Novel Tools and Methods

Synthetic Data Resource and Benchmarks for Time Cell Analysis and Detection Algorithms

Kambadur G. Ananthamurthy and Upinder S. Bhalla

https://doi.org/10.1523/ENEURO.0007-22.2023

National Centre for Biological Sciences - Tata Institute of Fundamental Research, Bellary Road, Bengaluru - 560065, Karnataka, India

Abstract

Hippocampal CA1 cells take part in reliable, time-locked activity sequences in tasks that involve an association between temporally separated stimuli, in a manner that tiles the interval between the stimuli. Such cells have been termed time cells. Here, we adopt a first-principles approach to comparing diverse analysis and detection algorithms for identifying time cells. We generated synthetic activity datasets using calcium signals recorded *in vivo* from the mouse hippocampus using two-photon (2-P) imaging, as template response waveforms. We assigned known, ground truth values to perturbations applied to perfect activity signals, including noise, calcium event width, timing imprecision, hit trial ratio and background (untuned) activity. We tested a range of published and new algorithms and their variants on this dataset. We find that most algorithms correctly classify over 80% of cells, but have different balances between true and false positives, and different sensitivity to the five categories of perturbation. Reassuringly, most methods are reasonably robust to perturbations, including background activity, and show good concordance in classification of time cells. The same algorithms were also used to analyze and identify time cells in experimental physiology datasets recorded *in vivo* and most show good concordance.

Significance Statement

Numerous approaches have been developed to analyze time cells and neuronal activity sequences, but it is not clear whether their classifications match, nor how sensitive they are to various sources of data variability. We provide two main contributions to address this: (1) a resource to generate ground truth labeled synthetic two-photon (2-P) calcium activity data with defined distributions for confounds such as noise and background activity, and (2) a survey of several methods for analyzing time cell data using our synthetic data as ground truth. As a further resource, we provide a library of efficient C++ implementations of several algorithms with a Python interface. The synthetic dataset and its generation code are useful for profiling future methods, testing analysis toolchains, and as input to computational and experimental models of sequence detection.

Introduction

The mammalian hippocampus is important for the formation of several kinds of memory, one of which is the association between stimuli occurring separately in time. Time cells were originally described using tuning curves from single-unit recordings of cellular activity when rats ran on a running wheel in between behavioral decisions (Pastalkova et al., 2008). These cells exhibited time tuning of the order of seconds. Several further studies have shown that small populations of hippocampal CA1 cells fire in time-locked sequences, "bridging" the time gap between stimulus and response in temporal delay tasks lasting several seconds (Pastalkova et al., 2008; MacDonald et al., 2011, 2013; Kraus et al., 2013). Cellular calcium imaging studies have also been used to report time cells, albeit at slower sampling rate (Modi et al., 2014; Mau et al., 2018). For example, similar interval tiling properties of hippocampal CA1 neurons were observed on much shorter, 500 ms timescales in a Trace Eyeblink Conditioning (TEC) task (Modi et al., 2014). Spontaneous sequential activity

Received January 2, 2022; accepted February 9, 2023; First published February 23, 2023.
The authors declare no competing financial interests.

Author contributions: K.G.A. and U.S.B. designed research; K.G.A. performed research; K.G.A. and U.S.B. analyzed data; K.G.A. and U.S.B. wrote the paper.

has also been reported in free-running animals (Villette et al., 2015). Such cells with a well-defined temporal firing field are commonly termed time cells (MacDonald et al., 2011; Eichenbaum, 2017). However, there is a wide diversity of methods used to detect and characterize time cells, and it is not clear how consistent these methods are in classifying cells as time cells. It is also unclear how sensitive each method may be to a range of physiological sources of variability and noise. A consistent set of benchmarks of classification performance is necessary to draw accurate and comparable conclusions from real physiology data across different methods and different laboratories. Our approach in the current study is not prescriptive, but pragmatic: we ask how existing methods work when we already know exactly which cells are time cells, and we determine how well each method deals with imperfect data.

The major approaches used to identifying time cells are tuning curves (peristimulus time histograms), temporal information (TI), principal component analysis with time offset, support vector machines, and bootstrap analysis of activity peaks. Several studies have used a temporal delay task lasting several seconds, in which a rat runs on a treadmill during the delay period. A temporal information metric (Mau et al., 2018) has been used to find individual time cells in such tasks. A distinct task involves monitoring recurrent sequences of activity during free-running treadmill recordings. Such datasets have been analyzed using offset principal component analysis (Kaifosh et al., 2013; Villette et al., 2015; Malvache et al., 2016), to first denoise two-photon (2-P) data, establish correlation coefficients, and detect hippocampal CA1 sequences. Time cells have also been reported for much shorter duration tasks (~500 ms) such as hippocampus-dependent trace conditioning (Tseng et al., 2004; Modi et al., 2014). Time cells in these 2-P datasets were identified using yet another method, in which bootstrapping was used to determine whether peak activity at a given time was different from chance. This method was termed ratio of ridge/background (Modi et al., 2014). Yet other methods have utilized support vector machines to categorize time cells (Ahmed et al., 2020). Additionally, while the applicability of a variety of algorithms for place cell detection has been previously compared (Souza et al., 2018), we have focused on methods which are fully automatable and which scale well to large datasets, specifically comparing algorithms to detect time cells.

Time cell detection is closely related to sequence detection, which has been fraught with statistical challenges. For example, detection of synfire chains has been the subject of some debate (Ikegaya et al., 2004; Lee and Wilson, 2004; Mokeichev et al., 2007; Schrader et al., 2008). Time cell detection is usually easier, in that in most experiments there is a well-defined initiating stimulus and a known delay or trace phase (however, see Villette et al., 2015). For any cell identified as a time cell, it is desirable to define a score to measure quality or reliability along with decodable time. Hence it is also valuable to be able to compare the score of a time cell across recordings and even between groups, using well defined, analog measures. Each algorithm currently used in the literature implements a different scoring method and it is as yet unclear whether comparable results would be observed with other metrics.

In the current study, we compare these diverse methods by estimating their performance on synthetic test datasets where we controlled all features of the data, including the identity and timing of each time cell. The development of a synthetic dataset serves two purposes. First, it facilitates principled comparison of different methods, since the ground truth is known. Second, it facilitates an analysis over many dimensions of input variance, corresponding to very different experimental and neural circuit contexts. Richness in variety of input data allows for better sampling of the performance of the analyses under many potential conditions. We have explored variance along the key dimensions of noise, timing imprecision, signal widths, frequency of occurrence, as well as several others. To strengthen the applicability of this synthetic data resource to real data, our generated output uses sampled experimental data.

Our experimental data, synthetic dataset, and code base are intended to be a resource for algorithm testing and optimization.

Materials and Methods

Animals, chronic implants, and behavioral training

All animal procedures were performed in accordance with the National Centre for 114 Biological Sciences Institutional Animal Ethics Committee (project ID NCBS115 IAE-2016/20(M)), in accordance with the guidelines of the Government of India (Animal Facility CPCSEA registration number 109/1999/CPCSEA) and equivalent guidelines of the Society for Neuroscience.

To chronically monitor the activity of the same population of hippocampal CA1 cells, we implanted two- to four-month-old male and female GCaMP6f mice [Tg(Thy1-GCaMP6f)GP5.17Dkim JAX stock #025393] with an optical window and head-bar using a protocol adapted from previously published methods (Dombeck et al., 2010). Briefly, anesthesia was induced with 2–3% isoflurane in a chamber, and subsequently maintained (breathing rate of ~1 Hz) with 1–2% isoflurane, directly to the mouse's nose using an inverted pipette tip. Surgery was performed on a

K.G.A. and the experiments received support from the Department of Biotechnology Grant BT/PR12255/MED/122/8/2016 (to U.S.B.). National Centre for Biological Sciences-Tata Institute of Fundamental Research (NCBS-TIFR) provided resources for analysis equipment and support to U.S.B. through intramural funds supported by the Department of Atomic Energy, Government of India, Project Identification No. TRI 4006.

Acknowledgments: We thank Daniel Dombeck for help with the chronic preparation for physiological 2-P calcium recording of hippocampal CA1 cells and William Mau for his Temporal Information calculation code. We also thank Vinu Varghese and Bhanu Priya Somashekar for helpful comments on this manuscript.

Correspondence should be addressed to Upinder S. Bhalla at bhalla@ncbs.res.in.

https://doi.org/10.1523/ENEURO.0007-22.2023

temperature-controlled table, maintained at 36.5°C, while the anaesthetized animal was cheek-clamped. After a haircut, a ~5 cm piece of scalp was cut open to reveal the skull. A ~3 mm circular craniotomy was then performed at a position 2 mm caudal and ~1.5 mm lateral to bregma, on the left hemisphere. After gently tearing off the dura, the underlying cortex was carefully aspirated till the corpus callosum (CC) layer, clearing out any blood using repeated washes of cortex buffer (Modi et al., 2014). A small thickness of corpus callosum fibers were then carefully aspirated till horizontal CC fibers were sparse but visible. The cortex buffer was then carefully suctioned out to dry the exposure till tacky. The exposure was then quickly sealed using a thin layer of Kwik–Sil and a coverslip attached to the bottom of a 3 mm steel cannula. This preparation left the CA1 cell body layer ~200 μm below the most exposed tissue. Finally, an imaging head-bar was surgically implanted and fixed to the scalp, using dental cement and skull screws, before the animal was brought out of anesthesia.

The animals were allowed to recover for 1–5 d after implantation, with a further 3–4 d of habituation to the rig. Following this simultaneous behavioral training and 2-P *in vivo* imaging was conducted.

Trace Eyeblink Conditioning (TEC)

We standardized a multi-session Trace Eyeblink Conditioning (TEC) paradigm to train head-fixed mice, based on previous literature (Siegel et al., 2015). TEC involves an association between a previously neutral conditioned stimulus (CS) with an eyeblink inducing unconditioned stimulus (US), across an intervening, stimulus-free, trace interval. Training involved 60 trials per session, one session a day, for approximately two weeks. The CS was a 50 ms blue LED flash while the US was a 50 ms air-puff to the left eye. The stimulus-free trace interval was 250–750 ms long, depending on the session. Additionally, a pseudorandom 10% of the trials were CS-only probe trials (no US) to test for learning. All behavior routines were controlled by programmed Arduinos. Eyeblinks were measured for every trial, by video camera (Point Gray Chameleon3 1.3 MP Monochrome USB3.0) based detection.

The conditioned response (CR) is observed as a preemptive blink before the US is delivered, in animals that learn the task. The analysis of the behavioral data was performed using custom written MATLAB scripts. In brief, each frame for every trial was:

1. Cropped to get the eye;
2. Binarized to get the pixels defining just the eye, and finally;
3. Given an FEC score from 0 to 1 (see below).

