



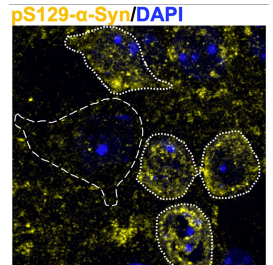
Jan 10, 2026

Version 1

A CellProfiler Pipeline for Quantification of p-SNCA in Mouse Striatal Cholinergic and Medium Spiny Neurons V.1

DOI

<https://dx.doi.org/10.17504/protocols.io.261ge13bwv47/v1>



Aaran Vijayakumaran^{1,2}, Yu-En Lin^{1,2}, Suzanne Pfeffer^{1,2}

¹Stanford University School of Medicine; ²Aligning Science Across Parkinson's



Suzanne R Pfeffer

Stanford University School of Medicine

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.261ge13bwv47/v1>

Protocol Citation: Aaran Vijayakumaran, Yu-En Lin, Suzanne Pfeffer 2026. A CellProfiler Pipeline for Quantification of p-SNCA in Mouse Striatal Cholinergic and Medium Spiny Neurons. **protocols.io**

<https://dx.doi.org/10.17504/protocols.io.261ge13bwv47/v1>

License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working

Created: January 09, 2026

Last Modified: January 10, 2026

Protocol Integer ID: 238354

Keywords: synuclein pathology from fluorescence imaging dataset, striatum of synuclein g51d, synuclein pathology, synuclein g51d, snca in mouse striatal cholinergic, mouse striatal cholinergic, synuclein, cholinergic neuron, cellprofiler pipeline for quantification, medium spiny neuron, medium spiny neurons this protocol, fluorescence imaging dataset, choline acetyltransferase, cell type specific psnca burden per unit area, cholinergic, cellprofiler pipeline, neuron, neuronal mask area, using cellprofiler, cellprofiler, mutant mice, brain section

Funders Acknowledgements:

Aligning Science Across Parkinson's

Grant ID: ASAP-000463

Abstract

This protocol describes an automated image analysis workflow using CellProfiler (1) to quantify Ser129 phosphorylated α -synuclein (pSNCA) within Cholinergic and Medium Spiny Neurons (MSNs) in the striatum of Synuclein G51D mutant mice (2). Brain sections are immunostained for Ser129 phospho- α -synuclein, DARPP-32 to identify MSNs, choline acetyltransferase (ChAT) to label cholinergic neurons, and DAPI to identify nuclei. Confocal Z-stacks are processed into consecutive three-slice maximum-intensity projections (MIPs) using a custom FIJI/ImageJ macro prior to segmentation and analysis. Quantitative outputs include integrated pSNCA intensity and neuronal mask area, which are normalised using a custom R script to yield cell type specific pSNCA burden per unit area. This workflow enables cell type resolved quantification of α -synuclein pathology from fluorescence imaging datasets.

(1) Stirling DR, Swain-Bowden MJ, Lucas AM, Carpenter AE, Cimini BA, Goodman A (2021). CellProfiler 4:improvements in speed, utility and usability. BMC Bioinformatics, 22 (1), 433.. PMID: 34507520 PMCID: PMC8431850.

(2) Kim Y, McInnes J, Kim J, Liang YHW, Veeraragavan S, Garza AR, Belfort BDW, Arenkiel B, Samaco R, Zoghbi HY. Olfactory deficit and gastrointestinal dysfunction precede motor abnormalities in alpha-Synuclein G51D knock-in mice. Proc Natl Acad Sci U S A. 2024 Sep 24;121(39):e2406479121. doi: 10.1073/pnas.2406479121.

Materials

Mouse brain immuno-staining was carried out as described (dx.doi.org/10.17504/protocols.io.bnwimfce) using anti-Ser129 phospho-synuclein antibody (1:1000 from Cell Signaling Technology #87281, RRID: AB_279886), anti-DARPP32 (1:1000 from Cell Signaling Technology #2306S, RRID: AB_A23479), anti-ChAT (1:200 from Millipore #AB144P-1ML, RRID: Ab_2079751).

