

Bioimage informatics

Population-scale three-dimensional reconstruction and quantitative profiling of microglia arbors

Murad Megjhani¹, Nicolas Rey-Villamizar¹, Amine Merouane¹, Yanbin Lu¹, Amit Mukherjee³, Kristen Trett², Peter Chong², Carolyn Harris², William Shain² and Badrinath Roysam^{1,*}

¹Department of Electrical and Computer Engineering, University of Houston, Houston, TX, USA, ²Center for Integrative Brain Research, Seattle Children's Hospital, Seattle, WA 98101, USA and ³Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11724, USA

*To whom correspondence should be addressed.

Associate Editor: Robert F. Murphy

Received on September 26, 2014; revised on February 12, 2015; accepted on February 16, 2015

Abstract

Motivation: The arbor morphologies of brain microglia are important indicators of cell activation. This article fills the need for accurate, robust, adaptive and scalable methods for reconstructing 3-D microglial arbors and quantitatively mapping microglia activation states over extended brain tissue regions.

Results: Thick rat brain sections (100–300 μm) were multiplex immunolabeled for IBA1 and Hoechst, and imaged by step-and-image confocal microscopy with automated 3-D image mosaicing, producing seamless images of extended brain regions (e.g. $5903 \times 9874 \times 229$ voxels). An over-complete dictionary-based model was learned for the image-specific local structure of microglial processes. The microglial arbors were reconstructed seamlessly using an automated and scalable algorithm that exploits microglia-specific constraints. This method detected 80.1 and 92.8% more centered arbor points, and 53.5 and 55.5% fewer spurious points than existing vesselness and LoG-based methods, respectively, and the traces were 13.1 and 15.5% more accurate based on the DIADEM metric. The arbor morphologies were quantified using Scorcioni's L-measure. Coifman's harmonic co-clustering revealed four morphologically distinct classes that concord with known microglia activation patterns. This enabled us to map spatial distributions of microglial activation and cell abundances.

Availability and implementation: Experimental protocols, sample datasets, scalable open-source multi-threaded software implementation (C++, MATLAB) in the electronic supplement, and website (www.farsight-toolkit.org). http://www.farsight-toolkit.org/wiki/Population-scale_Three-dimensional_Reconstruction_and_Quantitative_Profiling_of_Microglia_Arbors

Contact: broysam@central.uh.edu

Supplementary information: [Supplementary data](#) are available at *Bioinformatics* online.

1 Introduction

Microglia are immune cells of the mammalian central nervous system whose importance to brain function is receiving growing recognition (Fields, 2013; Streit, 2005). Normally, these cells are distributed throughout the brain in non-overlapping territories, and comprise up to 20% of the glial cell population (Fields, 2013; Gehrmann *et al.*, 1995). Their arbor morphologies are dynamic, informative of their internal state of activation and are related to tissue perturbations. Resting microglia exhibit motile and highly branched arbors that constantly screen for perturbations in their local environment. There is considerable heterogeneity in microglial populations, even in normal brain tissue. When perturbed by pathological stimuli, they undergo rapid changes in arbor morphology and migrate toward the lesion site (Ohsawa and Kohsaka, 2011). The arbor morphologies are indicators of cell activation, and are valuable regardless of any molecular markers of activation. For these reasons, there is a need for methods to analyze individual microglial arbors, profile arbor alterations and map spatial distributions over extended tissue regions. Capturing these phenomena given the small dimensions of microglial processes ($<2\ \mu\text{m}$) requires high-resolution microscopy. Accurately capturing the asymmetrical arbors of non-resting microglia, requires three-dimensional (3-D) imaging of tissue samples that are thick enough to include entire cells. Capturing perturbations to spatial distributions of these cells requires high-extent imaging, especially when the lesions are much larger than the field of view of the microscope (Rey-Villamizar *et al.*, 2014). Figure 1 illustrates the use of step-and-image confocal microscopy and computational image mosaicing to record large 3-D images of brain tissue. Our goal is to describe automated methods for quantifying microglial arbor morphologies and spatial distributions of activation in such images.

In considering arbor tracing methods, we note that there is a vast literature on tracing algorithms for neurons (Meijering, 2010). In contrast, the literature on microglial arbor analysis is much smaller (Galbreath, 2011; Rouchdy and Cohen, 2013; Wang *et al.*, 2011; Xiao and Peng, 2013). Importantly, the prior efforts have tackled

much smaller cell populations than reported here, cannot reconstruct microglia seamlessly and three-dimensionally across significant extents of brain tissue (multiple mm–cm), and do not guarantee that the reconstructions have a tree topology. Finally, the prior literature does not address population-scale quantitative analysis of microglial arbors as a function of activation. The prior literature on automated neuron tracing provides a valuable basis for reconstructing microglia, but there are important limitations that are specific to microglia that remain unaddressed. Most of the methods are based on pre-defined analytical models of the tubularity ('vesselness') and continuity of neurites. For example, Narayanaswamy *et al.* (2011) used the multi-scale curvelet transform to model neurites. Automated neurite tracing methods are generally based on model-based sequential tracing (Al-Kofahi *et al.*, 2008; Bas and Erdogmus, 2010; Peng *et al.*, 2010); probabilistic extraction of centerlines (Breitenreicher *et al.*, 2013; Gonzalez *et al.*, 2010; Türetken *et al.*, 2014, 2011) or on segmentation (Chothani *et al.*, 2011; Vasilkoski and Stepanyants, 2009). The neurite centerlines are usually detected by variations of skeletonization (He *et al.*, 2003; Peng *et al.*, 2014; Yuan *et al.*, 2009), or voxel coding (Chothani *et al.*, 2011; Jiménez *et al.*, 2014). For handling complex/dense neurite fields, various approaches have been proposed for negotiating branches and cross-overs (Al-Kofahi *et al.* 2008; Schmitt *et al.*, 2004; Wearne *et al.* 2005). The tracing performance depends on the quality of seed points, effective modeling of the peculiarities of the images being processed, and tracing control criteria (stopping, branching, negotiating cross-overs, etc.).