Every trial was then scored as a hit or miss, using the result of a two-sample Kolmogorov-Smirnov test between the FEC during the trace and pre-CS period (1% significance). The performance of an animal for a session was then established as the percentage of hit trials/total trials.

Definitions:

FEC: The fraction of eye-closed is estimated by counting the pixels defining the eye in every image of a time series, normalized by the maximum number of pixels defining the eye, in that session. Thus, every frame was given an analog score from 0 to 1, where,

- 0: fully opened eye
- 1: fully closed eye

CR: The conditioned response is the eye-closing transition during the trace period.

UR: The unconditioned response is the eye-closing transition when the US is delivered.

Performance: Percentage of hit trials/total trials. This allowed us to observe how the animals perform during and across sessions.

Two-photon imaging

We used a custom-built two photon laser-scanning microscope (Modi et al., 2014) to record calcium activity from 100–150 hippocampal CA1 cell bodies *in vivo*, at cellular resolution. We performed galvo-scans through the imaging window, over a field of view of ~$100 \times 100\ \mu m^2$, at 14.5 Hz, during TEC (Fig. 1*A*). An Arduino microcontroller was used to control the behavior routines, and it additionally sent a TTL trigger to initiate the imaging trials. The behavior and imaging were conducted simultaneously to record calcium activity when the animal was learning the task.

Time-series fluorescence data for various cells was extracted using Suite2P (Pachitariu et al., 2017). All further analysis and code development was done on MATLAB R2017b and batch analysis runs were performed on MATLAB R2021a. The average of the fluorescence values for cell specific pixels is then converted into the fold change relative to the baseline (dF/F0; F0 as 10th percentile), for every marked cell, in every trial (Fig. 1*B*). These dF/F traces were used for the rest of the analysis.

Curating a library of calcium events

For all synthetic data experiments, we used one good quality 2-P recording session's worth of data from one animal. We mapped our imaging dataset into a matrix of dF/F values for all cells, trials, and frames. We then identified calcium events as signal deviations that were above a threshold (mean ± 2*SD) for more than four consecutive frames (frame rate: 14.5 Hz or ~70 ms per frame). Once identified, we curated a library for each event by a cell, and saved the respective start indices and widths. Using this library, we generated synthetic data by inserting experimental calcium events into the time series for each simulated cell. This approach just uses a time series of signal bins and amplitudes, hence is signal-agnostic and could be applied to other imaging and recording modalities. In the interests of data integrity, our synthetic datasets were watermarked to be distinguishable from real physiology datasets.

Generating synthetic data

Synthetic data were generated using a custom-written MATLAB function script "generateSyntheticData()" in the provided code repository. We preallocated and set up a 3-D matrix of zeros (as cells, trials, frames), and added

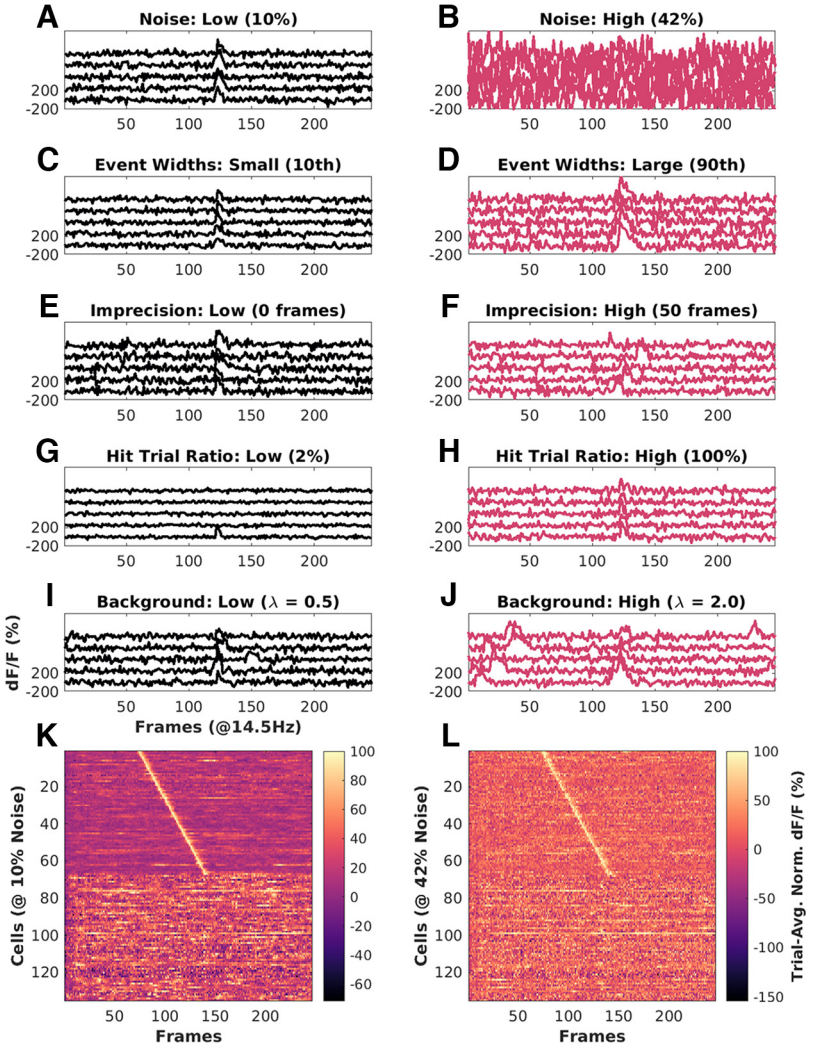

Figure 1. Key features of synthetic datasets. Left, Black panels, Low range of features. Right, Red panels, High range of features. ***A***, Noise = 10%. ***B***, Noise = 42%. ***C***, Event width: 10th percentile +/− 1 SD. ***D***, Event width 90th percentile +/1 SD. ***E***, Imprecision at 0 frames FWHM. ***F***, Imprecision at 50 frames FWHM. ***G***, Hit trial ratio from 0% to 2%. ***H***, Hit trial ratio from 0% to 100%. ***I***, ***J***, Background activity with the number of background spikes per background sampled from a Poisson distribution for with mean (λ), for ***I***: $\lambda = 0.5$ (low), and ***J***: $\lambda = 2.0$ (high). ***K***, ***L***, Trial-averaged Calcium traces from example synthetic datasets of 135 neurons, displayed as heatmap sorted by time of peak Ca signal. ***K***, Baseline physiology synthetic data trial-average with 10% noise (low) and high background activity ($\lambda = 2$ to 3 events/trial). ***L***, Same as ***K*** with 42% noise (high) and comparable background activity ($\lambda = 2$ to 3 events/trial). In both cases, 50% of the cells (top 67) are time cells and the remainder are not. Extended Data Figure 1-1 describes the most important parameters modulated for datasets in each of the three parameter regimes, "Unphysiological," "Canonical," and "Physiological," along with the false positives and false negatives, for each of the 10 implemented algorithms.

calcium events sampled from the Calcium Event Library at frames (time) determined by the synthesis algorithm. The input parameters to this algorithm included timing, noise, imprecision, event width selection, hit trial ratio, background activity, and several others. We aimed to cover the most likely conditions to affect timing and other experiment design properties. In more detail, we generated synthetic datasets using the following control parameters:

- Time cell percent

Value: Number between 0 and 100. This sets the number of cells that are assigned tuned calcium activity as a percentage of total cells, and controls the number of positive and negative class cells in the dataset.

- Cell order

Value: 'basic' or 'random.' In 'basic' mode, time cells are indexed lower than other cells. In 'random' mode, the indices of time cells and other cells are randomly selected. This should have no impact on algorithm detection but is useful for visualization.

- Max hit trial percent

Value: Number between 0 and 100. This sets the maximum possible fraction of total trials, during which a Time Cell will exhibit tuned calcium activity.

- Hit trial percent assignment

Value: 'fixed' or 'random.' In 'fixed' mode, the number of hit trials is set as defined by max hit trial percent. In 'random' mode, the number of hit trials is calculated by randomly picking a value from a range (½*max hit trials, max hit trials).

- Trial order

Value: 'basic' or 'random.' In 'basic' mode, the hit trials are indexed lower than miss trials. In 'random' mode, the indices of hit and miss trials are randomly selected. Specific patterns of hit and miss trials for a session have not been reported in physiology, so this feature is not implemented.

- Event width

Value: {0–100 percentile value, Integer N}. For each cell, this defines the selection of events based on width in frames. The percentile value is estimated from the histogram of all event widths. The variance of this selection is set by "N," which adds N*SD to the selection. All synthetic cells exhibit a range of different calcium events. This is considered an important parameter.

- Event amplification factor

Value: Number from 0 to $+\infty$. This allows additional control to multiplicatively amplify any chosen calcium event, before incorporation. Our library was curated from physiologically recorded signals. The default value is 1.

- Event timing

Value: 'sequential' or 'random.' In 'sequential' mode, the time of peak calcium activity is reflected by the indexing of the time cells. In 'random' mode, the time of peak calcium activity is randomly dispersed over the trial frame points.

- Start frame

Value: Number from 0 to total number of frames. This sets the timing of the first cell in a time cell sequence.

- End frame

Value: Number from 0 to total number of frames. This sets the timing of the last cell in a time cell sequence.

- Imprecision full width at half max (FWHM)

Value: Number from 0 to total number of frames. This sets the lower and upper bounds for the difference in timing of calcium activity across trial pairs for a time cell. We use this parameter to model trial to trial variability and is considered an important parameter to test.

- Imprecision type

Value: 'none,' 'uniform,' or 'normal.' In 'uniform' and 'normal' modes, the trial pair Imprecision is picked from a normal and uniform distribution, respectively. In 'none' mode, the trial pair Imprecision defaults to 0.

- Noise

Value: 'Gaussian' or 'none.' In 'Gaussian' mode, the noise is sampled as a time-series vector with points from a Gaussian distribution. In 'none' mode, the noise percent defaults to 0.

- Noise percent

Value: Number from 0 to 100. This allows scaling for any sample noise point, based on the max signal value for any cell.

- Add background spikes for time cells

Value: Boolean 0 or 1. This switch controls the incorporation of background (untuned) activity for putative time cells.

- Add background spikes for other cells

Value: Boolean 0 or 1. This switch controls the incorporation of background (untuned) activity for other (nontime) cells.

- Background distribution mean

Value: Number from 0 to $+\infty$. This sets the mean (λ) of the Poisson distribution to sample from when selecting how many background events to add per trial, for any given cell.

