PROOF-READING GUIDANCE IN CELL TRACKING BY SAMPLING FROM TRACKING-BY-ASSIGNMENT MODELS

Martin Schiegg, Ben Heuer, Carsten Haubold, Steffen Wolf, Ullrich Koethe, Fred A. Hamprecht

University of Heidelberg, HCI/IWR, Speyerer Str. 6, 69115 Heidelberg, Germany

ABSTRACT

Automated cell tracking methods are still error-prone. On very large data sets, uncertainty measures are thus needed to guide the expert to the most ambiguous events so these can be corrected with minimal effort. We present two easy-to-use methods to sample multiple proposal solutions from a tracking-by-assignment graphical model and experimentally evaluate the benefits of the uncertainty measures derived. Expert time for proof-reading is reduced greatly compared to random selection of predicted events.

***Index Terms*—** Cell tracking, uncertainty, machine learning, probabilistic graphical models

1. INTRODUCTION

Biomedical applications often require an accurate tracking of proliferating cells [1]. Existing cell tracking methods [2, 3, 4, 5, 6] perform well on sparse cell populations with high signal-to-noise ratio (SNR). However, they are far from achieving gold-standard-accuracy on high-dimensional data with low SNR. To analyze their data with highest reliability, biomedical specialists have to spend their precious time to proof-read the predicted cell lineages and correct them where necessary. When working with high-throughput setups, unguided proof-reading quickly becomes prohibitive. In response, what is needed is automatic guidance which presents the most ambiguous frame-to-frame assignments to the user, omitting trivial assignments.

Recently, numerous tracking-by-assignment methods have been proposed for cell tracking [3, 4, 5]: These structured models typically consist of a data term (local features) and an interaction term (*e.g.* lineage consistency constraints, penalties for large displacements, etc.). While the latter term is based on model assumptions, missing features, noisy signal, or poor image quality may corrupt the first term. When estimating the maximum a-posteriori (MAP) solution to obtain the most likely cell lineages, these cell tracking methods ignore this signal uncertainty.

In this article, we take into account the uncertainty in the measurements to derive ambiguity quantification measures in order to guide the user to those predicted assignments which are most uncertain. Both types of predicted events, cell migration (*moves*) and mitosis (*divisions*) are taken into consideration independently. In order to sample multiple proposal solutions from the tracking-by-assignment method, we present two methods, the application of Perturb-and-MAP [7] and a novel approach based on Gaussian process predictions for the unary terms. Both alternatives allow sampling multiple instances of the underlying graphical model on which, in turn, individual MAP solutions are estimated, *cf.* Fig. 1. We find that the variations in each assignment among these proposal solutions is an efficient guidance measure for proof-reading.

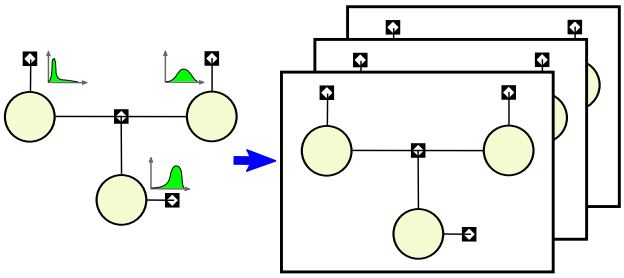


Fig. 1. To derive proof-reading priorities, we propose to perturb the unary potentials of a tracking-by-assignment cell tracking model according to *(i)* a Gumbel distribution [7], or *(ii)* a Gaussian distribution predicted from a Gaussian process. Instances of the graphical model can then be generated by sampling locally from the distributions of the unary potentials. Estimating the MAP solution of each individual graphical model instance allows computing robustness measures for each individual predicted event and helps to guide the biomedical specialist to the most ambiguous assignments.

