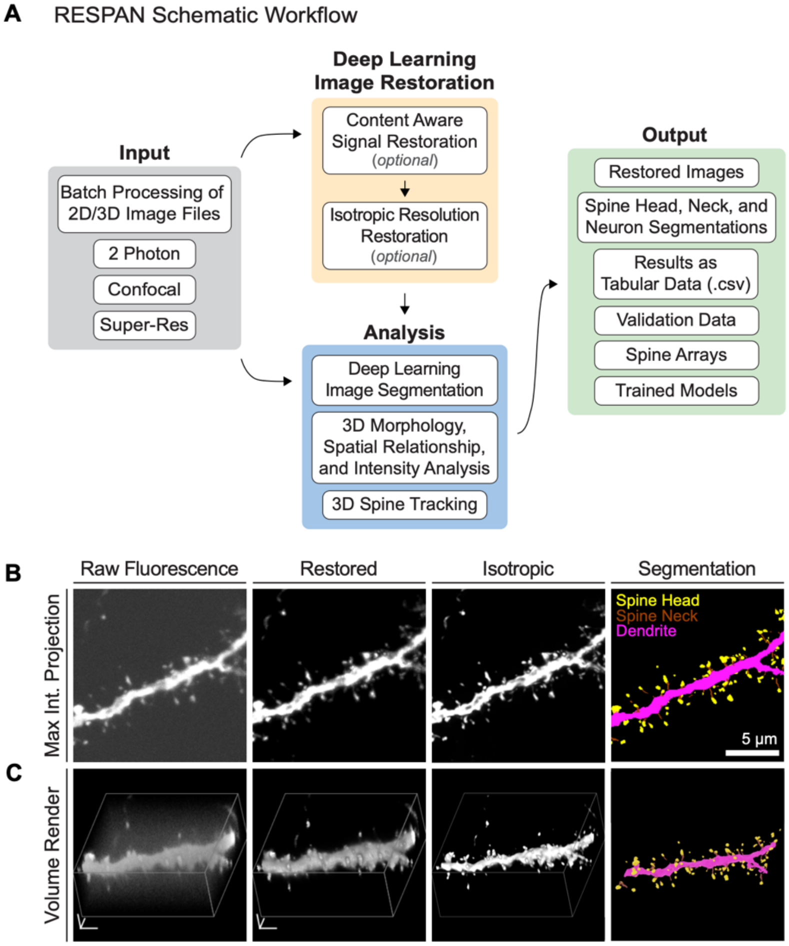
Restoration Enhanced Spine and Neuron (RESPAN) Analysis: Users Guide

About RESPAN

The RESPAN pipeline integrates GPU image processing and multiple deep-learning approaches to enable high-throughput fluorescent image segmentation, 3D reconstruction, and analysis of dendritic branches and dendritic spines. By integrating content-aware restoration and 3D convolutional neural network segmentation, RESPAN improves spine detection accuracy and sensitivity to morphological variations, particularly in challenging experimental conditions such as in vivo two-photon microscopy and rapid volumetric imaging of large tissue volumes. To ensure broad accessibility, RESPAN is provided both as Python code and as a standalone Windows application with an intuitive, unified GUI, allowing users to run batch analysis, train models, and perform analysis validation all within the same interface.


A
RESPAN Schematic Workflow
Input
Batch Processing of 2D/3D Image Files
2 Photon
Confocal
Super-Res
Deep Learning Image Restoration
Content Aware Signal Restoration (optional)
Isotropic Resolution Restoration (optional)
Analysis
Deep Learning Image Segmentation
3D Morphology, Spatial Relationship, and Intensity Analysis
3D Spine Tracking
Output
Restored Images
Spine Head, Neck, and Neuron Segmentations
Results as Tabular Data (.csv)
Validation Data
Spine Arrays
Trained Models
B
Raw Fluorescence
Restored
Isotropic
Segmentation
Max Int. Projection
Spine Head
Spine Neck
Dendrite
5 µm
C
Volume Render

Considerations

RESPAN currently only supports input data in TIFF format, requiring conversion of proprietary formats to TIFF prior to use in our software. To overcome this limitation, we provide a Fiji macro with a GUI to facilitate batch conversation to TIFF using OMERO Bio-Formats. For ease of use, we also generate our output data in TIFF, but the upcoming release will support Zarr format.

