Mesmerize: A highly versatile platform for calcium imaging analysis and creation of self-contained FAIR datasets.

Kushal Kolar[1*], Daniel Dondorp[1], Marios Chatzigeorgiou[1*]

Sars International Centre for Marine Molecular Biology, University of Bergen, Thormøhlensgt. 55, 5006 Bergen, Norway

*email: kushalkolar@gmail.com, Marios.Chatzigeorgiou@uib.no

Abstract

We present an efficient and expandable calcium imaging analysis platform that encapsulates the entire analysis process from raw data to interactive e-figures. It provides a graphical interface to the latest analysis methods for pre-processing, and signal extraction. We demonstrate how Mesmerize can be applied to a broad range of scientific questions by using datasets ranging from the mouse visual cortex, neurons, epidermis and TLCs of the protochordate *C. intestinalis*, and *C. elegans*.

Large-scale calcium imaging of neuronal activity in populated brain regions, or entire animals, has become an indispensable technique in neuroscience research. The analysis of calcium imaging datasets presents significant challenges with respect to signal extraction and data organization. Numerous commendable packages, such as the Caiman library[1], Suite2p[2] and SIMA[3] have been created to address pre-processing and signal extraction of imaging data. However, users often incorporate these tools in custom written scripts without a system to link the analysis procedures, raw and analyzed data. This non-ideal organization can greatly impede the reproducibility of the work even when the raw data are available[4,5]. Projects such as the Jupyter notebook[6] can help address some of these issues; however these platforms are not easily accessible to non-programmers. Moreover, these solutions do not have a system in place for FAIR, open & functionally linked data[4,7,8]. In order to address these organizational

challenges we created Mesmerize, a comprehensive platform that encapsulates data organization, analysis pipelines and interactive traceable e-figures to facilitate reproducibility and assist scientists in creating FAIR datasets within a system that is accessible to a broad audience of researchers. We aimed to create a platform, not a specific pipeline, which allows developers to add modules for pre-processing, analysis, and plotting. A high degree of flexibility will enable it to be adopted across a variety of labs beyond the realm of neuroscience and into the hands of cell biologists, developmental biologists and others.

Calcium imaging analysis usually requires the following components 1) pre-processing & signal extraction 2) data annotation and organization 3) signal analysis and 4) plotting. Mesmerize provides end-users with rich graphical interfaces for each of these components. Users who are familiar with Python can utilize the API, which facilitates smooth integration of customized modules. This allows the platform to be more broadly applicable and highly customizable. We built the graphical interfaces using the Qt framework (via PyQt bindings) due to its maturity and extensive developer community. All data structures are well-documented and built using pandas DataFrames[9] and numpy arrays[10], both highly prevalent and mature libraries. This allows developers to add new modules, and allows users to integrate data from Mesmerize in external analysis workflows.

Pre-processing and signal extraction often necessitate a system that allows users to explore their imaging data, perform pre-processing and signal extraction. This requirement is performed by the Mesmerize Viewer - a set of graphical interfaces that provide front-ends to pre-processing and signal extraction routines, and enables *in-place* annotation of experimental information, such as cell identities, mapping of stimulus or behavioral periods, and any other relevant information (Fig 1b). For example, our Ciona dataset includes annotations for seven different GCaMP6s promoters, eight anatomical regions, and twenty-one cell types (Supplementary table 1 & 2). The ability to conveniently tag a wealth of information through both command-line and graphical interfaces will allow researchers to efficiently curate and analyze complex datasets that are common in more genetically tractable

organisms. For instance, researchers can perform experiments that utilize tens of GCaMP promoters, various chemogenetic lines, UAS-GAL4 systems, multiple drugs, etc. altogether in an efficient and organized manner.

Front ends for CaImAn NoRMCorr[11] and CNMF(E)[12,13] are provided for motion correction and signal extraction respectively. These front ends have the potential to broaden the reach of cutting-edge packages, such as the CaImAn library[1], into the hands of more biologists who can now perform more accurate and in-depth data analysis. Viewer modules can also be used in conjunction with the Batch Manager to aid in data organization. Simple data structures, outlined in the API, allow the imaging data to potentially originate from any model organism, or organoid/organotypic culture under a 2D imaging setup (Fig 1a). A common limitation of conventional analysis software is the narrow range of options they provide. In Mesmerize image processing and signal extraction procedures are defined in Viewer modules, which can be expanded and customized. We provide instructions for efficiently creating custom Qt GUI modules that can interact with the Viewer work environment (http://www.mesmerizelab.org/developer_guide/viewer_modules.html). This flexibility allows scientists to conveniently integrate specialized pre-processing or signal extraction techniques.

Most analysis software do not aide users in the organization of their imaging samples. Mesmerize packages all data associated with an imaging sample, i.e. extracted signals, annotations etc, into a Project Sample (Fig 1c). These Samples constitute the Project Dataset, which can be explored and filtered to create experimental groups using a Project Browser (Fig 1d). Project Samples can be modified throughout the course of a project. Thereby, Mesmerize not only allows for efficient data annotation, but affords users the ability to append new annotations and change or supplement existing ones. This is crucial since biological questions and experiments are often in constant flux as new data are processed and analyzed.

A project dataset, or sub-dataset, can be loaded into a flowchart where users can build analysis pipelines by connecting analysis nodes (Fig 1e-g). We provide nodes to perform many common signal

processing routines, data handling/organization, dimensionality reduction, and clustering analysis. This default collection of nodes allows users to perform many common analysis procedures such as comparison of stimulus/behavioral periods (Fig 1e), peak detection (Fig 1f), and clustering analysis (Fig 1g). The flowchart utilizes a data object that maintains an analysis log describing the nodes and parameters the data were processed with. We provide documentation for efficiently writing new analysis nodes and using the data structures (http://www.mesmerizelab.org/developer_guide/nodes.html). The flowchart builds upon a pyqtgraph[14] widget, and the nodes that we provide use implementations from scipy and sklearn libraries for signal processing, dimensionality reduction and clustering[15,16]. We use common libraries so that developers can easily expand upon the available analysis nodes.

