Computer vision profiling of neurite outgrowth morphodynamic phenotypes

September 24, 2013

We developed an image processing pipeline to segments nuclei, somata and neurites and to track them from Time-Lapse High-Content Screens. In the following, we first describe the data and introduce some notations, then we give a detailed description of our processing pipeline. Finally, we describe our evaluation methodology to assess the quality of the segmentation results.

1 Data description and notations

We worked with sequences of two-channel images, one in which the cytoskeleton is marked with Lifeact-GFP. In the other, the nuclei are marked with NLS-mCherry, see the first column of figure 2.

The input to our approach is a series of T images $\mathcal{I} = \{I_1, \ldots, I_t, \ldots, I_T\}$ from which we extract K nucleus detections d_t^k. The tracking step described in Sec. 2.2 associates valid detections across time steps while rejecting spurious detections. Since each neuron contains only one nucleus, there is a one-to-one mapping between each valid nucleus detection c_t^i and a neuron X_t^i. Thus, the tracking task is to provide a set of neuron detections $\mathcal{X}^i = \{X_a^i, \ldots, X_t^i, \ldots, X_b^i\}$ defining an individual neuron i from time $t = a$ to $t = b$. As depicted in Fig. 1, each neuron detection X_t^i is composed of a nucleus c_t^i, a soma s_t^i, and a set of J neurites $N_t^i = \{n_t^{i,1}, \ldots, n_t^{i,j}, \ldots, n_t^{i,J}\}$. Thus, a complete neuron i at time step t is described by $X_t^i = \{c_t^i, s_t^i, N_t^i\}$.

2 Neuron segmentation and tracking

A three stage neurone segmentation and tracking approach is proposed. First, as described in section 2.1, we segment nuclei and the associated somata. Since each neuron contains only one nucleus, there is a one-to-one mapping between each valid nucleus and soma, yielding to a neurone cell body. Second, the cell body tracking step described in section 2.2, associates valid cell bodies across time steps while rejecting spurious ones. Finally, using the valid tracked cell bodies, neurites are segmented and tracked over time.

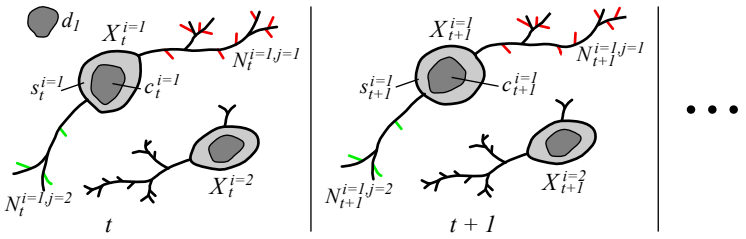

Figure 1: TODO TODO: no filopodia Neuron tracking notation. At time t a neuron i detection $X_t^i = \{c_t^i, s_t^i, N_t^i\}$ contains a nucleus c_t^i, a soma s_t^i, and a set of neurite-filopodia tuples $N_t^i = \{(n_t^{i,1}, F_t^{i,1}), \ldots, (n_t^{i,j}, F_t^{i,j}), \ldots, (n_t^{i,J}, F_t^{i,J})\}$ which contains J neurites and their associated filopodia shown in red for $j = 1$ and green for $j = 2$. A spurious nucleus detection d_1 is also shown. A neuron i is defined by a time-series of neuron detections $\mathcal{X}^i = \{X_a^i, \ldots, X_t^i, \ldots, X_b^i\}$. The tracking returns a set $\mathcal{X}^i$ for each neuron.