Batch process images and import files into CellProfiler

- 1 This protocol describes an automated **image analysis** workflow using CellProfiler to quantify Ser129-phosphorylated α -synuclein (pSNCA) within striatal Cholinergic and Medium Spiny Neurons (MSNs) in Synuclein G51D mutant mouse brain sections, focusing on the dorsolateral striatum.
 - 1.1 Use the FIJI/ImageJ preprocessing workflow described in [this related protocol](#) to generate three-slice maximum-intensity projections (MIPs) across the Z-stack. This workflow outputs one folder per image dataset, with each folder containing channel-specific MIP images generated from consecutive three-slice groups, suitable for direct import into CellProfiler.
 - 1.2 Open CellProfiler. Go to the Images module, drag and drop the folders containing the three-slice MIPs projected TIFF files as indicated. Select “no filtering” in the filter images option.
 - 1.3 Go to the Metadata module to extract information describing the images.
 - a. In the Metadata module, choose Yes for Extract metadata
 - b. Metadata extraction method: Extract from file/folder names
 - c. Metadata source: File name
 - d. Extract metadata from: All images
 - e. In Regular expression to extract from file name, enter: `‘MIP_(?P<startSlice>[0-9]+)_(?P<channel>[A-Za-z0-9]+)\.tif$’`. This regular expression extracts the starting Z-slice of each three-slice MIP (startSlice) and the channel identity (channel) from filenames such as MIP_13_pSNCA.tif. Click the magnifying glass icon at the right to check the accuracy of regular expression
 - f. Set Metadata data type to Text
 - g. Click Update to populate the metadata fields and verify correct extraction
 - 1.4 Go to the NamesAndTypes module to indicate a name for each channel.
 - a. Assign name to: Images matching rules
 - b. Process as 3D: No
 - c. Match “All” of the following rules
 - d. Select the rule criteria: “File/Does/Contain/DARPP32”
 - e. Name to assign these images: DARPP32
 - f. Select the image type: Grayscale image
 - g. Set intensity range from: Image metadata
 - h. Click on Add another image
 - i. Repeat for the other channels (pSNCA, DAPI and CHAT)
 - j. Click on “update” to populate the names and types field
 - 1.5 Go to the Groups module and choose No for Do you want to group your images?

Before analysis, only 3-slice MIPs containing detectable signal for Cholinergic and Medium Spiny Neurons within each field of view are included in downstream analysis.

Segment Medium Spiny Neuron (MSN) Objects

- 2 Next steps help identify the Medium Spiny Neurons stained with DARPP32 and therefore, represent the 'DARPP32' channel.
- 2.1
 - a. Click on the "+" sign at the bottom left next to Adjust Modules.
Add IdentifyPrimaryObjects module to the pipeline.
 - b. Use advanced settings? Yes
 - c. Select the input image: DARPP32
 - d. Name the primary objects to be identified: MSNObjects
 - e. Typical diameter of objects, in pixel units (Min, Max): 10,70
 - f. Discard objects outside the diameter range: Yes
 - g. Discard objects touching the border of the image: Yes
 - h. Threshold strategy: Global
 - i. Thresholding method: Robust Background
 - j. Lower outlier fraction: 0.05
 - k. Upper outlier fraction: 0.05
 - l. Averaging method: Mean
 - m. Variance method: Standard deviation
 - n. # of deviations: 1.0
 - o. Threshold smoothing scale: 1.3488
 - p. Threshold correction factor: 1.0
 - q. Lower and upper bounds on threshold 0.05, 1.0
 - r. Method to distinguish clumped objects: Intensity
 - s. Method to draw dividing lines between clumped objects: Intensity
 - t. Automatically calculate size of smoothing filter for declumping: Yes
 - u. Automatically calculate minimum allowed distance between local maxima: Yes
 - v. Speed up by using lower-resolution image to find local maxima: Yes
 - w. Display accepted local maxima: No
 - x. Fill holes in identified objects? After both thresholding and declumping
 - y. Handling of objects if excessive number of objects identified: Continue

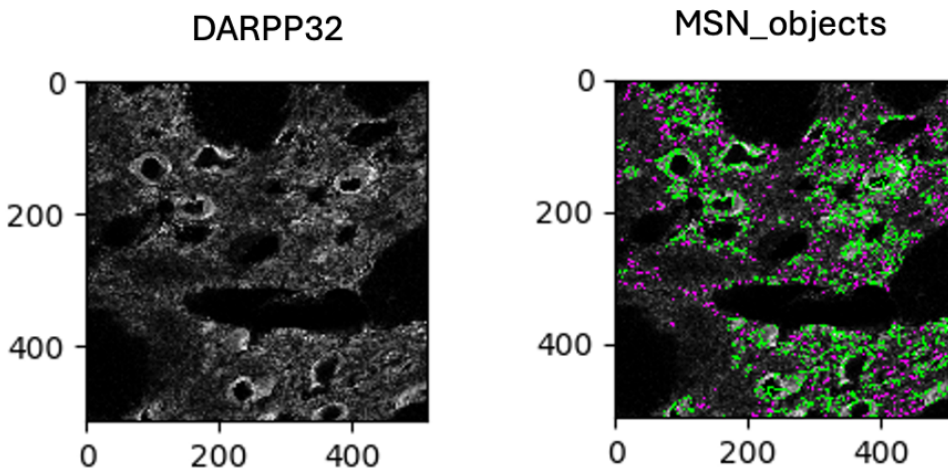


Figure 1: Example of input image of DARPP32 staining of medium spiny neurons (at left) and segmented MSN objects with background (at right). Green outlines represent valid objects; magenta denotes regions not segmented at this stage.

