Spotiflow: accurate and efficient spot detection for imaging-based spatial transcriptomics with stereographic flow regression

Albert Dominguez Mantes[1,2], Antonio Herrera[2], Irina Khven[2], Anjalie Schlaeppi[3], Eftychia Kyriacou[4], Georgios Tsissios[4], Can Aztekin[4], Joachim Lingner[4], Gioele La Manno[✉,2] and Martin Weigert[✉,1]

[1] *Institute of Bioengineering, School of Life Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland*
[2] *Brain Mind Institute, School of Life Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland*
[3] *Bioimaging and Optics Platform, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland*
[4] *Swiss Institute for Experimental Cancer Research, School of Life Sciences, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland*

The identification of spot-like structures in large and noisy microscopy images is an important task in many life science techniques, and it is essential to their quantitative performance. For example, imaging-based spatial transcriptomics (iST) methods rely critically on the accurate detection of millions of transcripts in low signal-to-noise ratio (SNR) images. While recent developments in computer vision have revolutionized many bioimage tasks, currently adopted spot detection approaches for iST still rely on classical signal processing methods that are fragile and require manually tuning. In this work we introduce Spotiflow, a deep-learning method that casts the spot-detection problem as a multiscale stereographic flow regression problem that yields subpixel-accurate localizations. Spotiflow is robust to different noise conditions and generalizes across different chemistries while being up to an order-of-magnitude more time and memory efficient than commonly used methods. We show the efficacy of Spotiflow by comprehensive quantitative comparisons against other methods on a variety of datasets and demonstrate the impact of its increased accuracy on the biological conclusions drawn from iST and live imaging experiments. Spotiflow is available as an easy-to-use Python library as well as a napari plugin at `http://www.github.com/weigertlab/spotiflow`**.**

✉ Correspondence to:
M.W. (martin.weigert@epfl.ch / @martweig / epfl.ch/labs/weigert-lab)
G.L.M. (gioele.lamanno@epfl.ch / @GioeleLaManno / gioelelamanno.com)

Introduction

Many methods in the life sciences generate images in which the detection and localization of spot-like objects is a crucial first analysis step for more complex downstream tasks, a problem commonly referred to as *spot detection* [1–4]. While spot detection has been the computational basis of many methods in genomics over the last decades [5, 6], the advent of imaging-based spatial transcriptomics (*iST*) has recently brought this problem to a significantly more challenging and computationally demanding domain [7] (Fig. 1a). In iST, RNA molecules are located in situ in large tissue sections during sequential imaging cycles, to generate gene expression maps at subcellular resolution [8–10]. Popular iST techniques such as MERFISH [8], seqFISH [10] or HybISS [9] require detection of millions of spots in gigabyte-sized images with high accuracy, high sensitivity, and computational efficiency. Due to the preservation of the native tissue context, any spot detection method has to address multiple imaging challenges such as autofluorescence background, aspecific probe binding, or inhomogeneous spot density (*cf.* Supp. Video 1). High sensitivity and accuracy are particularly important as for most iST methods transcript identity is combinatorially encoded in the sequence of multiple multi-channel images generated across different imaging rounds [8, 9]. As a result, suboptimal spot detection performance in one channel or imaging round can cause a significant drop in sensitivity and transcript identity misattribution [11].

Commonly used spot detection pipelines for iST often rely on classical threshold-based methods such as Laplacian-of-Gaussian (LoG) [12, 13] or radial symmetry [14]. While these approaches perform well on simulated or relatively clean data, they often struggle with realistic images that exhibit artifacts, autofluorescence, and varying contrast (*cf.* Results). While few deep learning-based methods have been proposed for this task [15–18], they are often hard to use and don't provide subpixel accuracy, with the notable exception of [17]. Consequently, currently used iST spot detection methods often lack robustness to challenging image conditions, are computional inefficient for large images, and require manual parameter (e.g. threshold) tuning for every channel and imaging round, which limits their applicability in large-scale iST experiments.

Here we introduce Spotiflow, a deep learning-based, threshold-agnostic, and subpixel-accurate spot detection method that outperforms other commonly used methods on a variety of iST and non-iST modalities while being up to an order-of-magnitude more time and memory efficient. Spotiflow is trained to predict *multiscale Gaussian heatmaps* and exploits a novel *stereographic flow* regression task from which sub-pixel accurate detections are obtained (Fig. 1b,c). Our method generalizes well to unseen samples and removes the requirement of manual threshold tuning in typical end-to-end iST workflows. Spotiflow is available as an easy-to-use Python library as well as a napari [19] plugin (*cf.* Supp. Video 2).