2.1 Nuclei and Somata segmentation

The first step in our approach is to extract a set of nucleus detections $\{d^1, \ldots, d^K\}$ over the image series. We worked with two-channel images where the cytoskeleton is marked with Lifeact-GFP and nuclei are marked with NLS-mCherry. The nuclei can be reliably detected as a Maximally Stable Extremal Region (MSER) [1] of the NLS-mCherry channel, and performing a morphological filling operation. The MSER detector finds regions that are stable over a wide range of thresholds of a gray-scale image. Assuming that the pixels below a given threshold are "black" and all those above or equal are "white"; if we are shown a sequence of thresholded images I_τ with frame τ corresponding to threshold τ, we would see first a white image, then "black" spots corresponding to local intensity minima will appear then grow larger. These "black" spots will eventually merge, until the whole image is black. The set of all connected components in the sequence is the set of all extremal regions. Once the extremal regions are extracted, a maximality criterion is defined and only a few extremal regions are kept, which are the Maximally Stable Extremal Regions.

To extract the Maximally Stable Extremal Regions form the NLS-mCherry channel, we used the `VLFeat` implementation of MSER[1]. Default parameters of the `VLFeat` implementation were used to segment the nuclei, except the minimal and maximal size of a nuclei at the given resolution, which were fixed to 70 and 170 [2]. The main advantage of MSER compared to the thresholding approach [2] is its robustness and insensitivity to contrast change.

Using the nuclei as seed regions, somata are segmented using a region growing and region competition algorithm on the Lifeact-GFP channel, called the *green* channel. This is done by computing geodesic distances from each nucleus based on the difference of image intensities. For a given image frame I, let $\{d^1, \cdots, d^K\}$ be the set of detected nuclei. For each detected nucleus d^k we

[1] publicly available at `http://www.vlfeat.org/`

[2] Nuclei are considered to have circular shapes with radius varying between 4 and 8 pixels

define a potential as follows:

$$\mathcal{P}_k(x) = \frac{1}{A \exp\left(-\frac{(I(x)-\mu_k)^2}{2\beta^2\sigma_k^2}\right) + 1}, \tag{1}$$

μ_k and σ_k are respectively the mean and standard deviation of the green intensities of the pixels describing d^k. Parameter β, is a multiplicative factor of the intensity standard deviation σ_k, and represents a tolerance of variation between the local foreground (soma) intensities and the local background intensities.

The geodesic distance $\mathcal{U}_k$ associated to the nucleus d^k is defined as the solution of the Eikonal equation

$$\|\nabla \mathcal{U}_k\| = \mathcal{P}_k \quad \text{such that} \quad \mathcal{U}_k(d^k) = 0. \tag{2}$$

From equation 1, one can see that the more the green intensity of a pixel $I(x)$ is different from the mean intensity μ_k of the detected nuclei , the higher the potential $\mathcal{P}_k$ would be, and from equation 2, the higher $\mathcal{U}_k$ would be.

The algorithm to compute the geodesic distances $\mathcal{U}_k$, is the Fast Marching algorithm [3, 4]. Our first approach to segment the somata regions consists on thresholding the geodesic distances $s^k = \{\mathcal{U}_k < \mathcal{T}_g\}$. However, as this requires computing all the K distances $\mathcal{U}_k$, the complexity depends on the number of detected cells, and may yield to intersecting somata regions (which was happening as well with the region growing approach proposed in [2]).

Instead, we used a more efficient approach that computes the geodesic distance from all the detected nuclei simultaneously by launching a region growing and region competition algorithm which complexity does not depend on the number of cells but only on the size of the image, and for which the obtained somata segmentations do not intersect. For that, we define a distance map to all the detected nuclei d^k as:

$$\mathcal{U}(x) = \min_{k=1,\cdots,K} \mathcal{U}_k(x). \tag{3}$$

The distance map $\mathcal{U}$ can be approximated by computing the solution of

$$\|\nabla \mathcal{U}\| = \mathcal{P} \quad \text{such that} \quad \mathcal{U}(d^k) = 0, \text{ for all } k, \tag{4}$$

where

$$\mathcal{P}(x) = \frac{1}{A \exp\left(-\frac{(I(x)-\mu_k)^2}{2\beta^2\sigma_k^2}\right) + 1}, \tag{5}$$

and where the index of the closest region k, is decided during the propagation of a variant of the Fast Marching algorithm which was introduced in [5]. The main difference between this variant of the Fast Marching algorithm and the original one in [3, 4], is that the potential values (more precisely the k index for our case) is not fixed before starting the propagation but is rather decided during the propagation. Computed that way, $\mathcal{U}$ defines an approximate geodesic distance to the detections d^k. It combines the local intensity differences and the euclidean