Our method is designed to address the needs of large-scale microglia reconstruction by exploiting microglia-specific constraints (e.g. known topology), and using algorithms that are specifically designed for mosaiced high-extent imaging of brain tissue (Tsai *et al.*, 2011). Specifically, we address the need to cope with biological and imaging variability within and across samples, without having to retune algorithm parameters. For this, we propose a machine learning approach for modeling the characteristics of microglia in actual images, rather than resorting to a pre-defined analytical model as in the prior work on neuron tracing. To overcome the difficulty of tuning parameters over large-extent images, we propose algorithms with very few parameters that are easy to interpret and adjust. Our method detects the microglial processes reliably using a machine-learning algorithm, and traces the arbors subject to a topological constraint. Appropriately designed cost terms guide our algorithm to exploit the fact that microglia, unlike neurons, are spatially localized (lacking long processes, e.g. axons). Whereas tight fiber cross-overs are common for neurite fields, they are rare for microglia. Furthermore, microglia do not form inter-connections with neighboring cells. The more pressing challenge in our work to correctly reconstruct microglia that are split across the microscope fields to enable accurate spatial distribution analysis. Our method uses these same constraints to handle such cells. To achieve scalability, our method is designed to exploit parallel computers. This results in an array of arbor reconstructions, one per IBA1+ microglial cell that can be visualized and proofread using 3-D visualization tools that are capable of handling thousands of cells at a time (Luisi *et al.*, 2011). The overall pipeline is very usable and scalable. The few adjustable parameters of our method affect the results smoothly, permitting stable and objective analysis of microglia morphologies. The resulting reconstructions are directly amenable to quantitative analysis using Scorcion's L-measure (Scorcion *et al.*, 2008). The quantitative arbor measurements are analyzed by Coifman's harmonic co-clustering to reveal morphologically distinct arbor classes that concord with the known microglia activation patterns. Application

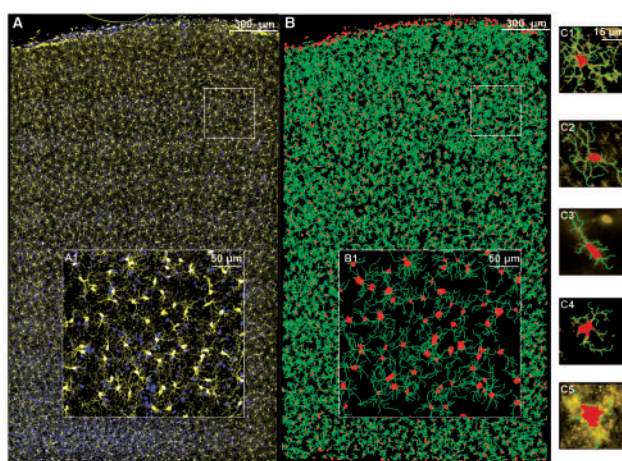


Fig. 1. Illustrating the problem of large-scale microglia reconstruction. (A) Maximum-intensity projection of a 3-D two-channel mosaic image (size $5903 \times 9874 \times 229$ voxels, 3117 microglia, yellow: IBA1, blue: Hoechst). (A1) Close-up of the boxed region. (B) Automated reconstructions with microglial soma in red and arbor reconstructions in green. (B1) Close-up of the reconstructions in the boxed region. (C1–C5) Close-ups of individual microglial arbor reconstructions illustrating the diversity of cellular morphologies

of these methods to brain tissue analysis will provide unprecedented measurements of microglia as a function of physiological and pathological stimuli.

2 Methods

Quantitative tissue-scale microglia profiling requires three-dimensional (3-D) imaging of large numbers of cells spread across significant lateral and axial extents of brain tissue (multi-millimeter lateral extent, and 100–300 μm depth). For the experiments reported here, coronal sections of 4% paraformaldehyde fixed rat brain tissue from the motor cortex region were cut into 100 μm thick sections, and fluorescently labeled for microglia-specific protein IBA1 to highlight microglia, and DNA stain Hoechst to highlight all nuclei). A Rolera EM-C² camera (QImaging, Surrey, Canada; 1004 \times 1002 pixels) mounted on an Olympus DSU spinning-disk confocal microscope with a $\times 30$ objective (N.A. 1.05, Silicone oil) was used for imaging individual 3-D image fields of interest (resolution of 0.267 $\mu\text{m}/\text{pixel}$, 14 bits/pixel (x, y), z -step size of 0.3 μm), and also to record a series of overlapping 3-D images for mapping tissue regions that are larger than the field of the microscope. For example, the dataset in Figure 1 is a mosaic constructed from 12 rows and 7 columns of overlapping image tiles, each covering a 267 μm square region. The tiles are collected with a 15–20% overlap, spanning a 2.615 mm \times 1.550 mm \times 0.1 mm region of tissue. The image tiles were combined into a 3-D montage of the whole field using a previously published method, and displayed in Figure 1A (Tsai et al., 2011). In addition to these large montages, randomly selected fields from different animals were also analyzed.

2.1 Sparsity-based modeling of microglia processes

Given 3-D images $I(x, y, z)$, our method learns the structure of microglial processes from actual images to construct a computational ‘dictionary-based’ model for microglial processes, and then utilizes this model to reconstruct the arbors. The term ‘dictionary’ refers to a set of basis vectors that can be combined linearly to represent the given image. These basis vectors are learned from image patches that are labeled as being foreground (microglia processes), or background. Several authors have shown that learning a dictionary from images rather than using a predetermined analytical model can lead to more effective image representations, and can therefore provide improved results in many practical image-processing applications such as restoration, segmentation and classification (Rubinstein et al., 2010). Our modeling approach is based on the sparse over-complete dictionary learning method described by (Zhang and Li, 2010) that was inspired by the K-singular value decomposition (K-SVD) algorithm (Aharon et al., 2006). Specifically, Zhang and Li (2010) described an extension of the K-SVD method, known as the *label-consistent* K-singular value decomposition method (LC-KSVD). In this method, small image patches that are labeled to indicate classes (e.g. foreground/background) are used as training examples. The typical patches in our work are quite small, typically, 15 \times 15 \times 3 voxels for the current image stacks. These labeled image patches are extracted from representative training images. The LC-KSVD algorithm has the important advantage of simultaneously and consistently learning a single discriminative dictionary (that is typically more compact compared to the K-SVD method), and a linear classifier. In our work, the dictionary typically consists of 675 – 2000 dimensional vectors (atoms).

A mathematical description of our approach is presented next. Let $X = \{x_1, \dots, x_N\} \in \mathcal{R}^{n \times N}$ denote a set of image patches drawn

from the 3-D image $I(x, y, z)$, where n denotes the number of voxels in each patch, and N denotes the total number of image patches. We wish to learn an over-complete dictionary, denoted $D = \{d_1, \dots, d_K\} \in \mathcal{R}^{n \times K}$ ($K > n$ making the dictionary over-complete) containing K basis elements, usually referred to as ‘atoms’ for building a sparse representation of X . The patch data X is approximated by $D\Gamma$, where $\Gamma = \{\gamma_1, \dots, \gamma_N\} \in \mathcal{R}^{K \times N}$ is a matrix that is chosen to respect a sparsity constraint. Specifically, we are interested in representations that minimize the number of non-zero entries in γ_i for representing signal x_i . In practice, the number of non-zero entries is a user-selected parameter that is known as the *sparsity constraint*, and denoted T . This requires solving the following optimization:

$$\langle D, \Gamma \rangle = \arg \min_{D, \Gamma} \|X - D\Gamma\|_F^2 \text{ s.t. } \forall i, \|\gamma_i\|_0 \leq T, \quad (1)$$

