ulaval.ca (A.E.)

Highlights
Unbiased detection/quantification of cells in brain atlas-aligned sections

Creation and analysis of point patterns based on *xy* coordinates of single cells

Calculation of density kernels and raster layers for analysis of spatial intensity

Calculation of cell distribution, covariance, and spatial modeling

Manrique-Castano & ElAli,
STAR Protocols 5, 102989
June 21, 2024 © 2024 The Author(s). Published by Elsevier Inc.
<https://doi.org/10.1016/j.xpro.2024.102989>



Protocol

Unbiased quantification of the spatial distribution of murine cells using point pattern analysis

Daniel Manrique-Castano^{1,2,3,*} and Ayman ElAli^{1,2,4,*}

¹Neuroscience Axis, Research Center of CHU de Québec - Université Laval, Quebec City, QC G1V 0A6, Canada

²Department of Psychiatry and Neuroscience, Faculty of Medicine, Université Laval, Quebec City, QC G1V 4G2, Canada

³Technical contact

⁴Lead contact

*Correspondence: damac36@ulaval.ca (D.M.-C.), ayman.el-ali@crchudequebec.ulaval.ca (A.E.)
<https://doi.org/10.1016/j.xpro.2024.102989>

SUMMARY

CNS injuries are associated with profound changes in cell organization. This protocol presents a stepwise approach to quantitatively describe the spatiotemporal changes in glial cell rearrangement in the injured murine brain, which is applicable to other biological contexts. Herein, we apply common immunolabeling of neurons and glial cells and wide-field microscopy imaging. Then, we employ computational tools for alignment to the Allen Brain Atlas, unbiased/automatic detection of cells, generation of point patterns, and data analysis. For complete details on the use and execution of this protocol, please refer to Manrique-Castano et al.¹

BEFORE YOU BEGIN

Adequate cell arrangement and spatiotemporal positioning during development and adulthood are a pre-requisite for optimal brain functioning in health and disease.² Ischemic stroke, which is one of the most prevalent CNS injuries, profoundly reshapes cell reorganization. Neuronal loss is associated with the activation of astrocytes and microglia, which are spatially rearranged in a fine-tuned manner to neatly demarcate and segregate the lesion site.³ The process of glial cell reorganization evolves overtime and decisively affects overall structural and functional outcomes.⁴

This protocol outlines a stepwise quantification of the spatiotemporal redistribution patterns of glial cells associated with neuronal loss in the context of ischemic stroke. We immunolabeled neurons, astrocytes, and microglia in serial brain sections using common markers, and we acquired wide-field images using a AxioScan Z1 slide scanner (Carl Zeiss Canada, ON, Canada). Next, we used open-source software, including FIJI⁵ and QuPath⁶ to perform image pre-processing, alignment, registration, and annotations to the Allen brain atlas. Using QuPath, we perform unbiased/automatic quantification of cells. Please note that in case no alignment to the Allen brain atlas is required, for example when ROIs are analyzed, unbiased cell detection can still be executed using alternative open-source software such as FIJI, CellProfiler or Python. As this protocol is based on mapping the xy coordinates of different objects, it has a wide range of applications in other biological contexts. For instance, it is suitable to simultaneously quantify the spatiotemporal distribution of multiple objects, including cells, organelles, vesicles, and other particles.

Later, we use R statistical software to process the dataset and perform point pattern analysis (PPA) using primarily the spatstat package.^{7,8} Since this package does not have a graphical user interface (GUI), this protocol provides a detailed annotated code for straightforward implementation. The user will also benefit from Quarto notebooks for single and batch processing of the images, point patterns and data following the same principles (<https://doi.org/10.5281/zenodo.10805534>).



Institutional permissions

Animal experiments to model ischemic stroke in mice using the transient middle cerebral artery occlusion (MCAo) were performed as previously described⁹ according to the Canadian Council on Animal Care guidelines, as implemented by the *Comité de Protection des Animaux de l'Université Laval-3* (CPAUL-3; Protocol # 20-470).

Brain staining and imaging

The conditions for staining are broadly flexible and most common approaches used in neuroscience are suitable. Therefore, this protocol assumes the user has performed a staining of interest and done imaging using wide-field or confocal microscopy. The steps presented in this article use staining of NeuN, GFAP and IBA1 in coronal 30 µm-thick brain sections.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
Fiji	Fiji developing team	https://imagej.net/software/fiji/
Aligning Big Brains & Atlases (ABBA)	BIOP	https://biop.github.io/ijp-imagetoatlas/
QuPath	Peter Bankhead et al.	https://qupath.github.io/
Warpy extension for QuPath	BIOP	https://github.com/biop/qupath-extension-warpy/releases
ABBA extension for QuPath	BIOP	https://github.com/BIOP/qupath-extension-abba/releases
Elastix	SuperElastix	https://github.com/SuperElastix/elastix/releases/tag/5.0.1
R-software	R Foundation	https://cran.r-project.org/mirrors.html
Spatstat package	Adrian Baddeley, Rolf Turner, and Ege Rubak	https://spatstat.org/
GitHub repository	Ayman ElAli Lab	https://github.com/elalilab/StarProtocol_PPA (https://doi.org/10.5281/zenodo.10805534)

MATERIALS AND EQUIPMENT

This protocol requires the installation of ABBA and associated plugins in QuPath (64-bit Windows, Linux and Mac, ≥ 16 GB) and a fast multicore processor (e.g., Intel Core i7). [This forum discussion](#) may be of interest for the experimenter.

