

METHODS

# Fast and accessible morphology-free functional fluorescence imaging analysis

Alejandro Estrada Berlanga<sup>1</sup>, Gabrielle Y. Kang<sup>2</sup>, Amanda Kwok<sup>2</sup>, Thomas Broggin<sup>3,4,5</sup>, Jennifer Lawlor<sup>6</sup>, Kishore V. Kuchibhotla<sup>6,7,8</sup>, David Kleinfeld<sup>4,10</sup>, Gal Mishne<sup>9,10\*</sup>, Adam S. Charles<sup>2,7,8,11\*</sup>

**1** Department of Bioengineering, University of Pennsylvania, Philadelphia, Pennsylvania, United States of America, **2** Department of Biomedical Engineering, Johns Hopkins University, Baltimore, Maryland, United States of America, **3** Frankfurt Cancer Institute (FCI), Goethe University Frankfurt, Frankfurt am Main, Germany, **4** Department of Physics, University of California San Diego, La Jolla, California, United States of America, **5** Department of Neurosurgery, University Hospital Frankfurt, Goethe University Frankfurt, Frankfurt am Main, Germany **6** Department of Psychological and Brain Sciences, Johns Hopkins University, Baltimore, Maryland, United States of America, **7** Department of Neuroscience, Johns Hopkins University, Baltimore, Maryland, United States of America, **8** Kavli Neuroscience Discovery Institute, Johns Hopkins University, Baltimore, Maryland, United States of America, **9** Halicioğlu Data Science Institute, University of California San Diego, La Jolla, California, United States of America, **10** Department of Neurobiology, University of California San Diego, La Jolla, California, United States of America, **11** Center for Imaging Science, Johns Hopkins University, Baltimore, Maryland, United States of America

\* [gmishne@ucsd.edu](mailto:gmishne@ucsd.edu) (GM); [adamsc@jhu.edu](mailto:adamsc@jhu.edu) (ASC)



**OPEN ACCESS**

**Citation:** Estrada Berlanga A, Kang GY, Kwok A, Broggin T, Lawlor J, Kuchibhotla KV, et al. (2026) Fast and accessible morphology-free functional fluorescence imaging analysis. *PLoS Comput Biol* 22(3): e1014038. <https://doi.org/10.1371/journal.pcbi.1014038>

**Editor:** Jinshan Xu, Zhejiang University of Technology, CHINA

**Received:** April 21, 2025

**Accepted:** February 17, 2026

**Published:** March 12, 2026

**Copyright:** © 2026 Estrada Berlanga et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data availability statement:** Code related to the development of the GraFT GUI can be found in the public repository <https://github.com/stradaa/GraFT-Application-Dev>. Documentation, short tutorials, and most up-to-date compiled application of our GraFT-App can be accessed

## Abstract

Optical calcium imaging is a powerful tool for recording neural activity across a wide range of spatial scales, from dendrites and spines to whole-brain imaging through two-photon and widefield microscopy. Traditional methods for analyzing functional calcium imaging data rely heavily on spatial features, such as the compact shapes of somas, to extract regions of interest and their associated temporal traces. This spatial dependency can introduce biases in time trace estimation and limit the applicability of these methods across different neuronal morphologies and imaging scales. To address these limitations, the Graph Filtered Temporal Dictionary Learning (GraFT) uses a graph-based approach to identify neural components based on shared temporal activity rather than spatial proximity, enhancing generalizability across diverse datasets. Here we present significant advancements to the GraFT algorithm, including the integration of a more efficient solver for the L1 least absolute shrinkage and selection operator (LASSO) problem and the application of compressive sensing techniques to reduce computational complexity. By employing random projections to reduce data dimensionality, we achieve substantial speedups while maintaining analytical accuracy. These advancements significantly accelerate the GraFT algorithm, making it more scalable for larger and more complex datasets. Moreover, to increase accessibility, we developed a graphical user interface to facilitate running and analyzing the outputs of GraFT. Finally, we demonstrate the utility of GraFT to imaging data beyond meso-scale imaging, including vascular and axonal imaging.

on our website <https://github.com/stradaa/GraFT-App>. Scripts for practical LASSO and compression optimization code are also available at <https://github.com/stradaa/GraFT-L1-Compression-Code.git>.

**Funding:** This work was supported by NIH R01EB026936 to GM, R01DC018650 to KVK, U19NS123717 to AEB, U19NS123717 to TM, U19NS123717 to DK, U19NS123717 to GM, R35NS097265 to TB, R35NS097265 to DK; NSF CAREER 2145247 to KVK; MathWorks (United States) through MATLAB Community Toolbox Training Fellowship to AEB; Swiss National Science Foundation to JL (P2SKP3\_164948 and P300PA\_177804) and by KAVLI through NDI Distinguished Postdoctoral Fellowship to JL. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** The authors have declared that no competing interests exist.

## Author summary

Calcium imaging enables the recording of large populations of cells in real-time within living organisms. However, current methods for analyzing these recordings often rely on the shapes of cells in the images, which can bias results and make it difficult to apply the same techniques across different types of data and scales. In our work, we introduce a significantly improved version of our algorithm, Graph Filtered Temporal Dictionary Learning (GraFT), that instead focuses on patterns of activity over time to identify groups of neurons. We made several key advances: we sped up the computations with new optimization tools, applied data compression techniques to improve efficiency, and built a user-friendly interface so that others can easily apply GraFT to their own work. We further showcase GraFT's strength in analyzing different types of imaging data and scales, from small blood vessels to long-range axons. By making analysis faster, more flexible, and more accessible, our contributions position GraFT as a robust and versatile approach to analyzing increasingly large and heterogeneous imaging datasets that support new directions in systems neuroscience.

## 1 Introduction

Scientific advancements in neuroscience are dependent on our capacity to observe and understand the intricate dynamics within neural tissue. Specifically, we need to analyze, and therefore discern, the activity of cells, neurites, neurotransmitters, and vasculature at both high spatial and temporal resolution, and at scale. To capture these complex dynamics across the varying scales of the brain, neuroscience has increasingly relied on advanced imaging techniques to record molecular and electrical signaling events indicative of brain function. Traditional non-invasive imaging techniques, while effective in their respective domains, often contend with limitations in terms of imaging speed, signal-to-noise ratio, and field-of-view scalability [1,2]. Amid these challenges, two-photon excitation microscopy (2P) remains a popular choice for its unique advantage of *in vivo* deep tissue penetration with optical sectioning, enabling imaging of both structural brain dynamics and cell function [3].

Genetically encoded reporters of neural activity can reliably record changes in fluorescence that represent the underlying neural-spiking-related biophysical changes in labeled neurons. Specifically, calcium is of great interest as calcium ion concentration within the cell changes with suprathreshold (neuronal firing) activity due to the influx of calcium ions into the neuron following the depolarization associated with action potentials [4,5]. The ability of calcium indicators coupled with multi-photon images to record *in vivo* at various scales—from dendrites and soma to widefield—at high resolutions has made calcium imaging an ideal approach for studying functional aspects of neural circuits [6,7].

In the wake of these imaging advances, the resulting data contains ever larger fields of view at high resolution (i.e., many pixels per image) and higher temporal resolution for longer sessions (many video frames). A significant current challenge is analyzing

the vast data volumes generated by such large-scale imaging efforts. Specifically, while traditional workflows relied on manual segmentation to identify single fluorescing objects and their temporal activities, manual annotation is infeasible in many modern datasets and so recent years have seen a shift towards automated analysis methods [8,9]. Current automated calcium imaging (CI) analysis [10–18] mainly decomposes recorded videos into two sets of variables: spatial profiles that represent neuronal occupancy, and their corresponding fluorescence time traces. These approaches, largely focused on somatic imaging, typically employ spatiotemporal regularization to reflect the sparse and well-localized nature of cell bodies. However, recent advances in CI, including finer-scale dendritic imaging and widefield imaging across the cortex, have diversified the range of observable neural structures and necessitated the development of more adaptable analysis techniques [18–20].

Relying solely on spatial structure can fail when the spatial morphology is complex, non-compact or does not match classical “neuron-shaped” components. In this context, the Graph Filtered Temporal Algorithm (GraFT) [18] was presented as a novel model for automated CI analysis that overcomes these shortcomings by replacing *spatial* regularization with regularization on a *graph* that captures correlations between per-pixel image time traces. The original publication demonstrated GraFT’s ability to analyze a variety of tissue morphologies across multiple spatial scales (e.g., widefield and dendritic imaging), and compared it against existing methods, establishing its validity and broad applicability to high-dimensional imaging data. Specifically, GraFT uses the graph-based relationships between pixels to learn a set of time traces that efficiently (i.e., sparsely) represent all pixels. The result is that each learned time trace corresponds to a single spatial profile, i.e., all the pixels whose temporal activity includes this time trace. These profiles are not constrained to be compact shapes, thus GraFT can identify components that are not spatially continuous or that span an entire FOV.

While GraFT has been successful in widefield, dendritic, somatic, and even functional ultrasound imaging (fUS) [21], the sparsity constraints and inherent matrix factorization aspect of the approach is computationally limiting for newer data taken at higher scales/resolution or at higher frame rates/longer duration. Current methods primarily deal with data limitations by downsampling [11,12], patching [17,18], or by simplifying the ROI detection problem (e.g., using deep learning methods) to a simple pixel-wise averaging instead of a full decomposition [22–24]. Downsampling is sub-optimal as it can reduce the ability to detect finer structures, such as axons, thinner dendrites, or spines. Patching can improve runtime by spreading the problem over more cores, but requires significant additional compute infrastructure as well as additional patch merging steps. Operating on space and time independently is also sub-optimal as summary images can be biased towards high-activity objects and averaging pixels can cause bleed-through of activity in densely labeled tissue.

In this work, we introduce a number of enhancements to the GraFT algorithm, aimed at speeding up its performance and improving its usability in the face of increasingly complex CI data. First, we integrate a more efficient solver for the L1 least absolute shrinkage and selection operator (LASSO) problem, optimizing the algorithm’s ability to accurately calculate spatial coefficients more rapidly and provide insights into parallelization parameters. Second, by incorporating principles of random projections from compressive sensing [25–27], we reduce the computationally-relevant data dimensions through a compression operation, while preserving the data quality, effectively reducing the computational workload without changing the output integrity. This approach significantly reduces the dimensionality of optimization problems within GraFT, including the calculation of weighted LASSO spatial coefficients and in the updating of the temporal dictionary. Furthermore, we implemented a Graphical User Interface that wraps the GraFT software to ensure it is accessible and easy to use, with both preprocessing capabilities and post-hoc visualization and labeling of results. We demonstrate GraFT on generate data as well as real-world cellular, vascular and axonal imaging, demonstrating GraFT’s broad applicability to neuronal morphologies. These improvements not only bolster the algorithm’s processing speed but also extend its applicability to larger and more diverse datasets, ensuring that GraFT remains a valuable alternative to other automated CI analysis tools in neuroscience.

## 2 GraFT algorithm

We first briefly review the GraFT framework for functional imaging analysis [18]. In Sec. 3–4 we present algorithmic speed-ups to this approach.

## 2.1 Overview

Given an  $N_x \times N_y$  FOV (containing  $N = N_x N_y$  pixels overall) imaged for  $T$  time-steps, the GraFT (Graph Filtered Temporal) algorithm is a method designed for extracting independent fluorescing objects and their temporal activity from functional microscopy videos. Specifically, GraFT leverages the principles of graph-regularized dictionary learning (DL), an unsupervised machine learning approach that decomposes signals as sparse combinations of elementary components (i.e., the dictionary), where the underlying graph structure promotes signals that are neighbors on the graph to have similar dictionary representations. Specifically for calcium imaging analysis, GraFT decomposes the pixel-by-time fluorescence data matrix ( $\mathbf{Y} \in \mathbb{R}^{N \times T}$ ) into a dictionary of time traces ( $\Phi \in \mathbb{R}^{T \times M}$ , where each column corresponds to the time trace of one of the  $M$  fluorescing components). These pixels are sparsely expressed across space, i.e., each pixel is associated with only a few times traces.