The ultimate result of almost any analysis procedure and scientific study is the creation of plots that convey the study's results from which interpretations and conclusions are made. The vast majority of plots in modern biology research are static, which makes it hard or impossible to connect plot datapoints with the raw data and analysis procedures[4,5]. Some recent ideas have been around to address these issues, and tools such as Jupyter[6] notebooks delivered via MyBinder[17] allow the data and analysis procedures to be shared[5]. However these methods are not readily accessible to non-programmers and more importantly they do not enforce the creation of FAIR and functionally linked datasets. Mesmerize allows users to easily create interactive plots through a GUI (Fig 1e-j) and share them in their interactive state. These interactive plots are attached to a *Datapoint Tracer* (Fig 1h& 1i, right panels) which highlights the spatial localization of the selected datapoint along with annotations, and the analysis log. A variety of plots are provided, as well as developer instructions for creating new plots (http://www.mesmerizelab.org/developer_guide/plots.html) that integrate with the Datapoint Tracer. The interactive plots are built using pyqtgraph, matplotlib and seaborn, which were chosen due to their maturity and extensive developer communities[14,18].

Mesmerize aims to address several common difficulties with reproducibility, data reusability, and organization. Critically, Mesmerize enables analysis procedures to be transparent at the level of individual data points. This is achieved by tagging the data with Universally Unique Identifiers (UUID) at various levels of analysis, which is one of the key principles of FAIR data. The ability to tag and keep track of an unlimited amount of information with each project sample *in-place* within the Viewer itself encourages users to comprehensively describe their data in detail. Secondly, A Mesmerize project is entirely self-contained within a single directory tree, thus enabling entire datasets, analysis workflows, and e-figures to be easily shared and readily accessible to the broader scientific community. A receiver can open a project and immediately explore e-figures, analysis procedures, and view the raw data associated with the datapoints on a figure. Finally, we make Mesmerize available as a Snap package (https://snapcraft.io/mesmerize), thus allowing a straightforward and accessible installation. This ease of opening a Mesmerize project and exploring datasets in conjunction with interactive plots will help scientists in making their data easily accessible and reusable.

In order to demonstrate the repertoire of tools provided by Mesmerize, we present calcium imaging of spontaneous activity in the whole brain of the protochordate *Ciona intestinalis* larvae, an emerging model in neuroscience. The *C. intestinalis* connectome[19] and single-cell transcriptomes[20,21] have recently been established, however there has not been a comprehensive functional study to investigate neuronal activity from its diverse neuronal populations. Its small nervous system, flat head, and the ability to label genetically defined populations of cells using promoters driving GCaMP6s expression allow us to approximate the identity of neuronal cells in reference to the connectome[19,22]. We also imaged the epidermis and Trunk Lateral Cells (TLCs), a population of migratory mesenchymal cells, to develop analysis methods that can be used to study a diverse range of dynamics which are of interest to cell and developmental biologists.

We were interested in describing calcium activity in cells and domains where typical neuronal spike trains have not been observed (Fig 2a); therefore, we implemented techniques which have not

previously been used to analyze calcium dynamics. These methods could also be used to understand calcium dynamics in other model systems. We introduce the application of Earth Mover's Distances[23] (EMD) between frequency domain representations of calcium traces data for hierarchical clustering as a simple and useful method for characterization of calcium dynamics across a diverse range of cell types. The EMD is commonly used for pattern recognition and image retrieval systems through histogram comparison[24]. The EMD allows for far superior categorization of calcium dynamics as opposed to Euclidean Distances (ED) as shown by the example distance matrices (Fig 2b & Supplemental Fig 1 & 2) and agglomerative coefficients (Supplementary Table 3). When used for hierarchical clustering, the EMD of frequency domain representations leads to better separation of disparate dynamics and aggregation of similar dynamics (Fig 2c & Supplemental Fig 3).

Using the EMD, we performed hierarchical clustering on traces obtained by imaging various neuronal and non-neuronal populations of cells in the *C. intestinalis* head. The resulting dendrogram was cut to form 10 clusters that depict stereotyped modes of calcium dynamics within these cell populations (Fig 2d). Cells within clusters 1-4 show relatively low levels of activity, cells of clusters 5-7 show intermediate activity, and cells within clusters 8-10 show high levels of activity and/or sharp peaks (Fig d-e). These results were used to train a Linear Discriminant Analysis (LDA) model, and the LDA means of each class shows the distinct frequency domain representations of each cluster (Fig 2f-g). Hierarchical clustering was also performed using the neuronal sub-dataset; the resulting dendrogram was cut to obtain 8 clusters (Supplementary Fig 4). The proportions of cells that exhibit each activity mode in this subset were then used to perform standard hierarchical clustering (Fig 2h and Supplementary Fig 5). We were able to distinguish activity between genetically defined populations of peripheral & sensory neurons, from populations that are located in the brain vesicle and form part of the Central Nervous System. Most interestingly, four cell types involved in peripheral sensory networks (palps, RTEN, ATENa and ATENp) exhibit similar modes of activity and cluster together. Palps provide feedforward excitation to the RTENs, which could explain the similarity in their modes

of activity. Cells that are mostly interneurons within the brain vesicle (PBV PNIN, PNIN, PR_TRIN, Antennal relay, Eminens and PR_RN) exhibit high levels of activity; this could reflect the possibility that they receive more complex inputs due to their intermediate position in a network. Put together, this reveals that spontaneous activity is sufficient to derive cell-specific functional fingerprints in *C. intestinalis* larvae. This simple but powerful technique can be used in other model systems to define discrete functional domains for specific populations or sub-types of neurons.