The term $\|X - D\Gamma\|_F^2$ denotes the squared signal reconstruction error where ‘F’ is the Frobenius norm. The reconstruction error is minimized subject to the L_0 sparsity constraint $\|\gamma_i\|_0 \leq T$. The K-SVD algorithm is an iterative approach for implementing this optimization in equation that operates in two steps. First, Equation (1) is minimized with respect to Γ to learn a dictionary D . Given D , the sparse coding algorithm computes the sparse representation Γ of signal X by solving:

$$\langle \Gamma \rangle = \arg \min_{\Gamma} \|X - D\Gamma\|_F^2 \text{ s.t. } \forall i, \|\gamma_i\|_0 \leq T. \quad (2)$$

An exact determination of sparse representations is proven to be an NP-hard problem (Natarajan, 1995). Several algorithms have been proposed in the literature to solve the above sparse coding problem, usually by relaxing the L_0 penalty to use the convex L_1 penalty instead, as in the basis pursuit method (Chen et al., 1998). The Focal Underdetermined System Solver (FOCUSS; Gorodnitsky and Rao, 1997) is very similar in using the L_p norm, with $P < 1$ as a replacement for the L_0 norm. Another approach is to use greedy algorithms like Matching Pursuit (Mallat, 1993), or Orthogonal Matching Pursuit (OMP; Pati et al., 1993), that select the dictionary atoms sequentially. In this work, we chose OMP to solve the sparse coding problem with computational efficiency in mind.

The next step is to classify the voxels as seed points using these sparse features. The sparse codes from Equation (1) can be directly used for classification. An effective classifier $f(\gamma)$ can be obtained by determining the model parameters $W \in \mathcal{R}^{m \times K}$, satisfying

$$\langle W \rangle = \arg \min_W \sum_i \mathcal{L}\{b_i, f(\gamma_i, W)\} + \lambda \|W\|_F^2,$$

where \mathcal{L} is the classification loss function, b_i are the labels (seed point / not a seed point) and λ is a regularization term that is incorporated to prevent overfitting. Widely used loss functions include the logistic function, hinge function and the quadratic. We used the linear predictive classifier in our study so $f(\gamma, W) = W\gamma$. However, separating the dictionary learning from the classifier learning will make D suboptimal for classification (Jiang et al., 2011). It is possible to jointly learn dictionary and classification models (Mairal et al., 2009; Yang et al., 2010) by implementing the following optimization:

$$\langle D, W, \Gamma \rangle = \arg \min_{D, W, \Gamma} \|X - D\Gamma\|_F^2 + \beta \|H - W\Gamma\|_F^2 \\ \text{s.t. } \forall i, \|\gamma_i\|_0 \leq T,$$

where $H = \{b_1, b_2, \dots, b_N\} \in \mathcal{R}^{m \times N}$ denotes the class labels for N examples and m classes. These approaches require relatively large dictionaries to achieve good classification (Jiang et al., 2011),

making it difficult to set the dictionary size (K). By introducing the label consistency term, a concise unified dictionary can be learned using the LC-KSVD algorithm. The resulting dictionary enables reliable classification using only a small, unified dictionary and a single multi-class linear classifier. The central idea behind this approach is to incorporate two additional terms to the objective function for learning the dictionary, specifically a label consistency term and a classification error term. The dictionary learned in this manner will adapt to the underlying structure of the training data, and generate discriminative sparse codes regardless of the size of the dictionary, as described by [Jiang et al. \(2011\)](#). Let $Q = \{q_1, q_2, \dots, q_N\} \in \mathcal{R}^{K \times N}$ denote such a discriminative set of sparse codes. Let A denote a linear transformation matrix that transforms the original sparse codes to the most discriminative sparse codes in the sparse feature space \mathcal{R}^K . Let $H = \{h_1, h_2, \dots, h_N\} \in \mathcal{R}^{m \times N}$ denote the class labels for the N examples for m classes. With this notation, the LC-KSVD algorithm can be expressed as the following expanded optimization problem:

$$\begin{aligned} \langle D, W, A, \Gamma \rangle = \arg \min_{D, W, A, \Gamma} & \|X - D\Gamma\|_F^2 + \alpha \|Q - A\Gamma\|_F^2 \\ & + \beta \|H - W\Gamma\|_F^2 \text{ s.t. } \forall i, \|\gamma_i\|_0 \leq T, \end{aligned} \quad (3)$$

where the first term $\|X - D\Gamma\|_F^2$ represents the squared reconstruction error. The second term $\|Q - A\Gamma\|_F^2$ represents the discriminative sparse-coding error. It is intended to penalize sparse codes Γ that deviate from the discriminative sparse codes Q . Intuitively, it forces the signals from the same class to have similar representations. For example, if q_i is the discriminative sparse code corresponding to the input signal x_i , then the non-zero values of q_i occur at those indices where the input signal x_i and the dictionary element d_k share the same label. The third term $\|H - W\Gamma\|_F^2$ represents the classification error, and $H = \{h_1, h_2, \dots, h_N\} \in \mathcal{R}^{m \times N}$ is the matrix of class labels for N samples and m classes. As a concrete example, for a two-class problem (Classes 1 and 2). Suppose the size of the dictionary K is 4, i.e., $D = \{d_1, \dots, d_4\}$, and the number of examples N is 4, i.e., $X = \{x_1, \dots, x_4\}$. To create a discriminative dictionary we want the examples of Class 1, (x_1, x_2) to be associated with the first two dictionary atoms $\{d_1, d_2\}$, and the remaining examples of Class 2, (x_3, x_4) to be associated with $\{d_3, d_4\}$. The Q matrix can then be written as follows:

$$Q = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

This matrix is set automatically in our implementation. For the above Q matrix, $H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$. The dictionary learned in this manner has excellent representational power, and enforces strong discrimination between the two classes (e.g. seed points versus non-seed points), as detailed later. Importantly, the dictionary is selected to provide a *sparse* representation of the image. We are interested in constructing a dictionary such that a small number of its atoms can be linearly combined to represent the image. Following ([Zhang and Li, 2010](#)) the above optimization can be rewritten as follows:

$$\langle D, W, A, \Gamma \rangle = \arg \min_{D, W, A, \Gamma} \left\| \begin{pmatrix} X \\ \sqrt{\alpha}Q \end{pmatrix} - \begin{pmatrix} D \\ \sqrt{\alpha}A \end{pmatrix} \Gamma \right\|_F^2 \text{ s.t. } \forall i, \|\gamma_i\|_0 \leq T$$

This can be recast concisely as follows:

$$\langle D', \Gamma \rangle = \arg \min_{D', \Gamma} \|X' - D'\Gamma\|_F^2 \text{ s.t. } \forall i, \|\gamma_i\|_0 \leq T$$

where

$$X' = \begin{pmatrix} X \\ \sqrt{\alpha}Q \end{pmatrix}, \text{ and } D' = \begin{pmatrix} D \\ \sqrt{\alpha}A \end{pmatrix} \quad (4)$$