Users should also be aware of the limitations of applying deep learning models to image data. Restoration models should be trained for the specific biological features, resolution, and imaging modality they are being used for. Otherwise, in addition to performing suboptimally, these models can hallucinate features that do not exist and degrade features that do exist. Users should also be aware of this characteristic if comparing conditions with appreciable changes in morphology and train models that capture the full diversity of biological features across conditions.

RESPAN requires limited specialized expertise and is distributed with trained models that can be readily adapted to most confocal image volumes with sufficient resolution to resolve spines. In cases where new imaging conditions or modalities are being used, new models should be trained, with experts reaching a consensus on manual annotations.

Experimental Design

To utilize RESPAN, image volumes should be acquired using a technique that provides sufficient resolution and optical sectioning to resolve spines, such as confocal or light-sheet microscopy. These datasets must include at least one channel representing the labeled dendrite or neuron. Additional channels can also be included and used for intensity measurements.

Requirements

Minimum specifications:

- A Windows computer with at least 32 GB of RAM.
- An NVIDIA GPU with a minimum of 8 GB VRAM to support computations.
- At least 500 GB of fast (SSD) storage space to handle image datasets and output files.
- An up-to-date operating system (Windows 10 or Windows 11).

For Optimal Performance:

- **Operating System**: Windows 10 Enterprise 64-bit.
- **Processor**: 16-core AMD Ryzen Threadripper PRO 5955WX.
- **Memory**: 256 GB DDR4 RAM.
- **Storage**: 1 TB Samsung 860 SSD.
- **Graphics Card**: NVIDIA GeForce RTX 4090 GPU with 24 GB VRAM.

Software

- RESPAN Executable available for download at: https://github.com/lahammond/RESPAN
    - RESPAN is provided as a standalone application with a graphical user interface (GUI), eliminating the need for programming expertise.

Required input data

- RESPAN accepts 2D or 3D TIFF (.tif) image files.
- Example data and models can be downloaded from

Installation of RESPAN

1. Download RESPAN:
    a. Navigate to the RESPAN GitHub repository: https://github.com/lahammond/RESPAN.
    b. Download the .zip file containing the RESPAN executable.
2. Download and install Anaconda (https://www.anaconda.com/)
3. Install RESPAN:
    a. Extract the contents of the .zip file to a suitable location on your computer (e.g. C:\Program Files\) using 7zip (https://www.7-zip.org/).
    b. Open the extracted folder and double-click on the RESPAN executable file (RESPAN.exe).
    c. The RESPAN GUI will initialize and begin the installation of all necessary packages within a custom RESPAN environment.
    d. Note: The installation process may take 5–10 minutes. Upon completion, a message saying "Initialization complete" will appear.

Running RESPAN from the GUI

This procedure demonstrates how to organize your data and run RESPAN for standard analysis using the GUI. This procedure excludes the usage for signal restoration and isotropic restoration.

Preparing data for RESPAN analysis

1. Convert Proprietary Formats (if necessary):
    a. If your image data is in a proprietary format (e.g., .ND2, .CZI), use the provided Fiji macro to batch convert the images to TIFF format before proceeding.
2. Organize Your Data:
    a. Store your images as TIFF files in subfolders within a single parent directory.
    b. Within each subfolder, include related images (e.g., multiple time points for a single dendrite/neuron or multiple dendrites/neurons for a single experimental condition).
    Note: This folder structure facilitates batch processing and comparative analysis across different conditions.
3. Prepare the Analysis Settings:
    a. Copy the provided Analysis_Settings.yml file into each subfolder containing your images. For standard use, this file does not require modification. Refer to Procedure 3 for advanced settings customization.

