There are a limited number of experimental tools for non-destructively discovering cell communication events in vitro and in vivo. Here, using tissue-specific genetically encoded calcium indicator (GECI) mice, we describe a protocol for preprocessing GECI fluorescence time-series measured by live cell imaging or intravital microscopy, detecting peaks of single-cell calcium fluorescence transients, and inferring putative cell communication events from peak synchrony.

Publisher’s note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.
Protocol
Inferring cell communication using single-cell calcium spatiotemporal dynamics

Nika Taghdiri1,3,* and Kevin R. King1,2,4,*

1Department of Bioengineering, Jacobs School of Engineering, University of California San Diego, La Jolla, CA, USA
2Division of Cardiology and Cardiovascular Institute, Department of Medicine, University of California San Diego, La Jolla, CA, USA
3Technical contact
4Lead contact
*Correspondence: ntaghdir@eng.ucsd.edu (N.T.), krking@ucsd.edu (K.R.K.)
https://doi.org/10.1016/j.xpro.2022.101647

SUMMARY
There are a limited number of experimental tools for non-destructively discovering cell communication events in vitro and in vivo. Here, using tissue-specific genetically encoded calcium indicator (GECI) mice, we describe a protocol for preprocessing GECI fluorescence time-series measured by live cell imaging or intravital microscopy, detecting peaks of single-cell calcium fluorescence transients, and inferring putative cell communication events from peak synchrony. For complete details on the use and execution of this protocol, please refer to Taghdiri et al. (2021).

BEFORE YOU BEGIN
Conditional calcium reporter mice are created by crossing mice expressing Cre under the transcriptional regulation of a cell type specific promoter (e.g., Csf1r-Cre, Tie2-Cre, or Alb-Cre to target macrophages, endothelial cells, or hepatocytes respectively) with mice expressing a Cre-inducible GECI (e.g., GFP-calmodulin-M13 peptide (GCaMP)). (Grienberger and Konnerth, 2012) In vitro experiments can be performed by isolating, culturing, stimulating, and imaging primary cells in culture. Alternatively, in vivo experiments can be performed using disease models and intravital imaging. In this protocol, we describe the computational steps required to transform stacks of time-lapse images with defined single cell ROIs into putative cell communication events based on the spatiotemporal correlations of single cell calcium dynamics (Taghdiri et al., 2021). Before the analysis, one must consider the following:

1. The choice of conditional Cre determines the cell type specificity.
2. The choice of GECI determines the temporal bandwidth, signal strength, background fluorescence, and fluorescence spectrum (Chen et al., 2013).
3. An incubation system is essential for cell survival during time lapse imaging in vitro. To avoid evaporation and to maintain pH, the imaging chamber should be humidified, temperature-controlled at 37°C, and CO2-controlled (assuming a bicarbonate buffer is used). If CO2 is unavailable, a HEPES-based buffer can be used to maintain pH.
4. One should select a magnification that can spatially resolve single cells so that individual calcium transients can be quantified and putative intercellular communication can be inferred.
5. Regions for imaging can be chosen to maximize cell density while preserving the ability to discern single cells. This can be accomplished using brightfield images or using a constitutively expressed fluorescence reporter protein that is independent of calcium activity if available (Gee et al., 2014).
6. Duration of monitoring depends on the biological goals and the frequency of calcium transients in the cell type of interest within the biological context being studied. For investigations of communication in macrophages, we imaged for durations ranging from 5 min to 60 min.

7. Sampling frequency is a critically important consideration during time-lapse imaging (Figures 1A and 1B). Undersampling (sampling at intervals longer than the characteristic time scale of calcium transients) should be avoided, as this will blunt or frankly eliminate biologically important signals. (See troubleshooting, problem 1).

Note: In practice, we select optimal sampling frequencies by first oversampling and then digitally downsampling. We then qualitatively compare the pre-downsampled and post-downsampled tracings, evaluating for blunting or elimination of potentially biologically significant signals. If the digitally downsampling signal minimally alters the original signal, then the downsampled frequency is considered adequate as a sampling frequency in future image acquisition experiments. In the specific case of Csf1rCreGCaMP5fl, this resulted in an optimal sampling interval of 0.5 s (2 Hz) (Taghdiri et al., 2021). We have not explored the dependencies of optimal sampling interval, but one would expect that it depends on choice of GECI reporter, cell type, and stimulus or biological context. An alternative approach is to determine the bandwidth of the biological signal by collecting a long oversampled recording and performing a Fourier transform to examine the frequency content of the signal. If there is an obvious gap between noise and biological signal, one can sample at twice the maximum biological frequency in accordance with the Nyquist sampling theorem.