distance. The somata segmentations are finally obtained by thresholding both $\mathcal{U}$ and the euclidean distance to the nuclei (those 2 shresholds are denoted $\mathcal{T}_g$ and $\mathcal{T}_e$ repectivelly). Formally, if $\{R^k\}$ is the Voronoi tessellation associated to $\mathcal{U}$ and the set of nuclei $\{d^k\}$ and if the euclidian distance to the set of nuclei is denoted $\mathcal{D}$ then the soma s^k associated to a detection d^k is defined as follows:

$$s^k = R^k \cap \{\mathcal{U} < \mathcal{T}_g\} \cap \{\mathcal{D} < \mathcal{T}_e\}. \tag{6}$$

Once this intersection is obtained, a morphological filling operation is applied to guarantee that a soma does not contain holes.

An example for nuclei and somata segmentation is depicted on figure 2. For all our experiments, we took $A = 1e7$, $\beta = 1.5$, $\mathcal{T}_g = 2e - 6$ and $\mathcal{T}_e = 7$.

At this point, cell body detections $d_t^i = (c_t^i, s_t^i)$ have been obtained form the whole sequence. A filtering step is applied to these detections to keep only the most reliable ones. First, detections that are too close to the image boundary (with distance form the centroid of the nucleus detection less than 10 pixels) are ignored. Second, the minimal tolerated circularity is 0.2. Finally, the maximal accepted eccentricity is 0.85. Those last 2 criteria are applied to the nuclei detections.

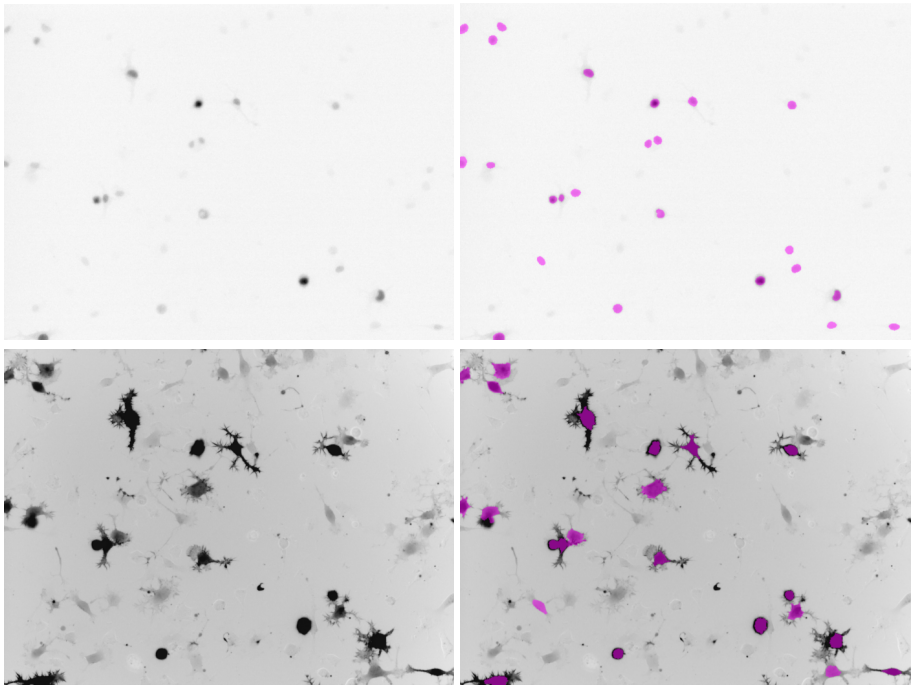

Figure 2: Cell body detection. On the first row, nuclei detections overlaid on top of the NLS-mCherry channel. On the second row, somata detections overlaid on top of the Lifeact-GFP channel. From left to right: the original image, and the automatic detections.

