Software for segmenting and quantifying calcium signals using multi-scale generative adversarial networks

Cellular calcium fluorescence imaging utilized to study cellular behaviors typically results in large datasets and a profound need for standardized and accurate analysis methods. Here, we describe open-source software (4SM) to overcome these limitations using an automated machine learning pipeline for subcellular calcium signal segmentation of spatiotemporal maps. The primary use of 4SM is to analyze spatiotemporal maps of calcium activities within cells or across multiple cells.

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

Software for segmenting and quantifying calcium signals using multi-scale generative adversarial networks

Hussein Moghnieh, 3,4 Sharif Amit Kamran, 1,2,4 Khondker Fariha Hossain, 2 Nyanbol Kuol, 1 Sarah Riar, 1 Allison Bartlett, 1 Alireza Tavakkoli, 2 and Salah A. Baker 1,5,6,*

1Department of Physiology and Cell Biology, University of Nevada, School of Medicine, Anderson Medical Building MS352, Reno, NV 89557, USA
2Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA
3Department of Electrical and Computer Engineering, McGill University, Montréal, QC H3A 0E9, Canada
4These authors contributed equally
5Technical contact: sabubaker@med.unr.edu
6Lead contact
*Correspondence: sabubaker@med.unr.edu
https://doi.org/10.1016/j.xpro.2022.101852

SUMMARY

Cellular calcium fluorescence imaging utilized to study cellular behaviors typically results in large datasets and a profound need for standardized and accurate analysis methods. Here, we describe open-source software (4SM) to overcome these limitations using an automated machine learning pipeline for subcellular calcium signal segmentation of spatiotemporal maps. The primary use of 4SM is to analyze spatiotemporal maps of calcium activities within cells or across multiple cells. For complete details on the use and execution of this protocol, please refer to Kamran et al. (2022).1

BEFORE YOU BEGIN

© Timing: <30 min

This section includes the minimal hardware requirements and the installation procedure.

Hardware requirements

List of NVIDIA graphics cards supporting CUDA 10.0 https://gist.github.com/standaloneSA/99788f30466516dbcc00338b36ad5acf.

Software requirements

- CUDA version 10.0: List of NVIDIA graphics cards supporting CUDA 10.0 https://gist.github.com/standaloneSA/99788f30466516dbcc00338b36ad5acf.

- Anaconda.

Anaconda python platform

Note: A video (Methods video S1) illustrating the steps to install Anaconda, download and install 4SM is available here: https://youtu.be/SIvdpFLGt1w.

4SM is functional on all operating systems (Windows, Linux, and macOS) with Anaconda installed.
1. Download and install Anaconda from:
   a. Windows 64bit: https://repo.anaconda.com/archive/Anaconda3-2021.11-Windows-x86_64.exe.
   b. Linux/macOS: https://www.anaconda.com/products/individual according to individual computer specifications.

   On Windows OS, make sure to select “Add Anaconda3 to the system PATH environment variable” during “Advanced Installation Options” as shown in Figure 1.

2. Download 4SM from https://github.com/SharifAmit/4SM/archive/refs/heads/star.zip (Figure 2A).
3. Unzip the downloaded file. The root folder of the unzipped file contains the source code and the scripts to install and run 4SM (Figures 2B and 2C).
4. Before using 4SM for the first time, it is necessary to run the setup scripts. This step will create Anaconda’s environment and install all Python dependencies.
   a. On Windows OS: double click the batch script file <windows_install_libs.bat>. This will install 4SM dependencies as shown in Figure 2C.
   b. On Linux based OS and macOS, open a shell window and run the following command:

   ```bash
   /bin/bash linux_install_libs.sh
   ```

**KEY RESOURCES TABLE**

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|----------------------|--------|------------|
| Software and algorithms | Fiji, version 2.0.0-rc-69/1.52 | NIH | https://fiji.sc/ |
| 4SM, Subcellular Fluorescence Analysis Software | Kamran et al., 2022 | https://doi.org/10.1016/j.isci.2022.104277 |
| STMapAuto, Ca2+ Analysis plugin | GitHub | https://doi.org/10.1016/j.ceca.2020.102260 |
STEP-BY-STEP METHOD DETAILS

4SM is designed to work primarily with spatiotemporal maps of fluorescence signals. Movies of dynamic fluorescence signals (i.e., cellular calcium signals) can be obtained from multiple imaging systems including: widefield, spinning disk confocal, 2 photon microscopy or macroscopy systems that can yield an image stack (movies of fluorescence signal over time). The fluorescence signals in the image stack can be plotted in 2D map (spatiotemporal map) as described in detail previously,1,2 and these maps can be effectively segmented and analyzed using the 4SM software.

4SM utilization

The full tutorial (Methods video S2) on how to launch and use 4SM for image segmentation can be found on the following link: https://youtu.be/uOI69A7bnyw.