The above optimization is conducted using the Orthogonal Matching Pursuit (OMP) algorithm. Given the nature of the L_0 norm, the non-zero values of q_i in the solution occur at those indices where the input signal x_i and the dictionary element d_k share the same label. We then solve [Equation \(4\)](#) using the KSVD method. The matrix D' is L_2 -normalized column-wise. We then compute the matrices $D = \{d_1, \dots, d_k\}$, $A = \{a_1, \dots, a_k\}$, and $W = \{w_1, \dots, w_k\}$ from D' . However, we cannot simply use D , A , and W for classifying new image patches since they are jointly normalized in D' , i.e. $\|d_k\|_2, \sqrt{\alpha}a_k, \sqrt{\beta}w_k = 1$. The normalized dictionary \hat{D} , transform parameters \hat{A} , and classifier parameters \hat{W} are computed as follows:

$$\begin{aligned} \hat{D} &= \left\{ \frac{d_1}{\|d_1\|_2}, \dots, \frac{d_k}{\|d_k\|_2} \right\}; \\ \hat{A} &= \left\{ \frac{a_1}{\|d_1\|_2}, \dots, \frac{a_k}{\|d_k\|_2} \right\}; \\ \hat{W} &= \left\{ \frac{w_1}{\|d_1\|_2}, \dots, \frac{w_k}{\|d_k\|_2} \right\}. \end{aligned}$$

The inputs to the seed detection algorithm are the set of training image patches X , and a corresponding label matrix H . The algorithm requires two adjustable parameters α and β . The setting of these parameters is described in the Experimental Results section. They merely serve to adjust the relative importance of the respective penalty terms in [Equation \(3\)](#). The outputs of the algorithm consist of the dictionary D , classifier weights W , transformation matrix A , and the sparse features Γ . The sample dictionary shown in [Figure 2](#), shows two distinct classes despite the variability, corresponding to the arbor processes (IBA1+ foreground), and the background. Once the dictionary is learned from the training images, it can be applied to seed detection in other (non-training) images that are collected using the same imaging protocols. Given the dictionary D , the sparse features Γ can be computed using [Equation \(2\)](#) by any sparse coding algorithm, including OMP. Once the sparse features Γ and the classifier W , are learned, the next step is to classify each pixel in the test image as a seed point or otherwise, based on the sparse image representation. For this, we first extract the image patch x_i , surrounding the voxel in question from the test image $I_t(x, y, z)$, and compute the sparse feature γ_i with respect to the learned dictionary D . Then, we classify the pixel using a linear classifier W as follows,

- i. Compute $W\gamma_i = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$,
- ii. Set $V(x, y, z) = \begin{cases} 1 \text{ or } w_1 \text{ (seed point)} & \text{if } w_1 > w_2 \\ 0 \text{ (background)} & \text{else} \end{cases}$,

where V is the seed point image whose voxel value is set to 1 or w_1 for pixels that are classified as a seed point, and to 0 otherwise. In

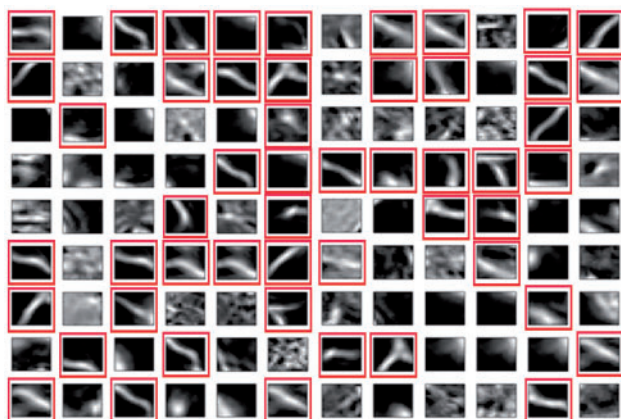


Fig. 2. A subset of the 3-D image patch dictionary entries displayed as maximum-intensity projections. It is clear from this illustration that some patches (highlighted by red boxes) clearly belong to parts of microglial arbors, whereas the other patches are either ambiguous or part of the background

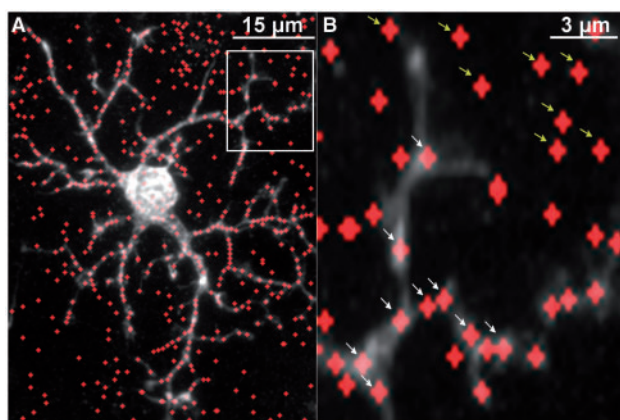


Fig. 3. Illustrating 3-D seed points used for arbor reconstruction. (A) Sample IBA1+ cell (grayscale) overlaid with initially detected seed points (red). (B) Close-up of boxed region in Panel A. The white arrows indicate some valid seed points that lie on the cellular arbor, and the yellow arrows some indicate invalid seed points. Only a subset of the seed points are marked by arrows here to avoid clutter

Section 3, we present experimental data showing that the seed points detected by the above algorithm (Fig. 3) are sufficiently reliable and accurate for reconstructing microglial arbors.

2.2 Automated microglia arbor reconstruction

Microglia consist of branching processes that form a tree, and the processes stem from a central *soma* that envelops the cell nucleus. Detecting the soma is the first step to reconstructing the microglia. For this, we proceed in two steps. First, we perform an automated segmentation of the Hoechst image channel (the nuclear label) to detect all cell nuclei (Al-Kofahi et al., 2010). From the segmentation results, we compute a set of features including the 3-D location, volume, shape factor and chromatin-intensity variance (a texture measure) for each nucleus. Next, we compute the total amount of the microglial marker IBA1 within a distance of eight voxels to each nucleus (Bjornsson et al., 2008). We then isolate the microglial cell nuclei based on these features, using an active learning-based classifier (Padmanabhan et al., 2014). Microglial cell somas that extend

beyond the nuclei are segmented using a level-set algorithm. The centroids of the somas are used as the starting point for the tracing algorithms.

The soma segmentations allow us to eliminate any false seed points that lie within the soma. The remaining seeds are used to reconstruct the cellular arbors. As noted above, individual microglia are known *a priori* to have a tree topology. Furthermore, microglia do not form inter-connections with neighboring cells (Rouchdy et al., 2011). These constraints enable us to develop fast and scalable algorithms that are appropriate for tracing the arbors of these cells. The tree assumption is also helpful from the standpoint of extracting quantitative measurements of arbors, since existing computational neuroanatomy tools, e.g. the L-measure (Scorcioni et al., 2008) expect a tree topology. Based on these considerations, we designate the centroids of the microglial somas as the *root points* for their respective cellular arbors. Starting from all root points, we simultaneously construct a forest of R minimum spanning trees (MST's), where R is the total number of detected root points. Each MST consists of *nodes* and directed *edges*. The nodes for a single cell include the root point, a subset of the seed points, referred to here as the *primary nodes* and denoted $v \in V_0$, an all intermediate pixels that are termed *secondary nodes*. The secondary nodes are used for calculating the costs associated with their neighbors (as explained further below).