Related Work Tracking-by-assignment models aim at solving the cell tracking task with a structured model spanning pairs of frames [3] or a larger, possibly the entire, time sequence [4, 5, 8]. These methods can be formulated as probabilistic graphical models which describe a Gibbs distribution defined by local potentials. To quantify uncertainty in such graphical models, sampling from this distribution by Markov chain Monte Carlo (MCMC) techniques would be the natural first choice. However, due to signal-sensitive local potentials, the Gibbs distribution may be "ragged" [9] and sampling will be expensive. In contrast, recent developments in machine learning turn to generating multiple instances of graphical

models and perform a MAP estimation on each of those instances to obtain proposal solutions [7], similar to k-best data association hypotheses [10]. We apply Perturb-and-MAP [7] to cell tracking, *i.e.* all unary potentials are perturbed with the same distribution, and we propose an alternative where the perturbations are based on local uncertainties in the features.

In cell tracking, assignment ambiguities have only been explored rarely. Rapoport *et al.* [11] find potential error types and remove all lineages which are likely to contain such errors to only show the most "trustworthy" lineages, *i.e.* aiming for a high precision at the cost of low recall. The authors in [12] exploit uncertainty in an active learning framework to request the most uncertain assignment from the user in order to improve the weights of the structured cell tracking method.

2. SAMPLING FROM TRACKING-BY-ASSIGNMENT MODELS

First, we briefly recapitulate tracking-by-assignment models, then review Perturb-and-MAP as "one shot approximate random sampling" approaches [7] and finally propose a novel related sampling method using Gaussian processes.

2.1. Tracking-by-Assignment Models

Tracking-by-assignment models typically decouple into two subsequent stages, a detection/segmentation stage followed by a tracking/assignment stage. The latter is often formulated as an energy minimization problem [4, 5],

$$\mathbf{y}^* = \underset{\mathbf{y}\in\mathcal{Y}}{\arg\min}\, E(\mathbf{y}) = \underset{\mathbf{y}\in\mathcal{Y}}{\arg\min}\, \boldsymbol{\theta}^\top \phi(\mathbf{y}) \qquad (1)$$
$$\textit{s.t.}\ \text{consistency constraints},$$

which minimizes real-valued costs (energy) E associated with specific configurations $\mathbf{y} \in \mathcal{Y}$ of the detections and cell-to-cell transitions to find the optimal assignments/detections $\mathbf{y}^*$. In general, the energy function can be expressed in terms of indicator functions $\phi(\mathbf{y})$ and model parameters $\boldsymbol{\theta}$ (which constitutes costs in unary and higher-order potentials). Linear constraints forbid inconsistent solutions such as more than one cell ancestor or more than two cell successors; these constraints can equivalently be added to the objective function. Thus we drop them in our following notations.

In [5], we propose to choose multi-state random variables, $\mathcal{Y} \in \{0, ..., m\}^D$, where m is the maximal number of cells expected to appear in one detection (falsely merged cells)[1] and D is the number of random variables in the graphical model.

Energy minimization problems as in (1) can be casted as maximum a-posteriori (MAP) estimation problems through the maximization of the underlying Gibbs distribution,

$$P(\mathbf{y}) = \frac{1}{Z}\exp\left(-E(\mathbf{y})\right) = \frac{1}{Z}\exp\left(-\boldsymbol{\theta}^\top\phi(\mathbf{y})\right), \qquad (2)$$

where Z is the normalizing partition function.

[1] For the ease of notation, we are here only referring to one variable type $\mathcal{Y}$, whereas in the original model in [5], a finer distinction is made.

2.2. Sampling through Perturb-and-MAP Random Fields

The key idea of Perturb-and-MAP [7] is to inject random noise $\boldsymbol{\varepsilon}$ into the Gibbs distribution in Eq. (2), *i.e.*

$$P_{\boldsymbol{\varepsilon}}(\mathbf{y}) = \frac{1}{Z}\exp\left(-(\boldsymbol{\theta}+\boldsymbol{\varepsilon})^\top\phi(\mathbf{y})\right), \qquad (3)$$

and then repeatedly find the MAP estimates of the perturbed objective. It has been shown [7] that if the perturbation variables $\boldsymbol{\varepsilon}$ are distributed according to a Gumbel distribution, the distribution defined in Eq. (3) approximates the Gibbs distribution in Eq. (2).

In practice, each cost of the (unary) potentials is perturbed independently and identically according to a Gumbel distribution. The scale parameter for this single distribution needs to be set manually. In the following, we propose to model each data term with a distinct Gaussian distribution that reflects the ambiguity of the local observations.