3 This step enhances the intensity of the DAPI-stained nuclei to improve segmentation accuracy.

- 3.1
- Add ImageMath module to the pipeline
 - Operation: Add
 - Name the output image: DAPI_Enhanced
 - Image or measurement? Image
 - Select the first image: DAPI
 - Multiple the first image by 5.0
 - Image or measurement? Image
 - Select the first image: DAPI
 - Multiple the first image by 1.0
 - Raise the power of the result by: 1.0
 - Multiply the result by: 1.0
 - Add 0 result: 0.0
 - Set values less than 0 equal to 0? Yes
 - Set values greater than 1 equal to 1? Yes
 - Replace invalid values with 0? Yes
 - Ignore the image masks? No

4 This step segments the enhanced, DAPI-stained nuclei

- 4.1
- Add IdentifyPrimaryObjects module to the pipeline
 - Use advanced settings? Yes

- c. Select the input image: DAPI
- d. Name the primary objects to be identified: Nucleiobjects
- e. Typical diameter of objects, in pixel units (Min, Max): 25,400
- f. Discard objects outside the diameter range: Yes
- g. Discard objects touching the border of the image: Yes
- h. Threshold strategy: Global
- i. Thresholding method: Robust Background
- j. Lower outlier fraction: 0.05
- k. Upper outlier fraction: 0.05
- l. Averaging method: Mean
- m. Variance method: Standard deviation
- n. # of deviations: 2.0
- o. Threshold smoothing scale: 1.3488
- p. Threshold correction factor: 1.0
- q. Lower and upper bounds on threshold 0.1, 1.0
- r. Method to distinguish clumped objects: Intensity
- s. Method to draw dividing lines between clumped objects: Intensity
- t. Automatically calculate size of smoothing filter for declumping: Yes
- u. Automatically calculate minimum allowed distance between local maxima: Yes
- v. Speed up by using lower-resolution image to find local maxima: No
- w. Display accepted local maxima: No
- x. Fill holes in identified objects? After both thresholding and declumping
- y. Handling of objects if excessive number of objects identified: Continue

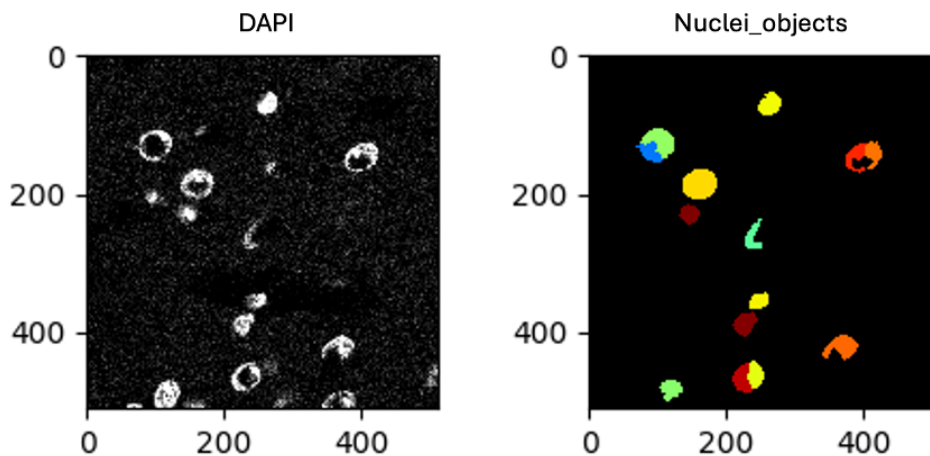


Figure 2: Example of input image of Nuclei stained with DAPI (at left) and segmented Nuclei_objects (at right). Partial nuclear structures are maintained in this analysis.

- 5 The next steps allow MSN objects to be reported at the single-cell level; nuclear segmentation provides a stable, continuous cellular reference that enables fragmented MSN signal to be grouped and merged on a per-cell basis.
- 5.1
 - a. Add RelateObjects module to the pipeline
 - b. Parent objects: Nucleiobjects
 - c. Child objects: MSNobjects
 - d. Calculate per-parent means for all child measurements? No
 - e. Calculate child-parent distances? None
 - f. Do you want to save the children with parents as a new object set? Yes
 - g. Name the output object: MSN_related_to_DAPI
- 6 The next step merges fragmented MSN objects to ensure a single MSN object is identified per nucleus and cell.
- 6.1
 - a. Add SplitorMergeObjects module to the pipeline
 - b. Select the input objects: MSNobjects
 - c. Name the new objects: MSN_per_DAPI
 - d. Operation: Merge
 - e. Merging method: Per-parent
 - f. Output object type: Disconnected
 - g. Select the parent object: DAPI_merged
- 7 Remove filled objects that touch the image border
- 7.1
 - a. Add FilterObjects module to the pipeline
 - b. Select the objects to filter: MSN_per_DAPI
 - c. Name the output objects: FilterObjects
 - d. Select the filtering mode: Image or mask border
 - e. Keep removed objects as a separate set: No