Each node is identified by its 3-D image coordinates. The directed edges, denoted $e_{ij} \in E$, link the nodes and their orientations always point towards the root point. Each edge is associated with a weight value that is used for determining whether or not a given pixel belongs in the arbor reconstruction. The process of growing each MST is based on an adaptation of Prim's algorithm (Mariano et al., 2013; Teacher and Griffiths, 2011). Starting from each root node, our algorithm connects each successive node to the nearest primary node (in the sense of the cost metric defined below), to form an edge (link). This process is repeated at each added node. The algorithm iterates as it expands all the trees starting from their respective roots concurrently to cover all the primary nodes V_0 . For each cell, the tree-growing procedure is stopped when a cumulative cost function (described next) exceeds a preset threshold. Once the computation of the MST's is complete, we obtain a forest of trees $T_k = (V_k, E_k)$, where k is the index of a tree, V_k is the list of all nodes belonging to tree k and E_k is the list of all edges belonging to tree k . Each tree captures the morphology of a single microglial cell.

2.3 Dynamic cost metric for linking nodes

The tree-growing process described above is guided by a cost metric that takes into account the normalized weight, denoted w_i , for the classified seed points, and the spatial proximity of points being considered for a connection. Intuitively, this method traces microglial processes by linking nodes forming the shortest geodesic path along the brightest (IBA1+) voxels representing the image foreground. With this in mind, we compute our connection costs using the Fast Marching Method (FMM; Sethian, 1996) that provides an efficient way to identify geodesic paths from each node to the corresponding root node. The FMM provides a fast approximation to the stationary Hamilton-Jacobi equation, commonly referred to as the Eikonal equation. This non-linear partial differential equation is used for path planning problems in non-homogeneous media. Given a domain $\Omega \subset \mathcal{R}^n$, the Eikonal Equation is of the form:

$$\begin{aligned} \|\nabla_{\Omega} C(x, y, z)\| &= F(x, y, z), \\ C(x_0) &= 0, \end{aligned} \quad (5)$$

where $C(x, y, z)$ is the cost function, $F(x, y, z)$ is a positive-valued

function, ∇ denotes the gradient operator, and $\|\cdot\|$ is the Euclidean norm. In our formulation, the function $F(x, y, z)$ is set to the inverse of the score given by the result of seed detection algorithm, instead of the voxel intensity. In our FMM implementation, a vertex v_i from a growing MST T_k can be thought to emanate ‘waves’ that propagate through the image with their speed of propagation proportional to the normalized weight of the classifier. The wave propagation from the vertex v_i is allowed to continue to the next closest vertex v_j , that is not part of any previously detected tree. The new vertex v_j , and the edge e_j at this point are included in V_k and E_k , respectively, resulting in an expansion of the tree T_k . This process is repeated concurrently for all trees (one per microglial cell) until all the primary nodes in V_0 are linked, or the minimum distance between all possible links exceeds a pre-defined cost threshold τ . Nodes that are deemed more expensive than this threshold are not added to a tree. This is a mechanism to prevent unreliable points from being erroneously added to the distal portions of microglia reconstructions. In other words, τ sets the overall sensitivity of the reconstruction method. A higher threshold makes the algorithm more sensitive to image-intensity values, and *vice versa*. Further details about the priority queue-based reconstruction algorithm and pseudo code summaries of our algorithm are provided in the Electronic Supplement.

3 Experimental results

For proofreading the 3-D arbor reconstructions, we used the open source FARSIGHT Trace editor (Luisi *et al.*, 2011). This tool allows the user to view the reconstructions from any angle, and edit the reconstructions if desired. The editing capability was not utilized for the experiments described here, since our end goal was an analysis of microglial activation patterns rather than exact reconstructions of each cell. The core of the dice and trace method is also embedded into this tool for tracing individual cells interactively—the user can trigger the reconstruction of an entire missed arbor by manually specifying the root point. Manually reconstructing thousands of cells, or even editing automatically generated reconstructions at this scale is not feasible and is time consuming. Our expectations of reconstruction accuracy are also appropriately pragmatic—we expect valid reconstructions with sufficient accuracy to analyze microglia activation patterns.

With these considerations in mind, our experiments are primarily designed to study the impact of parameter selection on reconstruction performance. For this, we randomly selected a population of 50 cells across multiple images from multiple animals. These cells were reconstructed manually by a pool of volunteers. Each cell was reconstructed by at least two different users using the Neuromantic tool (Myatt *et al.*, 2012), and saved into the standard SWC file format. This is a labor-intensive activity that takes between 40 and 60 min per cell. The next step was to compute a metric to compare the automatic reconstructions against the manual reconstructions. The DIADEM (Gillette *et al.*, 2011) metric is currently the most respected metric for comparing two different reconstructions. However there are a few limitations, for example, two reconstructions performed manually by different users do not necessarily concord with a perfect score of 1, and this metric was originally intended for neurons. Despite these limitations, we found this metric to be helpful in comparing reconstructions of the tree-like microglial arbors.

To compare the quality of the seed points we computed three scores for every cell in the test set. The first score is the number of seed points in the vicinity (we used a radius of 5 voxels) of the

Table 1. Validation of seed points

Algorithm	% seed points on traces	% seed points not on traces	Coverage (% of traces covered by seed points)
Wang <i>et al.</i> (2011)	40.1	59.8	94.3
Galbreath <i>et al.</i> (2011)	37.5	62.5	98.6
Our method	72.3	27.8	94.7

manual trace. Higher percentage values indicate greater concordance. However, this does not inform us about the quality of the seed points. The next score is the percentage of seed points that are in the background. This measure indicates the potential for spurious traces. The third score is the percentage of traces missed or not covered by the detected seed points. This measure indicates the potential for missed processes. The average scores for 25 cells are presented in Table 1.

3.1 Parameter selection

Seed point selection is the primary determinant of reconstruction performance. For this reason, our first experiments are designed to evaluate the impact of parameters that directly affect this step. The first step in seed detection is to learn the patch-based dictionary. Based on the expected range of diameters of the arbor processes, we used a fixed 3-D block (patch) size of $15 \times 15 \times 3$ voxels. The labels for learning the dictionary and training the classifier within the LC-KSVD algorithm were selected randomly from the manual traces. A total of 50 000 seed points from the image foreground were randomly selected from the training images, and the same numbers of pixels were selected from the background. The dictionary D and the classifier W were derived jointly using the LC-KSVD method.