2.2 Cell body tracking

The tracking algorithm searches through the full set of nuclei detections and iteratively associates the most similar pairs of detections, returning lists of valid detections corresponding to each neuron $\mathcal{X}^i$. This is accomplished by constructing a graph $\mathcal{G} = (\mathcal{D}, \mathcal{E})$ where each node $d_t^k \in \mathcal{D}$ corresponds to a detection. For each detection d_t^k in time step t, edges $e \in \mathcal{E}$ are formed between d_t^k and all past and future detections within a time window W. A weight w_e is assigned to each edge according to spatial and temporal distances, and a shape measure $w_e = \alpha||d_{t1}^k - d_{t2}^l|| + \beta|t1 - t2| + \gamma f(\nu_{t1}^k, \nu_{t2}^l)$ where $e^{k,l}$ connects d_t^k and d_t^l, and ν^k is a shape feature vector containing d_t^k's area, perimeter, mean intensity, and major and minor axis lengths of a fitted ellipse. f evaluates differences between a feature a extracted from d_t^k and d_t^l as $f(a^k, a^l) = \frac{|a^k - a^l|}{|a^k + a^l|}$. The tracking solution corresponds to a set of edges $\mathcal{E}' \subset \mathcal{E}$ that minimizes the cost $\sum_{e \in \mathcal{E}'} w_e$.

To minimize this cost function, we adopt a greedy selection algorithm outlined in Table 1 and summarized in Fig. 3 that iteratively selects an edge with minimum cost $\hat{w}_e$ and adds it to the set $\mathcal{E}'$, removing future and past connections from the detections $e^{k,l}$ connects. The algorithm iterates until the minimum cost $\hat{w}_e$ is greater than a threshold T. The track for neuron i is extracted from $\mathcal{E}'$ by traversing the graph $(\mathcal{G}, \mathcal{E}')$ and appending linked nucleus detections to $\mathcal{X}^i$.

Algorithm 1 Greedy tracking association algorithm

Start with an empty set $\mathcal{E}'$.
repeat
  Find edge $\hat{e}^{k,l}$ with minimum cost $\hat{w}_e$.
  Add $\hat{e}^{k,l}$ to $\mathcal{E}'$, linking detections d_{t1}^k and d_{t2}^l.
  Remove $\hat{e}^{k,l}$ from $\mathcal{E}$.
  if $t1 < t2$ **then**
    Remove edges between d_{t1}^k and *future* detections (where $t > t1$) from $\mathcal{E}$
    Remove edges between d_{t2}^l and *past* detections (where $t < t2$) from $\mathcal{E}$
  else
    Remove edges between d_{t1}^k and *past* detections (where $t < t1$) from $\mathcal{E}$
    Remove edges between d_{t2}^l and *future* detections (where $t > t2$) from $\mathcal{E}$
  end if
until $\hat{w}_e > T$

The parameters of this algorithm have been fixed empirically as follows: $W = 4$, $\alpha = 1$, $\beta = 50$, $\gamma = 40$ and $T = 200$. Only tracks containing at east 20 frames are kept. In addition to these parameters, a spatial connection constraint is applied during the construction of the graph $\mathcal{G}$. In fact, only detections that are at distance less than 50 pixels are connected to create the set of edges $\mathcal{E}$.

Once the tracking is achieved, tracks are sorted according to their total cumulated green intensities, and only the 40 best tracks are kept, as this is a reasonable number of stained enough moving cells which are quite visible for an experienced human eye.

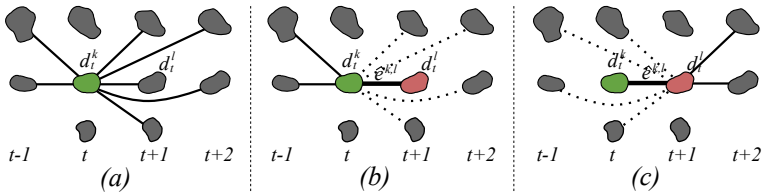

Figure 3: *Greedy Tracking. (a)* The algorithm begins with each detection fully connected to all future and past detections within a time window W. Above, only d_t^k's edges are shown. *(b)* Each iteration, the edge $\hat{e}^{k,l}$ with minimum cost $\hat{w}_e$ is added to $\mathcal{E}'$. Edges connecting d_t^k to future detections are removed from $\mathcal{E}$. *(c)* Edges connecting d_t^l to the past are removed from $\mathcal{E}$. The process is repeated until $\hat{w}_e > T$.