This approach can be extended with a user trained classifier that can then be utilized to predict cluster membership. For example, first a battery of specific promoters could be used to image spatiotemporally and genetically defined neuronal populations in a brain. Spontaneous-activity can be recorded for various cell populations/types in order to generate a library of FFT signatures, and FFT-EMD based hierarchical clustering can be used to train a classifier. Within Mesmerize, this library and classifier can be interrogated in subsequent experiments, where previously uncharacterized regulatory elements or pan-neuronal markers are used to systematically monitor. Spontaneous activity in reference to the benign stimulus can then be used to classify these cells into possible cell-types. Here we use our LDA classifier generated by imaging a range of promoters labeling neurons, putative support cells, epidermal cells and mesenchymal cells, in order to predict cluster membership using data derived from a ubiquitous pan-cellular GCaMP (EEF1A)[25]. Interestingly, as expected we find that EEF1A(+) cells that appear spatially localized on the animal's surface are classified as members of clusters that are highly enriched with CesA-positive cells (Supplementary Table 4 and Supplementary Fig 9), a well characterized marker of epidermal cells[26]. This shows that functional fingerprints derived through EMD based hierarchical clustering can be combined with supervised approaches to classify cells when specific promoters are impractical.

Mesmerize also provides an implementation for k-Shape clustering[27,28] to extract a finite set of discrete archetypical peaks from calcium traces (Fig 2i). These archetypical peaks allow traces to be reduced to sequences of discrete letters that can be modeled with techniques such as Markov Chains

(Fig 2j-m). These models are able to produce dynamics that visually resemble discretized real data (Fig 2k-l). Apart from Markov Chains, k-Shape clustering provides a contemporary approach to questions in other systems, such as examining stimulus-response profiles, behavioral periods, etc.

To show that Mesmerize can be used for studying other model systems, we analyzed spontaneous activity of *C. elegans* motorneurons (Fig 2n) and a portion of the CRCNS pvc-7 dataset which consists of in-vivo imaging of layer 4 cells in the mouse visual cortex[29] (Fig 2o). Orientation preferences (Fig 2p), spatial frequency tuning (Fig 2q) and temporal frequency tuning (Fig 2r) of the cells in the pvc-7 dataset were determined and mapped spatially onto the imaging field. This shows that Mesmerize can be used for studying biological questions that study neuronal dynamics in conjunction with stimulus or behavioral periods.

We demonstrate that Mesmerize is a platform that can be used to analyze calcium imaging data from a diverse range of cell types (including neurons, epidermal and mesenchymal cells). We show how Mesmerize offers contemporary analytical methods that can be used to combine functional fingerprinting (calcium signal) with genetic fingerprinting (e.g. promoters). The diverse applications, analysis techniques, and flexibility provided by Mesmerize shows that the platform can be adopted by a wide range of labs, by both developers and end-users. Importantly, the ability to produce FAIR data by the encapsulation of raw data, analysis procedures and interactive plots *en masse* will contribute towards making traceable interactive figures to become a commonplace practice.

Methods

Imaging

Larvae were embedded in low melting point agarose (Fisher BioReagents, BP1360-100) between two coverslips to minimize scattering and bathed in artificial sea water. Illumination was provided by a mercury lamp with a BP470/20, FT493, BP505-530 filterset. A Hamamatsu Orca

FlashV4 CMOS camera acquired images at 10Hz with exposure times of 100ms using a custom application[30].

Signal Extraction

Images were motion corrected using NoRMCorre and signal extraction was performed using CNMFE with parameters optimized per video. Extracted signals that were merely movement or noise were excluded. All parameters for motion correction and CNMFE can be seen in the available dataset. Cells were identified with the assistance of the connectome[19,22] to the best of our capability with 1-photon data. Only regions that covered cell bodies were tagged, axons were not tagged with cell identity labels.

Hierarchical Clustering

Analysis was performed using the Mesmerize flowchart. All traces extracted from CNMFE were normalized between 0 – 1. The Discrete Fourier Transform (DFT) of the normalized data was calculated using the scipy.fftpack.rfft function from the SciPy Python library[15]. The logarithm of the absolute value of the DFT data arrays were taken, and the first 1000 frequency domains (corresponding to frequencies between 0 – 1.66 Hz) were used for clustering. This cutoff was determined by looking at the sum of squared differences (SOSD) between the raw curves and interpolated Inverse Fourier Transforms of the DFTs with a step-wise increase in the frequency cutoff (Supplementary Fig 8). The SOSD changes negligibly beyond 1.5 Hz, and inclusion of higher frequencies would likely introduce noise. Earth Mover's Distance (EMD) was used as the distance metric through the OpenCV[31] EMD function and complete linkage was used for constructing the tree. This procedure can be opened in the dataset and is illustrated by supplementary graph 1 & 2. Agglomerative coefficients for the data in Fig 2b & Supplementary Fig 1 were calculated using the agnes function in R.

k-Shape Clustering

This method uses a normalized cross-correlation function to derive a shape-based distance metric[27]. The tslearn implementation is used in Mesmerize[28]. Peak curves were used as the input data for k-Shape clustering and the parameters can see seen in supplementary graph 4. Cluster 7 from the dataset was excluded.

Markov Chains

Cluster membership of peaks, as determined through k-Shape clustering, was used to express calcium traces as discretized sequences. These sequences were used to create Markov Chain models using the pomegranate Python library. One model was created for each cell type shown in supplementary table 3. The length of the generated sequences shown in Fig 2k-l was set as the average sequence length for the given cell type. The generated chains in Fig 2k-l were illustrated using k-Shape cluster means.