The seed detection algorithm has four parameters (T , K , α , β) for the dictionary learning step. The parameters α and β set the relative importance of the terms in Equation (3). For our experiments, these parameters were set to 5 and 8, respectively following (Jiang *et al.*, 2011). We observed that changing these parameters did not have a substantial impact on the type of dictionary learned. We also compared the DIADEM metric for K-SVD (by setting α to zero) versus LC-KSVD. Our results demonstrated that discriminative dictionary learning (LC-KSVD) outperforms the K-SVD method. Furthermore, the DIADEM metric did not vary significantly with respect to the size of the dictionary. During the classification stage, there is only one parameter—the sparsity constraint (T) that must be set by the user. This parameter can be set to the same value that was used for the learning stage. To examine how the reconstruction performance varies with different choices of T and K , we divided our dataset of 50 cells into two groups, a training set containing 25 cells and a test set containing 25 cells. We varied T from 2 to 15 and K from 675 to 1800, respectively, and plotted the average DIADEM metric for these 25 cells in Figure 4B and 4C. As pointed out in describing the LC-KSVD algorithm, the performance of the classification algorithm was not very sensitive to the size of the dictionary and the sparsity constraint, due to the discriminative term that was added during the dictionary and classifier learning stage. The performance of our algorithm was between 0.7 and 0.8 for different values of K and T , indicating that it is stable against parameter changes. The tracing algorithm is, by design, sensitive to one parameter—the cost threshold τ . Figure 4D presents a plot of the average

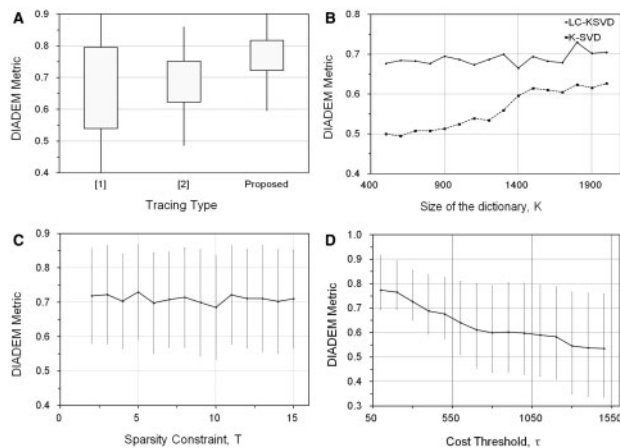


Fig. 4. Performance of our algorithm compared to manual reconstruction as measured by the DIADeM metric. (A) The mean DIADeM metric for our algorithm (0.77) was higher than the value for the algorithm of (Galbreath, 2011)[1](0.65) and the GVF based Vesselness-(Wang et al., 2011)[2] (0.68) based algorithm. (B) Our algorithm was insensitive to the size of the dictionary parameter K. (C) Our algorithm was insensitive to variations in the sparsity constraint T. (D) Increasing the cost threshold τ led to reduced performance, as expected. A threshold of 200 was chosen empirically for all the subsequent experiments based on this analysis

DIADeM metric for 25 different cells against the cost threshold. As expected, the performance of the algorithm was sensitive to this parameter. To test the robustness of the parameters to perturbations, we selected the parameters (K, T, τ) with the highest DIADeM metric on the training set and traced the test set. The algorithms performed equally well on this dataset with a mean DIADeM metric of 0.77 and a standard deviation of 0.07.

We also compared the performance of our algorithm with two different state of the art algorithms based on Laplacian of Gaussian (LoG) and vesselness measures for detecting the seed points (Galbreath, 2011; Wang et al., 2011) for microglia images. The results were found to be on par, and in some cases, better, as shown in Figure 4A. However our algorithm has the important advantage of performing reconstructions in a scalable and seamless manner across large 3-D image mosaics containing thousands of cells as exemplified by Figure 1B. This gives us the ability to perform arbor analytics seamlessly, on a large scale, over extended tissue regions.

3.2 Population-scale arbor morphology analysis by harmonic co-clustering of L-measure data

Given the reality that no single number captures all aspects of a complex cellular arbor, it is necessary to compute libraries of arbor features. Scorioni et al. (2008) have compiled a comprehensive library called the L-measure, containing approximately 130 features per arbor that quantify various aspects of arbor morphology (e.g. size and shape of the soma, number of stems, branching and tapering profiles, etc.). These features capture the seemingly abstract notion of ‘arbor morphology’ in the aggregate. In this work, we adopted this methodology for microglia populations, taking advantage of the fact that our reconstructions are guaranteed to be trees. This computation yields a large table of L-measure data with 130 columns and as many rows as there are cells. Interpreting this large table to extract usable insight is a difficult task for several reasons. First the data dimensionality is high, and individual L-measure features represent dissimilar quantities (lengths, areas, volumes, ratios, counts, etc.).

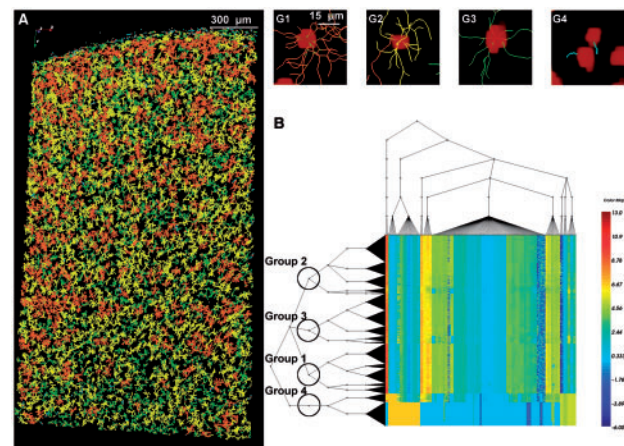


Fig. 5. Quantitative analysis of the microglial cell population by harmonic co-clustering of the L-measure data. (A) 3-D rendering of the reconstructed microglial field in Figure 1, with the arbor traces color coded to indicate the automatically identified groups (red: Group 1, Yellow: Group 2, Green: Group 3, Blue: Group 4). (B) Heatmap representation of the co-clustering results. Each row in the heatmap corresponds to one cell, and each column corresponds to one feature in the L-measure. The co-clustering identified four principal groups. Representative cells from these groups are shown in Panels G1–G4, and correspond to ramified cells (G1), elongated cells (G2), activated cells (G3) and amoeboid/round cells (G4), respectively