8. To quantify features of each single cell, one must first define regions of interest (ROI) surrounding each cell and save them to a file (*.roi). This can be performed manually by drawing each ROI by hand in Fiji/ImageJ or automatically using CellProfiler and its embedded modules as detailed below.

a. To manually define ROIs, open ROI manager in Fiji/ImageJ using Analyze > Tools > ROI manager. Choose Freehand Selection, encircle a cell, and select Add [t]. Additional cells can be encircled and added one by one. Save ROIs using Analyze > Tools > ROI manager > More > Save (.roi).

Figure 1. Example of undersampling and oversampling single-cell calcium transients
(A) Undersampling at 0.4 Hz (1 sample per 2.5 s) (top) compared to more optimal sampling at 2 Hz (bottom) sacrifices detail about the onset of the calcium transient.
(B) Oversampling at 15 Hz (top) compared to 2 Hz (bottom) does not obviously improve the signal.
b. To algorithmically define ROIs, we use CellProfiler, as detailed below (Chen et al., 2013; McQuin et al., 2018).
   i. Download CellProfiler software (macOS or Windows) and open it. You will see 4 preloaded modules at left.
   ii. Drag your image file onto the Images module and press Start Test Mode (lower left of window).
   iii. In the Metadata module, you will be prompted to extract metadata – answer yes.
   iv. In NamesAndTypes, you will be prompted to create a shorthand name for the image – input a shorthand name of your choosing.
   v. In the Groups module, you will be asked about grouping images – answer no.
   vi. Add 4 modules by clicking + to the right of Adjust modules. Select the following modules: Object Processing > IdentifyPrimaryObjects, EditObjectsManually, ConvertObjectsToImage, and File Processing > SaveImages.
   vii. Within the IdentifyPrimaryObjects module you must answer 3 questions. (1) After Use advanced settings? select No. (2) Select the shorthand name of your image. (3) Input Typical diameter of object (Min, Max), which can be determined by making interactive measurements. To do this, select the Images module, select Show Selected Image, and select Measure Length in the toolbar. Cells can then be measured interactively to determine minimum and maximum diameters.
   viii. Press Start Test Mode (lower left of window) to initiate object segmentation. Object segmentation can be optimized manually using the EditObjectsManually module. For additional details, press Show help for details. Advanced settings are also available to improve object selection.
   ix. Convert objects to an indexed mask (uint16) using the ConvertObjectsToImage module. Export it using the SaveImages module (*.tiff). Of note, the indexed mask is an image in which pixel values are set to the object number within object boundaries and 0 outside of objects.
   x. Open the indexed mask in Fiji/ImageJ (*.tiff).
   xi. Download the custom Fiji/ImageJ macro IndexedMaskToROI.ijm from Zenodo (see key resources table).
   xii. To use the macro, select Plugins > Macros > Run. When prompted, select IndexedMaskToROI.ijm. This will create ROIs from the indexed mask.
   xiii. Save ROIs using Analyze > Tools > ROI manager > More > Save (*.roi). For more information, please visit CellProfiler tutorial website.

Institutional permissions
All mouse experiments were approved and conducted under the oversight of University of California San Diego Institutional Animal Care and Use Committee (#17144) or approved by the Subcommittee on Animal Research Care at Massachusetts General Hospital.