Determining stimulus tuning of cell within the PVC-7 dataset

All stimulus periods were extracted and averaged per stimulus type (orientation or spatial/temporal frequency) for each cell. A line was fit to each of these means using linear regression. The tuning of a given cell was defined as the stimulus with the highest slope for the fitted regression line. This procedure can be explored in the dataset.

Promoters

To drive the expression of GCaMP6s population in different cell types in *Ciona intestinalis* larvae we used the following promoters:

Gene Unique ID	Gene Model ID	Name	Abbr.	Length
Cirobu.g00010959	KH.L128.92	Proprotein/Prohormone convertase 2	pc2	2.86kb
Cirobu.g00008038	KH.C7.211	CesA	cesa	2.2kb
Cirobu.g00014653	KH.S544.3	DMRT1	dmrt1	1.29kb

Cirobu.g00004616	KH.C2.42	Brn3b/POU4	brn3b	3.78kb
Cirobu.g00006491	KH.C4.403	HNK1[32]	hnk1	3.0kb
Cirobu.g00010171	KH.C9.608	PDE9	pde9	4.43kb
Cirobu.g00012642	KH.L42.6	CNG Channel 4	cng_ch4	1.48kb
Cirobu.g00003963	KH.C14.52	EEF1A1	eef1a	1.96kb

Sequences for several of these promoters were obtained from DBTGR[33].

Primer name	Primer sequence
PC2 GW-FW	g g g g a c a a c t t t g t a t a g a a a a g t t gCAGCAGTCAAAGGGTTTCTTGAAACAC
PC2 GW-RV	g g g g a c t g c t t t t t t g t a c a a a c t t gGCTGCTTTAAGAATTCTTCGTTTTTTCAC
CesA GW-FW	g g g g a c a a c t t t g t a t a g a a a a g t t gCCCGGTGCTTTGAAAATTGACAAG
CesA GW-RV	g g g g a c t g c t t t t t t g t a c a a a c t t gGAACTCGTATATCTTGATGGTTTGG
DMRT1 GW-FW	g g g g a c a a c t t t g t a t a g a a a a g t t gTCAGAACGAGGCGCTACATGATC
DMRT1 GW-RV	g g g g a c t g c t t t t t t g t a c a a a c t t gCACTGTTCTAAGCAAGGTATCAAGG
Brn3b/Pou4 GW-FW	g g g g a c a a c t t t g t a t a g a a a a g t t gCGACTGTAACAAGTTCTAAACAGAGC
Brn3b/Pou4 GW-RV	g g g g a c t g c t t t t t t g t a c a a a c t t gATATCGTATCAAAAAATATACAATAAGTCTG
HNK1 GW-FW	g g g g a c a a c t t t g t a t a g a a a a g t t gCAGCACGGGTTGAGTCAATGAAAC
HNK1 GW-RV	g g g g a c t g c t t t t t t g t a c a a a c t t gACGCACCAGGAAGTTAAATAAAACC
PDE9 GW-FW	g g g g a c a a c t t t g t a t a g a a a a g t t gATTCATGGCTGATATACCCGGTTG
PDE9 GW-RV	g g g g a c t g c t t t t t t g t a c a a a c t t gCTATGCTGTTGTAGAATCTGTATATAG
CNG4 GW-FW	g g g g a c a a c t t t g t a t a g a a a a g t t gCTCCGTTTCGTGGAAAACTCATTTTTC
CNG4 GW-RV	g g g g a c t g c t t t t t t g t a c a a a c t t gACTGGACTCTAGACACAGACAGC
EEF1A1 GW-FW	g g g g a c a a c t t t g t a t a g a a a a g t t gGTGACGGGAAAACGATAGTCG
EEF1A1 GW-RV	g g g g a c t g c t t t t t t g t a c a a a c t t gTTTGGAAGGTTGGGGTTAACC

The amplified PCR products were gel purified and inserted into P4-P1R vector using BP Clonase II. Positive clones identified by restriction digest were sequenced. Subsequently we performed a 4-way Gateway Recombination using one of the promoters in the 1st position, GCaMP6s in the 2nd positon and unc-54 3'UTR in the 3rd position. These were recombined into a pDEST II. Expression constructs were electroporated at a range of concentrations (80-120µg).

C. elegans strain generation and imaging

To generate construct drg1 [prab-3::GCaMP6m::NLS::unc-54 3'UTR] we performed a 4-way Gateway recombination reaction using LR Clonase II (Invitrogen). We recombined pDEST II with the

following entry clones: 1st position a 1.2kb promoter of rab-3 (a kind gift from Dr. Inja Radman, Chin lab, MRC LMB); 2nd position GCaMP6m fused to SV40NLS at the N-terminus and EGL-13 NLS sequence at the C-terminus and 3rd position unc-54 3'UTR. The resulting construct was injected into N2 animals at 100µg/µl to generate strain SCB1. C. elegans young adults were immobilized on 1% agarose pads (in M9) using DERMABOND (2-Octyl Cyanoacrylate) glue.

Dataset availability

The datasets are available as a Mesmerize project and can be downloaded from figshare:

C. intestinalis: https://doi.org/10.6084/m9.figshare.10289162

C. elegans: https://doi.org/10.6084/m9.figshare.10287113

PVC-7 as a Mesmerize dataset: https://doi.org/10.6084/m9.figshare.10293041

Notebooks that produce some of the figures and the Markov Chains are available on GitHub and can be used on binder.

https://github.com/kushalkolar/mesmerize_manuscript_notebooks

https://mybinder.org/v2/gh/kushalkolar/mesmerize_manuscript_notebooks/master

Author Contributions

K.K. wrote Mesmerize and analyzed all experiments. D.D. aided in the development of Mesmerize and provided critical input. Imaging experiments were performed by K.K. and M.C. GCaMP6s constructs were cloned by M.C. The manuscript was written by K.K. and M.C.