Implementation of a reference quality measure, Q

In order to compare the readouts from the various time-cell detection methods, we implemented a reference measure of quality (Q) of synthetic time cells that used the known inputs to the generation algorithm.

Based on preliminary analysis, we selected following five parameters as the most likely to affect the behavior and detection of time cells:

1. Noise
2. Event width
3. Imprecision
4. Hit trial ratio
5. Background activity

Accordingly, we were able to calculate a reference quality measure, using the following equation:

$$\text{RefQ} = \text{HTR} \times \exp - \{\alpha \times \text{MNP}/100 \times \text{EAF} + \beta \times \text{std. dev. EW}/\text{meanEW} + \gamma \times \text{std. dev. Imp}/\text{Stim Win}\}, \quad (1)$$

where HTR: hit trial ratio
MNP: max noise percent (%)
EAF: event amplification factor
EW: event widths (frames)
Imp: imprecision (frames)
Stim Win: stimulus window (frames)
α: 1
β: 1
γ: 10

The values of α, β, and γ, were set to have comparable effects of each of the terms inside the exponent. This reference Q was useful for debugging code and was the basis for a further metric for time cell classification discussed below. A representative synthetic activity trace for 'low' and 'high' values of each of these five parameters is shown in Figure 1.

All modulations for the datasets in this study along with the estimates for false positives and false negatives, across all algorithms are shown in Extended Data Figure 1-1.

Separate analysis modules were developed for three categories of analysis

We implemented three analysis modules: *ti*, *r2b*, and *peq*, shorthand for temporal information, ridge-to-background, and parametric equations. The *ti* module implements three algorithms from Mau et al. (2018). The *r2b* module implements two algorithms from Modi et al. (2014). The peq module computes estimates for noise, hit trial ratio, event width and imprecision, and estimates a Q score as above. All three methods were implemented in C++ with a PyBind11 interface to Python. This combination is fast and efficient in memory use, and also has the ease-of-use of Python. Thanks to the native MATLAB interface to Python, all three methods can also be called from MATLAB.

Synthetic datasets generated and analyzed in batch mode

We generated datasets pertaining to parameter sensitivity analysis by modulating one of the four main parameters and setting the others to noninterference levels. In this manner, we devised 99 cases to study in which one of the main parameters was varied. Note that in these cases the resultant activity was in an unphysiological regime because other sources of variation were kept to low levels so as not to interfere with the parameter of interest. With three randomized shuffles, we generated 297 unique datasets.

We wanted to use more realistic datasets, where we would modulate one of the four parameters while keeping the others to ranges typical of physiological data. We devised 12 canonical cases. With 10 randomized shuffles each, we generated 120 additional unique datasets in the canonical regime. Finally, we devised 12 physiological regime cases, identical to those in the canonical regime, with the addition of background (untuned) activity. This yielded another 150 datasets, with randomization.

Altogether, we had 567 unique datasets for our tests, each with 135 cells (total: 76,545 cells), 60 trials, and 246 frames/trial. Except when the percent time cells were modulated, all datasets featured 50% time cells.

We next implemented an analysis pipeline to run all the datasets through the time cell detection algorithms, yielding scores and predictions for each case. Finally, all the scores and predictions were collated for comparison and benchmarks as shown in the schematic (Fig. 2).

Metrics for time cell classification performance

Recall is inversely proportional to the number of false negatives (Type II error) and is the fraction of true positive class predictions over all positive class ground labels.

$$\text{Recall} = \text{TPR}/(\text{TPR} + \text{FNR}) \tag{2}$$

Precision is inversely proportional to the number of false positives (Type I error) and is the fraction of true positive class predictions over all positive class predictions.

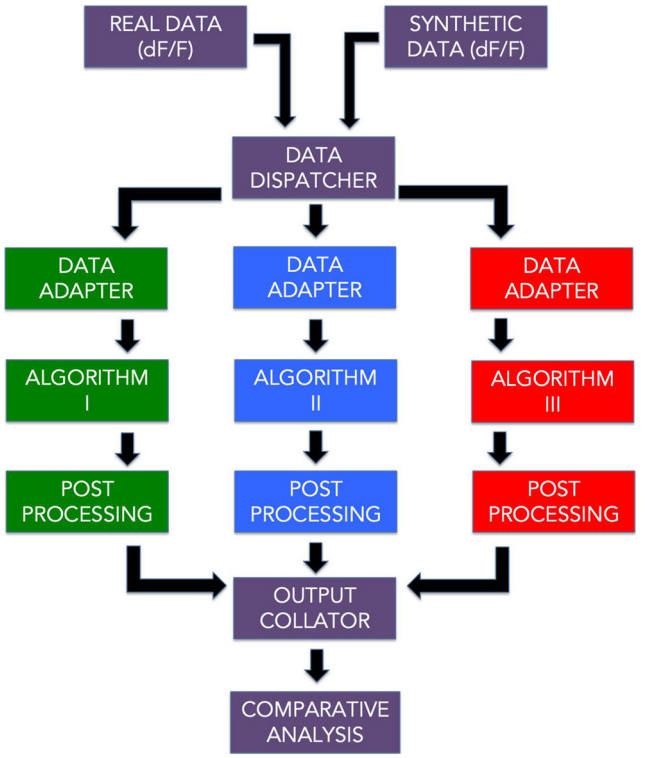

Figure 2. A schematic representation of the analysis pipeline. Physiology data as well as synthetic data were analyzed by 10 different implemented algorithms and the output was collated for comparative benchmarks.

$$\text{Precision} = \text{TPR}/(\text{TPR} + \text{FPR}). \tag{3}$$

F1 Score is the harmonic mean of recall and precision.

$$\text{F1 Score} = 2 * \text{Precision} * \text{Recall}/(\text{Precision} + \text{Recall}), \tag{4}$$

where

TPR: true positive rate

FNR: false negative rate

FPR: false positive rate

Here are the definitions for predictive/classification performance evaluation (Table 1).

Here are the important functions provided in the code base (Table 2).

Here are the MATLAB scripts running the comparative analysis and figure generation (Table 3).

Code and resource availability

The code/software described in the paper is freely available online at https://github.com/BhallaLab/TimeCellAnalysis. The code is available as Extended Data 1.

Results

We developed a pipeline (Fig. 2) with 10 different algorithm implementations for time cell detection, which involve scoring and then classifying cells.

Here, we describe the implementation of each of the methods.

Table 1: Definitions for predictive/classification performance evaluation

Ground truth	Prediction/classification	Remark
0/false/other cell	0/false	True negative (TN)
0/false/other cell	1/true	False positive (FP)
1/true/time cell	0/false	False negative (FN)
1/true/time cell	1/true	True positive (TP)

For each detection algorithm, the classification results were compared with known ground truth values to get the total number of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) cases.

Time cell scoring methods and classification

Temporal information: *tiBoot*, *tiMean*, *tiBoth*, *tiMean-O*, *tiBase-O* (Mau et al., 2018)

Here, we used the algorithm from Mau et al. (2018) as follows. There was an initial criterion of cells to have activity in at least 25% of trials. Their activity was summed into event time histograms with a bin size of three frames. The temporal information (TI) was estimated using Equation 5,

$$\mathrm{TI} = 1 \times \lambda \mathrm{j} \times \log 2 \lambda \mathrm{j} \times \mathrm{Pj}, \quad (5)$$

where, λ is the average transient rate for each cell;

λ j is the average transient rate for each cell in bin "j";

Pj is the probability that the mouse was in time bin "j."

Bootstrapping was used to determine whether each cell had a TI greater than chance. We circularly randomized the frame points to develop a random activity model (1000 iterations) and classified cells as time cells if $\lambda > \lambda$rand in >99% of the models for at least two consecutive bins. We implemented the activity filter from Mau et al. (2018); by considering the trial-averaged peak of the calcium traces for each of the cells, and testing for significance using bootstrapping (*tiMean*). A logical AND operation between the prediction lists for tiBoot and tiMean, provided us with the full Mau et al., 2018 Temporal Information based detection algorithm (*tiBoth*).

Additionally, we used Otsu's threshold (Otsu, 1979) on the temporal information scores as well as the trial-averaged peaks to get *tiBase-O* and *tiMean-O* using the MATLAB function "graythresh()" (https://in.mathworks.com/help/images/ref/graythresh.html). The purpose of adding the Otsu's threshold-based classification step was to study how well the scores could be classified with a fast thresholding method, rather than the computationally expensive bootstrap.

Table 2: List of important functions provided in the code base

Name	Description	Command line	Location	Language
synthesis Demo.m	Command line demo, output to file: "synthData-demo.mat". Generates a synthetic 2-P time cell dataset file	$ cd TimeCellAnalysis/rho-matlab/demos && matlab -nodisplay -nosplash -r "synthesisDemo; quit"	rho-matlab/ demos	MATLAB
ti_demo.py	Command-line demo, output to console.	$ python TcPy/ti_demo.py sampleData/sample_ synth_data.mat	TcPy	Python interface and C++ numerics
r2b_demo.py	Command-line demo, output to console. Runs Ridge-to-Background analysis from Modi et al. (2014). Reports R2B Mean and R2B Bootstrap classifications	$ python TcPy/r2b_demo.py sampleData/sample_ synth_data.mat	TcPy	Python interface and C++ numerics
peq_demo.py	Command-line demo, output to console. Runs parametric equation analysis from current study. Reports PEQ threshold classification, and estimates for noise, event width, imprecision, and hit trial ratio for dataset	$ python TcPy/peq_demo.py sampleData/sample_ synth_data.mat	TcPy	Python interface and C++ numerics
ground_truth_- check.py	Command-line demo, output to console. Uses synthetic data files to assess accuracy of classification by the various Mau and Modi algorithms	$ python TcPy/ground-truth_ check.py sampleData/sample_synth_data.mat	TcPy	Python interface and C++ numerics
Benchmark.py	Command-line demo, output to console. Simple time and memory benchmarks for the Mau, Modi, or PEQ algorithms	$ python TcPy/run_batch_ analysis.py sampleData/ sample_synth_data.mat	TcPy	Python interface and C++ numerics
run_batch_a- nalysis.py	Command-line production script, output to CSV files. Runs a batch analysis using all methods on a data file. Generates .csv filesfor TI, R2B, PEQ, and ground truth classifications	$ python TcPy/ti_demo.py sampleData/sample_ synth_data.mat	TcPy	Python interface and C++ numerics
pyBindMap.py	Provides an interface for MATLAB programmers, to the python/C__ fuynctions using two wrapper functions: runTIanalysis and runR2Banalysis	Utility function, not run from command line	TcPy	Python
dodFbF.m	Utility function to convert experimental raw 2p calcium activity data from Suite2P to df/F form.	Utility function, not run from command line	rho-matlab/ CustomFu- nctions	MATLAB

All these functions should be run from the cloned repository, TimeCellAnalysis.