The spatial expression of each component is captured in the corresponding column of the matrix  $\mathbf{A} \in \mathbb{R}^{N \times M}$ , which contains the sparse spatial coefficients that reflect the neuronal structure contributing to each time trace in  $\Phi$ .

GraFT starts by modeling the data as the product of the spatial profiles and time traces, i.e.,

$$\mathbf{Y} = \mathbf{A}\Phi^T + \mathbf{E}, \quad (1)$$

where  $\mathbf{E} \in \mathbb{R}^{N \times T}$  represents independent sensor noise that is modeled as Gaussian. In addition to the matrix factorization likelihood, GraFT further assumes that 1) there is minimal spatial overlap between components, i.e., each pixel can contain activity from only a few of the components in the FOV, and 2) that similar pixels (measured by temporal correlations) are part of similar components. The former constraint results in a sparsity constraint over the rows of  $\mathbf{A}$ , and the latter is implemented by graph-based regularization. GraFT incorporates these priors into the overall probabilistic model by first constructing a graph over the pixels and then employing a variational expectation-maximization approach to alternate between solving for the spatial profiles  $\mathbf{A}$  given the time traces  $\Phi$  and vice versa. Thus, GraFT operates via three main subroutines: 1) Construction of a graph between pixels, 2) Estimation of spatial profiles given an approximation of the time traces, and 3) Estimation of the time traces given the spatial profiles.

## 2.2 Graph construction

The graph regularization is a key step in GraFT whereby the model re-organizes the data away from the spatial domain into a correlation-driven graph. GraFT accomplished this by constructing a graph such that each node represents a single pixel and each edge weight connecting two nodes (pixels) represents the similarity of the two corresponding pixels' time traces. The graph captures the intrinsic structure of the data independent of pixel location in the FOV, allowing for the exploitation of the temporal correlations present in the imaging data without explicit morphological constraints. Thus two pixels on the same dendritic branch can be linked on the graph, even if they are spatially far apart. The graph is constructed by first computing the weighted graph affinity matrix  $\mathbf{W}$ . Each element  $W_{ij}$  between pixels  $i$  and  $j$  is computed with a Gaussian kernel as

$$W_{ij} = \exp \left\{ -\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{\sigma_{ij}^2} \right\}, \quad (2)$$

where  $\mathbf{y}_i$  is the time trace of pixel  $i$  and the bandwidth  $\sigma_{ij} = \sigma_i \sigma_j$  is chosen according to a local adaptive approach [28]. The graph is constructed via a k-nearest neighbor search such that the resulting graph is sparse (each node is connected to only a small number of other nodes). This prevents many weak connections from over-powering smaller numbers of strong connections in larger FOVs with more pixels. Finally, the rows of  $\mathbf{W}$  are normalized to yield the diffusion graph filter  $\mathbf{K}$  as  $\mathbf{K} = \mathbf{D}^{-1}\mathbf{W}$ , where  $\mathbf{D}$  is a diagonal degree matrix with entries summing each row of  $\mathbf{W}$ .

### 2.3 Model inference

After constructing the graph kernel  $\mathbf{K}$ , GraFT then alternates between updating the spatial profiles  $\mathbf{A}$  and the time traces  $\Phi$ . The update over  $\mathbf{A}$  is accomplished via a regularized least-squares minimization based on sparse coding, where the sparse codes are aligned to the graph. This optimization is performed through the Re-Weighted  $\ell_1$  Graph Filtering (RWL1-GF) optimization procedure. In RWL1-GF, the component time traces (columns of  $\Phi$ ) that best represent each pixel are inferred simultaneously across all pixels while promoting minimal overlap through a non-negative weighted LASSO. The LASSO weights, which promote or suppress a given component from appearing at a given pixel, are then updated at each iteration based on how many of the pixels that are neighbors on the graph are also well represented by that component.

Mathematically, RWL1-GF iterates between the two steps of estimating the sparse spatial coefficients and updating the weights

$$\hat{\mathbf{a}}_i = \arg \min_{\mathbf{a} \geq 0} \left\{ \frac{1}{2\sigma_y^2} \|\mathbf{y}_i - \Phi \mathbf{a}\|_2^2 + \lambda_0 \sum_k \lambda_{ik} |\mathbf{a}_{ik}| \right\} \quad (3)$$

$$\lambda_{ik} = \frac{\xi}{\beta + |\mathbf{a}_{ik}| + [|\mathbf{K}\hat{\mathbf{a}}_k|]_i}, \quad (4)$$

where  $\mathbf{a}_i$  is the strength of all spatial profiles at pixel  $i$ ,  $\lambda_i$  are the LASSO weights for the coefficients,  $\beta$  is a weight parameters, and  $\lambda_0$  is included for numerical stability. The weights  $\lambda$  in RWL1-GF incorporate non-local spatial information into per-pixel solutions and maintain sparsity via applying the graph-based filter to the current estimate of the spatial coefficients  $\mathbf{K}\hat{\mathbf{a}}_k$  [16]. At a high level, if the spatial profile coefficients for component  $k$  of the  $i^{\text{th}}$  pixel's graph neighbors are high, the corresponding weight  $\lambda_{ik}$  will be low, encouraging that component to also be active in pixel  $i$ . Conversely, if the neighbor's spatial profile coefficients have low weights, the LASSO weight for that component at pixel  $i$  will be high and bias that component away from being active at that pixel.

Once the spatial profiles  $\mathbf{A}$  are updated based on the time trace dictionary, the dictionary  $\Phi$  is then updated based on the current estimate of  $\mathbf{A}$ . The complete dictionary update for the estimate at the  $i^{\text{th}}$  update iteration  $\hat{\Phi}^i$  is given by

$$\hat{\Phi}^i = \arg \min_{\Phi \geq 0} \left\{ \|\mathbf{Y} - \Phi \mathbf{A}^T\|_F^2 + \gamma_1 \|\Phi\|_F^2 + \gamma_2 \|\Phi^T \Phi - \text{diag}(\Phi^T \Phi)\|_{sav} + \gamma_3 \|\Phi - \hat{\Phi}^{i-1}\|_F^2 \right\}, \quad (5)$$

where here the sum-absolute-value (SAV) norm is defined as  $\|\Phi^T \Phi - \text{diag}(\Phi^T \Phi)\|_{sav} = \sum_{i \neq k} \Phi_i^T \Phi_k$ . The first term in this objective is a data fidelity term such that the dictionary and the sparse coefficients faithfully reconstruct the data. The second term regularizes the magnitude of the time traces. The third term is a regularization which aims to prevent distinct components having highly correlated time traces. The final term is an optimization smoothness term preventing subsequent algorithmic iterations from changing too drastically. The parameters  $\gamma_1, \gamma_2, \gamma_3$  trade-off between the relative importance of each term to the optimization solution.

Put together, the full GraFT algorithm is summarized in Algorithm 1.

#### Algorithm 1 GraFT DL.

Input: data  $\mathbf{y}_0$ , parameters  $\{\beta, \xi, \lambda_0, \gamma_1, \gamma_2, \gamma_3\}$   
 Initialize  $\hat{\Phi}$  randomly  
 Construct a kNN graph with weights  $\mathbf{w}$  via Eq (2)  
 Normalize  $\mathbf{K} = \mathbf{D}^{-1} \mathbf{w}$

```

while not converged do
  for all voxels do
    Initialize  $\lambda_{ik} = 1 \forall \{i, k\}$ 
    for  $l \leq 3$  do
      Re-Weighed L1 Graph Filtering (RWL1-GF)
      Update  $\hat{\mathbf{a}}_i$  via Eq (3)
      Update  $\lambda_{ik}$  via Eq (4)
    end for
  end for
  Update  $\hat{\Phi}^t$  via Eq (5)
end while

```

### 3 Practical LASSO optimization

#### 3.1 Problem formulation

A key step in GraFT is the constrained, weighted LASSO, a regression technique that imposes a non-uniform  $\ell_1$  norm over the regression coefficients. This constraint encourages sparsity in the model by effectively shrinking most coefficients to zero, thereby performing simultaneous variable selection and estimation. LASSO minimizes the residual sum of squares subject to the sum of the absolute values of the coefficients being less than a fixed value. The standard LASSO optimization problem can be expressed as:

$$\min_{\mathbf{a}} \left\{ \frac{1}{2} \|\mathbf{y} - \Phi \mathbf{a}\|_2^2 + \lambda \|\mathbf{a}\|_1 \right\}$$

where  $\mathbf{y} \in \mathbb{R}^n$  is the response vector (i.e., the pixel's time trace),  $\Phi \in \mathbb{R}^{n \times p}$  is the matrix of predictors (i.e., the time trace dictionary),  $\mathbf{a} \in \mathbb{R}^p$  is the vector of coefficients (i.e., the pixel's "expression" of each of the traces in the dictionary), and  $\lambda \geq 0$  is a regularization parameter. Here,  $\|\mathbf{a}\|_1$  is the  $\ell_1$  norm of the coefficients, defined as  $\sum_{j=1}^p |a_j|$ . The  $\ell_1$  norm can be expressed as a linear combination of auxiliary variables, which allows the problem to be formulated as a quadratic program (QP) problem with linear constraints.

Reformulating LASSO as a QP is crucial, as it changes the optimization from a non-smooth L1-regularized least-squares problem into a smooth quadratic program with linear constraints. This change enables the use of mature QP algorithms that exploit sparsity in the constraint matrix (which is the expected case in GraFT), take advantage of efficient factorizations, and offer robust convergence guarantees. Thus, in principle, the QP reformulation allows GraFT to leverage state-of-the-art solvers that scale better in the high-dimensional, repeated-LASSO regime required for large-scale imaging datasets.

To formulate LASSO as a QP, we introduce auxiliary variables  $u \in \mathbb{R}^p$  such that

$$u_j \geq |a_j|, \quad j = 1, \dots, p,$$

which is equivalently written as

$$-u_j \leq a_j \leq u_j, \quad u_j \geq 0, \quad j = 1, \dots, p.$$

This substitution replaces the non-smooth  $\ell_1$  norm with a linear term, yielding a QP with linear constraints:

$$\begin{aligned} \min_{\mathbf{a}, u} \quad & \frac{1}{2} \|\mathbf{y} - \Phi \mathbf{a}\|_2^2 + \lambda \sum_{j=1}^p u_j \\ \text{subject to} \quad & -u_j \leq a_j \leq u_j, \quad u_j \geq 0 \quad \forall j. \end{aligned}$$

In standard QP form this becomes

$$\begin{aligned} \min_{\tilde{\mathbf{a}}} \quad & \frac{1}{2} \tilde{\mathbf{a}}^T \mathbf{H} \tilde{\mathbf{a}} + \mathbf{f}^T \tilde{\mathbf{a}} \\ \text{subject to} \quad & \mathbf{C} \tilde{\mathbf{a}} \leq \mathbf{b}, \end{aligned}$$

where

$$\tilde{\mathbf{a}} = \begin{pmatrix} \mathbf{a} \\ \mathbf{u} \end{pmatrix} \in \mathbb{R}^{2p}, \quad \mathbf{H} = \begin{pmatrix} \Phi^T \Phi & 0 \\ 0 & 0 \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} -\Phi^T \mathbf{y} \\ \lambda \mathbf{1} \end{pmatrix}.$$

Here  $\mathbf{1}$  is a vector of ones,  $\mathbf{C}$  encodes the linear inequalities  $-u_j \leq a_j \leq u_j$ ,  $u_j \geq 0$ , and  $\mathbf{b}$  represents the corresponding bounds ensuring non-negativity and other problem-specific requirements.