Documentation and source code

Mesmerize documentation: http://mesmerizelab.org/

GitHub repository: https://github.com/kushalkolar/MESmerize

Acknowledgements

We would like to thank Pietro Vertechi and Julius Parulek for technical advice and members of the Chatzigeorgiou lab for user feedback during Mesmerize's early development. We thank Mie Wong and Dario Sarra for comments on the manuscript. Work in MC's laboratory was funded by Sars Centre core budget.

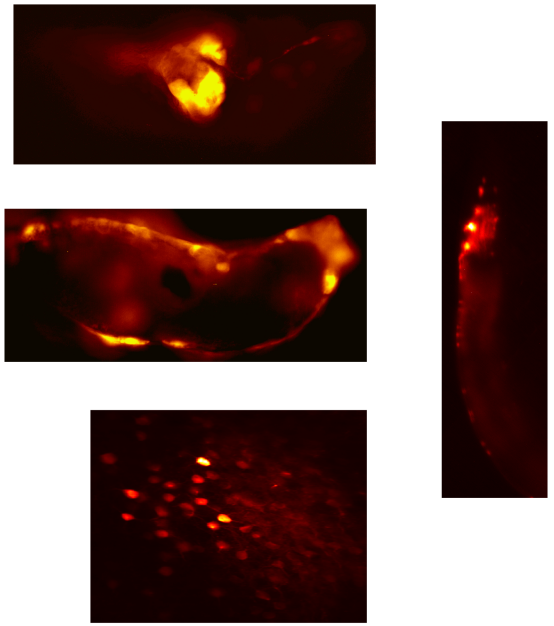
1. Giovannucci, A. *et al.* CaImAn an open source tool for scalable calcium imaging data analysis. *Elife* (2019) doi:10.7554/eLife.38173.
2. Pachitariu, M. *et al.* Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *bioRxiv* (2016) doi:10.1101/061507.
3. Kaifosh, P., Zaremba, J. D., Danielson, N. B. & Losonczy, A. SIMA: Python software for analysis of dynamic fluorescence imaging data. *Front. Neuroinform.* (2014)

doi:10.3389/fninf.2014.00080.

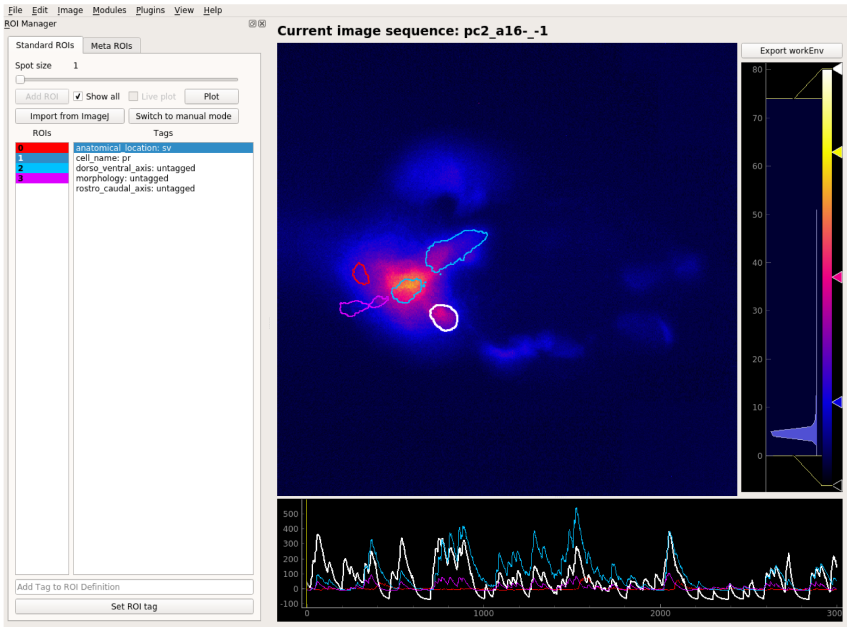
4. Jennings-Antipov, L. D. & Gardner, T. S. Digital publishing isn't enough: the case for 'blueprints' in scientific communication. *Emerg. Top. Life Sci.* (2018) doi:10.1042/etls20180165.
5. Perkel, J. M. Data visualization tools drive interactivity and reproducibility in online publishing. *Nature* (2018) doi:10.1038/d41586-018-01322-9.
6. Kluyver, T. *et al. Jupyter Notebooks—a publishing format for reproducible computational workflows. Positioning and Power in Academic Publishing: Players, Agents and Agendas* (2016). doi:10.3233/978-1-61499-649-1-87.
7. Stall, S. *et al.* Make scientific data FAIR. *Nature* (2019) doi:10.1038/d41586-019-01720-7.
8. Wilkinson, M. D. *et al.* Comment: The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* (2016) doi:10.1038/sdata.2016.18.
9. McKinney, W. Data Structures for Statistical Computing in Python. *Proc. 9th Python Sci. Conf.* (2010).
10. Van Der Walt, S., Colbert, S. C. & Varoquaux, G. The NumPy array: A structure for efficient numerical computation. *Comput. Sci. Eng.* (2011) doi:10.1109/MCSE.2011.37.
11. Pnevmatikakis, E. A. & Giovannucci, A. NoRMCorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. *J. Neurosci. Methods* (2017) doi:10.1016/j.jneumeth.2017.07.031.
12. Pnevmatikakis, E. A. *et al.* Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data. *Neuron* **89**, 285 (2016).
13. Zhou, P. *et al.* Efficient and accurate extraction of in vivo calcium signals from microendoscopic video data. *Elife* 1–37 (2016) doi:10.7554/eLife.28728.
14. Campagnola, L. pyqtgraph. www.pyqtgraph.org.
15. Virtanen, P. *et al.* SciPy 1.0--Fundamental Algorithms for Scientific Computing in Python. 1–22 (2019).
16. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* (2011).
17. Jupyter, P. *et al.* Binder 2.0 - Reproducible, interactive, sharable environments for science at scale. in *Proceedings of the 17th Python in Science Conference* (2018). doi:10.25080/majora-4af1f417-011.
18. Hunter, J. D. Matplotlib: A 2D graphics environment. *Comput. Sci. Eng.* (2007) doi:10.1109/MCSE.2007.55.
19. Ryan, K., Lu, Z. & Meinertzhagen, I. A. The CNS connectome of a tadpole larva of Ciona intestinalis (L.) highlights sidedness in the brain of a chordate sibling. *Elife* **5**, 1–34 (2016).
20. Cao, C. *et al.* Comprehensive single-cell transcriptome lineages of a proto-vertebrate. *Nature* (2019) doi:10.1038/s41586-019-1385-y.
21. Sharma, S., Wang, W. & Stolfi, A. Single-cell transcriptome profiling of the Ciona larval brain. *Dev. Biol.* (2019) doi:10.1016/j.ydbio.2018.09.023.
22. Ryan, K. & Meinertzhagen, I. A. Neuronal identity: the neuron types of a simple chordate sibling, the tadpole larva of Ciona intestinalis. *Current Opinion in Neurobiology* (2019) doi:10.1016/j.conb.2018.10.015.
23. Monge, G. Mémoire sur la théorie des déblais et de remblais. in *Histoire de l'Académie Royale des Sciences de Paris, avec les Mémoires de Mathématique et de Physique pour la même année* (1781).
24. Rubner, Y., Tomasi, C. & Guibas, L. J. Earth mover's distance as a metric for image retrieval. *Int. J. Comput. Vis.* (2000) doi:10.1023/A:1026543900054.
25. Sasakura, Y., Suzuki, M. M., Hozumi, A., Inaba, K. & Satoh, N. Maternal factor-mediated epigenetic gene silencing in the ascidian Ciona intestinalis. *Mol. Genet. Genomics* (2010) doi:10.1007/s00438-009-0500-4.

