Towards automated Neuroanatomy

Kui Qian, Elizabeth Friedman, David Kleinfeld, Yoav Freund

No Institute Given

Abstract. A fundamental goal of neuroanatomy is the identification of brain structures. Manual identification of structures is based on the spatial distribution of cell shape, size, orientation and density. With new technology it is possible to image entire brains at high resolution. However, manual identification of structures in these massive datasets is prohibitively time consuming. We present a machine learning method for automatic detection of brain structures. Our approach is to use, as a basic unit, the images of individual cells. We translate each cell image into a feature vector that includes aspect ratio, orientation and area, as well as additional features derived using a graph Laplacian. The algorithm uses the statistical distribution of these features vectors to identify brain structures.

1 Introduction

Our goal in the design of the system is to create models of the brain that can be explained to, and modified by, anatomists. In other words, we take a white box, rather than a black box approach.

This impacts the design in several important ways. First, image analysis methods in general, and CNNs in particular, are based on concept of a sliding window. In other words, the basic unit of analysis is a rectangular area, whose location is indpendent of the content of the image. In contrast our analysis unit is the image of a single cell. [1] By annotating cells we visually explain to the anatomist the basis for the computer's decision.

Second, we represent each cell by a set of features. the basic features are size, aspect ratio, and orientation. To characterize the cell shape beyond these basic features, we use an unsupervised learning method to find a dimensionality reducing mapping from cell image to ten additional parameters. We demonstrate that this unsupervised method provides a consistent parameterization across brains and across imaging modalities.

Third, explaining structure detection (xxxx).

2 Adaptive parametrization of cell shapes

In this section we describe the two main technical contributions of this paper: a method for learning a cell-shape feature-set that is optimized for a particular

[1] When cells are overlapping we sometimes get the image of several scells as the unit.

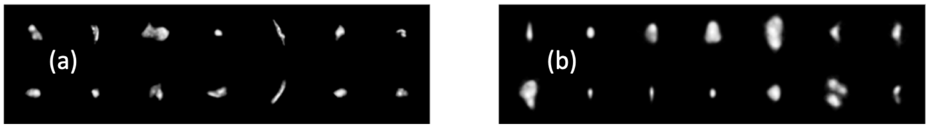

Fig. 1. (a) a sample of Cell patches, **(b)** a sample of the orientation-normalized, Kmeans-reduced and averaged cell images.

brain, and a method for mapping between feature-sets from different brains. Both methods are unsupervised, in other words, the input to these methods are sets of sections with no annotation or alignment.

The first step is to segment individual cells from the section image. For this we use a locally adaptive variant of Otsu's method [6]. This step is not perfect, over and under segmentations are common. However, our method relies on the statistical properties for many cells, which greatly reduces the sensitivity of the analysis to segmentation errors. Cell patches are typically 50×50 pixels which is a highly redundant representation. The second step is to map the cell image to a less redundant low-dimensional representation. Our goal here is to find a small set of features (10) that represent shapes of cells from a given brain. Our learned features provide a better representation of cells shapes which contrast with the feature extraction stage in [3], which uses a pre-trained CNN (Inception-BN).

We use Diffusion Maps (DM) [2, 4] [2] to find a low dimensional representation of cell shapes. While DM has gained popularity in many applications, the only work we are aware of regarding the use of DM for cell shape analysis is in the context of differentiation lineages [5]. The public implementation of DM is excellent. However, it is limited to datasets that fit in computer memory. As we typically extract from each brain hundreds of millions of cells, we need an efficient way to create a small number of representative patches. We use a streaming version of the Kmeans++ [1] seeding algorithm followed with a few iterations of Kmeans. Figure 1 shows samples of the original cell images and of cells after normalizing rotation, Kmeans++ and Kmeans averaging.

DM [4] creates a continuous non-linear mapping from single cell patches into a ten component feature vector. The ten components correspond to ten most dominant (lowest non-zero eigenvalue) eigen-vectors of the Laplace-Beltrami operator. We conceptualize the DM of a population of cells as a ten dimensional cloud of patches, each patch placed at the location defined by the corresponding feature vector. By projecting this cloud onto a two dimensional plane, we generate a visualization of the patch cloud (see figure 2).

After inspecting patch clouds from several brains, it became clear that the clouds corresponding to different brains are similar. More precisely, the clouds all have similar shapes when projected on different planes, but the order of and orientation can be different. This gave us the idea that there might be a single universal set of features, such that one can map the shape-features from any brain to this universal set of features.

[2] `https://github.com/DiffusionMapsAcademics/pyDiffMap`

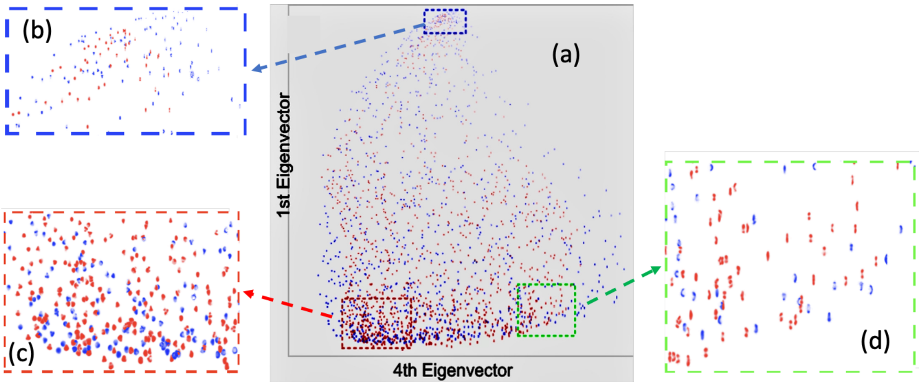

Fig. 2. Projecting the patch cloud and universal features. This figure demonstrates the continuous mapping generated by DM. **(a)** shows the projection of two patch clouds on the first and fourth eigenvectors produced by DM. The red patches correspond to cells stained by Thionin and imaged using brightfield, the blue patches correspond to cells stained with Neurotrace blue and imaged using fluorescent imaging. The brightfield cloud has been transformed to match the fluorescent cloud. Observe that the shape of the clouds match closely. Further, the insets show that the cell shapes match as well. **(b)** corresponds to very small cells. **(c)** correspond to large and round cells, and **(d)** correspond to thin cells or a row of 2-3 cells.