The most common need for microglia analysis is to organize the cell population into groups of similar cells so the data interpretation can be attempted one group at a time. Another practical need is to identify features and groups of features that distinguish groups of cells. Co-clustering is a natural strategy for achieving these goals simultaneously. The choice of an appropriate distance measure is a central decision for co-clustering. Traditional metrics, such as the Euclidean distance measure, are quite inadequate for at least two reasons. First, the various features in the L-measure are dissimilar quantities with different units (e.g. soma volume versus process length). Second, the dimensionality of the data is high. Third, the data is inherently noisy. For these reasons, we prefer the non-linear stochastic diffusion distance (Coifman and Lafon, 2006) to compare arbor morphologies. Coifman’s unsupervised harmonic co-clustering algorithm (Lu et al., 2014) uses this measure in conjunction with Haar wavelet smoothing to robustly identify groups of cells and features in a hierarchical manner. Importantly, this algorithm has very few tunable parameters, and usable results are produced with default parameter settings. The harmonic co-clustering algorithm clusters both cell reconstructions and cell features at multiple scales simultaneously. This reveals groups of cells with similar morphological properties, and the corresponding features that distinguish the cell groups. The results of co-clustering are presented as a heatmap and dual-tree representation in Figure 5B. Each row in the heatmap corresponds to one cell, and each column corresponds to one feature. The horizontal tree on the left-hand side of the heatmap shows the grouping of the cells, whereas the tree on the top shows the grouping of features. The analysis reveals that microglia in this dataset fall into four major groups (indicated by circles drawn on the horizontal tree). Close-up renderings of representative cells from the four groups are presented in Panels 5G1–5G4. They clearly correspond to the known microglial activation states including ramified/resting, elongated, activated, and amoeboid/round, respectively. The percentages of these four groups in the population are listed in

Table 2. Relative abundances of arbor morphology groups

	Group 1	Group 2	Group 3	Group 4	Total
Population	993	960	927	430	3310
Percentage	31%	29%	27%	13%	100%

Table 3. Selected features of arbor morphology groups

Group	Surface area (μm^2)	Volume (μm^3)	Segments	Stems	Branch point	Bifurcation	Skewness
1	159	174	35	8	14	13	30.1
2	78	84	19	5	7	6	28.3
3	42	44	9	2	4	5	19.6
4	2	1	0	0	0	0	0

Table 2. The ramified and elongated cells have the highest proportion in this data set and they are evenly distributed, whereas the activated and amoeboid cells are the smallest populations. The morphological complexities of the four groups are in decreasing order. We also calculated the mean values of the significant features from each group in Table 3. The overall size (surface area and volume) of the four groups are dramatically different from each other. Ramified cells and elongated cells are much larger than the activated cells and amoeboid cells. The complexities (segments, bifurcation, branch point and skewness) of different groups are also distinct. Ramified and elongated cells exhibit higher complexity compared to the other two types. This example illustrates the practical utility of our method.

4 Conclusion and Discussion

Our method provides a comprehensive population-scale solution for reconstructing, mapping, and quantitatively profiling microglial arbors in a scalable and seamless manner over extended 3-D regions of brain tissue. At the scale of individual cells, our method outperforms current approaches as summarized in Figure 4. At the tissue scale, our method provides novel capabilities. It can be used to study structural and functional alterations of microglia in response to diverse stimuli, for example, experimental drugs (Dambach et al., 2014), injury (Rey-Villamizar et al., 2014), and alcohol abuse (de la Monte and Kril, 2014). By reconstructing each and every cell in the image field, we avoid the problem of high variance associated with sub-sampling-based methods like unbiased stereology (Mouton, 2011), important for precise measurement of microglial responses at precisely chosen sites. Our algorithms are capable of adapting to novel image collections by machine learning, and are not restricted to a limited range of samples that fit a predefined set of analytical models. The ‘dice and trace’ method guarantees that our method can generate seamless reconstructions of the microglia that straddle imaging windows, and is indefinitely scalable with existing computers. This design will be crucial as large-scale brain tissue mapping efforts turn their focus towards the under-served ‘other brain’ composed of glia and vasculature (Fields, 2013).

Formulating the reconstruction problem in terms of minimal spanning trees is important—it not only guarantees that the reconstructions are trees but also avoids the difficult problem of breaking non-tree reconstructions. Our algorithm for dynamic linking of nodes improves upon the traditional FMM algorithm in several

ways. First, we construct multiple trees simultaneously from a common pool of nodes. This design not only takes advantage of how microglia are distributed in non-overlapping spaces, but also provides an approach to parallel computing for scalability. Second, the edge costs between nodes are not all defined when the algorithm begins execution. This is because our cost metric depends on the costs of the neighboring nodes. As the trees grow from their respective roots, more nodes are included in the queue dynamically, and their costs are updated as they are added. The final difference is that our algorithm incorporates a cost threshold τ , as explained above. Nodes that are deemed more expensive than this threshold are not added to a tree. This is a reliable mechanism to prevent distant and ambiguous pixels from being erroneously incorporated into our reconstructions.

Harmonic co-clustering is an effective exploratory analysis method that opens the door to subsequent group-specific analysis. This method is robust to noise in features, produces biologically meaningful yet concise results, and is scalable to large datasets (Lu et al., 2014). Our experimental results show that our method is able to identify morphologically distinct subpopulations of microglia that concord with the known biology. On a practical note, we have found our linked software implementation method (www.farsight-toolkit.org) to be particularly suited to microglia analysis. Each row of the harmonic co-clustering heat map is directly linked to an individual cell, and its corresponding L-measure data. This linking extends to other analytics tools (Xu et al., 2013), and a comprehensive 3-D visualization and proofreading system (Luisi et al., 2011). In summary, our method will enable comprehensive analysis of microglia morphology and distribution, and coupled with multi-channel 3-D imaging (Bjornsson, 2008), will provide unprecedented methods for mapping cellular-scale alterations as a function of development, function, disease, or injury.

Acknowledgements

We are grateful to Prof. Lawrence Carin at Duke University and Prof. Ronald Coifman at Yale University for advice on harmonic co-clustering analysis. We also thank Ms. Audrey Cheong for manual reconstruction assistance, Mr. Zachary Galbreath at Kitware Inc., for software process assistance and Dr. Raghav Padmanabhan, for his assistance in revising this manuscript. Funding: DARPA Grant N66001-11-1-4015, and NIH Grant R01EB014955.

Conflict of Interest: none declared.

References

Aharon, M. et al. (2006) K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54, 4311–4322.

Al-Kofahi, Y. et al. (2010) Improved automatic detection and segmentation of cell nuclei in histopathology images. *IEEE Trans. Biomed. Eng.*, 57, 841–852.

Al-Kofahi, Y. et al. (2008) Improved detection of branching points in algorithms for automated neuron tracing from 3D confocal images. *Cytom. Part A*, 73, 36–43.

Bas, E. and Erdogmus, D. (2010) Piecewise linear cylinder models for 3-dimensional axon segmentation in brainbow imagery. In: *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro.*, pp. 1297–1300.

Bjornsson, C.S. et al. (2008) Associative image analysis: a method for automated quantification of 3D multi-parameter images of brain tissue. *J. Neurosci. Methods*, 170, 165–178.