Activate the PTBIOP update site in FIJI to install the ABBA plugin. Then, install the QuPath Warpy extension (and its dependencies) and the QuPath ABBA extension (and its dependencies) by downloading the respective files, unzipping them, and dragging directly into the QuPath GUI. Restart QuPath and verify the installation by clicking in the extension/installed extensions menu. ABBA, Image Combiner Warpy, and ABBA should be listed.

Note: If the user requires to work with data from OMERO databases, install the QuPath OMERO RAW extension (and its dependencies) using [this](#) installation instructions.

ABBA uses elastix^{10,11} for automated registration. Download and install the latest release of elastix. Windows users require also Visual C++ redistributable according to their operative system.

When elastix is installed, open FIJI and run Plugins > BIOP > Set and Check Wrappers to indicate the location of elastix and Transformix executable files.

Note: Refer to <https://biop.github.io/ijp-imagetoatlas/> for further personalized instruction for ABBA installation in different platforms.

Finally, set up and download brain atlases in ABBA/FJI by selecting Plugins → BIOP → Atlas → ABBA - ABBA Start'. A window will allow the selection of the required brain atlas. We recommend to perform the alignment with the 'Adult Mouse Brain – Allen Brain Atlas V3p1'.

STEP-BY-STEP METHOD DETAILS

Image preprocessing

⌚ Timing: 5–10 min per image/unspecified for batch processing

In this step, the user performs image pre-processing to eliminate background and enhance cell features using ImageJ (Fiji).

1. The following ImageJ macro performs background subtraction, histogram equalization, conversion to 8-bit and apply the LUT.

```
> run("Subtract Background...", "rolling=50");

> run("Enhance Local Contrast (CLAHE)", "blocksize=127 histogram=256 maximum=3 mask=*None*
fast_(less_accurate)");

> run("8-bit");

> setMinAndMax(0, 80);

> run("Apply LUT");
```

Note: Perform the same steps for all the channels of interest. The user can carry out similar enhancing procedures using other selected software. A macro for batch processing of images located in a single directory can be found in our GitHub repository.

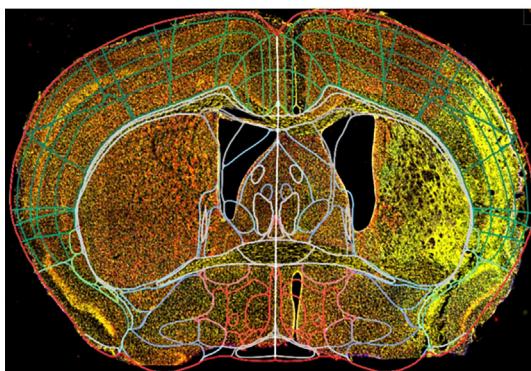
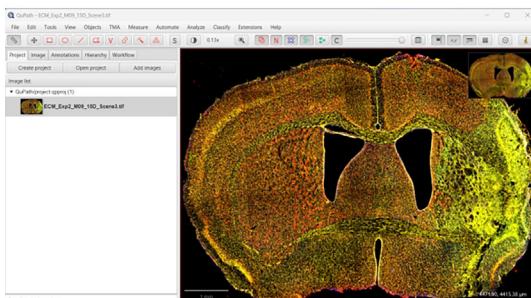
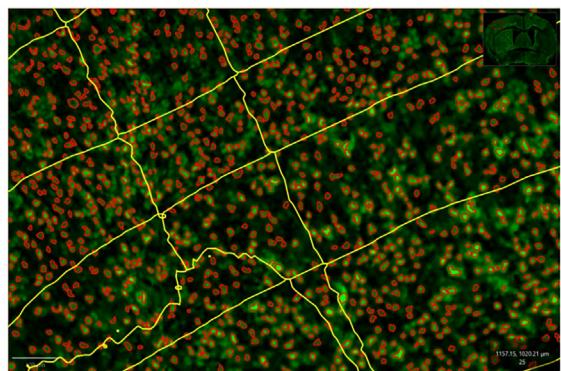
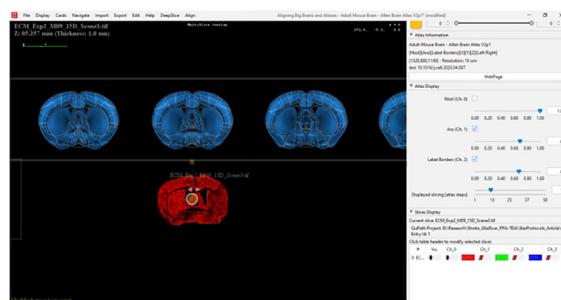
⚠ CRITICAL: Please note this should be adapted to the requirements and specific features of each image. Specifically, Enhance Local Contrast (CLAHE) performs histogram equalization that modifies pixel intensities. If the question of interest involves pixel intensity, this step should not be performed.

2. To execute the macro run Plugins → Macros → Interactive interpreter. Copy the macro in the window and click Macros → Run macro. Alternatively, run Plugins → Macros → Startup macros and create a new file under File → New. Please be sure that in the Language menu "ImageJ macro" is selected. Click the button Run to execute it.
3. Save the (multi-channel) image as a .tif file to preserve its quality and metadata.

⚠ CRITICAL: Make sure the image retains the metadata calibration in the metric system (microns, millimeters, centimeters), not pixels. Alignment to the Allen brain atlas requires a true reference scale.