2.3. Sampling through Gaussian Processes

The model parameters $\boldsymbol{\theta}$ introduced in Sec. 2.1 relate to the energies defined by unary or higher-order potentials in the graphical model. In many graphical models, the unary potentials are based on predictions of classifiers fed with local features. This is also true for the cell tracking method [5] studied in this article: The unary costs are results of a cell division classifier, a cell detection classifier, and a transition classifier. We propose to utilize Gaussian process classification[2] for the generation of unary potentials since they provide full predictive distributions for each query point.

In particular, let us denote the energy associated with some variable $Y_i \in \{0, ..., m\}$ by $\theta_i^k,\ k \in \{0, ..., m\}$. Note that θ_i^k are the costs which were then perturbed in Sec. 2.2. Typically, an unstructured classifier with parameters $\boldsymbol{\eta}$ is trained on a training set of input/output pairs, $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1,...,N} =: (\mathbf{X}, \mathbf{y})$, and the (mean) prediction $p(Y_i = k|\mathbf{x}_i, \mathbf{X}, \mathbf{y}, \boldsymbol{\eta})$ from local features $\mathbf{x}_i$ determines these costs θ_i^k for variable Y_i as

$$\theta_i^k = -w\log\left(p(Y_i = k|\mathbf{x}_i, \mathbf{X}, \mathbf{y}, \boldsymbol{\eta})\right), \qquad (4)$$

where w is a parameter of the graphical model. For a Gaussian process classifier, reviewed in the next paragraph, θ_i^k may either come from the mean of the predictive distribution in Eq. (6) specific for variable Y_i, or drawing M samples in Eq. (6) yields $\theta_{i,l}^k,\ l \in \{1, ..., M\}$. Drawing samples from the predictive distributions of each variable in the graphical model yields a multitude of perturbations of the graphical model as in Eq. (3) with $\boldsymbol{\varepsilon}$ approximated from the underlying Gaussian processes. Their non-parametric nature

[2] Gaussian process regression may also be used directly for the generation of unary potentials. We refer the reader to [13] for an introduction to Gaussian processes.

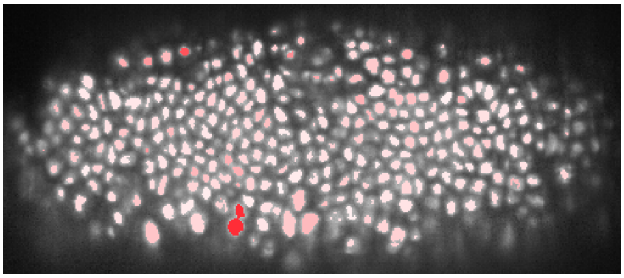


Fig. 2. One slice of the 3D+t *Drosophila* sequence: The more opaque the red color the higher the *classifier uncertainty* that the connected component contains exactly one cell.

avoids choosing a value for the perturbation variance, and the unary potentials in the tracking-by-assignment model are perturbed according to the predicted second-order uncertainty in the measured data, in other words the data term is perturbed locally w.r.t. the predictive uncertainty from the Gaussian process classifier. An example for the uncertainty of the cell detection classifier on our *Drosophila* dataset is shown in Fig. 2. Similarly to Sec. 2.2, these perturbed graphical models can finally be solved independently and distributedly to approximate samples from the Gibbs distribution.

Gaussian Process Classification Gaussian processes (GPs), denoted by $\mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, are stochastic processes each realization of which defines a multi-variate joint Gaussian distribution [13]. The mean function $m(\mathbf{x})$ is typically assumed to be constant zero whereas the covariance function $k(\mathbf{x}, \mathbf{x}')$ may be any Mercer kernel. GP classification aims at finding a mapping from an input space to a categorical output space given unstructured training data $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ and it can be shown [13] that the latent predictive distribution for query point $\mathbf{x}_*$ is given as

$$f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{k}_*^\top K^{-1}\mathbf{y}, k_{**} - \mathbf{k}_*^\top K^{-1}\mathbf{k}_*), \quad (5)$$