### 3.2 LASSO and QP solvers

Choosing the appropriate QP solver for the given LASSO is crucial to the optimization and utility of the GraFT algorithm. The performance of QP solvers can significantly vary depending on the size and sparsity of the constraint matrix, making it essential to select the right approach for the specific problem at hand. Two principal iterative methods include *active-set* and *interior-point* solvers. Originating from the simplex method for linear programming [29], active-set methods (ASM) iteratively select and adapt a set of constraints that directly impact the solution in a given iteration. These methods are particularly beneficial in applications that require warm-start strategies, such as sequential quadratic programming (SQP) and model predictive control (MPC). Some drawbacks to ASM include the potential exponential growth in worst-case complexity with the number of constraints and issues like active-set cycling when constraint qualifications are not met [30].

The Interior Point Method (IPM) originally gained popularity as a polynomial-time algorithm for linear programming (LP) and was later extended to general convex optimization problems. Unlike ASM, IPM models iterate within the feasible region of the problem, approaching the solution from the 'central path' [31]. In the primal case, IPM methods use a *barrier function* to ensure that iterations remain within the feasible region and gradually reduce this barrier to zero as the solution is approached.

A barrier function for the constraints  $\mathbf{C}\mathbf{a} \leq \mathbf{b}$  can be written as follows:

$$g(\mathbf{a}) = - \sum_{i=1}^m \log(b_i - (\mathbf{C}\mathbf{a})_i)$$

where  $m$  is the total number of constraints. The modified objective function becomes:

$$\min_{\mathbf{a}} \left( \frac{1}{2} \mathbf{a}^T \mathbf{H} \mathbf{a} + \mathbf{f}^T \mathbf{x} - \mu \sum_{i=1}^m \log(b_i - (\mathbf{C}\mathbf{a})_i) \right)$$

where  $\mu$  is the barrier parameter, which is systematically decreased between iterations, smoothly guiding the solution towards the boundary of the feasible region and ultimately to the optimal point. This approach allows for significant progress per iteration, though each iteration is computationally expensive.

The time complexity of IPMs is typically  $O(n^3)$  per iteration of input size  $n$  due to the matrix factorization involved in each step. However, because IPMs require fewer iterations to converge than ASM, they are efficient for large-scale problems. On the other hand, the time complexity of ASM is typically  $O(kn^2)$ , where  $k$  is the number of iterations. Each iteration involves solving a linear system of equations, which is computationally efficient for small to medium-sized problems but may struggle to scale.

The choice between ASM and IPM depends on the specific characteristics of the problem, including size and sparsity of the constraint matrix. An MPC study comparing these methods found that ASM is computationally faster with fewer variables and constraints, whereas IPM scaled better with larger, more complex problems [32,33]. Conversely, ASM is suitable for smaller problems and adjusts its working set of constraints iteratively to find the optimal set, which can be computationally intensive for larger datasets. The ability to selectively choose between IPM and ASM solvers can reduce the computational burden, thereby leading to faster convergence.

Thus, the selection of a QP solver has a profound impact on the efficiency and performance of GraFT algorithm. ASMs are advantageous for problems requiring warm-start strategies and small to medium-sized datasets (for example when GraFT is parallelized over patches of the FOV rather than the full FOV), whereas IPMs are better suited for large-scale problems due to their fewer iterations and robust performance. If the GraFT pipeline is intended to run repeatedly on growing datasets, the warm-start strategies available to ASM may reduce computation times. In contrast, for very large datasets characteristic of wide-field imaging or whole brain recordings, IPM's better scaling properties are likely to outweigh the per-iteration cost. Adjusting the solution choice based on the size, complexity and required runtime efficiency of the data significantly enhances the overall performance of the algorithm. Here, we have implemented both solvers for the GraFT solution and compare the solvers in Sec. 7.

#### 4 Compressive GraFT for fast processing

While judicious selection of LASSO optimizers enables a computation speed-up, further speedups can be obtained by solving an inherent bottleneck given by the size of the data. Projections of the data onto spatial and temporal components (i.e., inner products along the space or time dimension) must come in multiples of the number of pixels  $N$  or time-points  $T$ . To further speed up GraFT we thus turn to computation in a compressed representation. Specifically, GraFT operates on the assumption that each pixel's time trace is sparse in the dictionary of all the time traces of the components in the video (i.e., minimal spatial overlap). Sparse signals can be highly compressed into a very low-dimensional subspace without extensive distortion of their geometry. This property has been extensively studied in the compressive sensing literature and can be quantified as follows: If  $\mathbf{y}_1$  and  $\mathbf{y}_2$  are both  $K$ -sparse in a representational dictionary  $\Phi$ , then a linear projection  $\Psi$  that does not distort distances satisfies, for all  $K$ -sparse  $\mathbf{y}_1$  and  $\mathbf{y}_2$ ,

$$\alpha(1 - \delta)\|\mathbf{y}_1 - \mathbf{y}_2\|_2^2 \leq \|\Psi\mathbf{y}_1 - \Psi\mathbf{y}_2\|_2^2 \leq \alpha(1 + \delta)\|\mathbf{y}_1 - \mathbf{y}_2\|_2^2. \quad (6)$$

This condition is known as the Restricted Isometry Property (RIP) and has parameters  $\alpha$ , which measures the inherent scaling of the operator  $\Psi$  and  $\delta$ , a value between zero and one that captures how distorted the low dimensional representation can be. For  $\delta = 1$  two distinct points can map onto each other, while  $\delta = 0$  represents a perfect preservation of geometry. The RIP ensures that relative distances between sparse points are not distorted, and can hold for large compression ratios in the projection  $\Psi$ . The preserved geometry means that many computations can be performed in the compressed space, saving computation time by operating over smaller vectors.

We implement random projections in the GraFT pipeline by compressing the data along the temporal dimension:  $\tilde{\mathbf{Y}} = \Psi\mathbf{Y}$ . The projection matrix  $\Phi \in \mathbb{R}^{\tilde{T} \times T}$  reduces the full time trace from length  $T$  to length  $\tilde{T} < T$ . In the context of our implementation, both  $\Phi$  and corresponding temporally-compressed input  $\tilde{\mathbf{Y}}$  are computed once, prior to the initiation of the main dictionary learning loop. The core two steps that GraFT alternates between are then minimally changed from a mechanical standpoint, operating over the compressed version of the data instead of the full version.

The first primary savings comes in the form of a reduced space in which to estimate the coefficients at each LASSO step:

$$\mathbf{a}_{ij} = \arg \min_{\mathbf{a}} \left\{ \|\Psi\mathbf{y}_{ij} - \Psi\Phi\mathbf{a}\|_2^2 + \|\Lambda\mathbf{a}\|_1 \right\} = \arg \min_{\mathbf{a}} \left\{ \|\tilde{\mathbf{y}}_{ij} - \tilde{\Phi}\mathbf{a}\|_2^2 + \|\Lambda\mathbf{a}\|_1 \right\}, \quad (7)$$

where  $\tilde{\Phi} = \Psi\Phi$  is a compressed version of the component time traces. Note that since  $\mathbf{a}_{ij}$  is sparse, we expect that the compressed data will be sufficient to recover  $\mathbf{a}_{ij}$ .

The second savings comes in the update for the compressed dictionary  $\tilde{\Phi}$

$$\tilde{\Phi} = \operatorname{argmin}_{\tilde{\Phi}} \left\{ \|\tilde{\mathbf{Y}} - \tilde{\Phi}\mathbf{A}^T\|_2^2 + R(\tilde{\Phi}) \right\} \quad (8)$$

where  $R(\Phi)$  represents the additional regularizers in Eq 5. This problem is over a much lower dimensional space ( $\tilde{T} \times N$  instead of  $T \times N$ ), greatly speeding up the computation. This compressed approach results in the estimate of  $\mathbf{A}$  and  $\tilde{\Phi}$ , i.e., we obtain the spatial profiles as before but only a compressed version of the time traces. To recover the full time traces, we can solve Eq 5 once at the end of the optimization using the inferred  $\mathbf{A}$  values and the uncompressed data  $\mathbf{Y}$ .

## 5 GraFT graphical user interface

To enhance the usability and accessibility of the GraFT algorithm for the community, we developed a dedicated graphical user interface (GUI) that consolidates preprocessing, parameter setup, algorithm execution, and results visualization into a single workflow (Fig 1). The GUI is offered in two forms: a stand-alone compiled application for Windows systems and a MATLAB-based app compatible with MATLAB R2024a or newer (Users are encouraged to consult the GitHub GraFT wiki site for detailed tutorials on how to employ preprocessing features, parameter selection, and analysis of GraFT results.). Both versions are designed to simplify adoption of GraFT for imaging analysis across modalities and spatial scales.

Upon launching the GUI, users can load datasets in multiple file formats, including \*.h5, \*.mat, as well as common image formats (\*.tiff). The interface supports essential preprocessing steps such as interactive cropping for visually selecting regions of interest, manual or semi-automated mask selection to define or exclude specific areas (e.g., in sparse dendritic imaging), motion correction to address drift artifacts, and wavelet-based denoising to improve signal clarity.

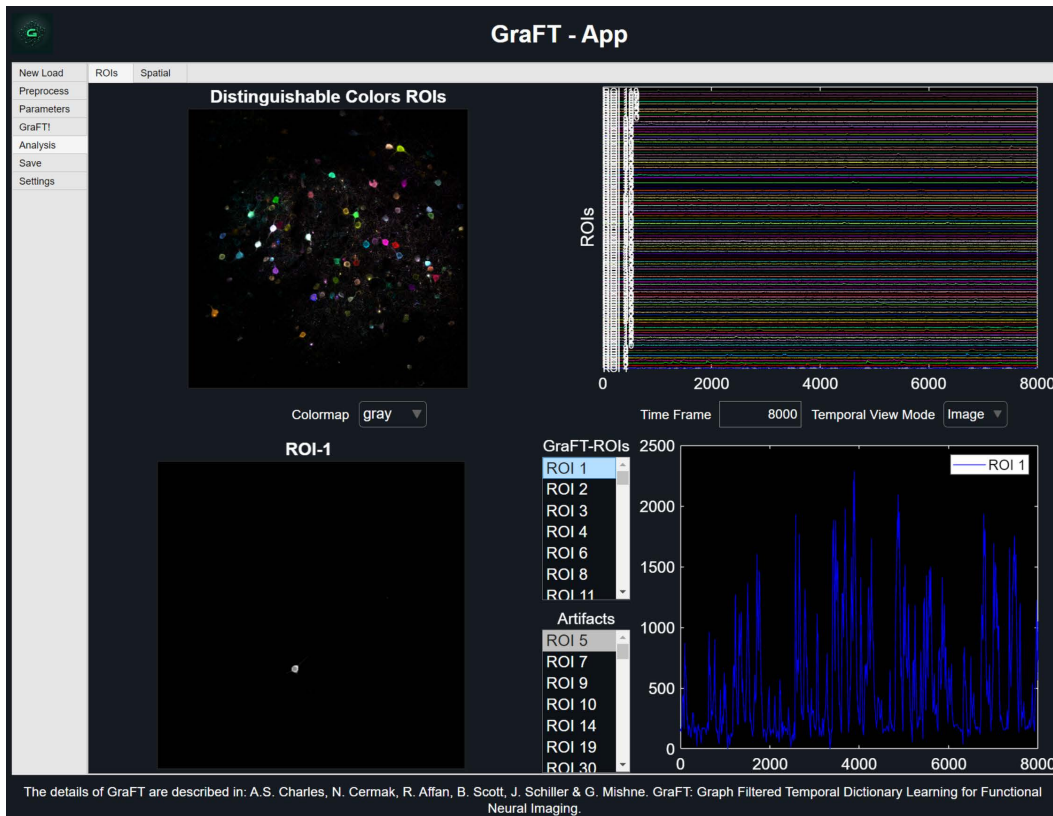
Prior to running the GraFT algorithm, users can save the algorithm parameter file with project name and author identifiers for reproducibility. Previously saved parameter configurations can also be loaded or shared with other users separately from the project, which is ideal for facilitating reproducible analyses and working collaboratively. After running GraFT algorithm, the analysis tool within GraFT-App allows for quick viewing of all identified spatial profiles and time traces, selected individual or groups of spatial profiles along with their time trace. We also include labeling functionality between real neural components and artifacts.