4. Open the .tif file and use the standard 'clear' and 'transform' functions from Fiji to eliminate undesired objects in the image and make image transformations (such as flipping and rotation) to facilitate alignment to the Allen Brain atlas (see [Figure 1](#)).
5. Save the changes in the same image file.

Note: Define a suitable route and descriptive naming conventions at this step. The route and image names must be maintained to avoid routing problems in subsequent steps. We recommend that the naming convention contain the ID/grouping factors of the research project (if

A Create a QuPath project**C Import annotations (QuPath)****B Brain alignment (ABBA)****D Unbiased cell detection (QuPath)****Figure 1. Cell detection pipeline**

- (A) Create a project in QuPath and load the images of interest.
- (B) Export the project to ABBA and align the slices to the Allen brain atlas. Export the annotations to QuPath.
- (C) Import the aligned slices to QuPath and load the images annotations.
- (D) Run a cell detection strategy according to staining features.

suitable). For example: AnimalID_Group1_Group2_Condition1_Section#. These metadata will be extracted in downstream processing in R to build adequate data frames.

Create a project in QuPath

⌚ Timing: 5–10 min per project

In this step, the user employs QuPath v0.4.4 to create a project for alignment and registration to the Allen brain Atlas.

6. Open QuPath and create a new project in a destinated folder. All the registration files will be stored in this place (see [Figure 1A](#)). A project contains a single or several images per animal.

Note: If just a single brain slice is analyzed per brain, the project will contain only one image. Otherwise, place all the images corresponding to a single brain in the same project to be aligned and annotated. In the analysis pipeline, projects are processed independently.

7. Drag and drop the desired image/s to the main window.
8. In the emergent window, select Bio-Formats builder and then click import. Optionally, the users can specify the type of image (fluoresce, bright field, other) they are using.

Note: In the ‘Image’ tab, the user can verify if the image is properly calibrated in the metric system (microns, millimeters, centimeters), not pixels. If the image does not contain the

required calibration metadata, go back to the image preprocessing step, and save appropriate files. We strongly recommend to setup the scale in FIJI, given that setting the scale in QuPath is associated with individual projects and the information is not stored in the image metadata.

9. After the images are listed, the user can close or minimize the QuPath window and go to FIJI.

Alignment to the Allen brain atlas

⌚ Timing: 10–30 min per project

In this step, the user imports the QuPath project to perform alignment to the Allen brain atlas (see [Figure 1B](#)).

Note: Given the broad functionality of ABBA, we recommend to check the documentation and tutorials in <https://biop.github.io/ijp-imagetoatlas/>. Describing the full operation of ABBA is out of the scope of the present protocol.

10. Open ABBA by executing ‘ABBA start’ in FIJI, select the atlas to perform the alignment and the slicing mode (coronal, sagittal, horizontal).
11. In the main menu, execute Import → ABBA- Import QuPath Project and select the ‘project.qpproj’ file in the QuPath project folder.
12. The slices/s will be loaded. Use scroll up/down to zoom in/out in the ABBA window.
13. Perform manual or automatic alignment of the slices having as a reference the displayed brain atlas.
14. After the alignment satisfy the expectations, execute sequentially Align → ABBA- Elastix Registration (Affine) and Align → ABBA- Elastix Registration (Spline).

Note: Brains resulting from neurodegenerative or injury models are subjected to morphological alterations (i.e. shrinkage) and commonly requires manual adjustments. To correct/modify the annotations, use the Align → ABBA- BigWarp Registration as detailed in <https://biop.github.io/ijp-imagetoatlas/>. Please be aware of these limitations evaluating brains individually; excessive deformation or folding can lead to numerous false positives during the cell detection process, especially if quantification is performed in specific regions. The precision of the alignment must be determined by the experimenter in each case.

15. Execute File → ABBA- Save State to save the registration.
16. Finally, execute Export → ABBA- Export Registration To QuPath Project.
17. The files annotations are saved in the QuPath folder and ready for importing in QuPath.

Cell detection and quantification in QuPath

⌚ Timing: 10–20 min

This major step assumes the user has performed alignment and registration to the Allen brain atlas. However, if registration is no required (i.e., analysis of ROIs) the user can create a QuPath project and execute the cell detection and quantification steps detailed in this section.

18. Open or return to the QuPath project window and execute Extensions → ABBA → Load Atlas Annotations into Open Image.
19. The user must decide if the hemispheres should be split and then select the property to name the brain regions. This decision depends on the objective of the study: whole brain measurements or per hemisphere measurements.

20. The aligned image with annotated brain regions is displayed in the main QuPath window (see [Figure 1C](#)).

Note: If the user evidence important deviations in the alignment, return to the previous major step to fix the alignment and registration in ABBA.

21. Execute Analyze → Cell detection → Cell detection.
22. In the emerging window, select the channel of interest and set up the parameters for accurate cell detection. Make sure the ‘make measurements’ box is checked at the bottom.

Note: Please note that these parameters are highly dependent of the staining of interest. Therefore, no default is recommended. The user should test the combination of parameters that best fit the sample features. Hovering the cursor over each parameter shows its function. QuPath also have machine-learning based classifiers to categorize cells according to marker expression (i.e dead/live, reactive/non-reactive, etc.). Each cell class can be quantified and mapped independently. Please refer to <https://qupath.readthedocs.io/en/0.4/> to access related tutorials.