Running RESPAN

1. Launch RESPAN:
    a. Double-click the RESPAN.exe file to open the GUI.
2. Configure the Analysis:
    a. In the GUI, select the parent folder containing your image subfolders. Results will be automatically saved in subfolders.
    b. Adjust any additional parameters as needed (see Box 1 for parameter details).
3. Start the Analysis:
    a. Click the "Run" button.
    b. The log window within the GUI will display updates as the analysis progresses.
    Note: An image volume of 100MB is expected to take 1 min to process in RESPAN.
4. Review the Results:
    a. Results are automatically saved in subfolders within your parent directory.
    b. Refer to the "Anticipated Results" section for details on output files and their locations.

Validating RESPAN outputs

This procedure guides you through validating RESPAN's segmentation outputs against a ground truth dataset.

Ground Truth Dataset:

If you have not generated a Ground Truth annotation dataset, please refer to notes below on **creating annotations** as a guide on how to generate these annotations for your specific datasets before you proceed.

CRITICAL STEP: If you use RESPAN-generated annotations as ground truth annotations, ensure these images and annotations are not included in the training dataset of any models intended for validation, to avoid bias.

Running the validation tool in RESPAN

1. Launch RESPAN:
    a. Open the RESPAN GUI.
2. Access the Validation Tab:
    a. Select the "Analysis Validation" tab at the top of the GUI.
3. Configure Validation Parameters:
    a. Select the Validation folder generated by RESPAN containing the predicted segmentation results.
4. Select the directory containing your ground truth annotations.
    a. Update the spine and dendrite detection parameters to match those used during the initial analysis.
5. Start the Validation:
    a. Click the "Run" button.
    b. The log window will display progress updates.
6. Review the Validation Results:
    a. An Analysis_Evaluation.csv file will be saved in the folder containing RESPAN's results. Refer to the "Anticipated Results" section for details on interpreting the validation output.

Advanced RESPAN Usage with Fine-Tuning and Restoration

Preparing data for RESPAN analysis

1. Organize Your Data:
    a. Follow steps 1–2 from **Preparing data for RESPAN analysis** to organize your images and copy the Analysis_Settings.yml file into each subfolder.

Modifying Analysis Settings

1. Edit the Analysis Settings File:
    a. Open Analysis_Settings.yml using a text editor (e.g., Sublime Text).
2. Customize Preprocessing Options:
    a. To enable CARE Restoration:
        i. Select *Use image restoration* on the GUI
        ii. In the *Analysis_Settings.yml* update the *rest_model_path* with the directory path to the appropriate restoration model.
    b. To enable SelfNet Isotropic Restoration:
        i. Select *Use axial restoration* on the GUI.
        ii. In the *Analysis_Settings.yml* update the *SelfNet model_path* with the directory path to the axial restoration model.
    c. For each channel you wish to preprocess with axial resolution restoration, set the corresponding “sn_restore” field to True. Then, update the “sn_model_path” field with the directory containing the restoration model to be used.

Running RESPAN

1. Start RESPAN and update the GUI as previously described.

Training an nnU-Net model with new data

Preparing data for nnU-Net training

Timing 1-3 hours

1. Organize image volumes containing the fluorescence channel used for spine and dendrite segmentation into a single folder. If image restoration will be used in RESPAN prior to segmentation, process these images using the same restoration models prior to adding them to this folder.
2. In a separate folder, organize corresponding image volumes containing label annotations, either created manually or by correcting label outputs from RESPAN. These files should have the same filename as their corresponding fluorescence channel image.

Training the model

Timing 12-24 hours

1. Start RESPAN, and at the top of the GUI, select the nnU-Net Training tab.
2. Within this tab, select the directory containing the fluorescence channel image volumes. Then select the folder containing the label annotation image volumes.
3. Click “Run” to initiate training.
4. When training is complete, a new segmentation model will be created in the directory containing the fluorescence channel image volumes.