KEY RESOURCES TABLE

| REGENT or RESOURCE          | SOURCE                                | IDENTIFIER          |
|-----------------------------|---------------------------------------|---------------------|
| Software and algorithms     |                                       |                     |
| Fiji/ImageJ                 | National Institute of Health, USA     | RRID: SCR_002285    |
| (Schindelin et al., 2012)   |                                       |                     |
| MATLAB                      | MathWorks- proprietary software      | R2020a– R2021b      |
| Statistics and machine      | MathWorks                             | R2021a -Version 12.1|
| learning toolbox            |                                       |                     |
| Signal processing toolbox   | MathWorks                             | R2021a -Version 8.6 |

(Continued on next page)
**STEP-BY-STEP METHOD DETAILS**

**Preprocessing of fluorescence time-series image stacks collected from GECI reporter cells**

© Timing: <30 min

The Preprocessing Step takes as input, a stack of fluorescence time-lapse images (*.tiff) and an associated ROI file (*.roi), and generates as output, a list of single cells (one for each ROI) with associated centroids and mean fluorescence time series. Fiji/ImageJ is first used to subtract background fluorescence from each image, extract centroids, and quantify mean fluorescence time series’ for each single cell (ROI). Next, MATLAB is used to calculate differential fluorescence time series, remove cells without transients, and smooth signals before peak-finding in step 2. Upon completion of this step, the signal is prepared for investigation of spatiotemporal correlations.

1. Open the fluorescence time series (*.tiff) in Fiji/ImageJ.
   a. Creating a Median Intensity Projection (MedIP) image.
   b. Subtract it from each frame of the fluorescence time series to remove background fluorescence.

   **Note:** Specifically, Image > Stacks > Z Project > Projection type (Median) to create the MedIP image, and Process > Image Calculator > Image 1 (Raw fluorescence) > Subtract > Image 2 (MedIP) to subtract the MedIP image (Figures 2 and 3) (See troubleshooting, problem 2).

2. Load the pre-saved ROIs using Analyze > Tools > ROI manager > More > Open.
3. Standardize the image orientation by setting the origin (0,0) to the lower left-hand corner using Image > Adjust > Coordinates.

   **Note:** A dialog box will appear. Input 0 after Left and 0 after Bottom.

4. Extract quantitative features (mean gray value and centroid) across all time points for each ROI using Analyze > Set Measurements > Mean Gray Value and Centroid followed by Analyze > Tools > ROI Manager > More > Multi Measure. Save the resulting file (*.csv) (Figure 3).
5. Download the cell communication inference script inferring_cell_communication.m (key resources table). Open the script in MATLAB and run it by pressing F5.
6. The MATLAB script will prompt the user to select a file (choose the .csv file saved above in sub-step 4).
   a. The script will automatically calculate $\Delta F/F[t]$, a matrix of normalized differential fluorescence time series’ for each cell.
   b. The script will automatically use $F[t]$, a matrix of raw fluorescence values defined at each time point for each cell.
   c. The script will automatically use $F_{med}$, a vector of median fluorescence values, one for each cell.

| REGENT or RESOURCE | SOURCE | IDENTIFIER |
|-------------------|--------|------------|
| Bioinformatics toolbox | MathWorks | R2021a - Version 4.15.1 |
| CellProfiler | Cimini Lab, The Broad Institute of MIT and Harvard | Version 3.1.9 |

| Other |
|-------|
| Fluoview FV1000 confocal microscope | Olympus | FV1000 |
| Raw time lapse images | Cell Methods paper | https://doi.org/10.1016/j.crmeth.2021.100132 |
| Scripts | GitHub / Zenodo | https://doi.org/10.5281/zenodo.6846833 |
\[ \Delta F / F(t) = (F(t) - F_{\text{med}}) / F_{\text{med}} \]

**Note:** Use of F\text{med} as a reference assumes that cells spend at least 50% at baseline. If this assumption is satisfied, then F\text{med} will provide a good estimate of cell-specific baseline dynamics. If it is only valid during part of the record, one should subset the time series and perform the analysis to derive F\text{med}. An alternative approach to calculating relative fluorescence is to subtract an estimate of initial fluorescence, F_0. However, this assumes that cells are not in the midst of a calcium transient at time zero (See troubleshooting, problem 1).