23. Execute ‘run’ and then click OK to process all image annotations (brain areas) (see [Figure 1D](#)).
24. Execute Measure → Export measurements to save the properties of the detected cells.

Note: Given the nature of common neuroscience approaches, this protocol provides a .groovy script (for QuPath) to perform individual or batch processing of images. In this same script, the user can define the brain regions of interest following the ontology and acronyms found [here](#). Also, please note that if alignment to the Allen brain atlas is not required, QuPath can be similarly used to perform automatic detections, cell counting, and the extraction of cell coordinates.

Clean the data sets for creating point patterns

⌚ Timing: 10–20 min

In this major step, the user subset the raw data obtained from QuPath to obtain relevant ID data and the cell coordinates. Please note the steps detailed here show the process for a single file (image). For batch processing of multiple files, we provide a completely annotated Quarto notebook as a guide.

25. Locate the .tsv cell “detections” files into the R working directory. Here, we operate with two files corresponding to GFAP and NeuN staining.

Note: QuPath—and the batch processing code provided in this protocol—generates two different files per brain. First, an “annotations” file containing a summary of the number of detected cells in each brain region of interest. Second, a “detections” file containing information about each cell, including xy coordinates. For PPA, just the “detections” files are of interest.

26. Load the file into the R environment.

```
> Gfap_Table <- read_tsv (file_route)  
> Gfap_Cells <- as.data.frame (Gfap_Table)  
> NeuN_Table <- read_tsv (file_route)  
> NeuN_Cells <- as.data.frame (NeuN_Table)
```

27. Handle the data to obtain the columns of interest. Use the same procedure for both markers. Please refer to the Quarto notebook for additional details. This step also obtains the metadata from the image name to set ID/grouping factors.

```
> Gfap_Cells <- Gfap_Cells %>%
  select(Image, ObjectID = Name, Region = Parent, Z = 'Allen CCFv3mm', Y = 'Allen CCFv3 Y mm', X = 'Allen CCFv3 Z mm') %>%
  sample_frac(.1) %>%
  separate(Image, into = c("NA1", "NA2", "MouseID", "DPI"), sep = "[_\\\\.]", extra = "drop", fill = "right") %>%
  select(-NA1, -NA2, MouseID, DPI, Region, ObjectID, -Z, X, Y)
```

28. Save the cleaned data frame for further purposes.

```
> write.csv(NeuN_Cells, "DataTables/NeuN_Cells.csv")
```

Create the point patterns

⌚ Timing: 10–20 min

In this major step, the user utilizes functions from the `spatstat` package⁷ to create scaled point patterns and define the observation window as a convex hull. For batch processing, we provide an annotated code to handle several files and save the result as an hyperframe to facilitate data handling.

29. Rotate the coordinates 180° for proper visualization in `spatstat`. Perform the same procedure for every marker.

```
> Gfap_Coords <- cbind(Gfap_Cells$X, Gfap_Cells$Y)
> Gfap_Coords <- secr::rotate(Gfap_Coords, 180)
> Gfap_Coords <- as.data.frame(Gfap_Coords)
> Gfap_Cells <- cbind(Gfap_Cells, Gfap_Coords)
```

30. Create the limits for the observation window using the desired limits.

```
> xlim <- range(NeuN_Cells$V1)
> ylim <- range(NeuN_Cells$V2)
```

Note: For this protocol, the observation window is delimited by NeuN staining. However, the user must choose the most suitable strategy considering the distribution of the markers and the research question. A recommended strategy is using a DAPI signal as an observation window, given that it covers the whole brain.

31. Create the point pattern using the *ppp* function from *spatstat*, specifying the limits of the observation window.

```
> Gfap_PPP <- ppp(  
  x = Gfap_Cells$V1,  
  y = Gfap_Cells$V2,  
  xrange = xlim,  
  yrange = ylim)
```

32. Rescale the point patterns using the original metadata from the image. In this case, 1.3 mm per 1000 pixels. Adjust for each case.

```
> unitname(NeuN_PPP) <- list ("mm", "mm", 1.3/1000)  
> NeuN_PPP <- spatstat.geom::rescale (NeuN_PPP)
```

33. Establish the observation window for the point pattern using the *convexhull* function.

```
> chull <- convexhull(Neurons_PPP)  
> Window(Cells_PPP) <- chull
```

Note: Establishing an observation window (convex hull) improves the estimation of the density and the calculations of the cell covariance. Therefore, the observation window should not be more extensive than the region/section of interest.

34. Use *plot(Neurons_PPP)* to visualize the point pattern (see [Figure 2B](#)).
35. The user can extract the mean intensity of the point patterns by executing:

```
> summary(NeuN_PPP)$intensity
```

Generate density kernels and tessellations

⌚ Timing: 5–10 min

This major step generates density kernels and tessellations using *spatstat* functions. These elements are useful to quantify and visualize the spatial intensity of cells and calculate cell covariance.