Table 3: List of paper figure generating scripts

Name	Description	Command line
paperFigures Synth.m	Plots all figures estimating algorithm performance for synthetic data analysis (paper Figs. 1, 4–6, and 8)	$ matlab -r "paperFiguresSynth"
paperFigures Real.m	Plots all figures estimating algorithm performance for real physiology data analysis (paper Fig. 7)	$ matlab -r "paperFiguresReal"
paperFigures Splits.m	For diagnostics; plots figures estimating algorithm performance over all the regimes (unphysiological, canonical, and physiologic)	$ cd ../src && matlab -r "paperFiguresSplits"

All these functions should be run from the cloned respository, TimeCellAnalysis/ρ-matlab/paperFigures.

Ratio of ridge/background, *r2bMean*, *r2bBoot*, *r2bBase-O* (Modi et al., 2014)

Here, we re-implemented the algorithm from Modi et al., 2014. The time of peak response for each cell was identified in averaged, nonoverlapping trials' worth of ΔF/F traces, in the CS-onset to US-onset period, or as specified. The rest of the trials were averaged and the summed area under the time of peak was estimated. The ridge was then defined to be a 200 ms window centered at the peak. Next, we calculated the summed area in the ridge window as well as the background (non-ridge frames) to get the ridge to background ratio. As a control condition, these traces were given random time-offsets and then averaged. An independent time of peak was identified for each random-offset, averaged trace and ridge to background ratio calculated for it. This bootstrapping was repeated 1000 times for each cell's data and averaged. The reliability score was then calculated individually, for each cell, as the ratio of the ridge to background ratio for aligned traces to the mean of that of the random-offset traces (*r2bMean*).

We also studied the significance of each cell's raw r2b values by comparing them to each of the r2b values of the randomized datasets, thresholding significance at the 99th percentile (*r2bBoot*). Finally, the raw r2b values were also thresholding using Otsu's Thresholding (*r2bBase-O;* Otsu 1979).

Parametric equations, *peqBase* and *peqBase-O* (in-house)

We developed this method to score cells in a manner similar to the reference quality, which uses the known ground truth of the input parameters given to the generator functions for the synthetic dataset. Rather than using the known inputs, this method computes the corresponding parameters read out or estimated from the dataset, whether synthetic or real. It is applicable to labeled or unlabeled datasets. It is defined as:

$$\mathrm{Q} = \mathrm{HTR} \times \exp - \{\alpha \times \mathrm{N/S} + \beta \times \text{std. dev. EW/mean EW} + \gamma \times \text{std. dev. Imp/Stim Win}\}, \quad (6)$$

where HTR: hit trial ratio
N/S: estimated noise/signal
EW: read out event widths (frames)
Imp: estimated imprecision (frames)
Stim Wind: stimulus window (frames)
α: 10
β: 1
γ: 10

While $10\times\alpha$ was required, β, and γ, were inspired by the same used for reference Q. Classification was then performed using Bootstrapping (as described above) as well as Otsu's threshold.

All of these implemented algorithms can handle unlabeled (real) or ground truth labeled (synthetic) data.

A schematic to describe the steps involved in each algorithm is shown (Fig. 3). We were then able to run all our synthetically generated datasets through each of the 10 implemented algorithms and perform comparative benchmarks.

Good predictive power in time cell quality scores despite different distributions

We ran each of the analysis methods on our synthetic datasets to assess how they scored the (known) time cells. There were four methods that provided a scoring function for time-cell classification: *tiMean*, *tiBase*, *r2bBase*, and *peqBase* (Fig. 4*A–D*). By inspection, these methods appeared to have distinct distributions. Below we describe how we compare the distributions using correlation analysis. In subsequent sections we describe other methods in our study that used these scores to generate a categorization through thresholding or bootstrap.

In these synthetic experiments, time cells were generated with a single calcium event per hit trial. Event insertions into the synthetic datasets were subject to noise, variable selection of event widths, trial-pair or timing imprecision, and hit trial ratio. We generated 99 unique unphysiological combinations (3× randomized shuffles) 12 unique canonical regime combinations (10× randomized shuffles), as well as 15 unique physiological regime combinations featuring background activity (10× randomized shuffles). In all, we performed our comparative analysis studies using 567 datasets, each with 135 cells, 60 trials/session, and 246 frames/trial at 14.5 Hz). We found that only *tiMean* and *tiBase* had a correlation coefficient of ~0.6, whereas other pairs were correlated below 0.4 (Fig. 4*E*).

Generalized linear regression (GLM) models were generated to look for the ideal thresholding value for the best classification predictions by each method. We used the MATLAB implementation of GLMs (fitglm(); https://in.mathworks.com/help/stats/fitglm.html). This is a linear model assuming a binomial distribution of categories (0 or 1, i.e., other cell or time cell; Collett, 2002). We obtained good predictive power for the four methods that provided a scoring function for time-cell classification. We generated Receiver Operating Characteristic (ROC) curves by going over the full range of thresholds for the range of scores for each method

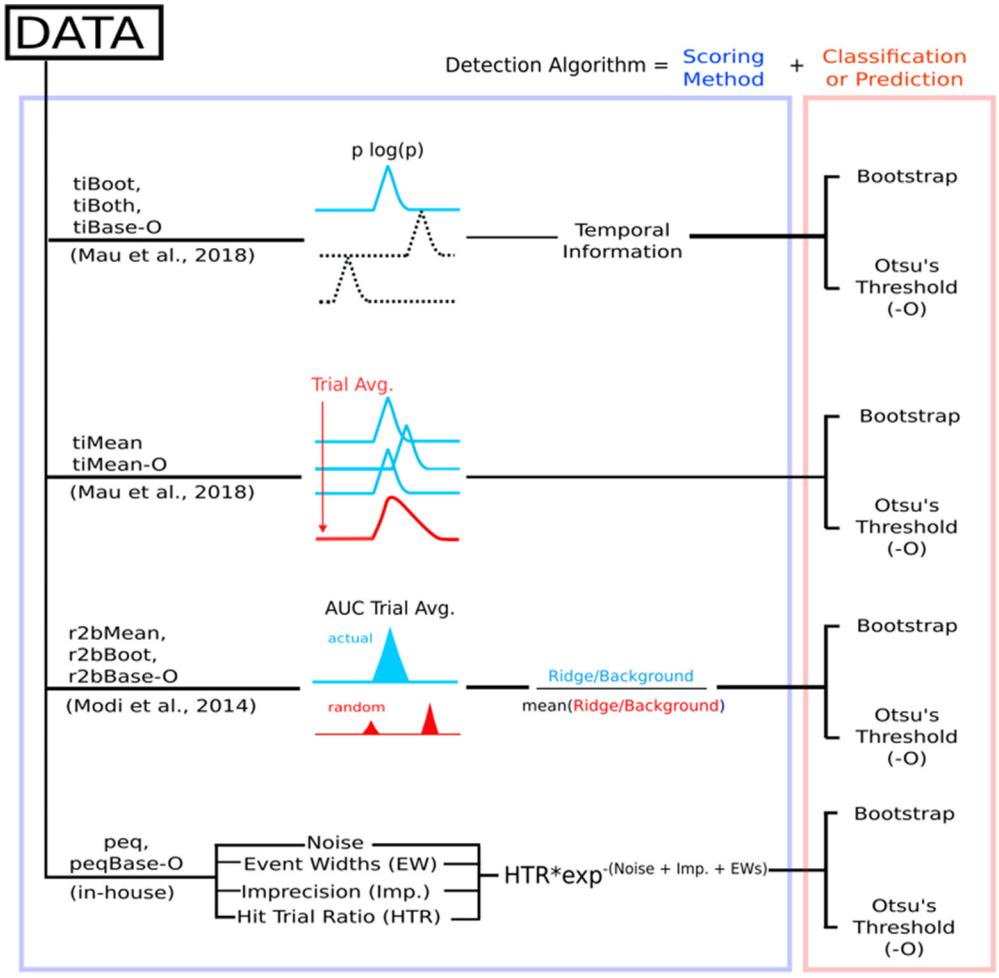

Figure 3. Schematic representation of the implemented algorithms, involving four different scoring methods followed by a classification step (bootstrapping or Otsu's automatic threshold) to have 10 complete time cell detection algorithms.

(ROC curves; Fig. 4*F*). We found that each distribution of scores had good predictive power, since ideal thresholds could be found to maximize TPR/FPR in all cases. We used the *tiBoth* categorization to distinguish time cells (Fig. 4*G*) from other cells (Fig. 4*H*), and plotted trial-averaged calcium traces to visually assess quality of classification as seen from raw data. Overall, each of our methods had distinct distributions of their base scores, but all had good predictive power for classification. The outcome of the classification steps is described in the next sections.

All algorithms exhibit near perfect precision with good recall

Next, we used the scores to classify the cells in our synthetic datasets, compared the predictions to ground truth, and established summaries for true and false cases. Confusion matrices were estimated to compare the predictions (classifications) for each algorithm, with reference to ground truth, and are shown (Fig. 5*A*,*B*). All methods exhibit very good precision (true positive classifications over the sum of all positive classifications), suggesting low false positive rates (Type I error; Fig. 5*C*). Most algorithms also generate good values for recall (true positive classifications over ground-truth positives). We observed F1 scores (harmonic mean of recall and precision) >0.75, all the way to 1 (perfect score), for most of the algorithms, as shown (Fig. 5*C*), suggesting overall usability.

We noticed moderate to strong correlation (>0.8) between the Boolean prediction lists for *tiMean*, *tiBoot*, *tiBoth*, *r2bMean*, and *r2bBoot* (Fig. 5*D*), but only weak to moderate correlation (<0.6) between the other pairs of predictions. The *tiMean-O* method does slightly better (correlation ~ 0.7 with the first five methods).

Algorithms differ in memory use and speed

Hardware and runtime requirements are a secondary, but practical concern when designing analysis of large datasets, and are specially relevant for experiment designs that require online analysis. We therefore looked at how memory use and runtime scaled on a per dataset basis when considering 67 or 135 cells per dataset ($2\times$).