Training a CARE-3D model for image restoration

Preparing data for CARE-3D training

Timing 30 min

1. Organize image volumes acquired using low SNR parameters into a single folder.
2. In a separate folder, organize corresponding image volumes acquired using high SNR parameters.

Training the model

Timing 3-5 hours

1. Start RESPAN, and at the top of the GUI, select the CARE Training tab.
2. Within this tab, select the directory containing the low SNR image volumes. Then, select the folder containing the high SNR image volumes.
3. Click “Run” to initiate training.
4. When training is complete, a new CARE restoration model will be created in the directory containing the low SNR image volumes.

Training a SelfNet model for axial restoration

Preparing data for SelfNet training

Timing 10 min

1. Organize image volumes containing the fluorescence channel used for spine and dendrite segmentation into a single folder. If image restoration will be used in RESPAN prior to axial resolution restoration, process these images using the same image restoration model prior to adding them to this folder

Training the new model

Timing 0.25 - 2 hours

1. Start RESPAN, and at the top of the GUI, select the SelfNet Training tab.
2. Within this tab, select the directory containing the image volumes organized in step 1.
3. Click “Run” to initiate training.
4. When training is complete, a new SelfNet restoration model will be created in the directory containing the image volumes.

Creating Annotations

If processing your data through RESPAN does not yield optimal results, it will be necessary to create new ground truth annotations to refine or train a new segmentation model. The most efficient way to do this, especially if the outputs only require minor corrections, is noted below.

Improving existing annotations

1. Open the raw input data and the annotations created during RESPAN processing in Fiji.
2. Ensure both images have the same bit depth (i.e. if annotations are 8-bit and the raw data is 16-bit, convert the annotations to 16-bit) via Image>Type> 16-bit
3. Merge the images as channels via Image > Color > Merge Channels…
4. Adjust the Brightness and Contrast (Shift-C) for the raw intensity channel so that you can see the spines and dendrites in your image – be careful not to over contrast the image; otherwise, your annotations may be inaccurate
5. Select the annotation channel. To observe the annotations, change the look up table to Glasby on Dark via Image > Lookup Tables > Glasby on Dark
6. To add and remove **spine** voxels, click Set in the Brightness and Contrast window, and set **Min displayed value** to 0 and **Max displayed value** to 1, then click OK
7. Select the **Brush** tool in the Fiji toolbar (double click to set width, usual 1px unless making significant changes)
   a. Click and drag to fill in pixels that reflect spines
   b. Hold down Alt while clicking and dragging to erase false positive voxels
8. For annotating **dendrites** repeat these steps but set the **Max displayed value** to 2
9. For annotating **necks** repeat these steps but set the **Max displayed value** to 3
10. For annotating **soma** repeat as above but set the **Max displayed value** value to 4

Anticipated Results

By following the RESPAN protocols, it is possible to segment and quantify the dendritic shaft and spines in an automatic and unbiased manner. The final RESPAN outputs include validation images, 3D renderings of dendrites and spines, a .swc tracing file of the dendrite or neuron, tabular results for each image, and a summary table for each condition or time series. Intermediary and 3D full-resolution validation data can also be saved if desired.

Intermediary data for validation

RESPAN will create a folder \Validation_Data, which contains image data used for validating the performance of the analysis and allowing faster reprocessing of datasets if spine filtering settings in the GUI need to be refined. Within this folder, the \Restored_Images subfolder contains image volumes following restoration. The \Segmentation_Labels subfolder contains the initial segmentation results with image volumes consisting of voxels labeled for dendrites, spines, and soma. If the provided models don't perform accurately, the annotations in these output images can be corrected and then used to train a new segmentation model. The \Validation_MIPs subfolder contains multi-channel Maximum Intensity Projection (MIP) images used to audit RESPAN's performance. The channels in these images are 1) input channel used for spine analysis, 2) all putative spines, 3) quantified spines with unqiue IDs matching tabular data, 4) quantified dendrites with unique ID matching tabular data, 5) dendrite/neuron skeleton, 6) distance from dendrite. Finally, if "Save 3D validation data" is enabled in the GUI, the \Validation_Vols subfolder will contain multi-channel full-resolution 3D volumes with the same set of six channels.