- Breitenreiter, D. et al. (2013) Hierarchical discriminative framework for detecting tubular structures in 3D images. *Inf. Process. Med. Imaging*, **23**, 328–339.
- Chen, S.S. et al. (1998) Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, **20**, 33–61.
- Chothani, P. et al. (2011) Automated tracing of neurites from light microscopy stacks of images. *Neuroinformatics*, **9**, 263–278.
- Coifman, R.R. and Lafon, S. (2006) Diffusion maps. *Appl. Comput. Harmon. Anal.*, **21**, 5–30.
- Dambach, H. et al. (2014) Glia and epilepsy: experimental investigation of antiepileptic drugs in an astroglia/microglia co-culture model of inflammation. *Epilepsia*, **55**, 184–192.
- Fields, R.D. (2013) Neuroscience: map the other brain. *Nature*, **501**, 25–27.
- Galbreath, Z.S. (2011) Tracing, extracting features, and classifying microglia from volumetric images of brain tissue. New York (Troy): Rensselaer Polytechnic Institute, 56p.
- Gehrmann, J. et al. (1995) Microglia: intrinsic immunoeffector cell of the brain. *Brain Res. Rev.*, **20**, 269–287.
- Gillette, T.A. et al. (2011) The DIADEM metric: comparing multiple reconstructions of the same neuron. *Neuroinformatics*, **9**, 233–245.
- Gonzalez, G. et al. (2010) Delineating trees in noisy 2D images and 3D image stacks. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2799–2806.
- Gorodnitsky, I.F. and Rao, B.D. (1997) Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm. *IEEE Trans. Signal Process.*, **45**, 600–616.
- He, W. et al. (2003) Automated three-dimensional tracing of neurons in confocal and brightfield images. *Microsc. Microanal.*, **9**, 296–310.
- Jiang, Z. et al. (2011) Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 1697–1704.
- Jiménez, D. et al. (2014) Improved automatic centerline tracing for dendritic and axonal structures. *Neuroinformatics*, **13**, 1–18.
- De la Monte, S.M. and Kril, J.J. (2014) Human alcohol-related neuropathology. *Acta Neuropathol.*, **127**, 71–90.
- Lu, Y. et al. (2014) Quantitative arbor analytics: unsupervised harmonic clustering of populations of brain cell arbors based on L-measure. *Neuroinformatics*, **13**, 47–63.
- Luisi, J. et al. (2011) The FARSIGHT trace editor: an open source tool for 3-D inspection and efficient pattern analysis aided editing of automated neuronal reconstructions. *Neuroinformatics*, **9**, 305–315.
- Mallat, S.G. (1993) Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.*, **41**, 3397–3415.
- Mariano, A. et al. (2013) Hardware and software implementations of Prim's algorithm for efficient minimum spanning tree computation. In: *Embedded Systems: Design, Analysis and Verification*. Springer, Heidelberg, pp. 151–158.
- Meijering, E. (2010) Neuron tracing in perspective. *Cytom. A*, **77**, 693–704.
- Mouton, P.R. (2011) Unbiased Stereology: A Concise Guide. The Johns Hopkins University Press, Baltimore, MD.
- Myatt, D.R. et al. (2012) Neuromantic - from semi-manual to semi-automatic reconstruction of neuron morphology. *Front. Neuroinform.*, **6**, 4.
- Narayanaswamy, A. et al. (2011) 3-D image pre-processing algorithms for improved automated tracing of neuronal arbors. *Neuroinformatics*, **9**, 219–231.
- Natarajan, B.K. (1995) Sparse approximate solutions to linear systems. *SIAM J. Comput.*, **24**, 227–234.
- Ohsawa K. and Kohsaka S. (2011) Dynamic motility of microglia: purinergic modulation of microglial movement in the normal and pathological brain. *Glia*, **59**, 1793–1799.
- Padmanabhan, R.K. et al. (2014) An active learning approach for rapid characterization of endothelial cells in human tumors. *PLoS One*, **9**, e90495.
- Pati, Y.C. et al. (1993) Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*. IEEE Comput. Soc. Press, pp. 40–44.
- Peng, H. et al. (2010) Automatic reconstruction of 3D neuron structures using a graph-augmented deformable model. *Bioinformatics*, **26**, i38–i46.
- Peng, H. et al. (2014) Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis. *Nat. Commun.*, **5**, 4342.
- Rey-Villamizar, N. et al. (2014) Large-scale automated image analysis for computational profiling of brain tissue surrounding implanted neuroprosthetic devices using Python. *Front. Neuroinform.*, **8**, 39.
- Rouchdy, Y. et al. (2011) Minimal path techniques for automatic extraction of microglia extensions. *Int. J. Comput. Vis. Biomech.*, **4**, 35–42.
- Rouchdy, Y. and Cohen, L.D. (2013) Geodesic voting for the automatic extraction of tree structures. Methods and applications. *Comput. Vis. Image Underst.*, **117**, 1453–1467.
- Rubinstein, R. et al. (2010) Dictionaries for sparse representation modeling. *Proc. IEEE*, **98**, 1045–1057.
- Schmitt, S. et al. (2004) New methods for the computer-assisted 3-D reconstruction of neurons from confocal image stacks. *Neuroimage*, **23**, 1283–1298.
- Scorioni, R. et al. (2008) L-Measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nat. Protoc.*, **3**, 866–876.
- Sethian, J.A. (1996) A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci. USA*, **93**, 1591–1595.
- Streit, W.J. (2005) Microglia and neuroprotection: implications for Alzheimer's disease. *Brain Res. Rev.*, **48**, 234–239.
- Teacher, A.G.F. and Griffiths, D.J. (2011) HapStar: automated haplotype network layout and visualization. *Mol. Ecol. Resour.*, **11**, 151–153.
- Tsai, C.-L. et al. (2011) Robust, globally consistent and fully automatic multi-image registration and montage synthesis for 3-D multi-channel images. *J. Microsc.*, **243**, 154–171.
- Türetken, E. et al. (2011) Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors. *Neuroinformatics*, **9**, 279–302.
- Türetken, E. et al. (2013) Reconstructing Loopy Curvilinear Structures Using Integer Programming. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1822–1829.
- Vasilkoski, Z. and Stepanyants, A. (2009) Detection of the optimal neuron traces in confocal microscopy images. *J. Neurosci. Methods*, **178**, 197–204.
- Wang, Y. et al. (2011) Novel 4-D open-curve active contour and curve completion approach for automated tree structure extraction. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 1105–1112.
- Wearne, S.L. et al. (2005) New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales. *Neuroscience*, **136**, 661–680.
- Xiao, H. and Peng, H. (2013) APP2: Automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics*, **29**, 1448–1454.
- Xu, Y. et al. (2013) Unsupervised inference of arbor morphology progression for microglia from confocal microscope images. In: *Proceedings of the IEEE 10th International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE, San Francisco, pp. 1356–1359.
- Yuan, X. et al. (2009) MDL constrained 3-D grayscale skeletonization algorithm for automated extraction of dendrites and spines from fluorescence confocal images. *Neuroinformatics*, **7**, 213–232.
- Zhang, Q. and Li, B. (2010) Discriminative K-SVD for dictionary learning in face recognition. In: *2010 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 2691–2698.