We ran the memory usage and runtime experiments on a gaming laptop (Lenovo Ideapad 3 Gaming) with a 6 core AMD Ryzen 5 4600H, 16 GB DDR4 RAM (3200 MHz) running MATLAB R2021a on Ubuntu 20.04. Note, however, that we have implemented all the time cell algorithms in

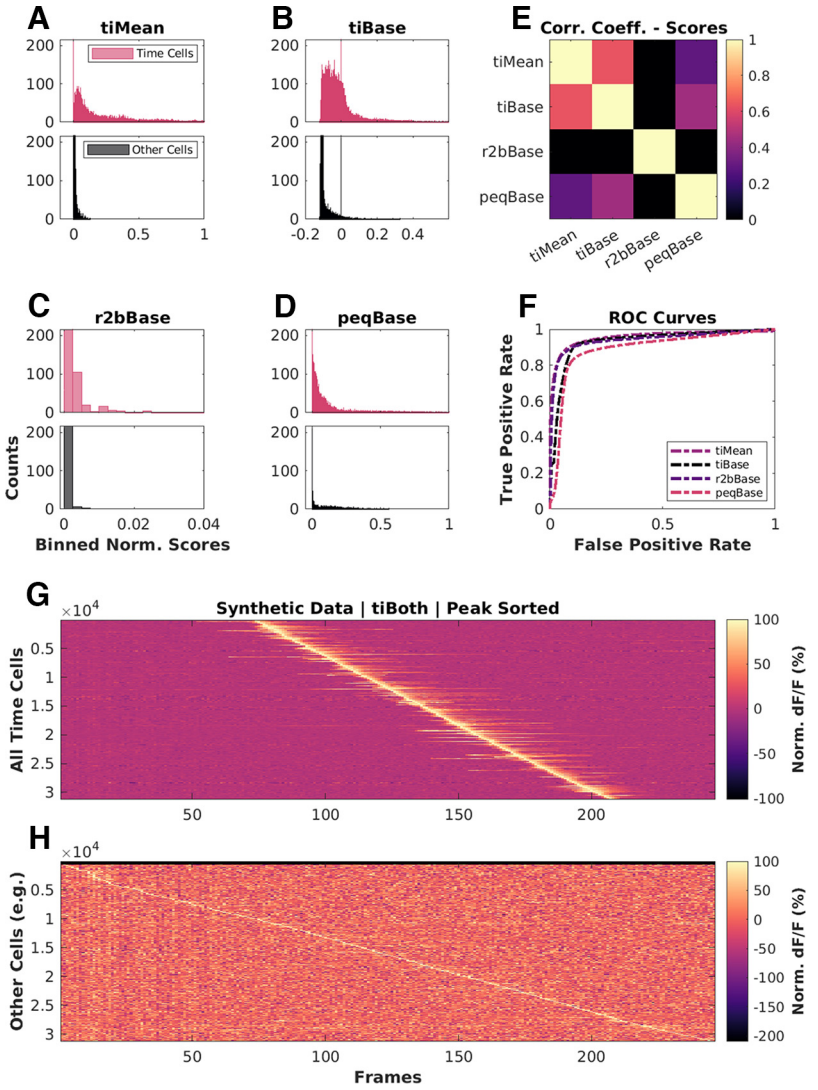

Figure 4. Base scores for different methods differ in their distributions but all have good predictive power. Scores for top (blue): time cells; bottom (red): other cells, across ***A***, *tiMean*; ***B***, *tiBase*; ***C***, *r2bBase*; ***D***, *peqBase*. ***E***, Pairwise correlation coefficients between the distributions of analog scores (pooling time cells and other cells) by each of the four scoring methods. ***F***, Receiver-operator characteristic (ROC) curves after generalized linear regression using the respective distributions of scores and comparisons with known ground truth. ***G***, ***H***, Trial-averaged calcium activity traces for cells classified as ***G***, time cells; ***H***, other cells.

serial and these do not use the additional cores. We found that most algorithms ran to completion requiring ~15 MB/dataset at a rate of ~1–4 s/dataset (135 cells/dataset). With 67 cells/dataset, the memory requirement and runtimes are approximately halved, suggesting that computational costs in memory and time were roughly linear with dataset size. We note that the analysis algorithms work independently for each cell. Thus, in principle, the analysis could be run in an embarrassingly parallel manner and should scale well on multicore architectures.

The synthesis of the main benchmarking datasets ($N = 567$ datasets or 76,545 total cells) required a more powerful analysis machine, running a 6 core AMD Ryzen 5 3600, 32GB of DDR4 RAM, running MATLAB R2021a on Ubuntu 20.04. Dataset batches up to ~30 datasets ($N = 40{,}500$ cells), however, could be easily handled by a less powerful laptop. The memory usage and runtime for 135 cells per dataset were accordingly, ~30 MB/dataset requiring ~1 s to complete. Thus, the methods scale readily to handle large datasets on modern hardware.

Physiologic range tests show sensitivity to noise but not to other features of the dataset

We next set out to see how these methods would work in estimated physiological ranges of signal confounds. Given our categorical labels on the synthetic data, we were able to split the datasets to look for the effects of the five main parameters: noise, event widths, imprecision, hit trial ratio, and background activity. We first computed the baseline physiology readouts keeping noise to 10%, event widths to the 60th percentile (±1 SD), imprecision to 0 frames, hit trial ratios to a range of 33–66%, and background activity to 0.9–1.2 events/trial for time cells

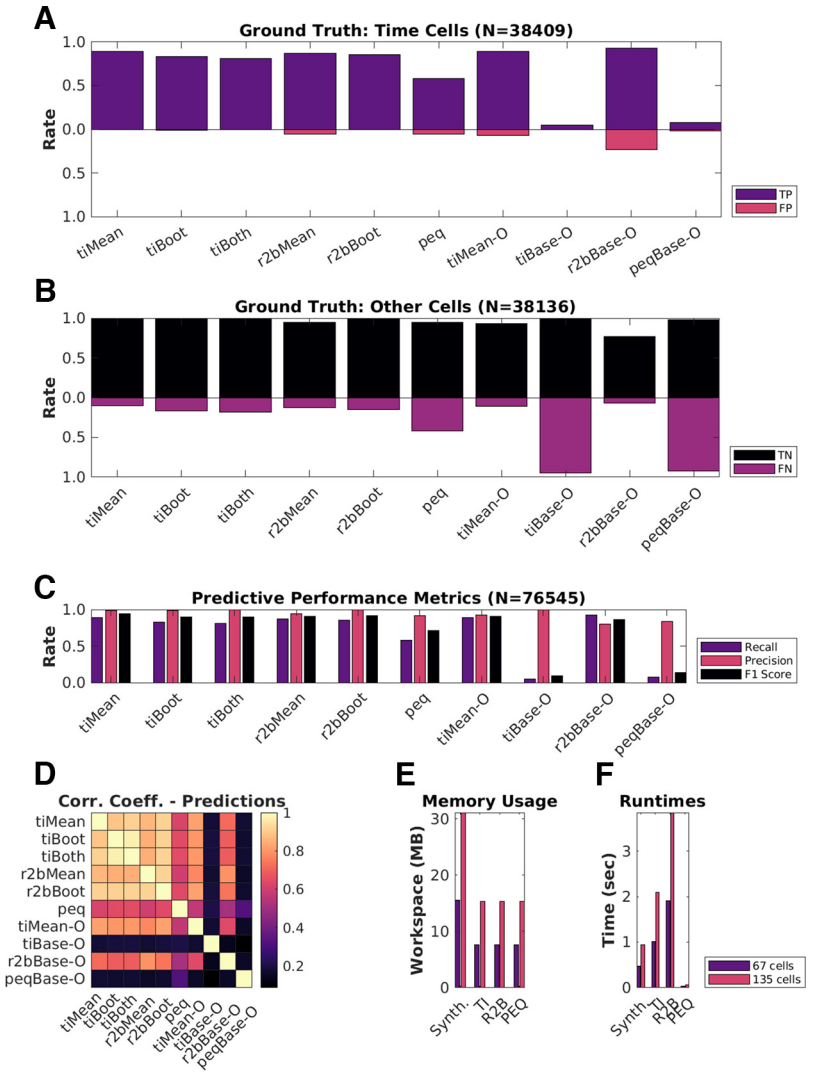

Figure 5. Good predictive performance by all algorithms. ***A***, ***B***, Classification performance of each of the 10 implemented detection algorithms. ***A***, True positives (TP; purple), false positives (FP; red). ***B***, True negatives (TN; black), false negatives (FN; purple). ***C***, Predictive performance metrics [Recall = TP/(TP + FN), Precision = TP/(TP + FP), and F1 Score = Harmonic mean of Recall and Precision] to consolidate the confusion matrices. ***D***, Pairwise correlation coefficients between the Boolean prediction lists by each of the 10 detection algorithms. Note that the first six methods correlate strongly. ***E***, Average memory usage per dataset by the implemented algorithms on datasets with either 67 cells (purple) or 135 cells (red). ***F***, Average runtimes per dataset by the implemented algorithms on datasets with either 67 cells (purple) or 135 cells (red).

(~50% of all synthetically generated cells, $N = 50$ baseline datasets, 135 cells/dataset, 60 trials/dataset). Next, we established dependency slopes for each of the algorithms, based on their predictions ($N = 10$ randomized shuffles for each case; Fig. 6*B–F*; Extended Data Figs. 6-1, 6-2).

Most methods exhibited a negative dependence of noise (range: 10% to 70%) on prediction F1 score (Fig. 6*B*). Although many methods are designed with some form of denoising strategy (trial-averaging, etc.), as expected all algorithms ran into classification difficulties at higher Noise levels. This reinforces the value of relatively high signal-to-noise recordings.

The relative insensitivity to event widths (Fig. 6*C*) is potentially useful for calcium imaging datasets where events may be slow, and in cases where slower tuning curves are expected. However, this criterion may need to be stringent for analyses that need to precisely identify fine differences in cell responses.

We observed that most algorithms were insensitive to how frequently time cells were active across trials in a session (HTR). This is possibly the reason for the potential confusion among physiologists with regard to how many time cells were expected in a recorded dataset.