1. To launch your application,
   a. On Windows OS: double click the batch script file <windows_start.bat> as shown in Figure 3B.
   b. On Linux based OS and macOS, open a shell window and run the following command:

   ```
   > sh linux_start.sh
   ```

   A new browser will open with 4SM app running on the following URL: http://localhost:8501.

   **Note:** If you are using an older version of Internet Explorer, copy the above URL and paste it in Google Chrome, Microsoft Edge, or Firefox.

2. Click on <Upload an Image> on the left panel as shown in Figure 4A.
3. Browse to the folder "src/test/stochastic" and select the 2 images available in this folder.
4. Set the Threshold, Connectivity, Height Calibration, Width Calibration, and Stride Values. For details about the definition of these settings, refer to Section “4SM parameters values”.
5. Click on “Run Prediction”.

The images selected will be displayed in the first column of the center panel as shown in Figure 4. The application will begin processing the images and finally the result of the processed images will be displayed in the center panel.

Quantification and calibrated quantification
Expanding the accordion menus shown in Figures 4D1 and 4D2 shows the quantification and calibrated quantification results respectively in the form of a table.

Figure 5 shows the quantification results of each processed image. Each row in the table represents the information pertaining to a detected region by the 4SM algorithm for the image specified in the column “Image”. The remainder columns in the resulting table are explained below:

Frequency: the number of fluorescence segmented events in each map.

Left and Top: event coordinates on the map.

Spatial spread: event travel distance (on the x-axes).
**Duration:** event travel time (on the y-axes).

**Area:** event area occupied in a map.

**Interval:** distance between events.

**Statistical analysis**

4SM produces statistical plots as shown in Figure 6. The x-axis of each plot displays the `<run_id>`, which is an autogenerated identifier produced by 4SM to label each run as shown in Figure 4C. This enables the user to toggle between different outputs (i.e., runs).

**Visualization of processed images**

The main panel of 4SM displays each input image and all its corresponding analyzed images as shown in Figure 4F.

**Displaying results from multiple runs**

4SM aggregates results from different runs into the same quantification results and plots. To allow proper statistical analysis at least 5 runs are recommended.
6. Repeat steps 2–5 and choose the folder <src/test/rythmic> instead of <src/test/stochastic>.

7. Figure 6B displays the plots from the current and previous runs.

Excluding results from multiple runs

Each run output, which consists of one or more images processed, is stored under src/<run_id> in the destination folder. To exclude the results, simply delete this folder.

4SM parameters values

The 4SM software can be optimized using multiple parameters inputs to enhance the segmentation output. These parameters can affect the quality and the speed of the segmentation process. A summary of 4SM parameters is shown in Table 1.

Choosing 4SM input parameters

Threshold: It is dependent on the image quality obtained and should contain most of the calcium signal with minimum noise. The ideal values vary dependent on the calcium signal. For our datasets, we pick a value between 30 and 50 to get the best binary segmentation results. A higher threshold value might diminish the subtle activation boundaries of the calcium signal, in turn will help to visualize strong signals, and discard weak signals.

Connectivity: Finding unique number of events based on connected components algorithm. A connectivity of 8 is mostly desired for the prediction, as it considers all possible directions for neighboring pixels for generating connected components.

Height and Width Calibration: This is the size of the cropping window that goes from left to right and top to bottom. This cropped image is inserted to the 4SM model and as the model expects the input to be at least 64 pixels, any value greater than that is accepted. Generally, $64 \times 64$ is the desirable size for these parameters.

Stride: The stride should be 8 or 16 if the system does not have a powerful GPU (such as NVIDIA RTX 2080 or higher), to give fast output of the prediction. And the user should choose stride 3 for the most accurate segmentation, however it will take longer.

EXPECTED OUTCOMES

4SM provides a platform for the analysis calcium fluorescence spatiotemporal maps. These 2D maps are a plot of cellular fluorescence dynamic signals and 4SM allows accurate quantification of these fluorescence events (i.e., Calcium signals). Single wavelength calcium probes (i.e., Fluo 4) or
genetically encoded calcium sensors (i.e., GCaMP6) signals are compatible with spatiotemporal maps and can take advantage of 4SM software analysis.

As outcome, the software framework demonstrates accurate calcium signal segmentation and fast analysis pipeline that can segment different fluorescence signal patterns. The software allows seamless data accessibility, quantification, and graphical visualization.

Considering the future needs for processing large datasets in a fast manner and enables large dataset analysis throughput, 4SM allows batch-mode analysis of multiple maps in an accurate fashion.