26. Sasakura, Y. *et al.* Transposon-mediated insertional mutagenesis revealed the functions of animal cellulose synthase in the ascidian Ciona intestinalis. *Proc. Natl. Acad. Sci. U. S. A.* (2005) doi:10.1073/pnas.0503640102.
27. Paparrizos, J. & Gravano, L. k-Shape: Efficient and Accurate Clustering of Time Series. *ACM SIGMOD Rec.* (2016) doi:10.1145/2949741.2949758.
28. Tavenard, R., Faouzi, J. & Vandewiele, G. tslearn: A machine learning toolkit dedicated to time-series data. https://github.com/rtavenar/tslearn (2017).
29. Garner, A. In vivo calcium imaging of layer 4 cells in the mouse using sinusoidal grating stimuli. (2014) doi:10.6080/K0C8276G.
30. Kolar, K. & Chatzigeorgiou, M. Simple GUI for acquiring images from a Hamamatsu Orca Flash 4.0 CMOS camera. (2019) doi:10.5281/ZENODO.3370464.
31. Bradski, G. The OpenCV Library. *Dr Dobbs J. Softw. Tools* (2000) doi:10.1111/0023-8333.50.s1.10.
32. Jeffery, W. R. *et al.* Trunk lateral cells are neural crest-like cells in the ascidian Ciona intestinalis: Insights into the ancestry and evolution of the neural crest. *Dev. Biol.* (2008) doi:10.1016/j.ydbio.2008.08.022.
33. Sierro, N. DBTGR: a database of tunicate promoters and their regulatory elements. *Nucleic Acids Res.* (2006) doi:10.1093/nar/gkj064.

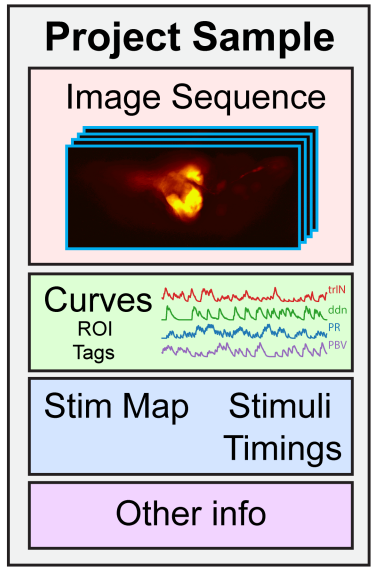
a



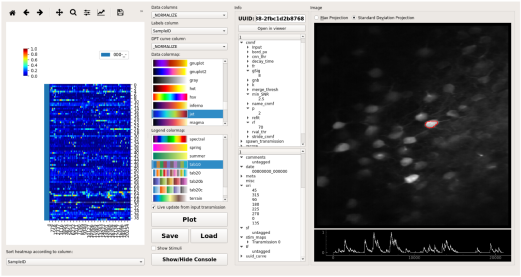
b


File Edit Image Modules Plugins View Help
ROI Manager
Standard ROIs
Meta ROIs
Spot size 1
Add ROI
Show all
Live plot
Plot
Import from ImageJ
Switch to manual mode
ROIs
Tags
anatomical_location: sv
cell_name: pr
dorso_ventral_axis: untagged
morphology: untagged
rostro_caudal_axis: untagged
Add Tag to ROI Definition
Set ROI tag
Current image sequence: pc2_a16-_-1
Export workEnv