## 6 Experimental Results

In this section, we evaluate GraFT's computational efficiency and reconstruction accuracy. We compare two LASSO solvers—quadprog (IPM) and MPC (ASM)—using the NeuroFinder dataset, showing that although both yield similar outputs, quadprog's runtime increases nearly twice as fast with added dictionary elements. We also assess the effect of compression ( $2\times$ – $1024\times$ ) on time trace and spatial profile reconstruction, demonstrating substantial speedups and reduced RAM usage with minimal loss in accuracy. Finally, we demonstrate GraFT on two non-cellular morphologies and modalities: widefield imaging of wide-field pial arterioles and two-photon axonal imaging.

### 6.1 QP computational results

First we isolate and test the differences in using different solvers for the LASSO optimization. Recognizing the limitations of each solver in certain scenarios, we compared the default GraFT IPM solver quadprog and its ASM counterpart. Among available ASM implementations, active-set methods are widely used in MPC, where warm-start strategies and faster convergence are crucial. The mpcActiveSetSolver from MATLAB's MPC Toolbox [34] is particularly relevant for GraFT



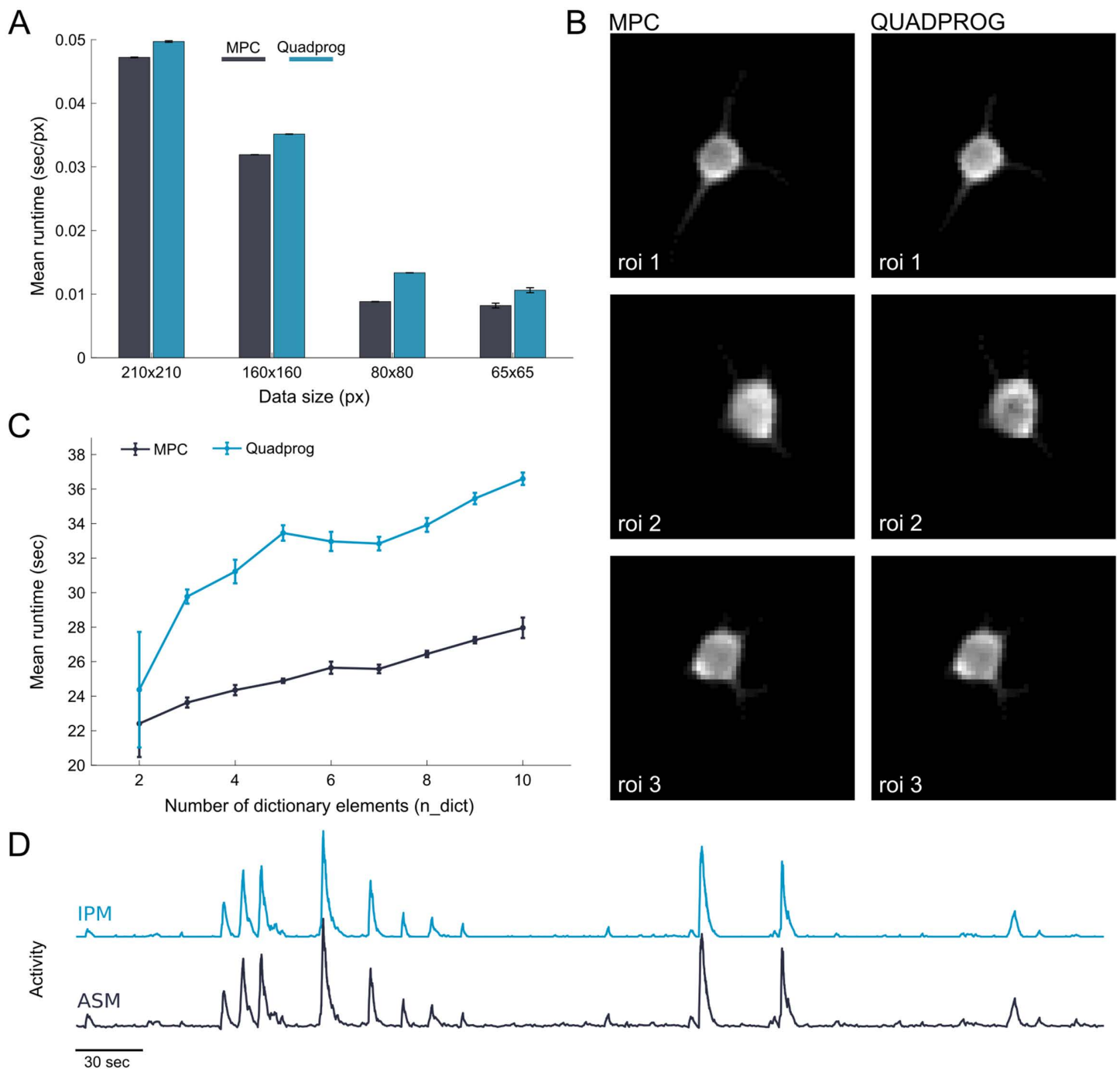
**Fig 1. Results display from GraFT-App.** (A) The distinguishable colors ROIs figure shows the overlay of all of the dictionary elements found, with their color matching their time trace in (B).

<https://doi.org/10.1371/journal.pcbi.1014038.g001>

because it is designed for repeated, structured QPs and employs the KWIK algorithm [35] for efficient active-set adjustments. This solver optimizes the selection process of constraints likely active at the optimal solution, thereby expediting convergence and reducing computational load [35]. The NeuroFinder dataset [36] was motion corrected using the Non-Rigid Motion Correction algorithm (NoRMCorre) and denoised using simple time trace wavelet denoising [37]. The processed ( $512 \times 512$ ) pixels  $\times$  8000 time-frames NeuroFinder data was then cropped into various sizes for testing.

We tested the GraFT algorithm runtime on ( $210 \times 210$ ,  $160 \times 160$ ,  $80 \times 80$ ,  $65 \times 65$ ) pixel  $\times$  8000 time frames with IPM and ASM solvers. In all data size comparisons, the ASM solver with MPC was faster than the IPM solver with little to no differences in quality of temporal and spatial results. Each data size was run 12 times, and runtime was tracked for each. The relative difference between the two solvers grows with decreasing data size (Fig 2A).

We further compared runtime differences of ASM and IPM solvers by varying the number of dictionary elements over a consistent ( $50 \times 50$ ) pixel window. In our patchGraFT implementation, as the number of dictionary elements increased, the runtime for the quadprog solver rose more steeply than that of MPC (Fig 2B). Linear regression on the experimental data revealed that the MPC solver's runtime increased by approximately 0.63 seconds per additional dictionary element, whereas the quadprog solver's runtime increased by about 1.18 seconds per additional dictionary element. This indicates that for every extra dictionary element (i.e., additional neural component to be identified), quadprog incurs nearly twice the additional computation time of MPC, highlighting a significant advantage for warm-start strategies with small datasets via ASMs solvers.



**Fig 2. Comparison of ASM (MPC) and IPM (quadprog) solvers.** (A) Runtime of solvers normalized by window size (pixel area). (B) Sample ROIs spatial results of the NeuroFinder dataset between both solvers. (C) Comparison of runtime between MPC and quadprog solvers by varying dictionary elements over 50x50 pixel area using patchGraFT. (D) Sample time trace of the same ROI (#3) from IPM and ASM solvers.

<https://doi.org/10.1371/journal.pcbi.1014038.g002>

## 6.2 Comparison between compressed and uncompressed GraFT

We evaluated the computational and accuracy performance of compressed GraFT using a  $490 \times 490 \times 100 \mu\text{m} \times 9,000$  time points (equivalent to 3 minutes) generated with NAOMi's default parameters (details in App. A). The degree of compression varied from  $2\times$  to  $1024\times$ , and we used both normalized and non-normalized sketching matrices.

First, we examined the preservation of time trace reconstruction in compressed GraFT using a non-normalized sketching matrix ( $\lambda = 0.7$ ), by leveraging the known ground truth events present in the synthetic NAOMi dataset. For each trace, we calculated the correlation coefficient between output GraFT trace and input NAOMi ground truth. Each GraFT trace was tagged by its highest correlation coefficient, allowing multiple GraFT traces to map to the same NAOMi trace. Compiling the trace correlation tags into a histogram (Fig 3A) displays in skewed right behavior. As compression level increases up to  $16\times$ , the trace correlation histogram begins to flatten at lower correlation levels, and GraFT identifies more traces. However, as compression level further increases up to  $512\times$ , increasingly many traces, previously unidentified traces with low correlation ( $\rho < 0.3$ ) to the ground truth NAOMi data are found, leading to an increased amount of raw identified traces.

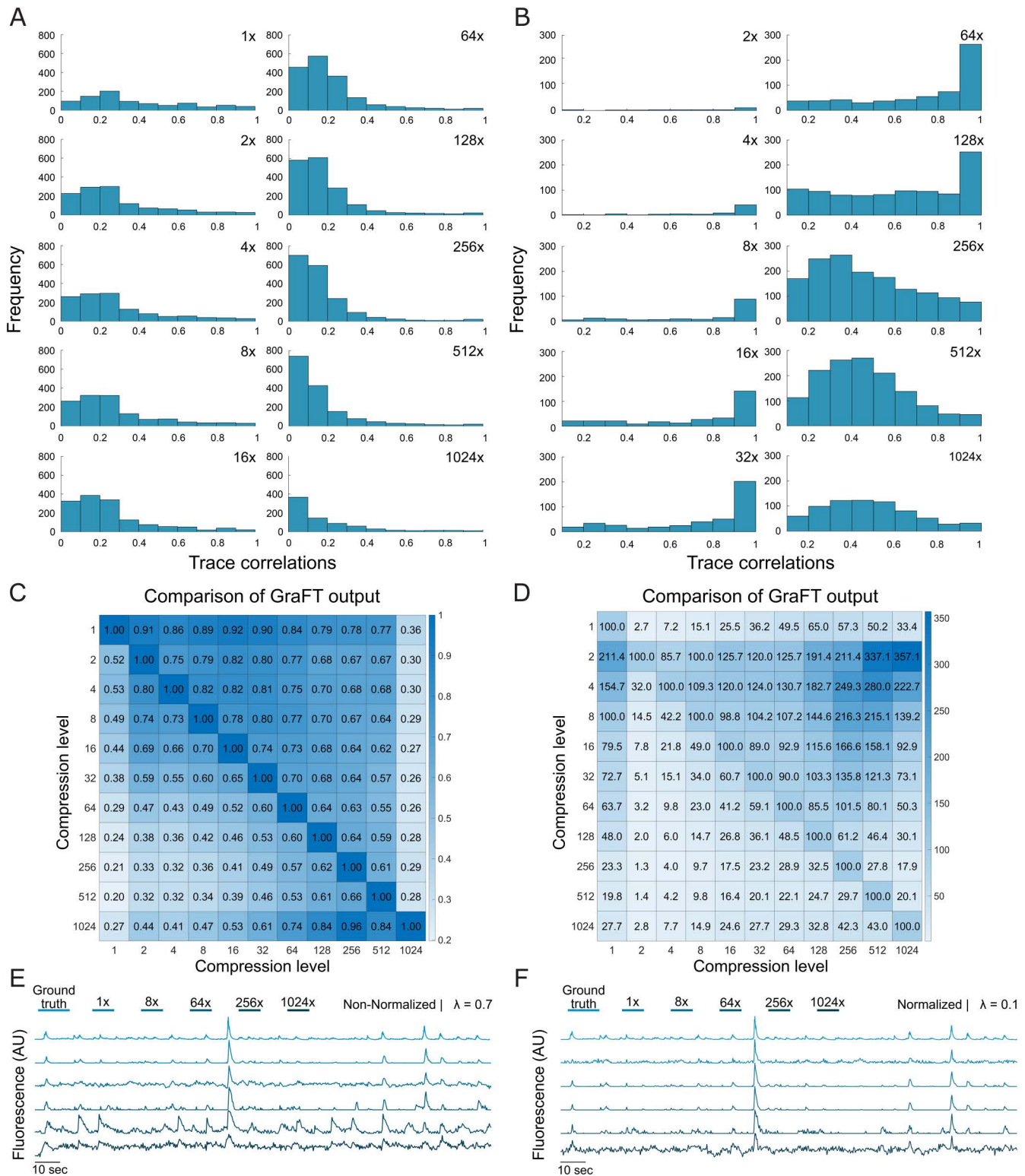
Examining individual time trace profiles, we see preservation in trace correlations across compression levels. Strong trace correlations (Fig 3E) retain high correlation across increasing compression magnitudes. Moderate trace correlations to the ground truth trace gradually decay in correlation, with the most noticeable drop in correlation starting at  $64\times$  compression (S1 Fig).