36. Generate density kernels for each point pattern with an adequate *sigma* parameter (see [Figure 2C](#)). Please refer to the Quarto notebook and the reference *spatstat* book for additional details.⁸

```
> NeuN_Density <- density(NeuN_PPP, sigma = 0.0002)
```

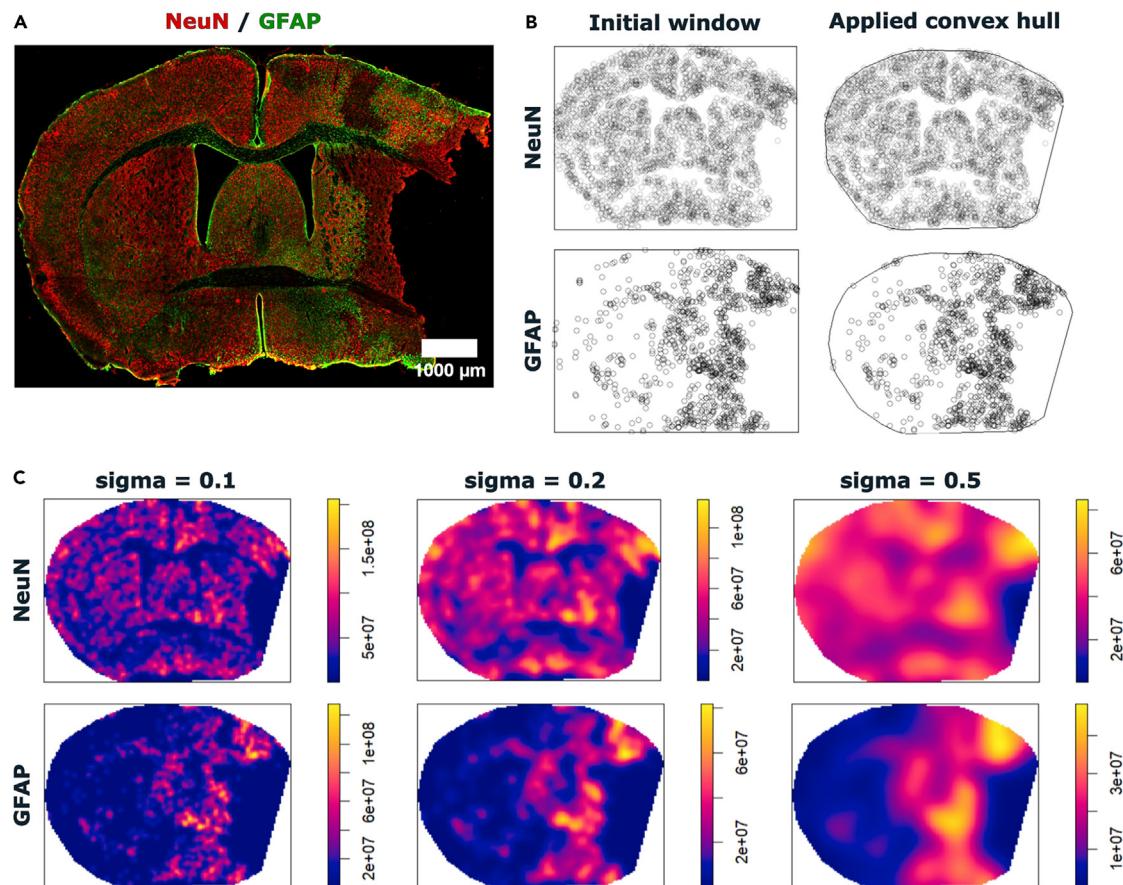


Figure 2. Creating point patterns and density kernels

(A) Staining of NeuN and GFAP intended for point patterns transformation.

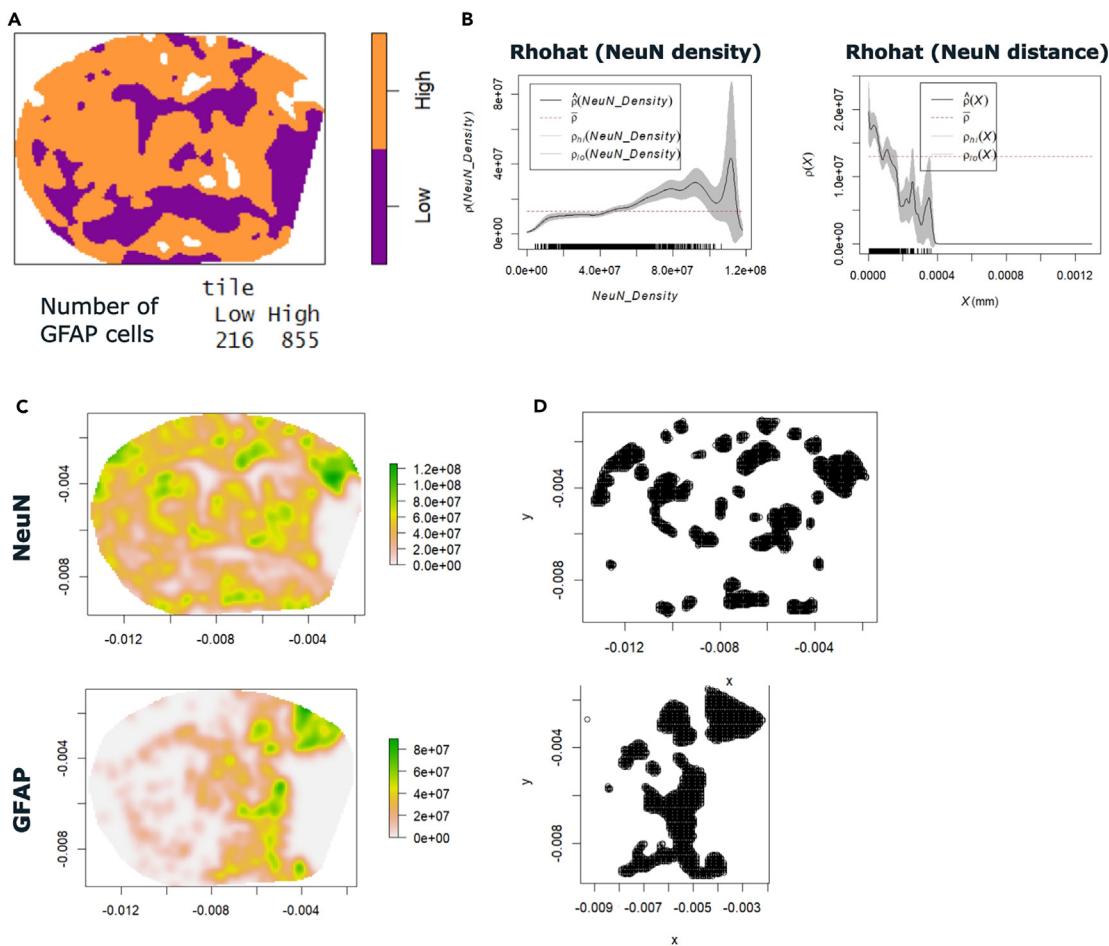
(B) The xy coordinates of detected cells in QuPath are transformed to point patterns using the *ppp* function from *spatstat*. A convex hull is applied to restrict the observation window.