2.3 Neurites segmentation and association

Given an image I_t and the set of somata present in it $S_t = \{s_t^1 \dots s_t^m\}$, our goal is to associate to each pixel u a label $J_t(u)$ that indicates to which soma it belongs. The probability of $J_t(u)$ can be deduced using Bayes' rule,

$$P(J_t(u) = i|S_t, I_t) = \frac{P(S_t, I_t|J_t(u) = i)}{\sum_{\eta=1}^{m} P(S_t, I_t|J_t(u) = \eta)}, \tag{7}$$

where we have assumed a uniform distribution on $P(J_t(u))$. The numerator is modeled as the probability of the path L that connects maximally the voxel u to the soma s_t^i, $P(S_t, I_t|J_t(u) = i) = \max_{L:u\to s_t^i} \prod_{\{l_r\}\in L} P(I_t(r)|l_r)$, where l_r are indicator variables for the locations forming the path L. We chose this model since an optimal maxima can be found by minimizing its negative likelihood using geodesic shortest path [3] and because it produces connected components.

The extraction of neurites from a time frame I_t proceeds in the following stages:

- Compute tubularity measure T_t as in [6]. In addition, detected but not tracked somata are ignored, that is the minimal tubularity value is assigned to "detected but not tracked" somata regions, in order to avoid segmenting their boundaries as neurites.

- Estimate the parameters of a sigmoid functions which is applied to the tubularity measure to obtain a potential P_t that drives the Fast Marching algorithm [3, 4]. The parameters of the sigmoid function are estimated using maximum likelihood.

- Launch simultaneously the front propagation Fast Marching algorithm from all the tracked somata. This is done by solving the Eikonal equation $\|\nabla \mathcal{U}_t\| = P_t$ and $\|\nabla \mathcal{L}_t\| = 1$. That yields a geodesic distance map $\mathcal{U}_t$, the associated tessellation $\mathcal{V}_t$, and the map of the Euclidian lengths $\mathcal{L}$ of the geodesics . The complexity of the algorithm, at this stage, does not depend on the number of tracked somata but depends only on the size of the image.

- Threshold the geodesic distance with a soft threshold $\mathcal{T}_s$, and then extract local maxima of $\mathcal{L}_t$ in each thresholded region[3]. We took local maxima of $\mathcal{L}_t$ as they likely correspond to continuations of elongated shapes (typically neurites). In fact, at a given threshold value of $\mathcal{U}_t$, elongated structures, such as neurites, has higher $\mathcal{L}_t$ values than their surrounding points. Here, we prefer to use a soft threshold and to eliminate the spurious branched in the next step using the hard threshold

- Using the Geodesic distance $\mathcal{U}_t$, back-propagate from all the local maxima of $\mathcal{L}_t$, by keeping only points for which the $\mathcal{U}_t$ value is above a hard threshold $\mathcal{T}_h$. The idea is that even if the candidate endpoints detected in the previous step are not necessarily all correct, they will quickly converge to interesting elongated structures during the back propagation.

- Finally, we instantiate a Minimum Spanning Tree from each root touching a soma, and create the associated neurite tree.

The different steps of our approach are illustrated in figure 4.