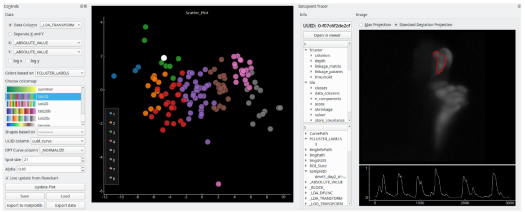
c


Project Sample
Image Sequence
Curves
ROI
Tags
trIN
ddn
PR
PBV
Stim Map
Stimuli
Timings
Other info

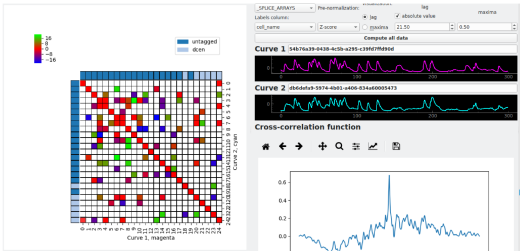
h


Plot
Save
Load
Show/Hide Console

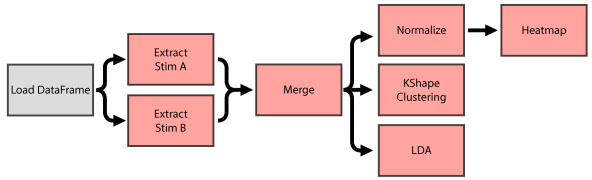
i



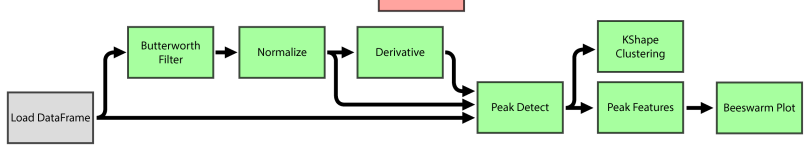
j


Curve 1
Curve 2
Cross-correlation function

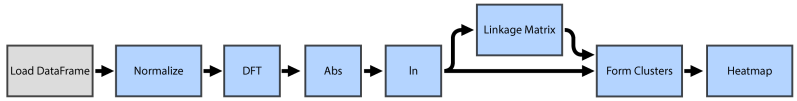
e


Load DataFrame
Extract Stim A
Extract Stim B
Merge
Normalize
KShape Clustering
LDA
Heatmap

f


Load DataFrame
Butterworth Filter
Normalize
Derivative
Peak Detect
KShape Clustering
Peak Features
Beeswarm Plot

g


Load DataFrame
Normalize
DFT
Abs
ln
Linkage Matrix
Form Clusters
Heatmap

d

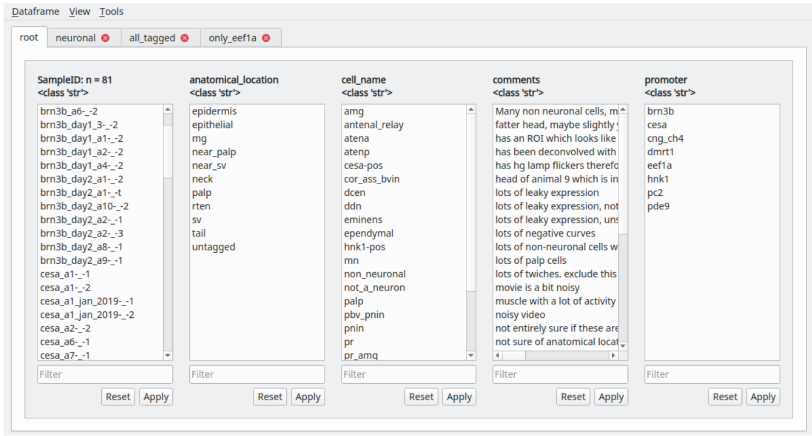

Dataframe View Tools
root
neuronal
all_tagged
only_eef1a
SampleID: n = 81
anatomical_location
cell_name
comments
promoter
Filter
Reset
Apply

Fig 1: (a) Raw imaging data that can originate from a variety of sources, examples shown from calcium imaging of *Ciona intestinalis* neurons & epithelial cells, mouse visual cortex neurons and *C. elegans* neurons. (b) Mesmerize Viewer shown with the ROI Manager. (c) Basic structure of a Project Sample which contains the image sequence, curves, along with information tagged to them, stimulus maps (if provided), and any other annotations. (d) Project samples can be organized using the Project Browser. (e-g) Illustrations of how the flowchart can be used to perform a diverse variety of analysis such as (e) extraction of stimulus/behavioral periods followed by various forms of analysis and visualization, (f) peak detection followed by k-Shape clustering and visualization of peak features, and (g) clustering of dynamics based on their frequency domain representations. (h) Interactive heatmap plot (left) with embedded Datapoint Tracer (right) showing spatial localization of a selected datapoint within the mouse visual cortex. (i) Data transformed by LDA visualized using an interactive scatter plot (center) with embedded Datapoint Tracer (right) showing spatial localization of a selected datapoint within the palps of *C. intestinalis*. (j) Interactive exploratory cross-correlation analysis (Datapoint Tracer not shown).