We found that the first six algorithms (*tiMean*, *tiBoot*, *tiBoth*, *r2bMean*, *r2bBoot*, and *peq*) gave equivalent predictions in ~66% of cases (Extended Data Fig. 6-1*A*). Next, we considered the various prediction lists across these top six algorithms and looked for consensus in time cell predictions from the most lenient threshold (“>=1” algorithm), incrementally through the most stringent threshold (“=10” algorithms). We thus established a

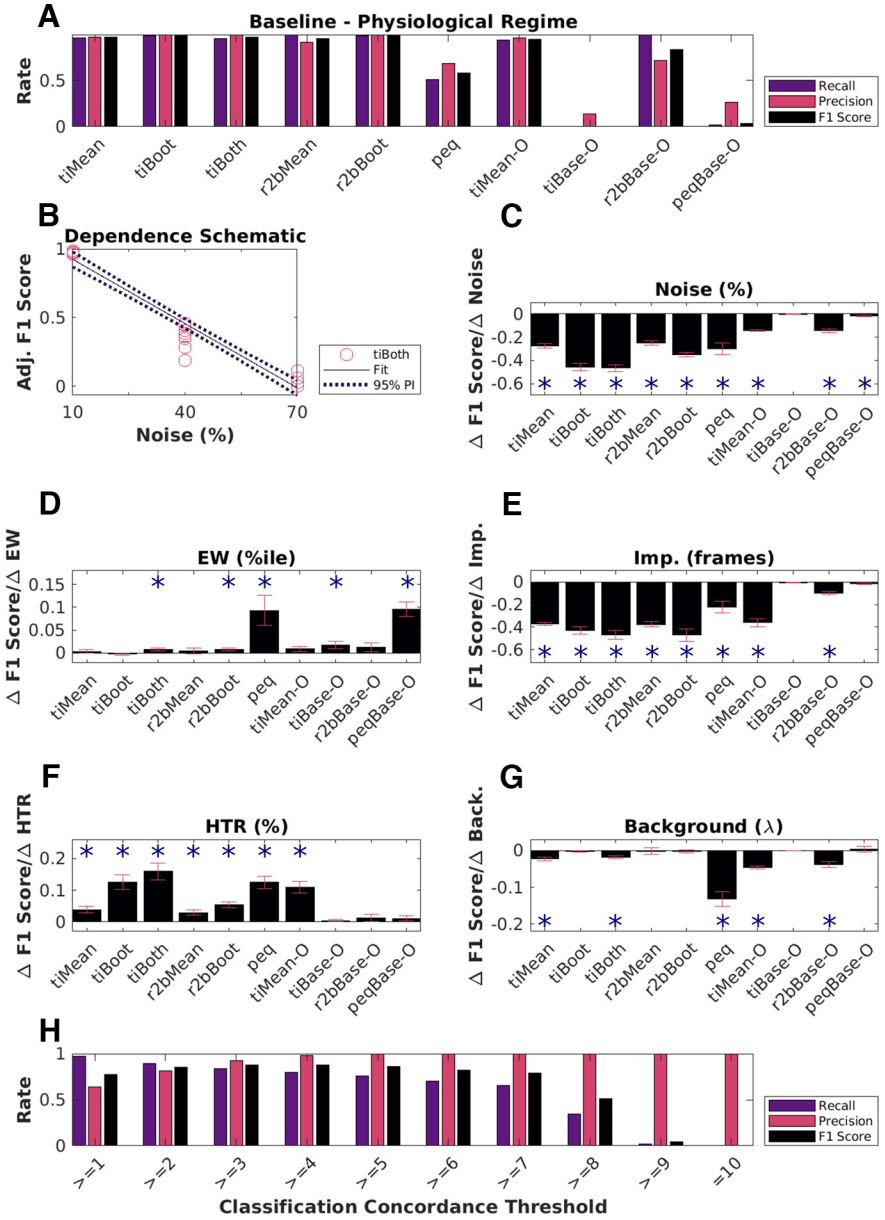

Figure 6. Physiological sensitivity analysis and concordance. ***A***, Classification performance scores for all algorithms with the baseline physiology synthetic datasets ($N = 6750$ cells). The first five methods perform well. Peq does poorly by all measures when confronted with physiology-range activity variability. Otsu's threshold method for score classification also does not work well for any method under physiological conditions. ***B***, Dependence of F1 score on noise as a schematic. This has an overall negative slope (dashed line) which was used for panel ***C***, TI-both. A similar calculation was performed for each method. Panels ***C–G***, Parameters were systematically modulated one at a time with respect to baseline and the impact on classification score for each algorithm was estimated by computing the slope, using repeats over 10 datasets each with an independent random seed. Significant dependence on the perturbing parameter was determined by testing whether the slope differed from 0 at $p < 0.01$, indicated by asterisks using the MATLAB function coefTest(). Plotted here are bar graphs with mean and error as RMSE normalized by the square root of N ($N = 10$ datasets). ***C***, Dependence on noise %. ***D***, Dependence on event width percentiles. ***E***, Dependence on imprecision frames. ***F***, Dependence on hit trial ratio (HTR; %). ***G***, Background activity (Poisson distribution mean, λ). ***H***, Classification performance using concordance for a range of classification thresholds. Extended Data Figure 6-1 describes the three-point line plot dependency curves for the F1 score for each of the implemented algorithms against each of the five main parameters modulated, as the mean of $N = 10$ datasets for each case, with error bars as SD. Extended Data Figure 6-2 showcases the linear regression fits for the same, with 95% prediction intervals (PIs), used to estimate the slopes of the various dependency curves.

Concordance based metric for time cell classification. We tested the predictive power of this Concordance based metric, which considers time cells based on consensus among the predictions from all the 10 implemented algorithms. We identified differences in the classification performance, across the full range of concordance thresholds (Fig. 6*H*). With lower threshold values (">=4" and below), we notice a slight drop in the Precision, indicating an

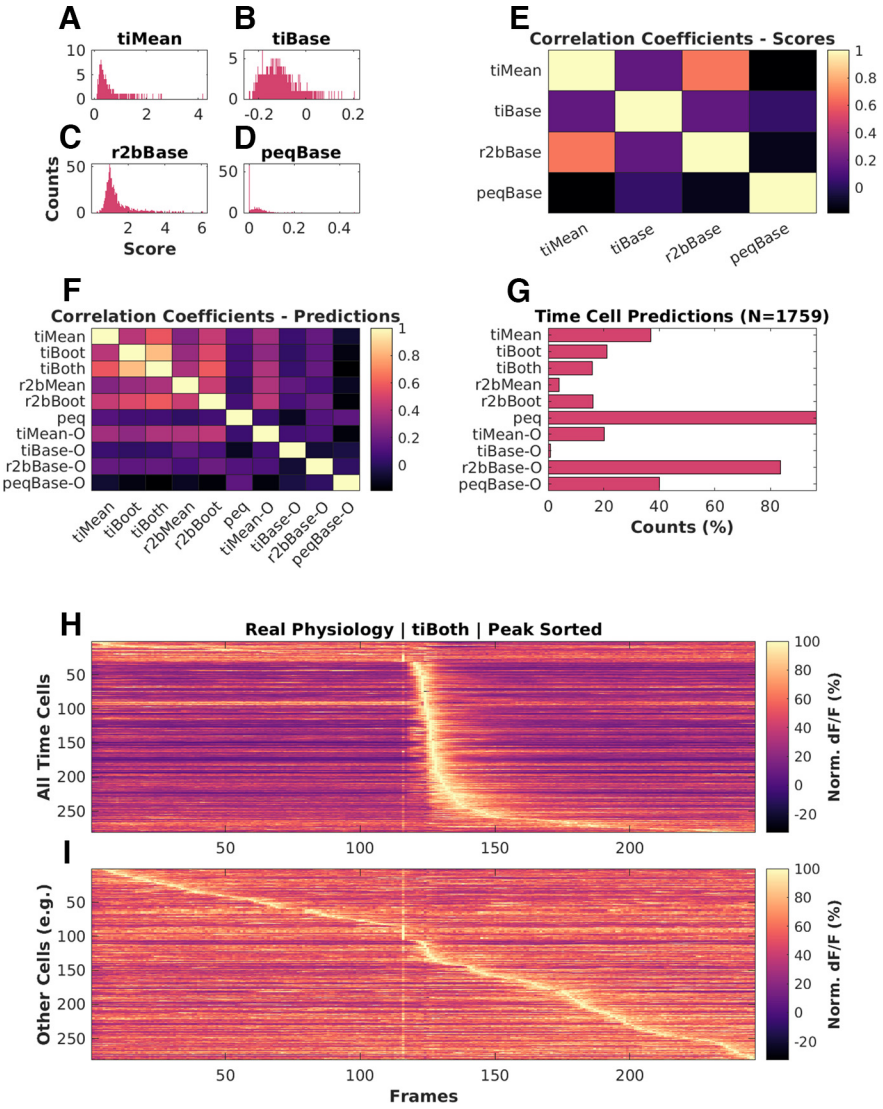

Figure 7. Analysis of experimental 2-P recordings of Ca^{2+} signals. ***A–D***, Histograms of scores for physiologically recorded *in vivo* calcium activity from hippocampal CA1 cells (total $N = 1759$), by (***A***) tiMean, (***B***) tiBase, (***C***) r2bBase, and (***D***) peqBase. ***E***, Pairwise correlation coefficients between the distributions of analog scores by the four scoring methods. ***F***, Pairwise correlation coefficients between the Boolean prediction lists by the 10 detection algorithms. ***G***, Numbers of positive class (time cell) predictions by each of the detection algorithms. ***H***, ***I***, Trial-averaged calcium activity traces for (***H***) time cells and (***I***) other cells. LED conditioned stimulus (CS) is presented at frame number 116, as seen by the bright band of the stimulus artifact. Most cells classified as time cells are active just after the stimulus. There is a characteristic broadening of the activity peak for classified time cells at longer intervals after the stimulus. Some of the cells at the top of panel ***H*** may be false positives because their tuning curve is very wide or because of picking up the stimulus transient. Similarly, some of the cells in the middle of panel ***I*** may be false negatives because of stringent cutoffs, although they appear to be responsive to the stimulus.

increase in false positive rate (Type I error). On the other hand, with increasing threshold values it is the Recall that drops, suggesting a higher false negative rate (Type II error). We find that a concordance threshold of ">=4" achieves the best recall, precision, and F1 scores, for time cell prediction (Fig. 6*F*). The utility of this approach is subject to the availability of resources to apply multiple algorithms to each dataset.

Time cells identified in real physiology recordings

We used the 10 different implemented algorithms on *in vivo* 2-P calcium recordings ($N = 13$ datasets, namely, 1759 isolated cells from three animals across chronically recorded datasets), to compare time cell classification between the algorithms. As we observed for the synthetic data, experimental 2-P Ca traces also yielded different base scores from the four different methods (Fig. 7*A–D*) Again, consistent with the synthetic data, the pairwise correlation was weak to moderate (Fig. 7*E*). When we consider the boolean prediction lists (Fig. 7*F*), we observed moderate pairwise correlation between *tiMean*, *tiBoot*, *tiBoth*, *r2bMean*, and *r2bBoot* (>0.5), and low or weak correlation between the other pairs (<0.5). This was consistent with observations for the synthetic data but the correlations were overall slightly

weaker. The total number of time cells predicted were also different across the implemented algorithms (Fig. 7*G*). Algorithms such as *r2bBase-O* and *peq*, which had more false positives (Fig. 5*B*) also had more cells classified as time cells. The converse was not true. *r2bMean*, which had moderate false negatives as well as false positives on the synthetic dataset, classified very few of the experimental set as time cells. The trial-averaged activity of the detected time cells (Fig. 7*H*; including false positives) and other cells (Fig. 7*I*), based on the predictions by *tiBoth*, are shown. The experimentally recorded time cells exhibited a characteristic widening of tuning curves (Pastalkova et al., 2008; MacDonald et al., 2011, 2013; Kraus et al., 2013; Mau et al., 2018) with tuning to later time points (Fig. 7*H*).