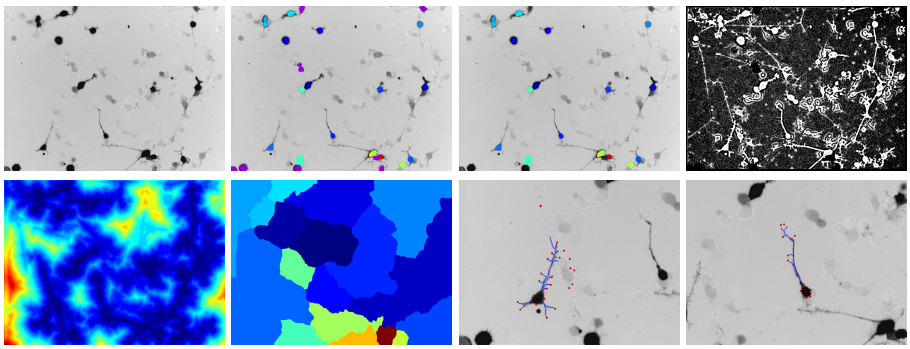

Figure 4: Neurites detection. From left to right, top to bottom : -Original image (only the green channel is displayed). -Both tracked and detected somata are overlaid on top of the original image. Detected but not tracked somata are coloured with dark violet. -Only tracked somata are overlaid on top of the original image. - Potential P_t, based on the tubularity measure of [6]. -Geodesic distance $\mathcal{U}_t$ computed from the tracked somata. Blue corresponds to low values and red to high values. - The Voronoi tessellation $\mathcal{V}_t$ associated to the geodesic distance. Each colour represent a region. -The last two figures are closeup looks of the original image illustrating the last steps of our neurite detection algorithm: first, local maxima of the soft thresholded regions are displayed in red, then the hard thresholded back propagation pixels are shown in blue.

Parameters of the neurite detection algorithm are as follows:

- Frangi [6] parameters are: `FrangiOpt.FrangiScaleRange = [1 2], FrangiOpt.FrangiScaleRatio = 1, FrangiOpt.FrangiBetaOne = .5, FrangiOpt.FrangiBetaTwo = 15,`,

[3]the local maxima are obtained using the Matlab function `imregionalmax`

- Geodesic distance thresholds: $\mathcal{T}_s = -\log(10^{-4})$, and $\mathcal{T}_h = -\log(0.2)$.
- Finally, any neurite containing less than 10 pixels have been ignored.

2.4 Neurite Tracking

Neurites are tracked by applying the algorithm described in Sec 2.2 using the centroids of the neurite trees instead of nucleus centroids, with the additional constraint that edges may only exist between neurites that emanate from the same tracked soma and only from consecutive time detections. The weight w_e of an edge connecting two neurites N_t^i and $N_{t'}^j$ is assigned according to spatial distance and a shape measure $w_e = w_{TCL} f(\text{TotalCableLenght}(N_t^i), \text{TotalCableLenght}(N_{t'}^j)) + w_{\text{Centroid}} \|\text{Centroid}(N_t^i) - \text{Centroid}(N_{t'}^j)\| + w_{\text{SomaContact}} \|\text{SomaContact}(N_t^i) - \text{SomaContact}(N_{t'}^j)\|$, where $w_{TCL} = 50$, $w_{\text{Centroid}} = 10$, $w_{\text{SomaContact}} = 5$, TotalCableLength$(N)$ is the total cable length of a neurite N, Cenrtoid(N) is its centroid, SomaContact(N) is its contact point with the soma, and $f(a^k, a^l) = \frac{|a^k - a^l|}{|a^k + a^l|}$. As opposed to the weights used for cell body tracking, the weights used here for neurites tracking do not include a temporal distance because the neurites are associated to already tracked cell bodies.

During the neurite tracking stage, only neurites that are considered stable have been taken into account. Neurites are considered stable if their total cable length is above 30 pixels. The threshold parameter of the tracking algorithm (Algorithm 1), have been adapted to neurites by taking $T = 800$.

3 Evaluation

The image processing pipeline, described in section 2, segments and tracks cell bodies and neurites. Questions one could raise to evaluate the quality of the segmentation results are:

1. What is the quality of the cell body components segmentation. More precisely, what is the quality of the nuclei and the somata segmentations. (Static evaluation)
2. What is the quality of the cell body tracking (cell identity assignment). In other words, does the algorithm track accurately the moving cells? and is it able not to switch identities of two neighbouring moving cells? (Dynamic evaluation)
3. Are the neurites segmented accurately? Neurite trees are complex structures, and are critical to our phenotypic study. How can we asses their quality is a crucial question to answer (Static evaluation)
4. Are the neurites tracked properly? or a simpler question could be : how accurate is the "neurite to cell body" assignment step? (Dynamic/Static evaluation ?)