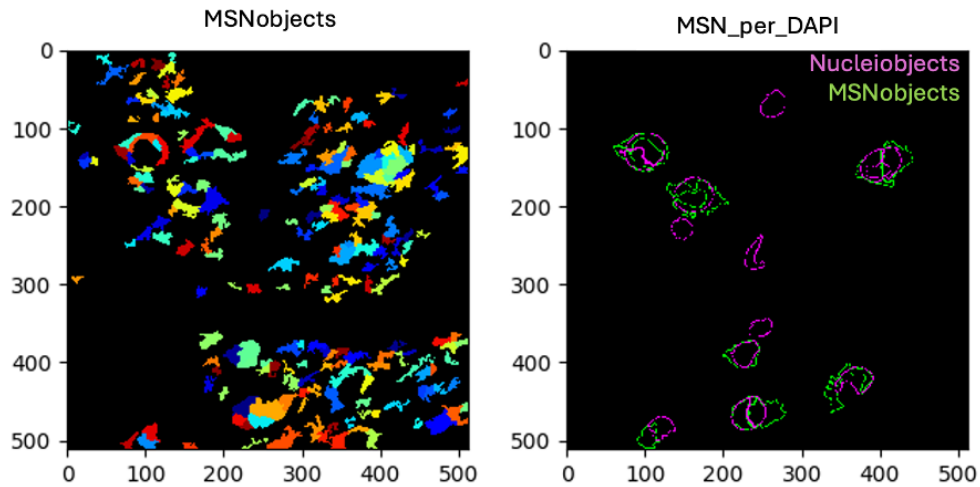


Figure 3: Initial noisy MSN segmentation is shown (left). After relating objects to nuclei (purple), final MSN objects are filtered to retain nuclei-associated objects, shown in green (right). Note that this underestimates MSNs in the images but helps to ensure that labeling is on an MSN.

Segment phospho-SNCA Objects

- 8 Next steps help enhance signal intensity to identify phospho-SNCA for downstream segmentation analysis.
 - 8.1
 - a. Add RescaleIntensity module to the pipeline
 - b. Select the input image: pSNCA
 - c. Name the output image: pSNCA_rescaled
 - d. Rescaling method: Stretch each image to use the full intensity range
 - 8.2 Add the EnhanceOrSupressFeature module
 - a. Select the input image: pSNCA_rescaled
 - b. Name the output image: pSNCA_enhanced
 - c. Select the operation: Enhance
 - d. Feature type: Speckles
 - e. Feature size: 15
 - f. Speed and accuracy: Slow
- 9 Identify phosphoSynuclein
 - 9.1
 - a. Add IdentifyPrimaryObjects module to the pipeline
 - b. Use advanced settings? Yes
 - c. Select the input image: pSNCA_enhanced

- d. Name the primary objects to be identified: pSNCAobjects
- e. Typical diameter of objects, in pixel units (Min, Max): 1,50
- f. Discard objects outside the diameter range: Yes
- g. Discard objects touching the border of the image: Yes
- h. Threshold strategy: Global
- i. Thresholding method: Otsu
- j. Two-class or three-class thresholding? Two classes
- k. Threshold smoothing scale: 4.0
- l. Threshold correction factor: 4.0
- m. Lower and upper bounds on threshold 0.1, 1.0
- n. Method to distinguish clumped objects: Intensity
- o. Method to draw dividing lines between clumped objects: Intensity
- p. Automatically calculate size of smoothing filter for declumping: Yes
- q. Automatically calculate minimum allowed distance between local maxima: Yes
- r. Speed up by using lower-resolution image to find local maxima: Yes
- s. Display accepted local maxima: No
- t. Fill holes in identified objects? After both thresholding and declumping
- u. Handling of objects if excessive number of objects identified: Continue