Tabular data

The \Tables folder will contain tabular data in .csv format. Each input volume will generate an individual results file with the filename "Detected_spines_{filename}.csv" and contents matching those described in Table 1. A summary of results from all files in the input folder is also generated with the filename "Detected_spines_summary.csv" and contents matching those described in Table 2.

Neuron tracing

If "Generate SWC file for dendrite/neuron" is enabled in the RESPAN GUI, the \SWC_files folder will contain a .swc tracing file for each input volume. These can be viewed and analyzed further in neuron analysis tools such as Vaa3D or Simple Neurite Tracer.

Spine arrays

RESPAN extracts a 3D image volume around each detected spine and saves these as arrays for further analysis and visualization. The \Spine_Arrays folder will contain four spine array images for each input volume with the following channels: 1) input channel used for spine analysis, followed by other channels in the image, 2) spine mask with unique ID matching tabular data, 4) dendrite mask. "Masked_Spines_MIPs_{filename}.tif" and "Masked_Spines_Slices_{filename}.tif" contain signal only within the detected spine as a MIP of the 3D volume, or a single slice centered on the detected spine, respectively. "Spine_MIPs_{filename}.tif" and "Spine_slices_{filename}.tif" also include signal outside the spine as a MIP or slice, respectively. If "Save 3D spine arrays" is selected, a "Spine_vols_{filename}.tif" image will be generated containing 3D volumes centered around each detected spine.

Analysis Evaluation

When performing analysis validation from the RESPAN GUI, an “Analysis_Evaluation.csv” file will be generated.

RESPAN Parameters

This Box describes the parameters needed in the RESPAN GUI and additional considerations.

Image Data and Model Selection

- **Select data directory:** the location of the parent folder, which contains subfolders, each with an analysis_settings.yml file and TIF files to be processed.
- **Select segmentation model directory:** the location of the nnU-Net model directory provided with this software or one trained on your data.

Spine and Dendrite Detection

- **Channel containing dendrite/neuron signal:** specify the channel best representing the dendrite/neuron. If processing a single channel dataset, leave as 1.
- **Minimum dendrite volume (µm3):** putative dendrites or neurons detected smaller than this volume and their associated spines will be excluded from the analysis.
- **Spine volume filter (µm3):** specify the minimum and maximum volume of spines to be analyzed. Putative spines detected outside this range will be excluded from the analysis.
- **Spine distance filter (µm):** maximum distance spines should extend from the dendrite. Spines located further than this distance will be excluded from the analysis.

Image Data and Model Resolution

- **Image voxel size XY (µm):** lateral sampling used for acquiring data.
- **Image voxel size XY (µm):** axial sampling or step-size used for acquiring data.
- **Model voxel size XY (µm):** lateral sampling in data used for training segmentation model.
- **Model voxel size XY (µm):** axial sampling in data used for training segmentation model.
- **Use voxel sizes in analysis_settings.yml:** If enabled, overrides the above scaling parameters in favour of those specified in the settings file. Useful when processing datasets acquired using multiple imaging conditions or requiring different restoration models.

Temporal Analysis and Spine Tracking

- **Spine tracking:** If enabled, allows subsequent tracking and temporal analysis of spines.
- **Histogram matching:** Used to match signal intensity across image volumes being used for spine tracking.
- **Registration method:** Method used for registering image volumes for spine tracking.