where $\mathcal{N}(\cdot,\cdot)$ denotes the Gaussian distribution, $[K]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $[\mathbf{k}_*]_i = k(\mathbf{x}_*, \mathbf{x}_i)$ with $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$, $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$, and the parameters of $k(\cdot,\cdot)$ are the hyperparameters $\boldsymbol{\eta}$ of the GP. For *binary* classification with class label $y \in \{0, 1\}$, the prediction for the latent f_* in Eq. (5) is "squashed" through a sigmoid function:

$$\begin{aligned} p(Y_* = +1|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \boldsymbol{\eta}) &= \mathbb{E}\left(\text{sigmoid}\left(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \boldsymbol{\eta}\right)\right) \\ &\approx \text{sigmoid}\left(\mathbb{E}(f_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}, \boldsymbol{\eta})\right). \end{aligned} \quad (6)$$

Applying the sigmoid function on multiple samples of f_* rather than its expected value yields samples from the non-Gaussian class-conditional distribution. In our experiments, we choose the error function as the sigmoid function. To train a GP classifier, the marginal likelihood of the model is optimized. Approximations need to be made due to the non-Gaussian likelihood function utilized in GP classification, see [13] for more details.

Events presented	0	10	50	100	150	250	500
GP sampling (5x)	68.3	69.3	71.8	74.9	78.5	80.5	-
GP sampling (20x)	68.3	70.0	75.3	77.9	81.9	84.2	87.2
GP sampling (50x)	68.3	69.8	75.0	79.8	82.4	84.1	86.6
GP sampling (100x)	68.3	69.8	75.1	80.0	82.5	84.1	86.6
Perturb-and-MAP (5x)	68.3	69.3	72.7	-	-	-	-
Perturb-and-MAP (20x)	68.3	69.1	73.4	76.4	80.8	-	-
Perturb-and-MAP (50x)	68.3	69.1	73.4	76.6	79.7	-	-
Perturb-and-MAP (100x)	68.3	69.3	73.3	76.6	79.1	-	-

Table 1. **Division accuracy** after N events presented to the user (and corrected if necessary); "-" indicates that the method did not generate enough events with positive uncertainty. The number "(tx)" is the number of samples drawn.

To extend the binary GP classifier to multiple classes, uncorrelated latent functions are assumed, one per class. The hyperparameters of these latent processes may either be learned jointly using the softmax [14] or independently as a set of binary classifiers in a one-vs.-all scheme. We use the latter in our experiments.

3. UNCERTAINTY IN CELL TRACKING

The goal in cell tracking is to find full cell lineages which, in turn, decompose over frame-to-frame *events*, assignments either between two or three cell candidates, denoted as *moves* or cell *divisions*, respectively. For manual proof-reading, we want to guide the user to the most uncertain events, while assuming that ground truth is not available for the test dataset. We propose to estimate the event uncertainty (also referred to as event *ambiguity*) by analyzing the proposal solutions drawn from the tracking-by-assignment model, as outlined in the previous sections. Both events, *moves* and *divisions* are modeled as random variables Y_i^{move} and Y_i^{div} in the graphical model proposed in [5], whose realizations in the nth proposal are denoted as $y_{i,n}^{\text{move}}$ and $y_{i,n}^{\text{div}}$; for brevity we omit the event type. To reason about the reliability of a selected labeling $\mathbf{y}_*$, we introduce the *labeling uncertainty measure* as

$$U(Y_i = y_{i,*}) = 1 - \frac{1}{N}\sum_{n=1}^{N} \mathbb{1}[y_{i,n} = y_{i,*}], \quad (7)$$

which normalizes the number of votes amongst the solutions of the perturbed models *against* labeling $\mathbf{y}_*$. In other words, the labeling uncertainty finds the most uncertain decisions, *i.e.* the predictions in the selected proposal which are likely to be wrong, compared to the labelings of the other proposals. Note that in order to guarantee a consistent solution, it is not possible to round the approximated marginal distributions for an "averaged" distribution. Instead, one proposal solution (*e.g.* the MAP solution of the unperturbed model) needs to be chosen to guarantee that no consistency constraint in Eq. (1) is violated. In the GP based sampling approach, this unperturbed proposal solution corresponds to the mean predictions.