7. The MATLAB script will automatically open filterDesigner app, an interactive tool used to design and evaluate digital filters, which will be used to smooth the signal.
   a. Change the filter type from finite impulse response (FIR) to infinite impulse response IIR. Use the default values or customize the filter for your data.
   b. Select Design Filter and export the file using File > Export (to Workspace, as coefficients) and check the Overwrite Variables box.
   c. Close the app. The resulting filter will be automatically applied to the fluorescence time series.
   d. Evaluate adequacy of filtering.
      i. The script will prompt the user to select an ROI.
      ii. The pre- and post-filtered fluorescence signals will be displayed.
      iii. If satisfied with the result, select “yes”; to reject and redesign the filter, select “no”.
      iv. For more information on filterDesigner select Help on its dialog box (See troubleshooting, Problems 1 and 3).
Peak finding and impulse train determination from Ca$^{2+}$ GECI time-series fluorescence

- **Timing:** &lt;30 min

8. The MATLAB script will guide the user through a process that identifies peaks in the fluorescence time series' and converts them into binary impulse trains for each cell (Figure 4).

**Figure 3. Example of fluorescence time-series preprocessing**

(A) Remove background. Specifically, **Image > Stacks > 2 Project > Projection type (Median), Process > Image Calculator > Image 1 (Raw fluorescence) > Subtract > Image 2 (MedIP)**.

(B) Load pre-saved ROIs. Specifically, **Analyze > tools > ROI manager > open**.

(C) Select quantitative features (e.g., mean gray value, centroid). Specifically, **Analyze > set measurement**.

(D) Record quantitative features for all ROIs. Specifically, **Analyze > tools > ROI manager > More > multi measure**.
a. Input the parameters for the MATLAB function findpeaks, namely threshold, Min peak Height, Min distance between peaks, Min prominence, Min width, and Max width. Default values are suggested.

b. Evaluate the performance. The MATLAB script displays fluorescence vs time with the identified peaks superimposed for the most active cells.

c. Accept the peaks by selecting “yes” if they match what one would label manually.

d. Reject by clicking “no” if there are extra peaks or missed peaks. Then proceed with optimization of parameters (See troubleshooting, Problems 1 and 4).

e. Optimize peak-finding parameters focusing primarily on Min prominence to adjust for fluorescence amplitude and Min width and Max width to adjust for fluorescence duration.

f. Once accepted, peaks are automatically transformed into a matrix of binary impulses, B[t], which contains a “1” at the time point of each peak for each cell, and a “0” everywhere else. A histogram of the number of impulses per cell will be automatically displayed. This will be used to fit a probability distribution function and to create synthetic cell impulse trains with the same statistics below.

△ CRITICAL: Optimization of peak-finding parameters is critical because fluorescence kinetics and amplitudes different across experimental systems since they depend on many factors including cell type, microenvironmental context, and choice of genetically encoded calcium reporter.

Inference of cell communication pipeline based on “excess synchrony” metric

∈ Timing: ~1 h

Step 3 of this protocol infers putative cell communication events from the spatiotemporal synchrony of single cell calcium impulse trains. The strategy is illustrated in Figure 5. An overview of the logic and approach are presented first, followed by the detailed sub-steps.

For purposes of this protocol, synchrony, S(w, t), is defined as the number of impulses within a time window of length w, beginning at time t. Synchronous impulses can occur either because of true biological information transfer or simply because of chance. By first modeling the “chance” component, we can then limit our search for cell communication to those times during which the experimentally observed synchrony exceeds the amount of synchrony expected “by chance”. We call the difference, after normalization to window size, “excess synchrony”, ∆S/w.

Note: The reason for normalizing is because synchrony increases monotonically with window size. Normalizing to w enables comparison of excess synchrony across different window sizes.

To determine the synchrony expected by chance, we model the statistics of experimental impulse trains from “real cells”, fit it to a probability distribution, and then sample the distribution to create
generated impulse trains from “generated cells” having the same statistics. This allows calculation of generated synchrony due to chance alone, $S_{\text{gen}}(w, \tau)$. To convert generated synchrony into a single number that can be easily subtracted from real experimentally observed synchrony, $S_{\text{real}}(w, \tau)$, to calculate excess synchrony, we define a threshold, $S_{\text{th}}$, at a specified percentile of $S_{\text{gen}}(w, \tau)$ using a z score. Excess synchrony, $\Delta S/w$, is then calculated by subtracting $S_{\text{real}}(w, \tau)$ and $S_{\text{th}}$ and normalizing to $w$. Selection of $S_{\text{th}}$ via the choice of z score is a critical step because high $S_{\text{th}}$ values will predict few cell communication events but have fewer false positives than low $S_{\text{th}}$ values, which will predict more cell communication events but include more false positives (predicted communication where synchrony is actually due to chance alone).