In order to evaluate our algorithm in a fair manner, one should bear in mind that, at detection time, we made the choice not to detect and track all visible cells and neurites. For instance, a neurone that enters the field of view for a short period of time (say during 5 frames, for instance) is not kept in the final set of tracks. In addition, as described in section 2.2, tracked cells are sorted according to their cumulated green intensity, and only the 20 best tracks are kept. We made this choice to favours cell body tracks that are bright with a long lifetime. This choice is also justified by the fact that the moving neurons appears with very different contrast: some are very bright and others are quite faint. Assuming that the algorithm will be able to segment and track accurately all the moving neurons is not realistic, and we preferred to focus our efforts in segmenting and tracking accurately the most robust neurons.

In order to evaluate the different critical components of our processing pipeline we need to annotate *ground truth* data. If one annotates all visible moving neurones, then our segmentation results will be penalised by the objects that have been purposefully ignored by the algorithm. Hence, evaluation scores obtained using such an over complete ground truth would not reflect the quality of the segmentation and tracking results. To sumup, a special care must be made for the ground truth annotation step and for the quantitative evaluation.

In the next sections, we will describe the ground truth annotation protocol and the evaluation methodology for each of the critical steps of our processing pipeline.

3.1 Evaluating Cell body detection

To manually segment nuclei and somata from dynamic sequences, we used `TrakEM2`[4], a plugin of `FIJI/ImageJ`[5]. To ease the annotation step, the input sequences have been encoded in a single multipage tiff file for each channel, and an xml file have been generated for each sequence to encode the segmentation and tracking data structure.

A set of instructions was given to 3 different experts to annotate the data:

- segment and track only cell bodies that are bright and that have a long enough lifetime
- segment carefully the nuclei and the somata
- keep track of the identities of the segmented cell bodies

Three sequences have been randomly selected from our dataset dataset. From these sequences, 29 cells have been tracked, representing 2152 annotated nuclei and somata.

A detected object, either a nucleus or a soma, is considered positive if there exist a ground truth object (of the same kind) overlapping sufficiently with it.

[4] publicly available at `http://www.ini.uzh.ch/~acardona/trakem2.html`
[5] `http://fiji.sc/Fiji`

More formally, a detection d is considered as a positive detection if there exist a ground truth object g such that $\frac{g \cap d}{g \cup d} > 80\%$.

On our ground truth dataset, and over the 3 annotated sequences, only 28 nuclei have not been detected, representing **1.3%** of miss detections.

3.2 Evaluating Cell body tracking

3.3 Evaluating Neurites detection

3.4 Evaluating Neurites association

4 Data structure

TODO

References

[1] Matas, J., Chum, O., Urban, M. & Pajdla, T. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, 36.1–36.10 (BMVA Press, 2002). Doi:10.5244/C.16.36.

[2] Gonzalez, G., Fusco, L., Pertz, O. & Smith, K. Automated quantification of morphodynamics for high-throughput live cell imaging datasets. *EPFL Technical Report* (2011).

[3] Cohen, L. & Kimmel, R. Global Minimum for Active Contour Models: A Minimal Path Approach. 666–673 (1996).

[4] Sethian, J. *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science* (Cambridge University Press, 1999).

[5] Benmansour, F. & Cohen, L. D. Fast object segmentation by growing minimal paths from a single point on 2d or 3d images. *Journal of Mathematical Imaging and Vision* **33**, 209–221 (2009).

[6] Frangi, A. F., Niessen, W. J., Vincken, K. L. & Viergever, M. A. Multiscale Vessel Enhancement Filtering. *Lecture Notes in Computer Science* **1496**, 130–137 (1998).