4. EXPERIMENTS AND RESULTS

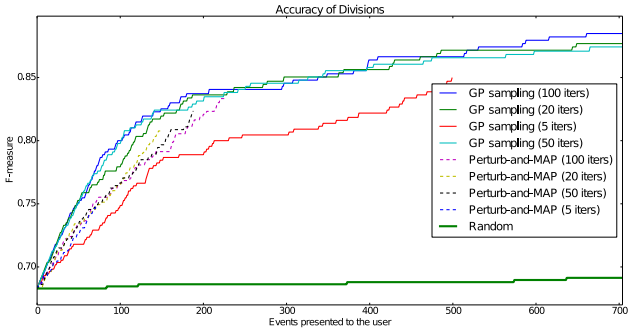
We evaluate the usefulness of the proposed uncertainty measures on a challenging 3D+t sequence from a developing *Drosophila* embryo taken from [5]. For this purpose, we adapt the tracking-by-assignment model from [5] (we set $m = 2$) and use Gaussian process classifiers for the detection, transition, and division priors using the GP implementations in the GPy package [15]. The parameters of the graphical model are trained using the structured max-margin learning implementation from [16]. We used the first 20 frames as training set and the remaining 80 frames as test set.

As performance measure, we choose the f-measure ($= \frac{2 \cdot \text{True Positives (TP)}}{2 \cdot \text{TP} + \text{False Negatives} + \text{False Positives}}$) since it balances the number of true events found (recall) and the rate of true predictions (precision). Note that although tracking-by-assignment is a two-stage model, we here only evaluate the second stage, namely the tracking model, since the ambiguity measures proposed only apply to that; we hence only compare to the ground truth actually seen by the tracking model. The numbers may deviate from those reported in [5] since a separate test set and different classifiers are used. We want to show in this evaluation that presenting the most ambiguous events to the user and asking for corrections may quickly boost performance. The samples are drawn from the graphical model offline, so no expensive human interaction time is needed during the generation of the uncertainty values. In our experiments, drawing a sample took ≈ 30 seconds on a contemporary workstation.

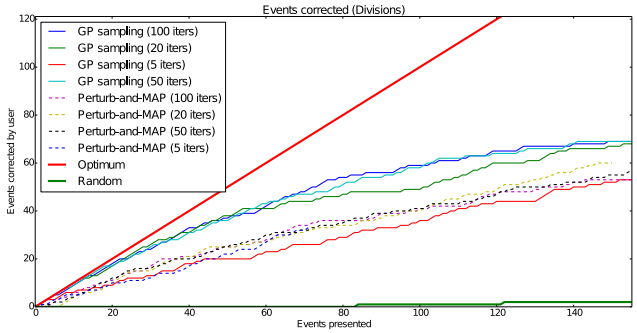
Our experimental setup is as follows: We iteratively present the event with the highest labeling uncertainty according to Eq. (7) to an oracle which corrects the event if necessary. For both divisions and moves, most of the highly uncertain events must be corrected by the oracle as shown in Fig. 3(b) and (d), and less corrections need to be made for less uncertain events. Our baseline is a random suggestion of events. Both presented sampling methods perform similarly well with a slight advantage for the GP sampling approach for division events. As expected, the more samples are drawn, the more meaningful are the uncertainty measures. As shown in Table 1 and Fig. 3(a), manually correcting the divisions proposed by the GP sampling method can boost the performance from 68.3% to 80.0% already after 100 presented examples. The f-measure of *moves* (*cf.* Fig. 3(c)) increases more slowly due to the large number of move events in the dataset.

5. CONCLUSIONS

We propose to apply Perturb-and-MAP [7] random fields and a novel Gaussian process sampling procedure to cell tracking models for the generation of proposal solutions. A labeling uncertainty measure is derived from the marginal predictions across these samples, which guides users to most ambiguous labels. In the future, we will try to fuse the proposal solutions (and the corrections made by the user) to consistent solutions.