Additional Options

- **Save 3D validation data:** Saves complete 3D datasets including labeled spines, which is useful for 3D validation, but significantly increases data storage requirements.
- **Save intermediate data:** Saves all intermediate data for troubleshooting challenging datasets.
- **Use image restoration:** If enabled, utilizes models specified in the analysis_settings.yml file to restore signal-to-noise using CARE.
- **Use axial restoration:** If enabled, utilizes models specified in the analysis_settings.yml file to restore signal-to-noise using SelfNet.
- **Generate SWC file for dendrite/neuron:** If enabled, generates an SWC file for the dendrite/neuron using Vaa3D. Requires Vaa3D to be installed and path to be correct in the analysis_settings.yml file.

Analysis Settings.yml Parameters

This Box describes the parameters in the Analysis_Settings.yml file that can be modified for the advanced use of RESPAN.

Parameters

- **input_resXY:** The input resolution in the X and Y axis.
- **input_resZ:** The input resolution in the Z axis.
- **model_resXY:** The resolution in the X and Y axis of the data used to train the model.
- **model_resZ:** The resolution in the Z axis of the data used to train the model.

Analysis:

- **tiles_for_prediction:** The number of tiles used when processing images. This can be adjusted to modify performance.
- **roi_volume_size:** The size of volume extracted for each detected spine (in μm)
- **GPU_block_size:** Parameters used to adjust the image volume used for GPU processing. This can be modified to improve performance depending on GPU specifications.

nnUnet

- **type:** The type of nnU-Net model being used for segmentation.
- **conda_path:** The directory where Anaconda is installed.
- **env:** The environment where nnU-Net is installed.

Vaa3D:

- **path:** The directory where Vaa3D is installed.

Channe1-4:

- **care_restore:** The variable used to enable image resoration for this channel (True/False).
- **care_model_path:** The directory containing the model for image restoration for this channel.
- **sn_restore:** The variable used to enable axial resoration for this channel (True/False).
- **sn_model_path:** The directory containing the model for axial restoration for this channel.

Table 1. Individual Spine Measurements

Column	Title	Contents
1	spine_id	Unique identifier for each detected spine.
2	x	X-coordinate of the spine in image space (pixels).
3	y	Y-coordinate of the spine in image space (pixels).
4	z	Z-coordinate of the spine in image space (pixels).
5	dendrite_id	Identifier for the dendrite associated with the spine.
6	geodesic_dist	Geodesic distance from the spine to its parent dendrite (µm).
7	spine_area	Cross-sectional area of the entire spine (µm^2).
8	spine_vol	Volume of the entire spine (µm^3).
9	spine_vol_m	Measured spine volume adjusted for voxel resolution (µm^3).
10	spine_sa_m	Spine surface area adjusted for voxel resolution (µm^2).
11	spine_length	Total morphological length of the spine from the dendrite base to the spine head tip (µm).
12	head_area	Cross-sectional area of the spine head (µm^2).
13	head_vol	Volume of the spine head (µm^3).
14	head_vol_m	Measured volume of the spine head adjusted for voxel resolution (µm^3).
15	head_sa_m	Surface area of the spine head adjusted for voxel resolution (µm^2).
16	head_length	Length of the spine head, measured from the neck boundary to the head tip (µm).
17	neck_area	Cross-sectional area of the spine neck (µm^2).
18	neck_vol	Volume of the spine neck (µm^3).
19	neck_vol_m	Measured volume of the spine neck adjusted for voxel resolution (µm^3).
20	neck_sa_m	Surface area of the spine neck adjusted for voxel resolution (µm^2).
21	neck_length	Length of the spine neck, measured from the dendrite boundary to the head boundary (µm).
22	spine_length_dm	Alternative spine length measurement derived from distance mapping (µm).
23	spine_C1_int_density	Fluorescence integrated density in Channel 1 within the entire spine.
24	spine_C1_max_int	Maximum fluorescence intensity in Channel 1 within the entire spine.
25	spine_C1_mean_int	Mean fluorescence intensity in Channel 1 within the entire spine.
26	dist_to_dendrite_dm	Geodesic distance from the spine to the dendrite (µm).
27	dist_to_soma	Geodesic distance from the spine to the soma (µm).
28	head_C1_mean_int	Mean fluorescence intensity in Channel 1 within the spine head.
29	head_C1_max_int	Maximum fluorescence intensity in Channel 1 within the spine head.
30	head_C1_int_density	Fluorescence integrated density in Channel 1 within the spine head.
31	neck_C1_mean_int	Mean fluorescence intensity in Channel 1 within the spine neck.
32	neck_C1_max_int	Maximum fluorescence intensity in Channel 1 within the spine neck.
33	neck_C1_int_density	Fluorescence integrated density in Channel 1 within the spine neck.