(C) Different sigma levels from the *spatstat's* *density* function provide alternative mapping resolution (density kernels) for the cells of interest.

Note: The parameter sigma specifies the smoothing bandwidth to be used. A smaller value provides a more "granular" mapping. The user should define this value according to the characteristics of the sample and the research objectives. If the scope is a fine mapping of cell density, a smaller sigma will serve better this purpose. Otherwise, a bigger sigma averages the density values over larger regions to provide a mapping with less spatial resolution (see Figure 2C).

37. Define quantiles (based on the selected density kernel) to generate tessellated regions. For this protocol, we provide a tessellation for "Low" and "High" number of NeuN⁺ cells (see Figure 3A). It is at the user's discretion to select the ranges and number of breaks.

```
> Quantiles <- c(0, 30000000, 80000000)
> Cut <- cut(NetN_Density_02,
breaks = Quantiles,
labels = c("Low", "High"))
> NetN_Tess <- tess(image = Cut)
```

**Figure 3. Analyzing cell covariance and creating raster layers**

(A) Tessellation of NeuN cells showing regions of low and high neuronal numbers. The `quadratcount` function enables the user to estimate the number of covariate cells (in this case GFAP).

(B) `Rho-hat` functions enable the user to estimate the spatial covariance between the cells of interest at different spatial intensity and distance scales.

(C) The `raster` function produces a set of raster layers as a pixel image that allows the user to separate regions based on the spatial intensity of cells.

(D) The isolated regions can be analyzed independently.

38. Print the tessellation using `plot (NeuN_Tess)`.

Calculate cell covariance based on relative distribution

⌚ Timing: 5–10 min

This major step calculates the cell covariance by quadrant counts in defined tessellations and relative distribution using `spatstat` functions.

39. Use the `quadratcount` function to obtain the number of GFAP⁺ cells in regions of low and high neuronal intensity, as estimates by the density kernel and defined tessellation.

```
> quadratcount (Gfap_PPP, tess = NeuN_Tess)
```

Note: The result displays the number of covariant cells (GFAP) in the tessellated regions. For this example: Low 216 / High 855. This means that regions with high NeuN density have about four times more GFAP⁺ positive cells (see [Figure 3A](#)).

40. Run the `rhohat` function to calculate the relative distribution/covariance of two cell types based on their spatial intensity (see [Figure 3B](#)).

```
> Gfap_Rhohat <- rhohat(Gfap_PPP, NeuN_Density)
> plot(Gfap_Rhohat)
```

Note: The output is a graphical result displaying the intensity of interest in y-axis and the covariant (in this case NeuN) in the x-axis. This object can be transformed to a function using 'as.function' to obtain precise estimates at each value of the covariant. Please refer to the Quarto notebook and the reference *spatstat* book for additional details.⁸

41. A relative distribution based on the distance (distance map) to the covariate can be calculated using the parameter `distfun` (see [Figure 3B](#)).

```
> Gfap_Rhohat_D <- rhohat(Gfap_PPP, distfun(NeuN_PPP))
> plot(Gfap_Rhohat_D)
```

Note: This will generate a map to the nearest covariant that can be handled in the same way as `rhohat` estimations.

Modeling cell covariance

⌚ Timing: 5 min

This major step performs a linear Poisson modeling (regression) for the spatial intensity of the cells of interest using the `ppm` function from *spatstat*.

42. Use the `ppm` function run a linear Poisson regression for the cells of interest. The user can define the relevant covariates, including point patterns, density kernels, and distance maps. Please refer to the Quarto notebook and the reference *spatstat* book for additional details.⁸.

```
> Gfap_Modeling1 <- ppm(Gfap_PPP ~ 1)
> Gfap_Modeling2 <- ppm(Gfap_PPP ~ NeuN_Density)
```

Note: A model with no predictor (`Gfap_PPP ~ 1`) will estimate the spatial intensity and its variance in the same point pattern, whereas a model with predictors will estimate the spatial intensity of the cell of interest conditional on the specified factors/covariates.

Creating raster layers

⌚ Timing: 5 min

This major step creates raster layers using the `raster` package. This is a practical approach to isolate points of interest based on the density kernel (spatial intensity). The result is a matrix of pixels representing information.