Figure 6. Graphical representation of the analysis
(A) Statistical analysis of the output of each run (image) are displayed in graphical format. The x-axis displays the run_id, which is a randomly generated number to identify the output being displayed.
(B) Statistical analysis of the output from 2 runs (images). In this example, plots from different runs are plotted on the same graph to allow interpretation and comparison of results.
LIMITATIONS

The current version of 4SM provides a solid analysis tool for dynamic calcium signals, however it has not been evaluated on other types of dynamic fluorescent signals such as those generated to visualize changes in ions, pH, protein trafficking, or voltage. A limiting factor of the software is its dependency on fast GPUs as it uses a machine-learning algorithm at its core, and this may limit the use of the software effectively. Although the segmentation accuracy of the software can be controlled by the user (i.e., stride size) and this feature allow the user to prioritize accuracy over speed and vice versa, the software is still dependent on system GPU capabilities. The calcium spatiotemporal maps rely on the quality of imaging acquisition, and we anticipate a better result of maps with minimum background noise levels.

TROUBLESHOOTING

Problem 1

The most common problem when running 4SM is the format of the input dataset. The input images should be 8-bit greyscale, larger than 64 x 64 pixels, and saved in one of the following formats: PNG, JPG and TIF.

Potential solution

User can utilize ImageJ or programmatically pre-process the input images using Python to fit the software requirements.

Problem 2

Blank browser page when 4SM launches in Internet Explorer.

Potential solution

Make sure to use Google Chrome, Microsoft Edge, or Firefox.

Problem 3

Error when double clicking the batch script file “windows_install_libs.bat”: ‘pip’ is not recognized as an internal or external command.

Potential solution

Re-install Anaconda and make sure to tick the box to add Anaconda to the system PATH as shown in Figure 1. Alternatively, the batch script can be executed in the Anaconda Command Prompt.

Problem 4

4SM does not start when double clicking the batch script file <windows_start.bat>.

---

Table 1. Summary of 4SM parameters

| Param name     | Values                                                      | Definition                                                                 |
|----------------|-------------------------------------------------------------|----------------------------------------------------------------------------|
| Threshold      | 1–254 (min and max pixel intensity 0 and 255)              | The threshold parameter controls the cutoff point for binarizing the segmentation results to background (0) and Calcium events (1). |
| Connectivity   | 4 or 8                                                      | Finding unique number of events based on connected components algorithm. The values consider either the 4 axial direction or 4 axial and 4 diagonal direction (total 8) for considering neighboring pixels part of the connected component. |
| Height calibration | 64 px or higher                                           | 4SM takes images with minimum 64-pixel height or larger                   |
| Width calibration | 64 px to higher                                            | 4SM takes images with minimum 64-pixel width or larger                  |
| Stride         | 3 to N, where N is the height or width of the image.       | STMaps can vary in image size, so our algorithm divides STMap images into smaller images by cropping with a fixed window size of 64 x 64 pixels. The cropping window goes from left-to-right and then top-to-bottom and can overlap on top of each other. This overlapping factor is called stride or step-size. As the stride is divided by the cropping height or width, it cannot be larger than the height or width. |

---

Threshold, Connectivity, Height calibration, Width calibration and Stride.
Potential solution
Re-install Anaconda and make sure to tick the box to add Anaconda to the system PATH as shown in Figure 1. Alternatively, the batch script can be executed in the Anaconda Command Prompt.

RESOURCE AVAILABILITY
Lead contact
Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Dr. Sal A Baker (sabubaker@med.unr.edu).

Materials availability
This study did not generate new unique reagents.

Data and code availability
The code used during this study are available at https://github.com/SharifAmit/4SM/tree/star and https://zenodo.org/badge/latestdoi/392980422.

SUPPLEMENTAL INFORMATION
Supplemental information can be found online at https://doi.org/10.1016/j.xpro.2022.101852.

ACKNOWLEDGMENTS
This project was supported by R01 DK120759 from the NIDDK, United States. The authors would like to thank Wesley A. Leigh and Guillermo Del Valle for analyzing Ca2+ imaging data and conducting imaging experiments.

AUTHOR CONTRIBUTIONS
Conception and design of the experiments: H.M., S.A.K., K.F.H., A.T., and S.A.B. Collection and analysis of data: S.A.K., K.F.H., A.B., S.R., N.K., H.M., and S.A.B. Interpretation of data: S.A.K., A.B., S.R., A.T., and S.A.B. Drafting the article: H.M., S.A.K., A.B., and S.A.B. Revising article critically for intellectual content: H.M., S.A.K., A.B., A.T., N.K., and S.A.B.

DECLARATION OF INTERESTS
The authors declare no competing interests.

REFERENCES
1. Kamran, S.A., Hossain, K.F., Moghnieh, H., Riar, S., Bartlett, A., Tavakkoli, A., Sanders, K.M., and Baker, S.A. (2022). New open-source software for subcellular segmentation and analysis of spatiotemporal fluorescence signals using deep learning. iScience 25, 104277. https://doi.org/10.1016/j.isci.2022.104277.
2. Leigh, W.A., Del Valle, G., Kamran, S.A., Drumm, B.T., Tavakkoli, A., Sanders, K.M., and Baker, S.A. (2020). A high throughput machine-learning driven analysis of calcium spatio-temporal maps. Cell Calcium 91, 102260. https://doi.org/10.1016/j.ceca.2020.102260.