---

**Figure 5. Diagram of cell communication pipeline inference steps**

(A–C) (A) “Real” experimental binary impulse trains are (B) modeled based on their impulse train statistics to create (C) “generated” binary impulse trains.

(D–F) Real and generated impulse trains are independently used to calculate synchrony, $S(w, \tau)$, specifically (E/F) $S_{\text{real}}(w, \tau)$ and $S_{\text{gen}}(w, \tau)$.

(G) A threshold called $S_{\text{th}}$ is defined based on generated synchrony and a user-generated z score.

(H) Excess synchrony, $\Delta S/w$.

(I and J) Cell communication events are identified at time $\tau_{\text{comm}}$’s and spatial locations $(X_{\text{comm}}$’s, $Y_{\text{comm}}$’s). Figure reprinted with permission from Taghdiri et al. (2021).
To determine the timing of putative cell communication events we derive a vector of communication times $\mathbf{t_{comm}}$ by grouping nearby values of $t$ for which excess synchrony $\Delta S/w$ is greater than zero. For each $\mathbf{t_{comm}}$, spatial analysis is limited to only those cells with calcium impulses between $\mathbf{t_{comm}}$ and $\mathbf{t_{comm}} + w$. k-Means clustering is performed and the centroid of the cluster with the greatest cell density is taken as the location of putative cell communication $\mathbf{X_{comm}}, \mathbf{Y_{comm}}$. The results of putative cell communication events and their component cells are reported in tabular form in a single *.csv file.

The following describes detailed sub steps with a focus on the how the user interacts with the script.

9. The MATLAB script will automatically open the interactive distributionFitter app to facilitate fitting a probability distribution function to the experimental impulse frequencies (Figure 6).
   a. In the parent window, select the Data tab and a daughter window will open.
   b. In the Data field of the daughter window, enter “ImpulseHistogram”, and then select Create Data Set.
   c. Returning to the parent window, select the NewFit… tab, which will open a new daughter window.
   d. In the Data field of the new daughter window, select “ImpulseHistogram”.
   e. In the Distribution field, select “negative binomial”.
   f. Finally, select Apply to create the probability distribution object and select Save to Workspace to save it.
   g. Close the app.
      i. The MATLAB script will perform a Chi square goodness of fit test using the built-in function chi2gof.
      ii. The MATLAB script will return a test decision for the null hypothesis that the experimental impulse histogram data comes from the specified probability distribution (selected from a dropdown in 9e).
      iii. If the test decision rejects the null hypothesis at the 5% significance level, then the app will reopen.
      iv. 9a–f must be repeated with a different distribution.
      v. Alternatively, the MATLAB script will proceed to automatically generate synthetic cell impulse trains by sampling from the approved distribution and placing the impulses at random time points. Therefore, the synchrony of resulting generated data will be normally distributed.

Note: The script will run automatically with the assistance of only two input parameters - window size, which determines temporal resolution, and a z-score, which determines the inclusiveness or exclusiveness of communication inference (default values are provided for both).
10. The user will be prompted, “Do you want to choose the window size automatically?” To use the default window size or specify it manually, answer “no.” Alternatively, to perform a systematic comparison of window sizes in search of the one yielding the maximum excess synchrony, answer “yes” (See troubleshooting, problem 5).

11. The user will be prompted to enter a z score, \( z \) (a default is provided), which will be combined with the normally distributed generated data of mean \( \mu \) and standard deviation \( \sigma \) to give a threshold synchrony \( S_{th} \) (See troubleshooting, problem 5).

\[
S_{th} = \mu + z \times \sigma
\]

\( \triangle \) CRITICAL: Optimization of the z score is critical. The default z score is set at 1.28, the 80th percentile. However, \( S_{th} \) is a tunable parameter, which allows the user to alter the stringency of this threshold. At lower percentiles (lower \( S_{th} \) values), the number of identified peaks will increase, resulting in more predicted events but more false positives (i.e., more synchronous impulses that are labeled communication will actually be due to chance). Conversely, at higher percentiles (higher \( S_{th} \) values), the number of identified peaks will decrease resulting in fewer predicted events but more risk of false negatives (i.e., missing true communication events). The z score should be optimized to balance the competing goals of maximizing the number of events detected (by lowering the z score) while minimizing the number of false positives (by increasing the z score). In practice, if one has a biologically negative control sample (i.e., few calcium transients per time) one can reduce the z score until an unacceptable number of cell communication events (of which all are considered false positives) are detected.