43. Pass the '*raster*' function to the density kernels of interest.
44. Plot the raster layer to verify the density intervals (raster layers) (see [Figure 3C](#)).
45. Extract the points of interest (based on a threshold) using the '*rasterToPoints*' function.

```
> Gfap_Raster <- raster(Gfap_Density_02)
> plot(Gfap_Raster)
> Gfap_High <- rasterToPoints(Gfap_Raster, fun=function(x){x>30000000})
> plot(Gfap_High)
```

Note: With the extracted raster layers, the user can perform several procedures, including distance and centroid measurements (see [Figure 3D](#)).

Estimating the relation between points

Timing: 10 min

With this major step, the user can quantify the interaction between the time points to determine if the point pattern follows a random spatial distribution or exhibit aggregation, or inhibition. We evaluate this spatial correlation using the K-function.

46. To visualize the distance between points, use the '*distmap*' function (see [Figure 4A](#)).

```
> Gfap_Dist <- distmap(Gfap_PPP)
> Window(Gfap_Dist) <- chull
> plot(Gfap_Dist)
> plot(Gfap_PPP, add = TRUE, col = "white", pch = 18, cex= 0.4)
```

47. The '*pairdist*' and '*nndist*' functions return matrix of pairwise distances and a vector of nearest neighbor distances, respectively.

```
> Gfap_pairwise <- pairdist(Gfap_PPP)
> Gfap_nndist <- nndist(Gfap_PPP)
```

48. Use the *Kest* or *Kinhom* functions to calculate the K-function. For interpretation of this graphical results, please refer to the reference book.

```
> Gfap_Kest <- Kest(Gfap_PPP)
> Gfap_Kinhom <- Kinhom(Gfap_PPP)
```

Note: The user can pass additional parameters to the plot function to improve visualization. Please refer to the supplementary Quarto notebook.

EXPECTED OUTCOMES

This protocol will enable the user to obtain a comprehensive quantification of spatiotemporal cell distribution in any biological tissue. Furthermore, the protocol provides new approaches to estimate cell-to-cell covariance in an observational window, as well as calculate the interaction within the point pattern. The results can be saved as R objects or data frames that can be shared in any

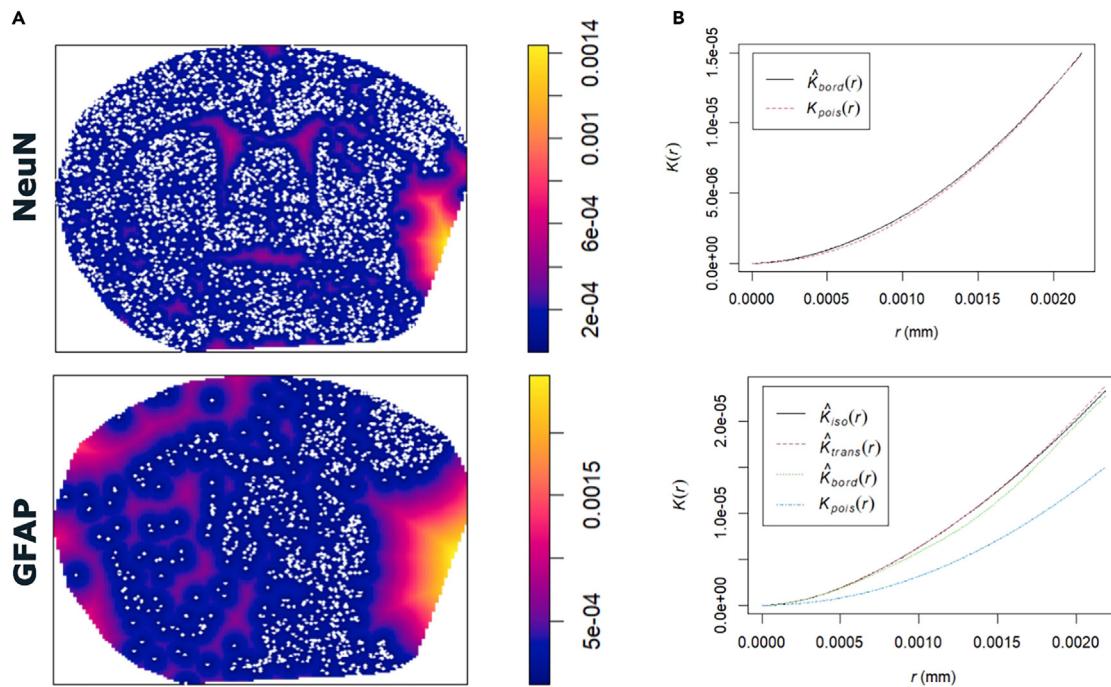


Figure 4. Analyzing the interaction within points

(A) Distance functions for NeuN and GFAP map the distance between cells.

(B) The *Kinhom* function from *spatstat* enables the user to identify random, clustering, or inhibitory patterns in a given observation window.

repository. Complete annotated codes for batch processing are available in the GitHub repository (<https://doi.org/10.5281/zenodo.10805534>).

QUANTIFICATION AND STATISTICAL ANALYSIS

We strongly recommend using statistical modeling based on parameter estimation and uncertainty to analyze grouped data. Complete annotated codes for this purpose are available in the GitHub repository (<https://doi.org/10.5281/zenodo.10805534>). Please note that 'rhohat' calculations yield graphical summaries that must be interpreted using context and scientific knowledge of the phenomena. Otherwise, the point process models (ppm) in 'spatstat' are modeled as a Poisson linear regression on the logarithmic scale. The resulting intercept, slopes and interactions should be interpreted as regression coefficients. The reference book for *spatstat* contains a comprehensive explanation on this regard.⁸ We recommend taking into consideration the hierarchy of the data for modeling and interpretation of the results.