The output traces from compressed GraFT were also compared against other GraFT compression level outputs to assess self-agreement of the GraFT algorithm (Fig 3B). Specifically, we select one GraFT output as the pseudo-ground truth (listed on the y-axis) and another GraFT output as the baseline run (listed on the x-axis). The correlations between this pseudo-ground truth and comparison were calculated per trace. We performed the same highest correlation tagging procedure as before, but computed the number of GraFT traces with a correlation  $\rho > 0.5$  in the comparison as a percentage with respect to the number of traces identified in the y-listed compression level GraFT run. With this method,  $> 100\%$  relationships indicate that more traces are found in the comparison than the ground truth and that multiple traces may map to the same ground truth trace. Higher percentages in the heatmap correspond to higher levels of self-agreement in the output GraFT dictionary, and we see consistency across compression levels. This suggests that increasing the compression level identifies similar traces as lower compression levels. Using the first row, we see that at least 75% of time traces identified by uncompressed GraFT are also identified with correlation coefficient  $\rho > 0.5$  in non-normalized, compressed GraFT up to  $512\times$  compression. Additionally for all combinations of levels up to  $64\times$  compression, lower compression levels have at least 70% agreement with higher compression levels (upper triangular). This indicates that higher compression levels consistently identify similar traces to lower compression level but, when combined with the observation that the lower triangular has lower agreement percentages, also indicates that a higher raw number of traces are identified.

For comparison of normalized GraFT time trace reconstruction, the sparsity parameter was changed to account for compression (from  $\lambda = 0.7$  to  $\lambda = 0.1$ ). With the lower  $\lambda$  parameter, notably fewer traces are identified overall (S1C Fig), but the remaining traces identified by normalized GraFT have higher correlation to the ground truth across compression levels (Fig 3B). However, we see that the percentage of highly correlated traces in respect to uncompressed GraFT as the pseudo-ground truth has a maximal performance around 65% at  $128\times$  compression (Row 1, Fig 3D). Although the maximal performance is less than with non-normalized GraFT (which had maximal performance around 80% at the same  $128\times$  compression level), the strength of the correlation is much higher for the identified traces (Fig 3B). The individual time traces demonstrate similar preservation of trace correlation strengths across compression levels, with more stable preservation of mid-correlated traces (S1B Fig).

## 6.3 Comparison of spatial profile reconstruction in graFT

For evaluation of spatial profiles, we used a  $500 \times 500 \times 100 \mu\text{m} \times 20,000$  time point at 30 Hz (approximately equivalent to 11 minutes) synthetic NAOMi dataset for analysis. The individual spatial profiles generated from non-normalized GraFT



**Fig 3. Effects of time compression on temporal components identified by non-normalized and normalized GraFT.** Histogram of highest time trace correlations to the NAOMi ground truth for (A) non-normalized GraFT and (B) normalized GraFT. Heatmap for (C) non-normalized and (D) normalized GraFT results across compressions levels. Each entry is the percentage of traces in the row-wise compression result that correlated at  $\rho > 0.5$  with the column-wise result. Example highly correlated traces ( $\rho > 0.7$  for all compression levels) in (E) non-normalized GraFT and (F) normalized GraFT.

<https://doi.org/10.1371/journal.pcbi.1014038.g003>

preserve important structures but have increased background noise as compression level increases. We note that time traces with  $\rho < 0.5$  correlation to the ground truth still have relevant spatial information at  $64 \times$  compression (Fig 4A). However, when examining the full volume, there are strong patch boundaries (where individual regions of the image are brighter or darker than neighboring patches) and a reduction in captured cells (Fig 4B). Spatial profile visualization can be improved via post-processing techniques adjusting filter sensitivity or contrast, indicating that the “missing” spatial profiles are still identified but experiencing pixel intensity modulation from the introduction of the sketching matrix (Fig 4C).

By implementing normalization among the sketching matrix, we are able to see less noisy spatial profile visualization with the default display parameters. Additionally, we are able to directly visualize the effect of modifying of the sparsity coefficient,  $\lambda$ , where decreasing the coefficient identifies more spatial profiles, and correspondingly use the results from  $\lambda = 0.1$  for subsequent analysis on spatial profiles corresponding to the time traces highlighted in Fig 3. We see that non-normalized GraFT results have more background contamination in the case of the higher correlated trace and worse recognition of a cell ROI for the mid correlated trace (Fig 4D).

#### 6.4 GraFT speedup

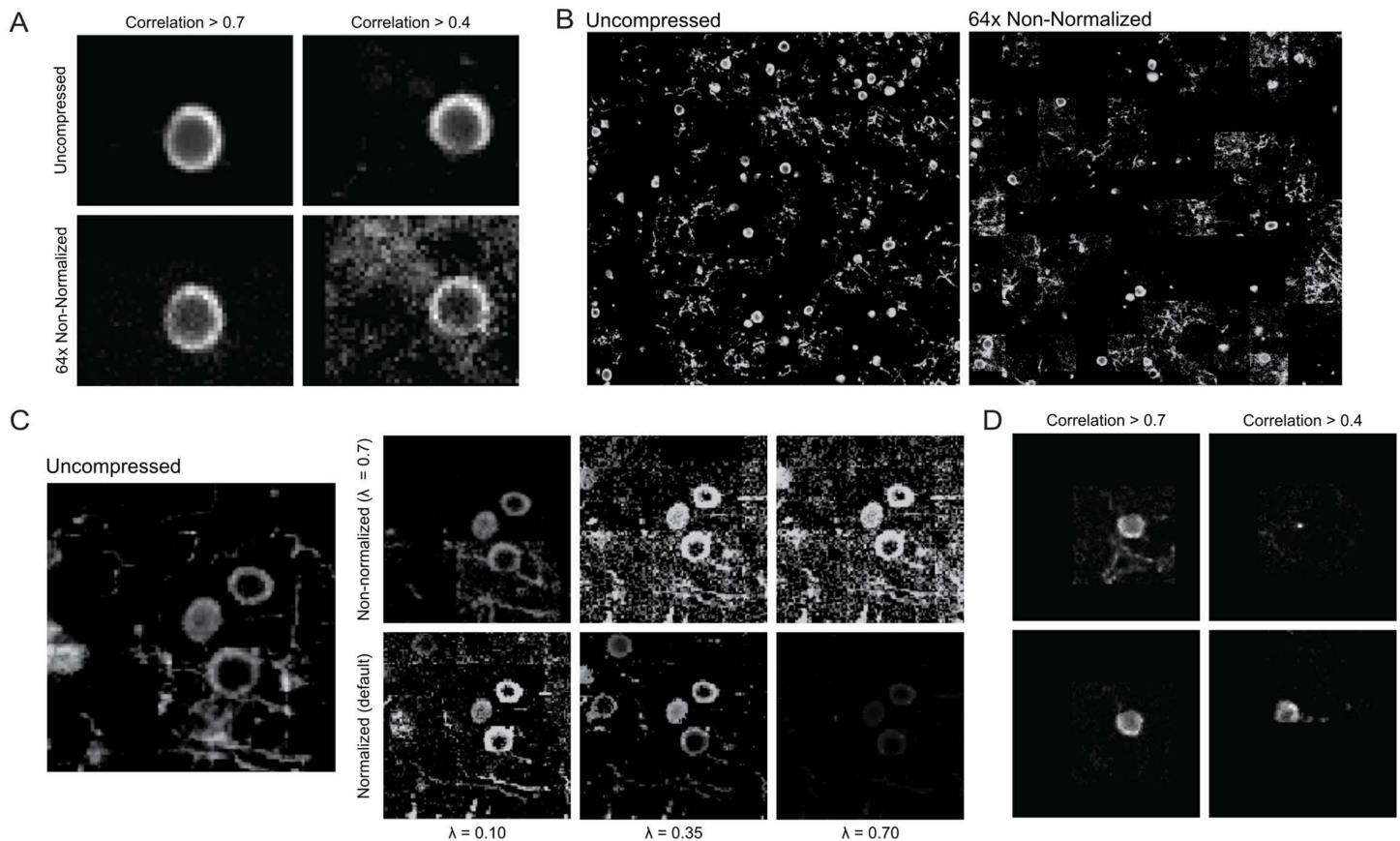
We used the inbuilt MATLAB profiler to evaluate the run time and RAM usage of GraFT at different compression levels. In addition to extracting the metrics for the total algorithm, we also extracted the metrics of the dictionary update step (which was expected to show the most improvement). The relative metric at each compression level was calculated by dividing the compressed metric by the uncompressed metric. We see a comparable relationship between relative algorithm speedup and compression level across the dictionary update step ( $8 \times$  relative speed up at  $64 \times$  compression). Generally, relative speed up follows a root based trend, where speed up is equivalent to the square root of compression level at the time intensive step. The notable speed up in terms of total algorithm runtime is less significant, stabilizing at  $3 \times$  relative speed up at  $64 \times$  compression. On the other hand, relative RAM usage demonstrates consistent root based trends in both the time intensive and overall algorithm step (Fig 5).

#### 6.5 Analysis of diverse morphologies with GraFT

GraFT is designed as a morphology-agnostic approach for functional imaging analysis. Here we highlight two non-somatic imaging datasets, to complement the cellular data in previous sections.

**6.5.1 Vasculature widefield data.** We use GraFT to analyze calcium imaging of pial arterioles of a mouse at rest from the study in [38], in order to identify sub-structures within the pial arteriole network and their corresponding time traces. Using a masking option, we restrict the analysis only to pixels that fall within the mask of the pial arterioles, resulting in 6197 pixels by 2223 time frames (see supplementary for details on mask construction). In Fig 6A we present the extracted spatial profiles. GraFT extracts five profiles that all exhibit bilateral symmetry and partition the arterial network into distinct regions. Note that due to the graph-based nature of GraFT, a spatial profile need not be contiguous and can be composed of multiple segments. In Fig 6B we present the time trace dictionary, where each trace corresponds to one of the extracted spatial profiles in Fig 6A. Finally, Fig 6C shows Pearson’s correlation across the spatial and temporal components, exhibiting low correlation between the spatial profiles. Profiles that are closer together spatially are more correlated temporally.

**6.5.2 Axonal data.** We use GraFT to analyze calcium imaging of cholinergic basal forebrain (CBF) axons imaged in the Primary Auditory Cortex (A1) of the mouse. Data was collected as in [39], resulting in a  $312 \times 192 \mu m^2$  area at a frame rate of 15.63 Hz. The mean image of a region of the field of view as well as the overlay of the primary components identified by GraFT are shown in Fig 7A (left and right, respectively). GraFT extracts 4 spatial profiles that are distributed across the field of view (Fig 7B), reflecting the distributed nature of axonal trees in cortical target regions. For the same field of view, suite2p [12] ROI segmentation is an order of magnitude larger (48 components total) which would require significant post-processing to merge segments belonging to the same axonal track. The fluorescent trace of each spatial



**Fig 4. Effects of time compression on spatial profiles identified by non-normalized and normalized GraFT.** (A) Plot of spatial profiles comparing uncompressed GraFT and 64 ×, non-normalized GraFT which had highly correlated time traces ( $\rho > 0.7$ ) and medium correlated time traces to the ground truth ( $0.4 < \rho < 0.7$ ). (B) Plot of all spatial profiles with corresponding time trace correlations  $\rho > 0.5$ . (C) Spatial profile comparison within a defined ROI between non-normalized,  $\lambda = 0.7$ , 64 × compressed GraFT and normalized, 64 × GraFT. The non-normalized ROI was post-processed using different filtering parameters to improve visualization, while the normalized ROI used the default visualization parameters across different sparsity  $\lambda$  coefficients. (D) Comparison of non-normalized,  $\lambda = 0.7$ , GraFT (top) and normalized GraFT,  $\lambda = 0.1$ , GraFT (bottom) spatial profiles comparing to the time traces shown in Fig 3.