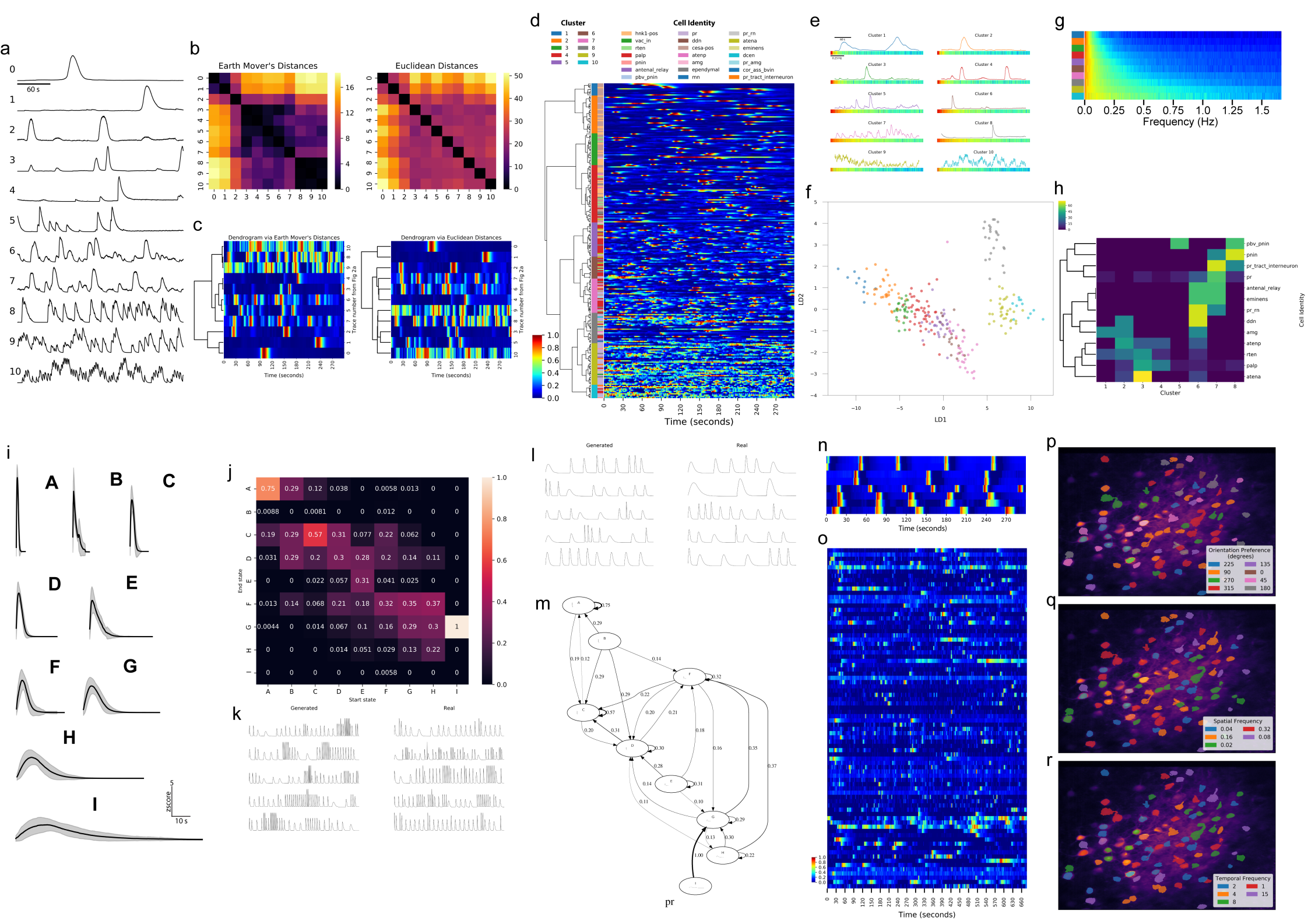

a
b
Earth Mover's Distances
Euclidean Distances
c
Dendrogram via Earth Mover's Distances
Dendrogram via Euclidean Distances
Trace number from Fig 2a
Time (seconds)
d
Cluster
Cell Identity
hnk1-pos
vac_in
rten
palp
pnin
antenal_relay
pbv_pnin
pr
ddn
cesa-pos
atenp
amg
ependymal
mn
pr_rn
atena
eminens
dcen
pr_amg
cor_ass_bvin
pr_tract_interneuron
Time (seconds)
e
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Cluster 8
Cluster 9
Cluster 10
f
LD1
LD2
g
Frequency (Hz)
h
Cluster
Cell Identity
i
A
B
C
D
E
F
G
H
I
zscore
10 s
j
End state
Start state
k
Generated
Real
l
Generated
Real
m
pr
n
Time (seconds)
o
Time (seconds)
p
Orientation Preference (degrees)
q
Spatial Frequency
r
Temporal Frequency

Fig 2: (a) Eleven calcium traces from various neuronal & non-neuronal populations in the *Ciona* head. (b) Distance matrices of the data in [a]. (c) Dendrogram from the data in [b] beside a heatmap illustrating the calcium traces. (d) EMD Hierarchical clustering of neuronal and non-neuronal cells in the head of *C. intestinalis.* Left colorbar legend indicates cluster, right colorbar legend indicates cell identity. Heatmap shows normalized traces. (e) Most center-like members from clusters shows in [d]. Time-domain calcium trace is shown on top of the corresponding frequency domain representation. (f) LDA projection trained on cluster labels derived from [d]. Spot colors represent cluster membership from [d]. (g) LDA means of the 10 clusters. (h) Hierarchical clustering of data in Supplementary Fig 5. Colormap indicates percentages. (i) Cluster means from k-Shape clustering of peaks from neuronal and non-neuronal cells in the head of *C. intestinalis*. Clusters are assigned alphabetical labels according to their half peak width. Error bands show intra-cluster standard deviation. (j) State Transition Matrix of Markov Chain created from discretized sequences from PR cell calcium traces. Colors represent transition probability. (k) Examples of generated (left) and real (right) sequences for PR cells. (l) Examples of generated & real sequences for palp cells. (m) State Transition graph of Markov Chain for PR cells, same data as [j]. Transition probability less than 0.1 were excluded to reduce clutter. (n) Traces from motorneurons in *C. elegans*. (o) Heatmap of normalized calcium traces from cells in the visual cortex of a mouse viewing moving bars at various orientations. (p-r) Spatial localization of cells in [o] with colored patches showing their orientation tuning [p], spatial frequency tuning [q] and temporal frequency tuning [r].