LIMITATIONS

The accuracy of the point pattern analysis is highly limited by the precision in cell detection and quantification, and the establishment of a defined observation window.

TROUBLESHOOTING

Problem 1

Errors importing/exporting the results from ABBA.

Potential solution

Update ABBA and QuPath and check <https://biop.github.io/ijp-imagetoatlas/> to retrieve additional changes in the functionality.

Problem 2

Errors executing functions from *spatstat*.

Potential solution

Update the package or check for potential issues <https://github.com/spatstat/spatstat>.

RESOURCE AVAILABILITY

Lead contact

Dr. Ayman ElAli (ayman.elali@crchudequebec.ulaval.ca).

Technical contact

Dr. Daniel Manrique-Castano (damac36@ulaval.ca).

Materials availability

This protocol does not contain unique materials or reagents.

Data and code availability

We provide complete analysis pipelines and supplementary documentation in our GitHub repository: (<https://doi.org/10.5281/zenodo.10805534>).

ACKNOWLEDGMENTS

This work was supported by grants from the Canadian Institutes of Health Research (CIHR) (#169062; #186148) (all to A.E.A.). D.M.-C. is the recipient of a postdoctoral fellowship from the *Fonds de recherche du Québec - Santé* (FRQS, 318466). A.E.A. holds a Tier 2 Canada Research Chair in molecular and cellular neurovascular interactions. We thank the platform of High-Speed Image Analysis at the research center of CHU de Québec-Université Laval.

AUTHOR CONTRIBUTIONS

Conceptualization, D.M.-C. and A.E.A.; methodology, D.M.-C.; software, D.M.-C.; validation, D.M.-C.; formal analysis, D.M.-C.; investigation, D.M.-C.; resources, A.E.A.; data curation, D.M.-C.; writing – original draft, D.M.-C.; writing – review and editing and visualization, D.M.-C.; supervision, A.E.A.; project administration, A.E.A.; funding acquisition, A.E.A.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Manrique-Castano, D., Bhaskar, D., and ElAli, A. (2023). Dissecting glial scar formation by spatial point pattern and topological data analysis. *bioRxiv* 49, D1029. <https://doi.org/10.1101/2023.10.04.560910>.
2. Song, L., Pan, S., Zhang, Z., Jia, L., Chen, W.-H., and Zhao, X.-M. (2021). STAB: a spatio-temporal cell atlas of the human brain. *Nucleic Acids Res.* 49, D1029–D1037. <https://doi.org/10.1093/nar/gkaa762>.
3. Manrique-Castano, D., and ElAli, A. (2021). Neurovascular reactivity in tissue scarring following cerebral ischemia. In *Cerebral Ischemia*, R. Pluta, ed. (Exon Publications). <https://doi.org/10.36255/exonpublications.cerebralischemia.2021.neurovascularreactivity>.
4. Adams, K.L., and Gallo, V. (2018). The diversity and disparity of the glial scar. *Nat. Neurosci.* 21, 9–15. <https://doi.org/10.1038/s41593-017-0033-9>.
5. Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al. (2012). Fiji: an open-source platform for biological-image analysis. *Nat. Methods* 9, 676–682. <https://doi.org/10.1038/nmeth.2019>.
6. Bankhead, P., Loughrey, M.B., Fernández, J.A., Dombrowski, Y., McArt, D.G., Dunne, P.D., McQuaid, S., Gray, R.T., Murray, L.J., Coleman, H.G., et al. (2017). QuPath: Open source software for digital pathology image analysis. *Sci. Rep.* 7, 16878. <https://doi.org/10.1038/s41598-017-17204-5>.
7. Baddeley, A., and Turner, R. (2005). spatstat: an R package for analyzing spatial point patterns. *J. Stat. Softw.* 12. <https://doi.org/10.1863/jss.v012.i06>.
8. Baddeley, A., Rubak, E., and Turner, R. (2005). *Spatial Point Patterns: Methodology and Applications with R* (Boca Raton: Chapman and Hall/CRC).
9. Jean-LeBlanc, N., Menet, R., Picard, K., Parent, G., Tremblay, M.-É., and ElAli, A. (2019). Canonical Wnt Pathway Maintains Blood-Brain Barrier Integrity upon Ischemic Stroke and Its Activation Ameliorates Tissue Plasminogen Activator Therapy. *Mol. Neurobiol.* 56, 6521–6538. <https://doi.org/10.1007/s12035-019-1539-9>.
10. Klein, S., Staring, M., Murphy, K., Viergever, M.A., and Pluim, J.P.W. (2010). elastix: a toolbox for intensity-based medical image registration. *IEEE Trans. Med. Imaging* 29, 196–205. <https://doi.org/10.1109/TMI.2009.2035616>.
11. Shamonin, D.P., Bron, E.E., Lelieveldt, B.P.F., Smits, M., Klein, S., and Staring, M.; Alzheimer's Disease Neuroimaging Initiative (2013). Fast parallel image registration on CPU and GPU for diagnostic classification of Alzheimer's disease. *Front. Neuroinform.* 7, 50. <https://doi.org/10.3389/fninf.2013.00050>.