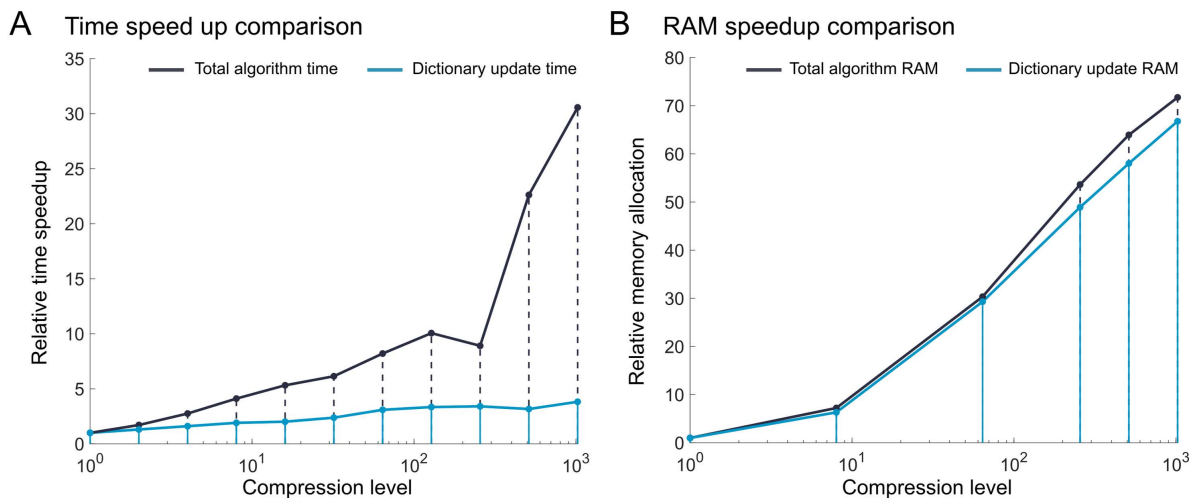
<https://doi.org/10.1371/journal.pcbi.1014038.g004>

GraFT component is displayed in Fig 7C for the example imaging session. Large increases in fluorescence reflect ongoing cholinergic tonic activity (highlighted with the average activity of the component and its corresponding mean image). Tonic activity is component-specific indicating that GraFT, identified distributed spatial profile with distinct temporal profiles.

## 7 Conclusions and discussions

In this work we build on the GraFT methodology in several ways. First, we introduce computational improvements that increase the speed of component identification. Second, we improve the usability of GraFT through a compiled graphical user interface. Finally, we demonstrate the extent of GraFT's ability to identify components in novel large-scale imaging applications, including vascular and axonal imaging.

The computational advances we present target the bottlenecks of performing the weighted  $\ell_1$  regularized optimization in the spatial profile estimation and the least-squares time trace update. For the former, we tested multiple solvers, identifying significant gains by switching to an MPC solver over a more traditional IPM. For the latter, we implemented a



**Fig 5. Time speedup and RAM allocation improvement by compression level. (A)** Relative time speedup as a function of compression. **(B)** Relative RAM usage as a function of compression.

<https://doi.org/10.1371/journal.pcbi.1014038.g005>

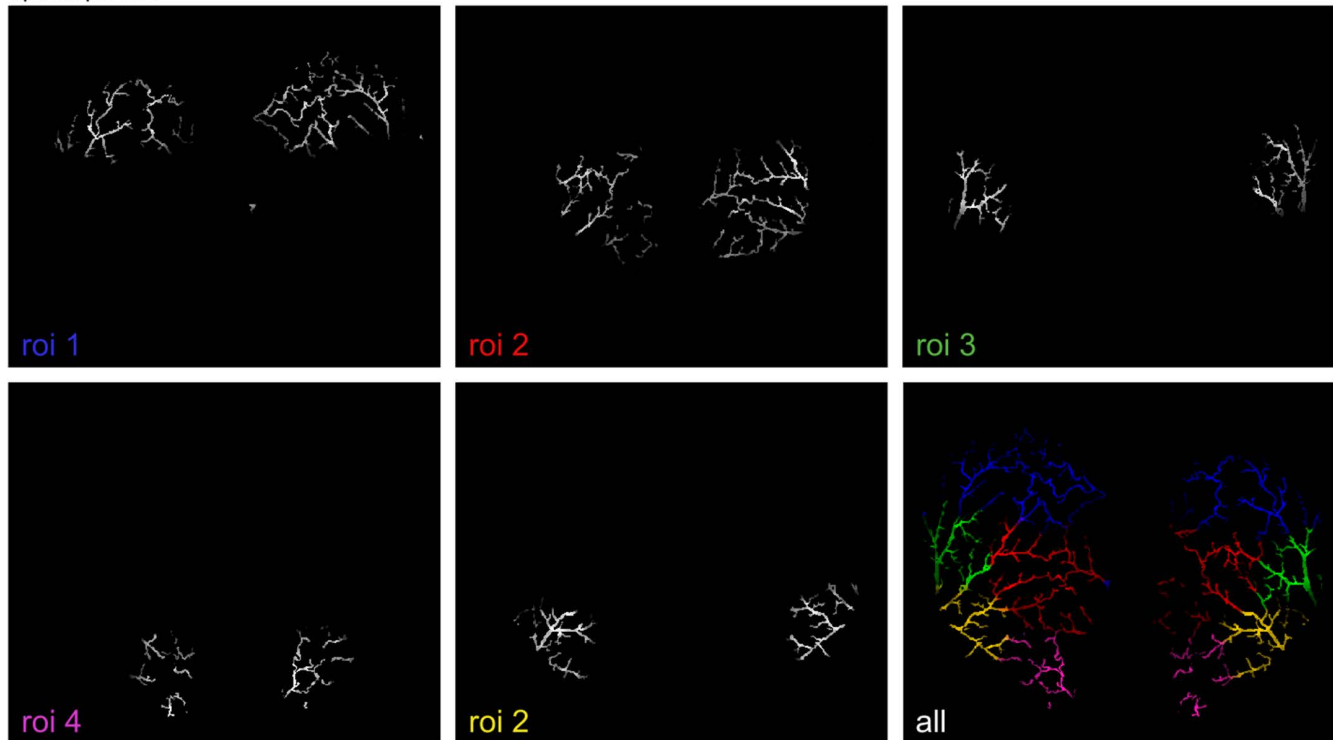
compressive computation, reducing the effective size of the data in the least squares problem and therefore the computation time.

The comparative analysis of ASM and IPM solvers for the LASSO optimization within the GraFT algorithm reveals distinct tradeoffs that are highly dependent on dataset characteristics. The ASM solver, implemented via MATLAB's `mpcActiveSetSolver`, consistently demonstrated faster runtimes compared to the IPM-based `quadprog` approach, particularly for small to medium-sized data windows. This improved performance is largely attributed to its effective warm-start strategy and lower per-iteration computational cost. While IPM methods benefit from robust convergence properties and are theoretically well-suited for large-scale problems due to fewer iterations, their higher per-iteration computational expense becomes a limiting factor when applied to smaller or less complex datasets. In contrast, the ASM method's ability to rapidly adjust its working set of constraints makes it a more efficient choice in scenarios where rapid, repeated optimizations are required.

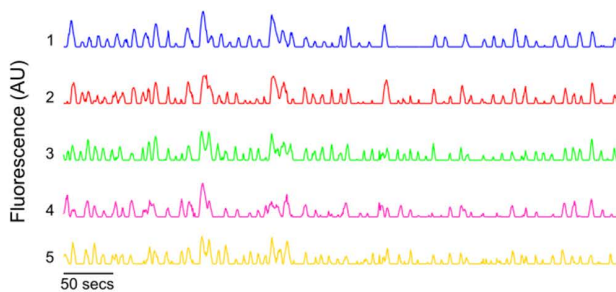
While the speedups afforded by changing the LASSO optimization are direct replacements of specific steps in the code, the compressive computation changes the data size and magnitudes, therefore changing the balance of sparsity and data variance. We thus recommend tuning the sparsity coefficient,  $\lambda$ , and normalized compressed GraFT in a majority of cases. Normalizing GraFT in this way provides much better stability at higher compression levels and spatial profile comparison, making it the appropriate choice for larger data that can take the most advantage of high compression levels. However, in a specific case where time traces information is strongly prioritized over spatial information, non-normalized GraFT remains a good alternative due to its strong self-consistency of trace preservation up to 64X compression and its ability to identify a more comprehensive collection of time traces.

Our newly developed GUI significantly enhances the accessibility and efficiency of GraFT by providing user-friendly parameter configuration, algorithm execution, and preliminary result filtering within a single application. Both the stand-alone and MATLAB-based versions are also helpful in maintaining flexibility with existing pipelines. Future releases will expand data-format support, including Neurodata Without Borders (NWB) compatibility [40], to further extend the platform's utility and maintain its user-oriented design.