12. The MATLAB script will calculate Excess Synchrony \( \Delta S/w \) is calculated (Figure 7).

\[
\Delta S/w = [S_{real}(w, \tau) - S_{th}] / w
\]

13. The MATLAB script will return the timing of putative cell communication events which is vector of communication times \( t_{comm} \) by grouping nearby values of \( \tau \) for which excess synchrony \( \Delta S/w \) is greater than zero (Figure 8).

14. The script will open a window prompting the user to select a criterion for unsupervised learning algorithm that is used to solve clustering problems.

15. The script will calculate an optimal number of clusters \( k \), evalclusters.

16. The script will use optimal \( k \) to perform k-Means clustering, kmeans function in MATLAB (See troubleshooting, problem 6).

17. The script will return \( t_{comm} \) and perform spatial analysis on only those cells with calcium impulses between \( t_{comm} \) and \( t_{comm} + w \). Then the script will perform k-Means clustering and report the centroid of the cluster with the greatest cell density as the location for putative cell communication \( X_{comm}, Y_{comm} \) (Figure 9).

18. Upon completion the script will create an output file called FinalResults.csv and save all associated plots and figures. The FinalResults file contains a *.csv file with the following column headings: EventIndex, StartTime, EndTime, Duration, NumOfCells, CellIndex, CentroidX, CentroidY.

**Note:** For clarity, each row refers to a distinct cell and its participation in each cell communication event. Therefore, EventIndex, StartTime, EndTime, Duration, Number of Cells are properties of the event, while CellIndex, CentroidX, CentroidY are defined for the individual cells within each event (Figure 10).
This protocol describes an imaging-based method for quantifying single cell calcium reporter responses and inferring cell communication based on "Excess Synchrony". A computational pipeline analyzes the images and creates a list of putative cell communication events with associated timing, location, and component cells. This can then be studied using genetic or pharmacologic modulators of specific cell communication mechanisms to validate and probe mechanism via perturbation. The protocol is best suited for investigators with basic signal processing and computational experience and who have access to time-lapse fluorescence recordings of cells that are potentially communicating.

Figure 7. Determination of real and generated synchrony from single-cell impulse trains
Synchrony is defined as the sum of impulses within a window of length w beginning at time $\tau$. "Real" synchrony $S_{\text{real}}$ derives from real experimental single impulse trains. "Gen" synchrony $S_{\text{gen}}$ derives from generated single impulse trains. Excess synchrony is shown without normalization $\Delta S$. Figure reprinted with permission from Taghdiri et al. (2021).

EXPECTED OUTCOMES
QUANTIFICATION AND STATISTICAL ANALYSIS
Quantification and statistical analyses are described throughout the protocol.

LIMITATIONS
A limitation of our study is that we lack gold standard methods to validate whether correlated dynamics are indeed the result of cell communication. Some of our studies In vivo show spatially isolated communication events where propagation of calcium dynamics is evident by inspection. However, in other experiments where calcium reporters are highly dynamic and densely packed in tissue, it is difficult to confirm. Application of mutual information theory and causal inference may be helpful for determining whether correlated calcium dynamics are cell autonomous or are the result of cell-to-cell or environment-to-cell communication (Quinn et al., 2011; Wibral et al., 2013). The computational pipeline is applicable to fluorescence live cell time series imaging with high cell density. However, a dense and very active population would degrade the confidence to find the high spatiotemporal cells as a putative cell communication using k-Means clustering. A second important limitation is that our method cannot distinguish information transfer between cells from information transfer from the environment to neighboring cells (an external stimulus). An important assumption in the current instantiation is that there is only one event per time point within one field of view. To accommodate cases where the density and frequency of cell communication is high, one can separate the image into sub-images and run the pipeline on each sub-image. In this way, one can discover simultaneous but spatially distinct cell communication events within the same field of view.