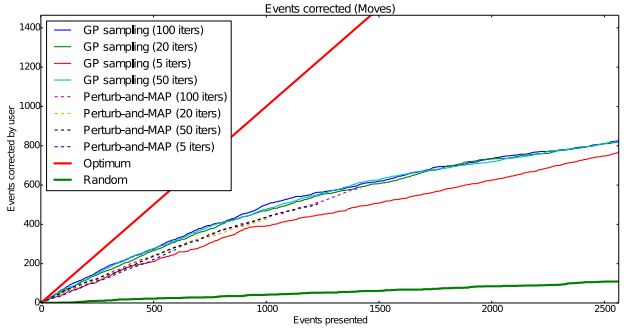


(a) Division: F-Measure



(b) Division: Corrected Events

(c) Move: F-Measure



(d) Move: Corrected Events

Fig. 3. Comparison of sampling methods and the resulting labeling uncertainties. The curves terminate prematurely if all remaining uncertainty estimates are zero. "iters" stands for the number of samples generated from one model. See main text for details.

6. REFERENCES

[1] Erik Meijering, Oleh Dzyubachyk, Ihor Smal, and Wiggert A van Cappellen, "Tracking in cell and developmental biology," in *Seminars in cell & developmental biology*. Elsevier, 2009, vol. 20, pp. 894–902.

[2] Omar Al-kofahi, Richard J Radke, Susan K Goderie, Qin Shen, Sally Temple, and Badrinath Roysam, "Automated Cell Lineage Construction: A Rapid Method to Analyze Clonal Development Established with Murine Neural Progenitor Cells," pp. 327–335, February 2006.

[3] Dirk R. Padfield, Jens Rittscher, and Badrinath Roysam, "Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis," *Medical Image Analysis*, vol. 15, no. 4, pp. 650–668, 2011.

[4] Ryoma Bise, Zhaozheng Yin, and Takeo Kanade, "Reliable Cell Tracking by Global Data Association," in *ISBI*, 2011.

[5] Martin Schiegg, Philipp Hanslovsky, Bernhard X Kausler, Lars Hufnagel, and Fred A Hamprecht, "Conservation tracking," in *ICCV*, 2013.

[6] Fernando Amat, William Lemon, Daniel P Mossing, Katie McDole, Yinan Wan, Kristin Branson, Eugene W Myers, and Philipp J Keller, "Fast, accurate reconstruction of cell lineages from large-scale fluorescence microscopy data," *Nature methods*, 2014.

[7] George Papandreou and Alan L Yuille, "Perturb-and-MAP random fields: Using discrete optimization to learn and sample from energy models," in *ICCV*, 2011.

[8] Florian Jug, Tobias Pietzsch, Dagmar Kainmüller, Jan Funke, Matthias Kaiser, Erik van Nimwegen, Carsten Rother, and Gene Myers, "Optimal joint segmentation and tracking of Escherichia coli in the mother machine," in *MICCAI-BAMBI*. 2014.

[9] Subhransu Maji, Tamir Hazan, and Tommi Jaakkola, "Efficient boundary annotation using random maximum a-posteriori perturbations," in *AISTATS*, 2014.

[10] Bret Kragel, Shawn Herman, and Nick Roseveare, "A comparison of methods for estimating track-to-track assignment probabilities," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 1870–1888, 2012.

[11] Daniel H. Rapoport, Tim Becker, Amir Madany Mamlouk, Simone Schicktanz, and Charli Kruse, "A novel validation algorithm allows for automated cell tracking and the extraction of biologically meaningful parameters.," *PloS one*, vol. 6, no. 11, pp. e27315, Jan. 2011.

[12] Xinghua Lou, Martin Schiegg, and Fred A. Hamprecht, "Active structured learning for cell tracking: Algorithm, framework, and usability," *IEEE Trans. of Medical Imaging*, 2014.

[13] Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[14] Christopher K.I. Williams and David Barber, "Bayesian classification with Gaussian processes," *PAMI*, vol. 20, no. 12, 1998.

[15] the GPy authors, "GPy: A Gaussian process framework in Python," `https://github.com/SheffieldML/GPy`, 2014.

[16] Jan Funke, *Automatic Neuron Reconstruction from Anisotropic Electron Microscopy Volumes*, Ph.D. thesis, Institute of Neuroinformatics, ETH Zurich.