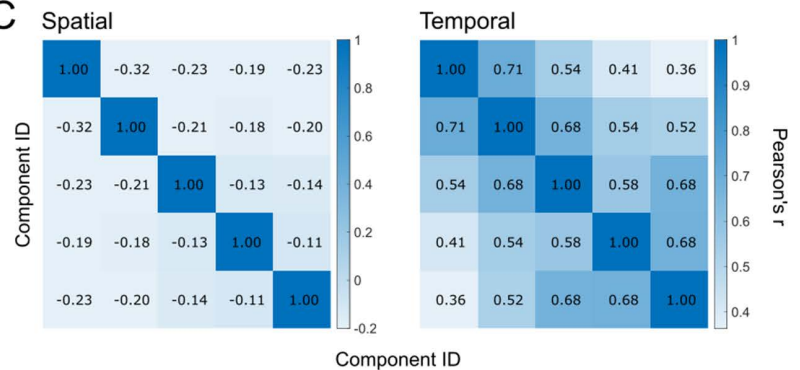
A Spatial profiles



B Time traces



C



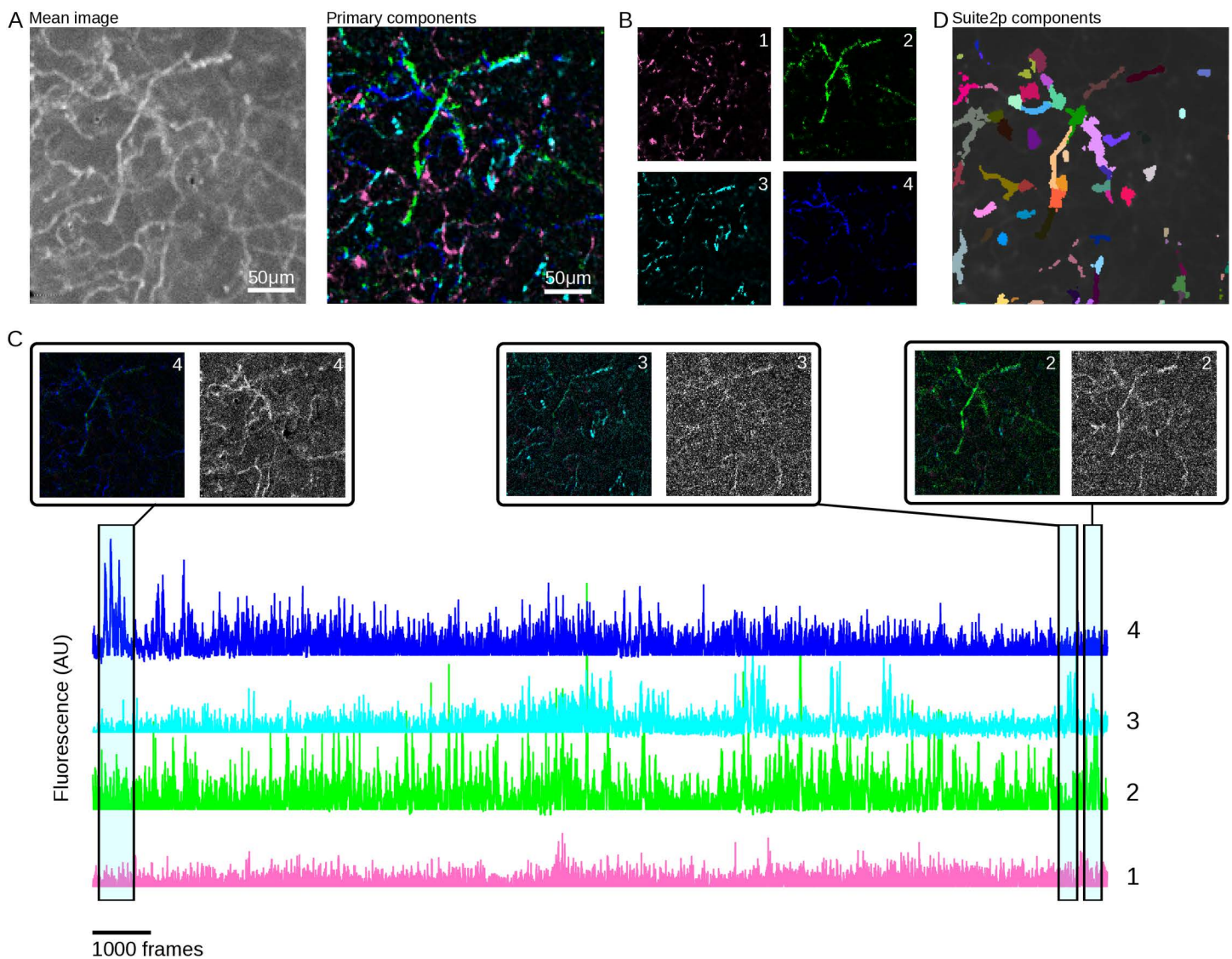
**Fig 6. GraFT algorithm results on calcium imaging of pial arterioles.** (A) Spatial profiles found along with an overlay of all five ROIs with distinguishable colors. (B) Respective time traces of the spatial profiles detected. (C) Component spatial and temporal linear correlations using Pearson's correlation coefficient ( $r$ ).

<https://doi.org/10.1371/journal.pcbi.1014038.g006>

A Data

NeuroFinder

We analyzed one of the NeuroFinder datasets [36], which contained two-photon calcium imaging at the somatic level, as well as manually annotated neuron spatial profiles. We ran GraFT on a  $445\mu\text{m} \times 445\mu\text{m}$  field of view in area vS1 of mouse visual cortex, recorded at 8 Hz.



**Fig 7. Results of GraFT on axonal imaging.** (A) Mean image (left) and four identified components (right) show that GraFT finds networks of axons across the full field-of-view. (B) The four components shown separately for clarity. (C) Example component time traces, color coded to match panel B. Local temporal averages for three time-points, each with the components displayed with intensity weighted by the respective integrated fluorescence over that time-window, and the local video average of the same time window show that the full axonal group is active when the components have high activity levels. (D) Suite2p results on the same data breaks up the axons into multiple segments, similar to dendritic imaging [18].

<https://doi.org/10.1371/journal.pcbi.1014038.g007>

## NAOMi

For evaluation of compression on GraFT, we generated both the  $(490 \times 490 \times 100) \mu\text{m} \times 9,000$  time point (equivalent to 3 minutes) and  $(500 \times 500 \times 100) \mu\text{m} \times 20,000$  time point (approximately equivalent to 11 minutes) in NAOMi with a sampling rate of 30Hz (`spike_opts.dt=1/30`). Specifically, the imaging was simulated using a 0.6NA Gaussian beam (`psf_params.NA=0.6`), 0.8NA objective (`psf_params.objNA=0.8`), and 40 mW power (`tpm_params.pavg=40`). The imaging depth was set to the middle of the simulated volume (`vol_params.vol_depth=100`).

## Vasculature

We analyze a single trial from the dataset collected in [38]. Imaging was performed in mice with GCaMP8.1 in arteriole smooth muscles at 4.47 Hz. We used an extended thinned skull window that spanned an approximately 10 mm by 10 mm region of skull over the cortical mantle. To create the pial arteriole mask, we took the standard deviation over time of the trial's first 1250 diffusion-filtered GCaMP8.1 fluorescence images. The result was normalized and contrast-enhanced using MATLAB's "adapthisteq" function. Adobe Photoshop was used to sharpen, threshold, and manually connect pial arteriole segments that fell below the threshold. The thin-skull window edge was then manually defined and saved as the "rim" mask. The pial arteriole mask was used to define a vessel graph using procedures developed for reconstruction of the 3-dimensional microvascular connectome [41]. Briefly, the two-dimensional mask was skeletonized and diameter was estimated using MATLAB's "bwskel" and "bwdist" respectively. Skeleton pixels, which define the vessel centerline, were classified as either an endpoint, a link that connects two neighbors, or a node that connects three or more neighbors. Skeleton pixels were downsampled and dilated by the vessel diameter to define skeleton segments. We then calculated the raw fluorescence signal  $F(x, t)$  as the median intensity of pixels within each skeleton segment at each time point. Finally,  $\frac{\Delta F(x, t)}{\langle F(x, t) \rangle}$  was calculated and stored for subsequent analyses, where  $\Delta F(x, t) = F(x, t) - \langle F(x, t) \rangle$  and  $\langle F(x, t) \rangle$  denotes the time-averaged intensity at each location  $x$ . Additional experimental details can be found in [38].

## Axonal data

Cholinergic axonal imaging data were collected as described in [39] from the auditory cortex of ChAT-Cre mice (stock no. 006410, Jackson Laboratory; 2–3 months old). To label cholinergic neurons and their axonal projections, an AAV vector encoding the calcium indicator Axon-GCaMP6s (1  $\mu$ L; AAV5-hSynapsin1-FLEX-axon-GCaMP6s, Addgene) was injected into the left basal forebrain (coordinates from bregma: AP -0.5 mm, ML 1.8 mm, DV 4.5 mm). To express a calcium indicator in auditory cortical neurons, an AAV vector encoding jRGECO1a (~0.8–1.5  $\mu$ L; AAV1-syn-jRGECO1a, Addgene) was injected into layer 2/3 of the left auditory cortex. A 3 mm circular glass window (Warner Instruments) was implanted over the exposed cortical surface to permit optical access (centered 1.75 mm anterior to lambda on the ridge line). Two-color, two-photon imaging of cholinergic axons and auditory cortical neurons was performed using a resonant-scanning two-photon microscope (Neurolabware) equipped with a X16 objective (Nikon), angled at 45–55°, and a Spectra-Physics Insight X3 laser (980 nm excitation,  $\leq 40$  mW laser power). A tunable lens enabled near-simultaneous imaging of two planes—layer 1 (60–100  $\mu$ m below dura) and layer 2/3 (150–200  $\mu$ m below dura)—with a field of view of  $312 \times 192 \mu\text{m}^2$  per plane at 15.63 Hz. Animals underwent longitudinal two-photon imaging daily during learning of an auditory discrimination task (see [42,43]). The data analyzed in this manuscript correspond to the layer 1 plane from the first day of learning in one animal (male, 2 months old at imaging start), comprising 40,807 frames from the green channel (cholinergic axons only).

## B Implementation

### B.1 Method hyper-parameters

GraFT hyper-parameters used for our somatic two-photon imaging, vasculature, and axonal data are shown in Table 1.

### B.2 Computing environment

The solver optimization analyses were done on a Windows PC tower with 13th Gen Intel(R) Core(TM) i7-13700K 3.4 GHz processor, 64 GB of DDR4 RAM and a GeForce3080 GPU. We wrote custom MATLAB scripts to iteratively test and time each GraFT.m and patchGraFT.m scripts with the different solvers.

### B.3 Solver comparison in optimization

To improve computational performance, we implemented two separate quadratic programming solvers tailored to the structure of our LASSO subproblem: MATLAB's Optimization Toolbox quadprog solver, and the MPC Toolbox's

**Table 1. Parameters used for experimental results between the three different sample datasets analyzed. Highlighted in gray are parameters that changed across datasets.**

GraFT Parameters			
Parameter	NeuroFinder (Default)	Vasculature	Axonal
$\lambda$	0.7	0.7	2
$\lambda_{\text{Forb}}$	0.2	0	0.2
$\lambda_{\text{Cont}}$	0.1	1	0.1
$\lambda_{\text{Corr}}$	0.1	0.5	0.1
$\beta$	0.09	0.09	0.09
learn eps	5e-4	0.02	5e-4
nonneg	true	true	true
nneg dict	true	true	true
grad type	full_ls_cor	full_ls_cor	full_ls_cor
likely form	gaussian	gaussian	gaussian
maxiter	0.01	0.01	0.01
step s	1	1	1
step decay	0.995	0.995	0.995
max learn	50	50	50
GD iters	1	1	1
updateEmbed	false	false	false
normalizeSpatial	true	true	true
reduce dim	true	true	true
corrType	embedding	embedding	embedding

<https://doi.org/10.1371/journal.pcbi.1014038.t001>

`mpcActiveSetSolver`. The former is a general-purpose convex QP solver that supports various algorithms, including interior-point and active-set methods, and is widely used across a range of convex problems. In contrast, `mpcActiveSetSolver` leverages the KWIK algorithm, a reduced-Hessian active-set method originally developed for real-time control applications. The KWIK method, based on the dual Goldfarb-Ilnani algorithm and optimized for reduced Hessian SQP [35], offers computational advantages in problems with few degrees of freedom but many constraints—conditions often present in our L1-constrained formulation. While `quadprog` ensures generality and numerical stability, the specialized structure and warm-start capabilities of `mpcActiveSetSolver` offer a substantial speedup, particularly in iterative regimes where active sets change slowly. We exploit both solvers depending on problem size and conditioning, with `mpcActiveSetSolver` providing the greatest gains in large-scale regimes with structured sparsity.

## Supporting information

**S1 Fig. Additional influences of GraFT compression level on identified time trace components.** Example medium correlated time traces ( $0.4 < \rho < 0.7$ ) across compression levels in (A) non-normalized GraFT and (B) normalized GraFT. (C) Heatmap for normalized GraFT results across compression levels. Values in the heatmap are the actual number of identified traces.  
(TIF)

## Acknowledgments

The authors would like to thank Jacob Duckworth for his assistance with the vasculature dataset.

## Author contributions

**Conceptualization:** Alejandro Estrada Berlanga, Gabrielle Y. Kang, Gal Mishne, Adam S. Charles.

**Data curation:** Alejandro Estrada Berlanga, Gabrielle Y. Kang, Jennifer Lawlor, Amanda Kwok.

**Formal analysis:** Alejandro Estrada Berlanga, Gabrielle Y. Kang.

**Funding acquisition:** David Kleninfeld, Kishore V. Kuchibhotla, Gal Mishne, Adam S. Charles.

**Investigation:** Alejandro Estrada Berlanga, Gabrielle Y. Kang, Amanda Kwok.

**Methodology:** Alejandro Estrada Berlanga, Gabrielle Y. Kang, Gal Mishne, Adam S. Charles.

**Project administration:** Gal Mishne, Adam S. Charles.

**Resources:** Thomas Broggini, Jennifer Lawlor, Kishore V. Kuchibhotla, David Kleninfeld.

**Software:** Alejandro Estrada Berlanga, Gabrielle Y. Kang.

**Supervision:** Gal Mishne, Adam S. Charles.

**Validation:** Alejandro Estrada Berlanga, Amanda Kwok.

**Visualization:** Alejandro Estrada Berlanga, Gabrielle Y. Kang, Amanda Kwok.

**Writing – original draft:** Alejandro Estrada Berlanga, Gabrielle Y. Kang.

**Writing – review & editing:** Estrada Berlanga, Gabrielle Y. Kang, Amanda Kwok, Thomas Broggini, Jennifer Lawlor, Kishore V. Kuchibhotla, David Kleninfeld, Gal Mishne, Adam S. Charles.