Figure 8. Example of how varying $S_\text{th}$ affects predicted cell communication events
$S_\text{th}$ corresponding to the 50th, 80th, and 100th percentiles of $S_\text{gen}$ are illustrated. Communication is predicted where $S_\text{real}$ exceeds $S_\text{th}$ (e.g., where excess synchrony is greater than zero). The initiation of each putative communication event where $S_\text{real}$ sustains exceeds $S_\text{th}$ is deemed a putative communication event assigned to the start time $\tau_{\text{comm}}$.

Figure reprinted with permission from Taghdiri et al. (2021).
TROUBLESHOOTING

Problem 1
Fewer than expected binary impulses are observed when compared to visual inspection of the fluorescence time lapse images or the resulting raw fluorescence tracings (step 6, before you begin and step 7 step-by-step method details).

Potential solution

- If no calcium transients are observed, confirm the presence of Cre and GECI reporter genes by genotyping the mouse.
- Calcium transients may be unappreciated if their amplitudes are too low. If the background-corrected-amplitude is low compared to the noise, or only uses a small portion of the available dynamic range, increase the excitation source or the gain of the detector. We recommend maximizing the amplitude until your maximum biological signal of interest is near saturation. To minimize background, use phenol-red free medium and reduce other sources of autofluorescence.
- Image normalization can cause some calcium transients to be reduced in amplitude if the reference fluorescence is too high. A potential solution is to change the reference from median to initial fluorescence. Alternatively, one can reference to the median or initial fluorescence of a subset of the time-series that contains more characteristic cellular calcium responses.
- If the calcium transients appear blunted or if one perceives some are missing, it is possible the signal is undersampled. This is solvable by increasing the sampling rate during image acquisition. Alternatively, blunted, or missing transients may be the result of overfiltering, in which case, the user should increase the passband frequency during the digital filter design substep in order to preserve more signal detail. For more information on optimization of digital filters in MATLAB, please refer to the documentation for the function designfilt. Missing transients may also result from inadequate optimization of the findpeaks function above.
- If the cell type has not been characterized before, it is possible the cell type does not exhibit calcium transients in the biological and experimental context being tested. In this case, consider different stimuli, different culture media, or different time points for analysis. In vivo, consider modifying the disease model the time points analyzed.

Problem 2
The background fluorescence is steadily decreasing during calcium transient imaging due to photo-bleaching (step 1 step-by-step method details).
Potential solution
Instead of subtracting a constant median fluorescence from each time point (step 1 and Figure 3), one can remove dynamic background fluorescence by subtracting a windowed moving mean of the background fluorescence using the `movmean` function in MATLAB.

Problem 3
The signal is overly smooth after digital filtering (step 7 step-by-step method details).

Potential solution
Increase the lowpass filter passband frequency during filter design to preserve more signal detail if there are concerns that potentially important biological signals were removed during filtering. Reanalyze the pre- and post-filtered signals to confirm improvement.

Problem 4
More peaks are identified by the `findpeaks` function than one would expect from visual inspection of the fluorescence time series (step 8 step-by-step method details).
Potential solution
Change default parameters within the `findpeaks` function to alter the identified peaks and the location of impulses. The `findpeaks` function in MATLAB accepts six parameters, `MinPeakHeight`, `MinPeakWidth`, `MaxPeakWidth`, `Threshold` (the minimum height differences between adjacent peaks), `MinProminence`, and `MinDistPeak` (Figure 11). To optimize peak-finding focus primarily on `Min prominence` to adjust for fluorescence amplitude and `Min width` and `Max width` to adjust for fluorescence duration.

Problem 5
The predicted number of cell communication events are much greater or far fewer than the expected number of events (steps 10 and 11 step-by-step method details).

Potential solution
Reduce the z score, $z$, which will decrease $S_{th}$, and increase $\Delta S/w$ (Figure 8). This will result in more inclusive and less stringent criteria, which will predict more cell communication events but be more susceptible to including more synchrony that merely occurs by chance. Conversely, increasing the $z$ will increase $S_{th}$, and decrease $\Delta S/w$. This will result in a more restrictive and more stringent criteria, which will predict fewer cell communication events but be more susceptible to erroneously missing true cell communication events.