Overall, four of the algorithms from the literature seemed consistent in their classifications as well as having reasonable numbers of classified time cells. These were the three algorithms from Mau et al. (2018; *tiMean*, *tiBoot*, and *tiBoth*), and the *r2bBoot* method derived from Modi et al. (2014). This is broadly in agreement with their performance on the synthetic datasets.

Discussion

We have developed a full pipeline for comparing time cell detection algorithms. This starts with synthetic datasets for benchmarking, in which we program in the ground truth of cell identity and timed activity, and a range of perturbations characteristic of experiments. These include noise, event widths, trial-pair timing imprecision, hit trial ratio, and background activity. This resource is, in itself, a key outcome of the current study, and though it is designed for 2-P calcium imaging data it can be extended to rate-averaged single-unit recordings. We built a pipeline for running and comparing the outcome from five methods derived from two previous studies, and one from the current work. These algorithms were applied to synthetic and experimental datasets and compared against each other and, where possible, against ground truth. We observed that most algorithms perform well and substantially agree in their time cell classification, but there were different degrees of sensitivity to different forms of signal variability, notably noise and imprecision.

The value of synthetic data in experimental science

Synthetic neural activity datasets are valuable in at least two main ways: evaluating algorithms for detection of important activity features, and for delivering stimuli to *in vitro* and simulated neurons, so as to provide a more physiological context in which to study input-output properties (Abbasi et al., 2020). While we have deployed our synthetic dataset for the specific purpose of comparing time cell detection algorithms, we suggest that it could also be useful for evaluating sequence analysis algorithms (Ikegaya et al., 2004; Foster and Wilson, 2006; Villette et al., 2015). Beyond the domain of neuronal data analysis, such synthetic datasets act as a test-bed for critique and development of analysis algorithms meant for deployment on real-world or typical use case data. They have been used previously to benchmark unsupervised outlier detection (Steinbuss and Bohm, 2020), explainable machine learning (Liu et al., 2021), intrusion detection systems (Iannucci et al., 2017), 3D reconstruction algorithms (Koch et al., 2021), among several others. We report the first use of synthetic data pertaining to cellular physiology in the context of identifying time cells from network recordings. Moreover, our experiments study important operational differences across several previously published and new detection algorithms.

Our dataset may also be valuable for the second use case, stimulus delivery. There is a growing body of work on network detection of sequences (Ikegaya et al., 2004; Foster and Wilson, 2006; Csicsvari et al., 2007; Jadhav et al., 2012; Villette et al., 2015; Malvache et al., 2016) or even single-neuron sequence selectivity (Branco et al., 2010; Bhalla, 2017). More realistic input activity patterns with a range of physiological perturbations may be useful probes for such experimental and theoretical studies. Further, experimenter-defined neural activity inputs through optogenetic stimulation has already begun to use more complex temporal patterns than static or periodic illumination (Schrader et al., 2008; Dhawale et al., 2010; Bhatia et al., 2021). Our approaches to synthetic sequential neuronal activity generation may be useful to add more physiological dimensions to the sequential activity employed in such studies.

Further dimensions of time cell modulation

Our experiments allowed us to probe for parametric dependence systematically across published and new algorithms. We observed little or no dependence of the predictive performance (F1 score) of the various algorithms to event widths, hit trial ratios, and background activity. We did observe the F1 scores for most algorithms to be negatively dependent on noise and imprecision. On the one hand, this is a useful outcome in that different methods yield similar time-cell classification. It is a limitation, however, if the network uses such response features for coding, since it means that these methods are insensitive to relevant response changes. Further potential coding dimensions were not explored. Thus, several potential behavioral correlates of tuned cells (Ranck, 1973), could not be studied in our experiments. Such correlates include but are not limited to measurements of spatial navigation (O'Keefe and Dostrovsky, 1971; O'Keefe and Nadel, 1978; Wilson and McNaughton, 1993) and decision-making (Foster and Wilson, 2006; Csicsvari et al., 2007; Diba and Buzsáki, 2007; Davidson et al., 2009; Karlsson and Frank, 2009; Gupta et al., 2010; MacDonald et al., 2013; Villette et al., 2015), as well as navigation across tone frequencies (Aronov and Tank, 2014). While each of these further inputs would be interesting to incorporate into synthetic datasets, this requires that the time cell generation algorithm itself incorporate some form of simulation of the neural context. This is beyond the scope of the current study.

A specific limitation of our dataset is that it assumes that time is encoded by individual neurons. This leaves out population encoding schemes in which no one cell responds with the level of precision or consistency that would clear the criteria we use. For example, many of the same studies that use the methods tested here also use

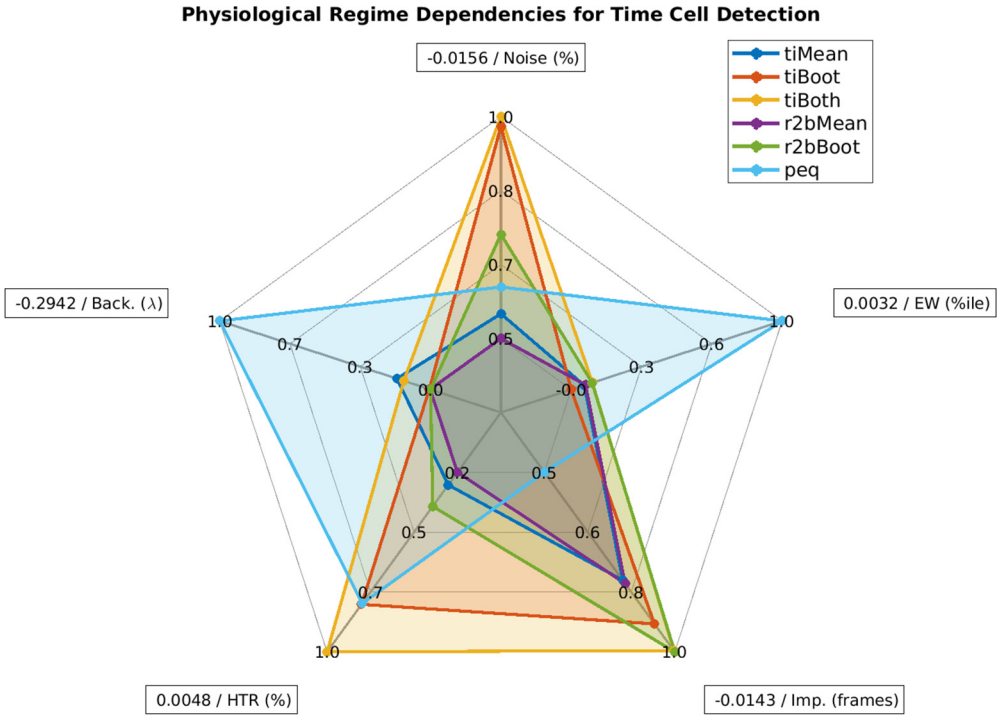

Figure 8. Spider plot summary. Relative sensitivity of the six best detection algorithms (*tiMean, tiBoot, tiBoth, r2bMean, r2bBoth*, and *peq*) to the five main parameters for data variability, noise (%), event widths (%ile), imprecision (frames), hit trial ratio (%), and background activity (λ). A perfect algorithm would have very small values (i.e., low sensitivity) for each of the parameters and, thus, occupy only the smallest pentagon in the middle. Note that even the maximal absolute value of sensitivity for most parameters (outer perimeter) is quite small, indicated in boxes at the points of the spider plot.

neural network decoders to report time (Mau et al., 2018). Such decoders might detect time encoding without time cells. A similar situation of individual versus network coding appears for the closely related problem of sequence representation. Place cell replay sequences have been shown to be modulated by the prevalence of location specific aversive (Wu et al., 2017) as well as appetitive stimuli (Bhattarai et al., 2020). Such physiological findings have been the subject of theoretical models of behavior planning (Foster, 2017; Mattar and Daw, 2018), and have been reported to improve performance on multiple Atari games by artificial neural networks (Mnih et al., 2015) featuring salience detection and experience mapping. We suggest that synthetic data for such higher-order encoding schemes might be a useful tool, and could draw on the approaches in the current study.

Comparative analysis benchmarks and concordance

A particularly challenging time cell classification problem is when the same cells may play different timing roles, such as forward and reverse replay. This is made more difficult because of the relative rarity of forward replay sequences over the more typical reverse replay (Diba and Buzsáki, 2007; Foster, 2017). Preplay is also a topic of some debate (Dragoi and Tonegawa, 2013; Foster, 2017). At least one possible problem in such debates is the degree of consistency between time cell or sequence classifiers. Our pipeline allows for (1) error correction in case of nonconcordant classifications, (2) suggest candidate algorithms with a dependence on dataset features like event widths, imprecision, and hit trial ratio, as well as (3) the possibility to expand the detection regime in more realistic physiological datasets using concordance.

Which algorithms to use?

We did not set out to rank algorithms, but our analysis does yield suggestions for possible use domains based on sensitivity to experimental perturbations (Fig. 8). In cases where runtime and compute resource use is a concern, we recommend using the temporal information method with Bootstrap along with the activity filter (*tiBoth*). Combinations of *tiBoth* with *r2bBoot* may be useful where there are rare and potentially multimodally tuned time cells (Pastalkova et al., 2008; Villette et al., 2015), either to combine their classification for stringent time cell identification, or to pool their classified cells. While it is tempting to use Otsu's threshold as a very fast alternative to bootstrapping, we found that none of the Otsu variants of these methods did a good job of classification. Ultimately, five of our algorithms *tiMean*, *tiBoot*, *tiBoth*, *r2bMean*, and *r2bBoot*: all based on either Mau et al. (2018) or Modi et al. (2014), have very good Precision, and classify with very few false positives (low Type I error). Many methods are susceptible to classification errors if the dataset has high noise.

Here we also implemented the parametric equation (*peq*) algorithm. It is not very good for time cell classification per se, as it is prone to false positives and is susceptible to noise and low hit trial ratios. However, it generates useful

additional estimates of the four key parameters of real data, namely, noise, hit trial ratio, event width and imprecision. This is useful for a first-pass characterization of the properties of the dataset.