## References

1. Chaudhary S, Moon S, Lu H. Fast, efficient, and accurate neuro-imaging denoising via supervised deep learning. *Nature Communications*. 2022;13(1):5165. <https://doi.org/10.1038/s41467-022-32886-w>
2. Nguyen JP, Shipley FB, Linder AN, Plummer GS, Liu M, Setru SU, et al. Whole-brain calcium imaging with cellular resolution in freely behaving *Caenorhabditis elegans*. *Proc Natl Acad Sci U S A*. 2016;113(8):E1074–81. <https://doi.org/10.1073/pnas.1507110112> PMID: 26712014
3. Frostig RD, Frostig RD. *In vivo* optical imaging of brain function. 2 ed. Taylor and Francis. 2010.
4. Baker PF, Hodgkin AL, Ridgway EB. Depolarization and calcium entry in squid giant axons. *J Physiol*. 1971;218(3):709–55. <https://doi.org/10.1113/jphysiol.1971.sp009641> PMID: 5133953
5. Grinvald A, Omer DB, Sharon D, Vanzetta I, Hildesheim R. Voltage-sensitive dye imaging of neocortical activity. *Cold Spring Harbor Protocols*. 2016;2016(1):pdb.top089367. <https://doi.org/10.1101/pdb.top089367>
6. Kerr JND, Greenberg D, Helmchen F. Imaging input and output of neocortical networks in vivo. *Proc Natl Acad Sci U S A*. 2005;102(39):14063–8. <https://doi.org/10.1073/pnas.0506029102> PMID: 16157876
7. Chen T-W, Wardill TJ, Sun Y, Pulver SR, Renninger SL, Baohan A, et al. Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*. 2013;499(7458):295–300. <https://doi.org/10.1038/nature12354> PMID: 23868258
8. Benisty H, Song A, Mishne G, Charles AS. Review of data processing of functional optical microscopy for neuroscience. *Neurophotonics*. 2022;9(4):041402. <https://doi.org/10.1117/1.NPh.9.4.041402> PMID: 35937186
9. Mishne G, Charles AS. Deep and shallow data science for multi-scale optical neuroscience. In: *Neural Imaging and Sensing 2024*, 2024. 502. <https://doi.org/10.1117/12.3026093>
10. Maruyama R, Maeda K, Moroda H, Kato I, Inoue M, Miyakawa H, et al. Detecting cells using non-negative matrix factorization on calcium imaging data. *Neural Netw*. 2014;55:11–9. <https://doi.org/10.1016/j.neunet.2014.03.007> PMID: 24705544
11. Pnevmatikakis EA, Soudry D, Gao Y, Machado TA, Merel J, Pfau D, et al. Simultaneous Denoising, Deconvolution, and Demixing of Calcium Imaging Data. *Neuron*. 2016;89(2):285–99. <https://doi.org/10.1016/j.neuron.2015.11.037> PMID: 26774160
12. Pachitariu M, Packer AM, Pettit N, Dalgleish H, Hausser M, Sahani M. Suite2p: beyond 10,000 neurons with standard two-photon microscopy. *bioRxiv*. 2016.
13. Haeffele BD, Vidal R. Structured Low-Rank Matrix Factorization: Global Optimality, Algorithms, and Applications. *IEEE Trans Pattern Anal Mach Intell*. 2020;42(6):1468–82. <https://doi.org/10.1109/TPAMI.2019.2900306> PMID: 30794507
14. Mishne G, Coifman RR, Lavzin M, Schiller J. Automated cellular structure extraction in biological images with applications to calcium imaging data. *bioRxiv*. 2018. 313981.

15. Inan H, Erdogdu MA, Schnitzer M. In: 2017. 2905–14.
16. Mishne G, Charles AS. Learning Spatially-correlated Temporal Dictionaries for Calcium Imaging. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019. 1065–9. <https://doi.org/10.1109/icassp.2019.8683375>
17. Giovannucci A, Friedrich J, Gunn P, Kalfon J, Brown BL, Koay SA, et al. CalmAn an open source tool for scalable calcium imaging data analysis. *Elife*. 2019;8:e38173. <https://doi.org/10.7554/eLife.38173> PMID: [30652683](https://pubmed.ncbi.nlm.nih.gov/30652683/)
18. Charles AS, Cermak N, Affan RO, Scott BB, Schiller J, Mishne G. GraFT: Graph Filtered Temporal Dictionary Learning for Functional Neural Imaging. *IEEE Trans Image Process*. 2022;31:3509–24. <https://doi.org/10.1109/TIP.2022.3171414> PMID: [35533160](https://pubmed.ncbi.nlm.nih.gov/35533160/)
19. Saxena S, Kinsella I, Musall S, Kim SH, Meszaros J, Thibodeaux DN, et al. Localized semi-nonnegative matrix factorization (LocaNMF) of wide-field calcium imaging data. *PLoS Comput Biol*. 2020;16(4):e1007791. <https://doi.org/10.1371/journal.pcbi.1007791> PMID: [32282806](https://pubmed.ncbi.nlm.nih.gov/32282806/)
20. Li M, Gui S, Huang Q, Shi L, Lu J, Li P. Density center-based fast clustering of widefield fluorescence imaging of cortical mesoscale functional connectivity and relation to structural connectivity. *Neurophotonics*. 2019;6(4):045014. <https://doi.org/10.1117/1.NPh.6.4.045014> PMID: [31853460](https://pubmed.ncbi.nlm.nih.gov/31853460/)
21. El Hady A, Takahashi D, Sun R, Akinwale O, Boyd-Meredith T, Zhang Y, et al. Chronic brain functional ultrasound imaging in freely moving rodents performing cognitive tasks. *J Neurosci Methods*. 2024;403:110033. <https://doi.org/10.1016/j.jneumeth.2023.110033> PMID: [38056633](https://pubmed.ncbi.nlm.nih.gov/38056633/)
22. Weng W, Zhu X. INet: Convolutional Networks for Biomedical Image Segmentation. *IEEE Access*. 2021;9:16591–603. <https://doi.org/10.1109/access.2021.3053408>
23. Petersen A, Simon N, Witten D. Scalpel: extracting neurons from calcium imaging data. *Ann Appl Stat*. 2018;12(4):2430–56. <https://doi.org/10.1214/18-AOAS1159> PMID: [30510612](https://pubmed.ncbi.nlm.nih.gov/30510612/)
24. Klibisz A, Rose D, Eicholtz M, Blundon J, Zakharenko S. Fast, Simple Calcium Imaging Segmentation with Fully Convolutional Networks. *Lecture Notes in Computer Science*. Springer International Publishing. 2017. p. 285–93. [https://doi.org/10.1007/978-3-319-67558-9\\_33](https://doi.org/10.1007/978-3-319-67558-9_33)
25. Wang F, Li P. Efficient Nonnegative Matrix Factorization with Random Projections. In: *Proceedings of the 2010 SIAM International Conference on Data Mining*, 2010. 281–92. <https://doi.org/10.1137/1.9781611972801.25>
26. Damle A, Sun Y. Random projections for non-negative matrix factorization. 2014. <https://arxiv.org/abs/1405.4275>
27. Ding C, Li T, Jordan MI. Convex and semi-nonnegative matrix factorizations. *IEEE Trans Pattern Anal Mach Intell*. 2010;32(1):45–55. <https://doi.org/10.1109/TPAMI.2008.277> PMID: [19926898](https://pubmed.ncbi.nlm.nih.gov/19926898/)
28. Zelnik-Manor L, Perona P. Self-tuning spectral clustering. In: *Advances in neural information processing systems*, 2004.
29. Dantzig GB. *Linear programming and extensions*. Princeton, NJ: Princeton University Press. 1998.
30. Nocedal J, Wright S. *Numerical Optimization*. 2nd ed. Springer Series in Operations Research and Financial Engineering. New York, NY: Springer; 2006.
31. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge University Press. 2004.
32. Lau MSK, Yue SP, Ling KV, Maciejowski JM. A comparison of interior point and active set methods for FPGA implementation of model predictive control. In: *2009 European Control Conference (ECC)*, 2009. 156–61. <https://doi.org/10.23919/ecc.2009.7074396>
33. Kim S-J, Koh K, Lustig M, Boyd S, Gorinevsky D. An Interior-Point Method for Large-Scale  $\ell_1$ -Regularized Least Squares. *IEEE J Sel Top Signal Process*. 2007;1(4):606–17. <https://doi.org/10.1109/jstsp.2007.910971>
34. Inc TM. MATLAB version: 9.13.0 (R2022b); 2022. Available from: <https://www.mathworks.com>
35. Schmid C, Biegler LT. Quadratic programming methods for reduced hessian SQP. *Computers & Chemical Engineering*. 1994;18(9):817–32. [https://doi.org/10.1016/0098-1354\(94\)e0001-4](https://doi.org/10.1016/0098-1354(94)e0001-4)
36. Berens P, Theis L, Stone JS, Sofroniew N, Tolias A, Bethge M, et al. Standardizing and benchmarking data analysis for calcium imaging. In: *Computational and Systems Neuroscience Meeting (COSYNE 2017)*. p.66–7.
37. Pnevmatikakis EA, Giovannucci A. NoRMCorre: An online algorithm for piecewise rigid motion correction of calcium imaging data. *J Neurosci Methods*. 2017;291:83–94. <https://doi.org/10.1016/j.jneumeth.2017.07.031> PMID: [28782629](https://pubmed.ncbi.nlm.nih.gov/28782629/)
38. Brogini T, Duckworth J, Ji X, Liu R, Xia X, Mächler P, et al. Long-wavelength traveling waves of vasomotion modulate the perfusion of cortex. *Neuron*. 2024;112(14):2349–2367.e8. <https://doi.org/10.1016/j.neuron.2024.04.034> PMID: [38781972](https://pubmed.ncbi.nlm.nih.gov/38781972/)
39. Zhu F, Elnozahy S, Lawlor J, Kuchibhotla KV. The cholinergic basal forebrain provides a parallel channel for state-dependent sensory signaling to auditory cortex. *Nat Neurosci*. 2023;26(5):810–9. <https://doi.org/10.1038/s41593-023-01289-5> PMID: [36973512](https://pubmed.ncbi.nlm.nih.gov/36973512/)
40. Rüböl O, Tritt A, Dichter B, Braun T, Cain N, Clack N. NWB:N 2.0: An Accessible Data Standard for Neurophysiology. *bioRxiv*. 2019. <https://doi.org/10.1101/523035>
41. Ji X, Ferreira T, Friedman B, Liu R, Liechty H, Bas E, et al. Brain microvasculature has a common topology with local differences in geometry that match metabolic load. *Neuron*. 2021;109(7):1168–1187.e13. <https://doi.org/10.1016/j.neuron.2021.02.006> PMID: [33657412](https://pubmed.ncbi.nlm.nih.gov/33657412/)
42. Kuchibhotla KV, Hindmarsh Sten T, Papadoyannis ES, Elnozahy S, Fogelson KA, Kumar R, et al. Dissociating task acquisition from expression during learning reveals latent knowledge. *Nat Commun*. 2019;10(1):2151. <https://doi.org/10.1038/s41467-019-10089-0> PMID: [31089133](https://pubmed.ncbi.nlm.nih.gov/31089133/)
43. Drieu C, Zhu Z, Wang Z, Fuller K, Wang A, Elnozahy S, et al. Rapid emergence of latent knowledge in the sensory cortex drives learning. *Nature*. 2025;641(8064):960–70. <https://doi.org/10.1038/s41586-025-08730-8> PMID: [40108473](https://pubmed.ncbi.nlm.nih.gov/40108473/)