The window size, $w$, plays a critical role in distinguishing between calcium impulses that are due to cell communication and those that occur by chance. The pipeline optimizes it algorithmically by evaluating the maximum excess synchrony $\Delta S/w$ for all values of $w$ and selecting the window size that yields a local maximum with additional constraints on minimum and maximum values (Figure 12). Choose algorithmic window size optimization by answering “yes” when asked. Of note, this may either increase or decrease the number of communication events and the number of cells involved in each event.

Problem 6
Cells predicted to be synchronous appear spatially disperse rather than localized or are predicted in unexpected locations (step 14 step-by-step method details).

Potential solution
Once the cells are selected, spatial clustering is performed algorithmically using k-Means clustering without regard for the timing of calcium transients. The optimal k value is determined using the MATLAB function `evalclusters`, which allows for different clustering evaluation criterion — silhouette (default), gap, CalinskiHarabasz, or DaviesBouldin. The MATLAB script exposes this option to the user can compare clustering results for different criterion if the default does not yield expected or intuitive results. Finally, if a technical explanation is not uncovered, a biological explanation may also be possible. For example, it is possible that distant cells are communicating across seemingly long distances via fine processes.

Figure 11. Example of peak finding optimization
(A) Peaks identified using `findpeaks` with default parameters suggested by MATLAB.
(B) Peaks identified using `findpeaks` with optimized parameters suggested by inferring cell communication pipeline. Please refer to `findpeaks` function documentation in MATLAB if you want to adjust the parameters.
RESOURCE AVAILABILITY

Lead contact
Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Kevin.R. King (krking@ucsd.edu).

Materials availability
All unique/stable reagents generated in this study are available from the lead contact without restriction. Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact Kevin.R. King (krking@ucsd.edu).

Data and code availability
- This paper does not report Standardized datatypes. All data reported in this paper will be shared by the lead contact upon request.
- All original code has been deposited at https://github.com/Nikatag/inferring_cell_communication_pipeline and is publicly available.
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

ACKNOWLEDGMENTS
The work was funded by NIH R00HL129168 (K.R.K.) and DP2AR075321 (K.R.K.). We would also like to thank the UCSD Nikon Imaging Center for assistance with analysis.

AUTHOR CONTRIBUTIONS
N.T. and K.R.K. developed the method, performed all experiments and analysis, and wrote the manuscript.
DECLARATION OF INTERESTS

K.R.K. is Founder, Director, on Board of Directors, and holds equity in Nightingale Labs, which has no scientific overlap with the material in this protocol.

REFERENCES

Chen, T.W., Wardill, T.J., Sun, Y., Pulver, S.R., Renninger, S.L., Baohan, A., Schreiter, E.R., Kerr, R.A., Orger, M.B., Jayaraman, V., et al. (2013). Ultrasensitive fluorescent proteins for imaging neuronal activity. Nature 499, 295–300.

Gee, J.M., Smith, N.A., Fernandez, F.R., Economo, M.N., Brunert, D., Rothermel, M., Morris, S.C., Talbot, A., Polumbo, S., Ichida, J.M., et al. (2014). Imaging activity in neurons and glia with a Pol2a-based and cre-dependent GCaMP5G-ires-tdTomato reporter mouse. Neuron 83, 1058–1072.

Grienberger, C., and Konnerth, A. (2012). Imaging calcium in neurons. Neuron 73, 862–885.

McQuin, C., Goodman, A., Chernyshev, V., Kamentsky, L., Cimini, B.A., Karhohs, K.W., Doan, M., Ding, L., Rafelski, S.M., Thirstrup, D., et al. (2018). CellProfiler 3.0: next-generation image processing for biology. PLoS Biol. 16, e2005970.

Quinn, C.J., Coleman, T.P., Kiyavash, N., and Hatsopoulos, N.G. (2011). Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. J. Comput. Neurosci. 30, 17–44.

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.

Taghdiri, N., Calcagno, D.M., Fu, Z., Huang, K., Kohler, R.H., Weissleder, R., Coleman, T.P., and King, K.R. (2021). Macrophage calcium reporter mice reveal immune cell communication in vitro and in vivo. Cell Rep. Methods 1, 100132.

Wibral, M., Pampu, N., Priesemann, V., Siebenhuhner, F., Seiwert, H., Lindner, M., Lizer, J.T., and Vicente, R. (2013). Measuring information-transfer delays. PLoS One 8, e55809.