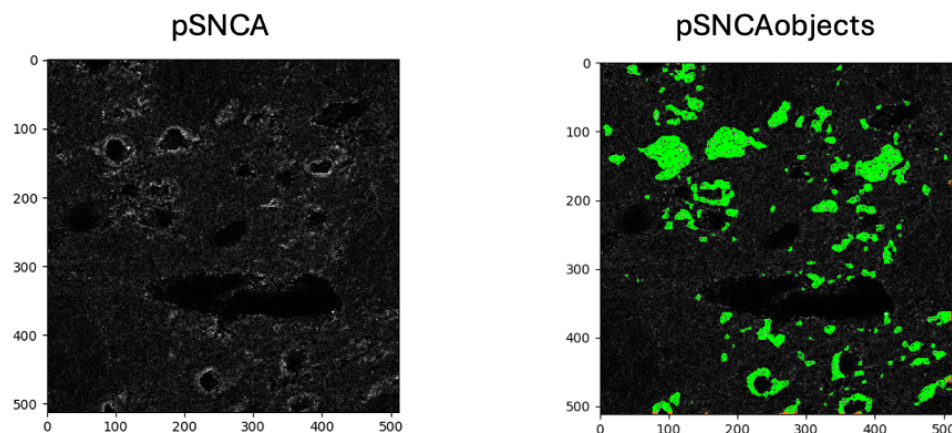


Figure 4: Example of input image of pSNCA (at left) and segmented pSNCAobjects (at right). Green structures are considered valid objects; pSNCA intensity is later determined within these areas.

Segment pSNCA only within MSNs

- 10 To measure pSNCA within MSNs, the RelateObjects module is used to associate p-SNCA objects with MSN cell masks, thereby excluding background segmentation and restricting p-SNCA measurements to signal located within MSNs.

- 10.1
 - a. Add the RelateObjects module to the pipeline
 - b. Parent objects: MSNObjects_filled
 - c. Child objects: pSNCAobjects
 - d. Calculate per-parent means for all child measurements? No
 - e. Calculate child-parent distances: None
 - f. Do you want to save the children with parents as a new object set? No
- 11 Filter p-SNCA objects to retain only those located within MSN cell masks.
 - 11.1
 - a. Add FilterObjects module to the pipeline
 - b. Select the objects to filter: pSNCAobjects
 - c. Name of the output objects: pSNCA_within_MSNs
 - d. Select the filtering mode: Measurements
 - e. Select the filtering method: Limits
 - f. Category: Parent
 - g. Measurement: MSNObjects_filled
 - h. Filter using a minimum measurement value: Yes
 - i. Minimum value: 0.5
 - j. Filter using a maximum measurement value: No
 - k. Keep removed objects as a separate set: No

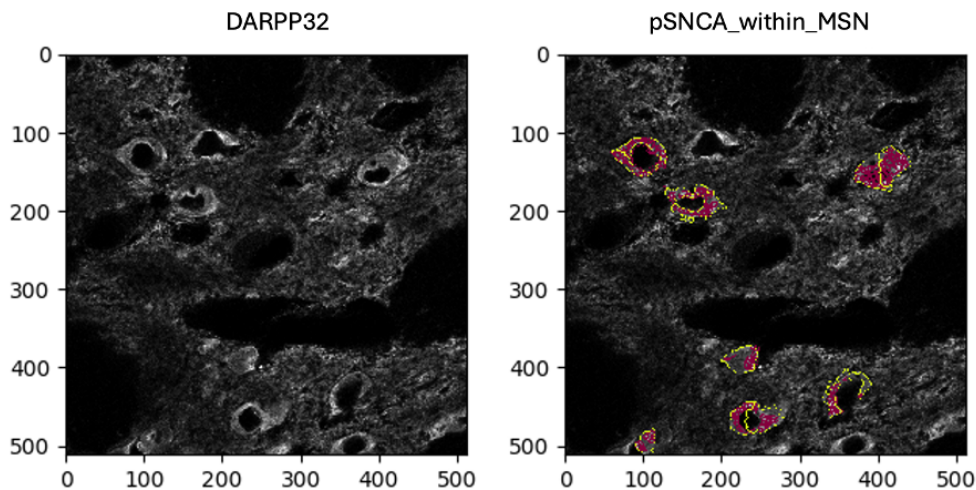


Figure 5. Example DARPP32 input image (left) with segmentation outlines of MSNs (yellow) and pSNCA objects (dark red) overlaid on the same image at right.

Segment Cholinergic Neuron (CHAT) Objects



12 This step enhances the intensity of the CHAT-stained Cholinergic neurons to improve segmentation accuracy.

- 12.1
- Add ImageMath module to the pipeline
 - Operation: Add
 - Name the output image: CHAT_Enhanced
 - Image or measurement? Image
 - Select the first image: CHAT
 - Multiple the first image by 5.0
 - Image or measurement? Image
 - Select the first image: CHAT
 - Multiple the first image by 5.0
 - Raise the power of the result by: 1.0
 - Multiply the result by: 1.0
 - Add 0 result: 0.0
 - Set values less than 0 equal to 0? Yes
 - Set values greater than 1 equal to 1? Yes
 - Replace invalid values with 0? Yes
 - Ignore the image masks? No