Such a universal set of features provides several benefits. First, it provides a way to visualize the difference between regions in terms of the density, types and orientations of cells. This gives the anatomist a way to understand the decisions made by the computer and potentially correct them. This is in contrast with black box methods such as deep neural networks which provide no explanations of their decisions.

Second, using a universal set of features decouples any downstream processing from variations in section preparation and imaging. Such decoupling is particularly valuable when switching between different imaging modalities such as fluorescence vs. brightfield. In the next section we show that structure Detection benefits from the universal set of features.

After the transformation to universal features, patch clouds from different imaging modalities overlap almost perfectly. This is demonstrated in figure 2. The transformations we used to between patch clouds are simple affine transformations. To find a good transformation between two clouds we use a simple RMS-based formulation (details in Appendix A). The important thing is that this transformation, like DM, is unsupervised. All that is needed are the two DM feature vectors and the cell patches. The result is an unsupervised learning method for learning features. The first step is DM and the second step is finding a transformation from the generated feature set to a universal feature set.

3 Supervised Learning

4 Computer generated suggestions of new landmarks

References

1. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. Tech. rep., Stanford (2006)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation **15**(6), 1373–1396 (2003)
3. Chen, Y., McElvain, L.E., Tolpygo, A.S., Ferrante, D., Friedman, B., Mitra, P.P., Karten, H.J., Freund, Y., Kleinfeld, D.: An active texture-based digital atlas enables automated mapping of structures and markers across brains. Nature methods **16**(4), 341–350 (2019)
4. Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proceedings of the national academy of sciences **102**(21), 7426–7431 (2005)
5. Haghverdi, L., Buettner, F., Theis, F.J.: Diffusion maps for high-dimensional single-cell analysis of differentiation data. Bioinformatics **31**(18), 2989–2998 (2015)
6. Otsu, N.: A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics **9**(1), 62–66 (1979)

A Transformation between Diffusion Maps

A.1 Problem Description and Solution

The mathematical formulation of finding a linear transformation between two different diffusion maps can be described as follows: Given a set of vector pairs $(a_1, b_1), \ldots, (a_n, b_n)$ where each of the vectors a_i, b_i are in a d dimensional space R^d, find an offset vector $\mu \in r^d$ and a linear transformation M which is a $d \times d$ matrix so that the following cost function is minimized:

$$cost(\mu, M) = \frac{1}{n}\sum_{i=1}^{n} ||\mu + Ma_i - b_i||_2^2 \tag{1}$$

We first compute the hessian matrix of 1 with respect to μ and M to see if the problem can be directly solved.

$$\frac{\partial^2 cost(\mu, M)}{\partial\mu\partial\mu^T} = 2I \rightarrow p.d \tag{2}$$

$$\frac{\partial^2 cost(\mu, M)}{\partial M\partial M^T} = \frac{2}{n}\sum_{i=1}^{n} a_i a_i^T \rightarrow p.s.d \tag{3}$$

The results show that both hessian matrices are Positive Semi-definite matrices and thus we can find the solution by setting derivatives to 0. We have:

$$(\frac{1}{n}\sum_{i=1}^{n} a_i a_i^T - \frac{1}{n}\sum_{i=1}^{n} a_i \times \frac{1}{n}\sum_{i=1}^{n} a_i^T)M^T = \frac{1}{n}\sum_{i=1}^{n} a_i b_i^T - \frac{1}{n}\sum_{i=1}^{n} a_i \times \frac{1}{n}\sum_{i=1}^{n} b_i^T \tag{4}$$

B Pseudo-code for Finding Significant Regions

1. Calculate the statistical significance map of the whole 3D shape relative to the background as $Mask$. Traverse the whole shape with a cube (200 microns cubed) by a step size of 100 microns.
2. Define M to mark regions in the 2D shape and R to record cubes of each region.
3. Find seeds and expand them to foreground regions according the following rules:
   - Find the cube with the highest KS significance in $Mask$ of unmarked regions ($M == 0$) as the seed. Append that cube to a list of potential starting cubes C.
   - Pop one starting cube from C. Starting from the chosen cube, slide by a stride of half window size to get 8-connected neighborhood cubes as foreground candidates.
   - The foreground candidates are added to the foreground and the list C if $dist(x, S_B) > \alpha \cdot dist(x, S_F)$ where S_F represents the CDFs of cells in the foreground and S_B represents the CDFs of cells in the background. α is set to be 3.
   - Repeat the above two steps until the list C is empty.
4. Find 100 regions via 200 micron cubed seeds, then find 30 regions via 100 micron cubed seeds.
5. After finding regions, for pairs of regions in R:
   - If more than 80% of a region A is covered by another one B, then A is merged to B.
   - If A and B have intersection areas, A and B are merged together if $dist(A, B) < threshold$ else, $A \cup B$ belongs to A if $dist(A \cup B, A) < dist(A \cup B, B)$ otherwise belongs to B.

C List of Additional Images

We have demonstrated the ability of our method in structure detection taking the 6N/R structure as example. Images of two other structures are also provided.

1. `5N_L.png`: Detection figure for the 5N/L structure
2. `5N_R.png`: Detection figure for the 5N/R structure