Sequence detection in large-scale recordings and scaling of analysis runs

The discovery of replay over the past two decades, has benefitted from the technological advances made in increasing the cellular yield of network recordings and has been reviewed previously (Foster, 2017). Further advances such as with the large scale recordings of $\sim 10^3$ single units by electrical recording using Neuropixels (Jun et al., 2017), fast volumetric fluorescence scanning with up to $\sim 10^4$ cells using resonant electro-optic imaging (Poort et al., 2015; Pachitariu et al., 2017; Bowman and Kasevich, 2021), $\sim 10^3$ mesoscopes (Sofroniew et al., 2016), as well as advances in automated cell region of interest (ROI) detection, denoising, and neuropill subtraction (Pachitariu et al., 2017; Pnevmatikakis et al., 2016) only increase the scale and size of datasets, likely leading to longer analysis runtimes. In addition to our recommendations above for the temporal information/boot method for scalable time-cell analysis, our C++/Python implementations may also be useful in further optimizing these methods. Our implementations allow for relatively fast analysis of the same datasets with multiple algorithms.

References

Abbasi S, Maran S, Jaeger D (2020) A general method to generate artificial spike train populations matching recorded neurons. J Comput Neurosci 48:47–63.

Ahmed MS, Priestley JB, Castro A, Stefanini F, Solis Canales AS, Balough EM, Lavoie E, Mazzucato L, Fusi S, Losonczy A (2020) Hippocampal network reorganization underlies the formation of a temporal association memory. Neuron 107:283–291.e6.

Aronov D, Tank DW (2014) Engagement of neural circuits underlying 2D spatial navigation in a rodent virtual reality system. Neuron 84:442–456.

Bhalla US (2017) Synaptic input sequence discrimination on behavioral timescales mediated by reaction-diffusion chemistry in dendrites. Elife 6:e25827.

Bhatia A, Moza S, Bhalla US (2021) Patterned optogenetic stimulation using a DMD projector. Methods Mol Biol 2191:173–188.

Bhattarai B, Lee JW, Jung MW (2020) Distinct effects of reward and navigation history on hippocampal forward and reverse replays. Proc Natl Acad Sci U S A 117:689–697.

Bowman AJ, Kasevich MA (2021) Resonant electro-optic imaging for microscopy at nanosecond resolution. ACS Nano 15:16043–16054.

Branco T, Clark BA, Häusser M (2010) Dendritic discrimination of temporal input sequences in cortical neurons. Science 329:1671–1675.

Collett D (2002) Modeling binary data. London: Chapman and Hall.

Csicsvari J, O'Neill J, Allen K, Senior T (2007) Place-selective firing contributes to the reverse-order reactivation of CA1 pyramidal cells during sharp waves in open-field exploration. Eur J Neurosci 26:704–716.

Davidson TJ, Kloosterman F, Wilson MA (2009) Hippocampal replay of extended experience. Neuron 63:497–507.

Dhawale AK, Hagiwara A, Bhalla US, Murthy VN, Albeanu DF (2010) Non-redundant odor coding by sister mitral cells revealed by light addressable glomeruli in the mouse. Nat Neurosci 13:1404–1412.

Diba K, Buzsáki G (2007) Forward and reverse hippocampal place-cell sequences during ripples. Nat Neurosci 10:1241–1242.

Dombeck DA, Harvey CD, Tian L, Looger LL, Tank DW (2010) Functional imaging of hippocampal place cells at cellular resolution during virtual navigation. Nat Neurosci 13:1433–1440.

Dragoi G, Tonegawa S (2013) Distinct preplay of multiple novel spatial experiences in the rat. Proc Natl Acad Sci U S A 110:9100–9105.

Eichenbaum H (2017) On the integration of space, time, and memory. Neuron 95:1007–1018.

Foster DJ (2017) Replay comes of age. Annu Rev Neurosci 40:581–602.

Foster DJ, Wilson MA (2006) Reverse replay of behavioural sequences in hippocampal place cells during the awake state. Nature 440:680–683.

Gupta AS, van der Meer MAA, Touretzky DS, Redish AD (2010) Hippocampal replay is not a simple function of experience. Neuron 65:695–705.

Iannucci S, Kholidy HA, Ghimire AD, Jia R, Abdelwahed S, Banicescu I (2017) A comparison of graph-based synthetic data generators for benchmarking next-generation intrusion detection systems. 2017 IEEE International Conference on Cluster Computing (CLUSTER), Honolulu, HI, pp. 278–289, https://doi.org/10.1109/CLUSTER.2017.54.

Ikegaya Y, Aaron G, Cossart R, Aronov D, Lampl I, Ferster D, Yuste R (2004) Synfire chains and cortical songs: temporal modules of cortical activity. Science 304:559–564.

Jadhav SP, Kemere C, German PW, Frank LM (2012) Awake hippocampal sharp-wave ripples support spatial memory. Science 336:1454–1458.

Jun JJ, et al. (2017) Fully integrated silicon probes for high-density recording of neural activity. Nature 551:232–236.

Kaifosh P, Lovett-Barron M, Turi GF, Reardon TR, Losonczy A (2013) Septo-hippocampal GABAergic signaling across multiple modalities in awake mice. Nat Neurosci 16:1182–1184.

Karlsson MP, Frank LM (2009) Awake replay of remote experiences in the hippocampus. Nat Neurosci 12:913–918.

Koch S, Worchel M, Silva C, Alexa M (2021) Hardware Design and Accurate Simulation of Structured-Light Scanning for Benchmarking of 3D Reconstruction Algorithms (Vol. 2021). Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1 (NeurIPS Datasets and Benchmarks 2021). Retrieved from https://openreview.net/forum?id=bNL5VlTfe3p.

Kraus B, Robinson R, White J, Eichenbaum H, Hasselmo M (2013) Hippocampal "time cells": time versus path integration. Neuron 78:1090–1101.

Lee AK, Wilson MA (2004) A combinatorial method for analyzing sequential firing patterns involving an arbitrary number of neurons based on relative time order. J Neurophysiol 92:2555–2573.

Liu Y, Khandagale S, White C, Neiswanger W (2021) Synthetic benchmarks for scientific research in explainable machine learning. https://github.com/abacusai/xai-bench.

MacDonald CJ, Lepage KQ, Eden UT, Eichenbaum H (2011) Hippocampal "time cells" bridge the gap in memory for discontiguous events. Neuron 71:737–749.

MacDonald CJ, Carrow S, Place R, Eichenbaum H (2013) Distinct hippocampal time cell sequences represent odor memories in immobilized rats. J Neurosci 33:14607–14616.

Malvache A, Reichinnek S, Villette V, Haimerl C (2016) Awake hippocampal reactivations project onto orthogonal neuronal assemblies. 353:1280–1283.

Mattar MG, Daw ND (2018) Prioritized memory access explains planning and hippocampal replay. Nat Neurosci 21:1609–1617.

Mau W, Sullivan DW, Kinsky NR, Hasselmo ME, Howard MW, Eichenbaum H (2018) The same hippocampal CA1 population simultaneously codes temporal information over multiple timescales. Curr Biol 28:1499–1508.e4.

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D,

Legg S, Hassabis D (2015) Human-level control through deep reinforcement learning. Nature 518:529–533.
Modi MN, Dhawale AK, Bhalla US (2014) CA1 cell activity sequences emerge after reorganization of network correlation structure during associative learning. Elife 3:e01982.
Mokeichev A, Okun M, Barak O, Katz Y, Ben-Shahar O, Lampl I (2007) Stochastic emergence of repeating cortical motifs in spontaneous membrane potential fluctuations in vivo. Neuron 53:413–425.
O'Keefe J, Dostrovsky J (1971) The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. Brain Res 34:171–175.
O'Keefe J, Nadel L (1978) The hippocampus as a cognitive map. Oxford: Clarendon Press.
Otsu N (1979) A threshold selection method from gray level histograms. IEEE Trans Syst Man Cyber 9:62–66.
Pachitariu M, Stringer C, Dipoppa M, Schröder S, Rossi LF (2017) Suite2p: beyond 10,000 neurons with standard two-photon microscopy. BioRxiv 061507. https://doi.org/https://doi.org/10.1101/061507.
Pastalkova E, Itskov V, Amarasingham A, Buzsáki G (2008) Internally generated cell assembly sequences in the rat hippocampus. Science 321:1322–1327.
Pnevmatikakis EA, Soudry D, Gao Y, Machado TA, Merel J, Pfau D, Reardon T, Mu Y, Lacefield C, Yang W, Ahrens M, Bruno R, Jessell TM, Peterka DS, Yuste R, Paninski L (2016) Simultaneous denoising, deconvolution, and demixing of calcium imaging data. Neuron 89:285–299.
Poort J, Khan AG, Pachitariu M, Nemri A, Orsolic I, Krupic J, Bauza M, Sahani M, Keller GB, Mrsic-Flogel TD, Hofer SB (2015) Learning enhances sensory and multiple non-sensory representations in primary visual cortex. Neuron 86:1478–1490.
Ranck JB, J (1973) Studies on single neurons in dorsal hippocampal formation and septum in unrestrained rats. I. Behavioral correlates and firing repertoires. Exp Neurol 41:461–531.
Schrader S, Grün S, Diesmann M, Gerstein GL (2008) Detecting synfire chain activity using massively parallel spike train recording. J Neurophysiol 100:2165–2176.
Siegel JJ, Taylor W, Gray R, Kalmbach B, Zemelman BV, Desai NS, Johnston D, Chitwood RA (2015) Trace eyeblink conditioning in mice is dependent upon the dorsal medial prefrontal cortex, cerebellum, and amygdala: behavioral characterization and functional circuitry. eNeuro 2:ENEURO.0051-14.2015.
Sofroniew NJ, Flickinger D, King J, Svoboda K (2016) A large field of view two-photon mesoscope with subcellular resolution for in vivo imaging. Elife 5:e14472.
Souza BC, Pavão R, Belchior H, Tort ABL (2018) On information metrics for spatial coding. Neuroscience 375:62–73.
Steinbuss G, Bohm K (2020) Generating artificial outliers in the absence of genuine ones-a survey. arXiv. https://doi.org/10.48550/arXiv.2006.03646.
Tseng W, Guan R, Disterhoft JF, Weiss C (2004) Trace eyeblink conditioning is hippocampally dependent in mice. Hippocampus 14:58–65.
Villette V, Malvache A, Tressard T, Dupuy N, Cossart R (2015) Internally recurring hippocampal sequences as a population template of spatiotemporal information. Neuron 88:357–366.
Wilson MA, McNaughton BL (1993) Dynamics of the hippocampal ensemble code for space. Science 261:1055–1058.
Wu CT, Haggerty D, Kemere C, Ji D (2017) Hippocampal awake replay in fear memory retrieval. Nat Neurosci 20:571–580.