13 This step helps segment the enhanced Cholinergic Neurons stained with anti-ChAT antibodies

- 13.1
- Add IdentifyPrimaryObjects module to the pipeline.
 - Use advanced settings? Yes
 - Select the input image: CHAT_Enhanced
 - Name the primary objects to be identified: CHATObjects
 - Typical diameter of objects, in pixel units (Min, Max): 40,400
 - Discard objects outside the diameter range: Yes
 - Discard objects touching the border of the image: No
 - Threshold strategy: Global
 - Thresholding method: Robust Background
 - Lower outlier fraction: 0.05
 - Upper outlier fraction: 0.05
 - Averaging method: Mean
 - Variance method: Standard deviation
 - # of deviations: 2.0
 - Threshold smoothing scale: 1.5
 - Threshold correction factor: 1.0
 - Lower and upper bounds on threshold 0.0, 1.0
 - Method to distinguish clumped objects: Intensity
 - Method to draw dividing lines between clumped objects: Intensity

- t. Automatically calculate size of smoothing filter for declumping: Yes
- u. Automatically calculate minimum allowed distance between local maxima: Yes
- v. Speed up by using lower-resolution image to find local maxima: Yes
- w. Display accepted local maxima: No
- x. Fill holes in identified objects? After both thresholding and declumping
- y. Handling of objects if excessive number of objects identified: Continue

14 This step merges fragmented ChAT segmentations to ensure one ChAT-defined cell per object

- 14.1
- a. Add SplitorMergeObjects module to the pipeline
 - b. Select the input objects: CHATObjects
 - c. Name of the new objects: CHAT_merged
 - d. Operation: Merge
 - e. Merging method: Distance
 - f. Maximum distance within which to merge objects: 20
 - g. Merge using a grayscale image? No

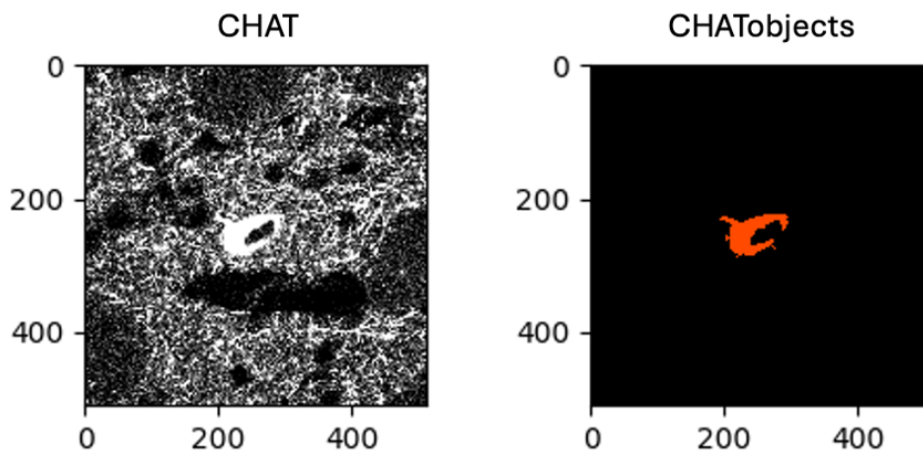


Figure 6: Example of input CHAT image after image enhancement (at left) and segmented CHATObjects (at right).

Segment pSNCA only within Cholinergic Neurons

15 To measure pSNCA within CHAT neurons, the RelateObjects module is used to associate p-SNCA objects with CHAT cell masks, thereby excluding background segmentation and restricting p-SNCA measurements to signal located within MSNs.

- 15.1
- a. Add the RelateObjects module to the pipeline
 - b. Parent objects: CHAT_merged
 - c. Child objects: pSNCAObjects

- d. Calculate per-parent means for all child measurements? No
- e. Calculate child-parent distances: None
- f. Do you want to save the children with parents as a new object set? No

16 Filter pSNCA objects to retain only those located within CHAT cell masks

- 16.1
- a. Add FilterObjects module to the pipeline
 - b. Select the objects to filter: pSNCAobjects
 - c. Name of the output objects: pSNCA_within_CHAT
 - d. Select the filtering mode: Measurements
 - e. Select the filtering method: Limits
 - f. Category: Parent
 - g. Measurement: CHATobjects
 - h. Filter using a minimum measurement value: Yes
 - i. Minimum value: 0.5
 - j. Filter using a maximum measurement value: No
 - k. Keep removed objects as a separate set: No