Table 2. Detected Spines Summary

Column	Title	Contents
1	Filename	Name of the analyzed image file.
2	res_XY	Lateral (XY) resolution of the image in micrometers per pixel (µm/pixel).
3	res_Z	Axial (Z) resolution of the image in micrometers per pixel (µm/pixel).
4	dendrite_id	Unique identifier for the segmented dendrite.
4	dendrite_length	Total length of the segmented dendrite, in micrometers (µm).
5	dendrite_vol	Total dendritic volume, in cubic micrometers (µm³).
6	total_spines	Total number of detected spines along dendrite.
7	spines_per_um	Number of spines per micrometer of dendrite length (spines/µm).
8	spines_per_um3	Number of spines per cubic micrometer of dendrite volume (spines/µm³).
9	avg_spine_area	Mean cross-sectional area of the detected spines, in square micrometers (µm²).
10	avg_spine_vol	Mean volume of the detected spines, in cubic micrometers (µm³).
11	avg_spine_vol_m	Mean mesh volume of spines (µm³).
12	avg_spine_sa_m	Mean surface area of spines (µm²).
13	avg_spine_length	Mean length of the spines, in micrometers (µm).
14	avg_head_area	Mean cross-sectional area of the spine heads (µm²).
15	avg_head_vol	Mean volume of the spine heads (µm³).
16	avg_head_vol_m	Mean mesh volume of spine heads (µm³).
17	avg_head_sa_m	Mean surface area of spine heads (µm²).
18	avg_head_length	Mean length of the spine heads (µm).
19	avg_neck_area	Mean cross-sectional area of the spine necks (µm²).
20	avg_neck_vol	Mean volume of the spine necks (µm³).
21	avg_neck_vol_m	Mean mesh volume of spine necks (µm³).
22	avg_neck_sa_m	Mean surface area of spine necks (µm²).
23	avg_neck_length	Mean length of the spine necks, in µm.
24	avg_spine_length_dm	Mean spine length derived from distance-map or similar metric, in µm.
25	avg_spine_C1_int_density	Mean integrated density in Channel 1 within all spines.
26	avg_spine_C1_max_int	Mean maximum fluorescence intensity in Channel 1 within all spines.
27	avg_spine_C1_mean_int	Mean fluorescence intensity in Channel 1 within all spines.
28	avg_dist_to_dendrite_dm	Mean distance from spines to the dendrite shaft, in µm.
29	avg_dist_to_soma	Mean distance from spines to the soma, in µm.
30	avg_head_C1_mean_int	Mean fluorescence intensity in Channel 1 within spine heads.
31	avg_head_C1_max_int	Maximum fluorescence intensity in Channel 1 within spine heads.
32	avg_head_C1_int_density	Mean integrated density in Channel 1 within spine heads.
33	avg_neck_C1_mean_int	Mean fluorescence intensity in Channel 1 within spine necks.
34	avg_neck_C1_max_int	Maximum fluorescence intensity in Channel 1 within spine necks.
35	avg_neck_C1_int_density	Mean integrated density in Channel 1 within spine necks.