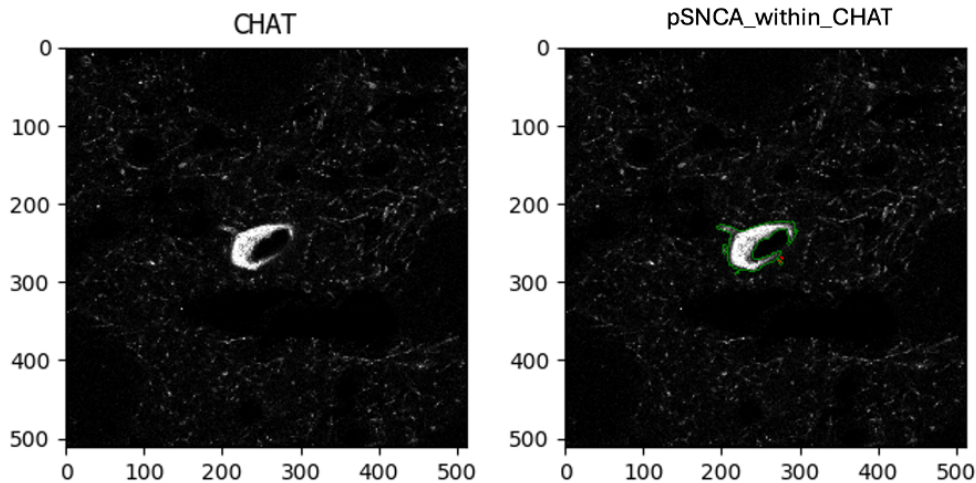


Figure 7. Example CHAT input image (left) with segmentation outlines for CHAT (green) and pSNCA objects (red) overlaid on the same image at right.

Measure Intensity and Object Size Shape

- 17 This step is used to measure the intensity of pSNCA within Cholinergic and Medium Spiny Neurons.



- 17.1
 - a. Add MeasureObjectIntensity module to the pipeline
 - b. Select images to measure: pSNCA
 - c. Select objects to measure:
pSNCA_within_CHAT
pSNCA_within_MSNs
- 18 This step is used to measure the object area of the Cholinergic and Medium Spiny Neuron masks.
- 18.1
 - a. Add MeasureObjectSize module to the pipeline
 - b. Select object sets to measure:
CHAT_merged
MSN_per_DAPI
 - c. Calculate the Zernike feature: Yes
 - d. Calculate the advanced features: No
- 19 Add ExportToSpreadsheet module from the + at the bottom to export the measurements into files that can be opened in Excel:
- 19.1
 - a. Select the column delimiter: Comma
 - b. Output file location: choose a folder where images will be saved
 - c. Add prefix to file name: Yes
 - d. File name prefix: Add experiment identifier
 - e. Overwrite existing files without warning? No
 - f. Add image metadata columns to your object data file? Yes
 - g. Add image file and folder names to your object data file? Yes
 - h. Representation of Nan/Inf: NaN
 - i. Select the measurements to export? Yes
 - j. Click Press button to select measurements: Yes
 - k. Calculate the per-image mean values for object measurements? No
 - l. Calculate the per-image median values for object measurements? No
 - m. Calculate per-image standard deviation values for object measurements? No
 - n. Create GenePattern GCT file? No
 - o. Export all measurement types? Yes
- 20 Downstream quantitative analysis is performed using a custom R script (see below), in which integrated pSNCA fluorescence intensities is summed per dataset separately for ChAT-positive and DARPP32-positive neurons and normalized to the corresponding total segmented cell mask area, yielding cell type-specific pSNCA intensity per unit area.
- 20.1 **R Script Used:**

library(tidyverse)

```
library(readxl)
library(janitor)
library(stringr)

# FILE PATHS (edit these)
chat_intensity_file <- "~/Desktop/CHAT_N_3_2026.xlsx"      # has
intensity_integrated_intensity_snca
chat_area_file <- "~/Desktop/CHAT_N_3_area_2026.xlsx"    # has areashape_area (or
AreaShape_Area)

msn_intensity_file <- "~/Desktop/MSN_N_3_2026.xlsx"
msn_area_file <- "~/Desktop/MSN_N_3_area_2026.xlsx"

# HELPERS
extract_dataset_id <- function(x) {
  str_extract(x, "(?<=/For_analysis/)\d+")
}

# Robustly find the area column name after clean_names()
# (it will usually become "area_shape_area")
get_area_col <- function(df) {
  nm <- names(df)
  if ("area_shape_area" %in% nm) return("area_shape_area")
  if ("areashape_area" %in% nm) return("areashape_area")
  stop("Couldn't find an area column. Expected 'area_shape_area' or 'areashape_area'
after clean_names().")
}

# FUNCTION: build per-dataset table with total_intensity, total_area, intensity_per_area
summarise_intensity_and_area <- function(intensity_path, area_path, cell_type_label) {

  # ---- intensity ----
  df_int <- read_excel(intensity_path) %>%
    clean_names() %>%
    mutate(dataset_id = extract_dataset_id(path_name_snca)) %>%
    filter(!is.na(dataset_id)) %>%
    mutate(dataset_id = as.integer(dataset_id))

  intensity_by_dataset <- df_int %>%
    group_by(dataset_id) %>%
    summarise(
      total_intensity = sum(intensity_integrated_intensity_snca, na.rm = TRUE),
      n_snca_objects = n(),
```

```
.groups = "drop"
)

# ---- area ----
df_area <- read_excel(area_path) %>%
  clean_names()

area_col <- get_area_col(df_area)

# Use path_name_snca if present; otherwise fall back to pathname_snca (or others)
# After clean_names(), your example becomes path_name_snca
if (!("path_name_snca" %in% names(df_area))) {
  stop("Area file is missing 'path_name_snca' after clean_names(). Check the column
name in the area sheet.")
}

df_area <- df_area %>%
  mutate(dataset_id = extract_dataset_id(path_name_snca)) %>%
  filter(!is.na(dataset_id)) %>%
  mutate(dataset_id = as.integer(dataset_id))

area_by_dataset <- df_area %>%
  group_by(dataset_id) %>%
  summarise(
    total_area = sum(.data[[area_col]], na.rm = TRUE),
    n_masks    = n(),
    .groups = "drop"
  )

# ---- join + compute ----
out <- intensity_by_dataset %>%
  left_join(area_by_dataset, by = "dataset_id") %>%
  mutate(
    cell_type = cell_type_label,
    intensity_per_area = total_intensity / total_area
  ) %>%
  arrange(dataset_id)

out
}
```

RUN FOR CHAT + MSN

```
chat_by_dataset <- summarise_intensity_and_area(chat_intensity_file, chat_area_file,
"ChAT")
msn_by_dataset <- summarise_intensity_and_area(msn_intensity_file, msn_area_file,
"MSN")

# Save Excel-ready CSVs
write_csv(chat_by_dataset, "~/Desktop/CHAT_pSNCA_per_area_per_dataset.csv")
write_csv(msn_by_dataset, "~/Desktop/MSN_pSNCA_per_area_per_dataset.csv")

print(chat_by_dataset)
print(msn_by_dataset)

# COMBINE + PLOT (individual datasets + mean)
combined <- bind_rows(chat_by_dataset, msn_by_dataset)

ggplot(combined, aes(x = cell_type, y = intensity_per_area)) +
  geom_jitter(width = 0.12, size = 3, alpha = 0.85) +
  stat_summary(
    fun = mean,
    geom = "crossbar",
    width = 0.5,
    fatten = 0,
    linewidth = 1
  ) +
  labs(
    x = "",
    y = "Total pSNCA intensity / Total mask area (per dataset)",
    title = "pSNCA density per dataset (dots) + mean"
  ) +
  theme_classic(base_size = 13)

# Optional summary table (mean/SD/SEM across datasets)
summary_by_type <- combined %>%
  group_by(cell_type) %>%
  summarise(
    mean_intensity_per_area = mean(intensity_per_area, na.rm = TRUE),
    sd_intensity_per_area = sd(intensity_per_area, na.rm = TRUE),
    sem_intensity_per_area = sd_intensity_per_area / sqrt(n()),
    n_datasets = n(),
    .groups = "drop"
  )

print(summary_by_type)
```



```
write_csv(summary_by_type,  
"~/Desktop/Summary_pSNCA_per_area_CHAT_vs_MSN.csv")
```

Protocol references

1. Kim Y, McInnes J, Kim J, Liang YHW, Veeraragavan S, Garza AR, Belfort BDW, Arenkiel B, Samaco R, Zoghbi HY. Olfactory deficit and gastrointestinal dysfunction precede motor abnormalities in alpha-Synuclein G51D knock-in mice. *Proc Natl Acad Sci U S A*. 2024 Sep 24;121(39):e2406479121. doi: 10.1073/pnas.2406479121. Epub 2024 Sep 16. PMID: 39284050; PMCID: PMC11441490.
2. Stirling DR, Swain-Bowden MJ, Lucas AM, Carpenter AE, Cimini BA, Goodman A (2021). CellProfiler 4:improvements in speed, utility and usability. *BMC Bioinformatics*, 22 (1), 433.. PMID: 34507520 PMCID: PMC8431850.
3. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez JY, White DJ, Hartenstein V, Eliceiri K, Tomancak P, Cardona A. Fiji: an open-source platform for biological-image analysis. *Nat Methods*. 2012 Jun 28;9(7):676-82. doi: 10.1038/nmeth.2019. PMID: 22743772; PMCID: PMC3855844.
4. R Core Team (2021). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.