Results

To compute spot coordinates from a given microscopy image, Spotiflow uses a convolutional neural network (U-Net [20]) that is trained to predict two distinct but synergetic targets: *Gaussian heatmaps* and the *stereographic flow*(*cf.* Fig. 1b, Supp. Fig. 1). The first target, *Gaussian heatmaps*[21], are real-valued images of different resolutions in which each pixel can be interpreted as the probability of that position being a spot center (*cf.* Fig. 1c, Supp. Fig. 1, Supp. Note 3.1). We predict a multiscale hierarchy of heatmaps by processing their respective network decoder feature maps, which jointly contributes to the optimized training loss. We found this approach to be beneficial for training convergence, especially when only few spots are present (*cf.* Methods, Supp. Fig. 1, Supp. Note 1). The second target, which we denote *stereographic flow*, is a problem-adapted representation of the closest-spot vector field that, for every position, points to the closest spot. The stereographic flow is defined as the inverse stereographic projection of the two-dimensional local offset vector field in $\mathbb{R}^2$ onto the unit three-dimensional sphere S^2. Crucially, this embedding maps all offsets for points far away from spot locations to a common value (the south pole of the unit sphere) therefore avoiding the problem of indeterminate offset prediction for distant locations (*cf.* Fig. 1c, Methods, Supp. Fig. 2, Supp. Note 3.1, Supp. Video 3). To produce the final spot coordinates from a given prediction, we use the peaks of the highest-resolution heatmap to obtain preliminary spot locations, which we refine with the inverted stereographic flow (*cf.* Methods), achieving subpixel precision and substantially lower localization errors (*cf.* Supp. Table 1).

We systematically assessed the performance of Spotiflow on multiple datasets in comparison with other commonly used methods. Specifically, we compared against the Laplacian-of-Gaussian (LoG/starfish) implementation used in the popular *iST* framework *starfish* [12], Big-FISH [13], the radial symmetry based method RS-FISH [14], and the deep learning-based method deepBlink [17]. We first generated two synthetic datasets of diffraction-limited spots (*cf.* Fig. 1d): one using a simple Gaussian PSF model (*synthetic-simple*), and another using a more realistic image formation model including autofluorescence and optical aberrations (*synthetic-complex*, Methods). We found that on *synthetic-simple* all methods achieved close to perfect scores (F_1-score of 0.967–0.995), which is expected due to the limited complexity of the simulated images (Fig. 1d, Supp. Table 3). However, on the more realistic *synthetic-complex* dataset, classical methods showed a substantial performance drop ($F_1 = 0.758 - 0.836$) while Spotiflow achieved the best detection accuracy ($F_1 = 0.929$) followed by the other deep-learning based method (deepBlink, $F_1 = 0.915$). This demonstrates the advantages of our learned approach for more complex datasets and highlights the limitations of overly simplistic simulations in benchmark scenarios. Similarly, when generating images at different noise conditions and spot densities, we found Spotiflow to consistently outperform other methods, demonstrating its effectiveness in adverse imaging conditions (Supp. Fig. 3).

We next assessed Spotiflow on several publicly available and in-house generated manual annotated datasets from multiple iST modalities (MERFISH, HybISS, smFISH, *cf.* Supp. Table 2, Supp. Fig. 4). We observed that Spotiflow again outperforms all other methods, including deepBlink, achieving a consistently high detection rate and localization accuracy on all modalities (*cf.* Fig. 1d, Supp. Fig. 5, Supp. Note 3.3). The performance difference to threshold-based classical methods is particularly prominent for the HybISS and MERFISH datasets, which contain substantial background signal ($F_1 = 0.861/0.796$ *vs.* $0.531/0.790$ for *e.g.* LoG-starfish). Interestingly, a Spotiflow model jointly trained on all diverse datasets (*general*) achieves an almost equal performance to models trained on individual datasets, demonstrating the inherent capacity of Spotiflow to capture diverse image characteristics in a single model. We evaluated the utility of Spotiflow for non-iST modalities by annotating two datasets from single frames of live-cell recordings of HeLa cells with labeled telomeres and the telomeric repeat *TERRA* (*cf.* Methods). As before, Spotiflow outperforms all other methods both in detection quality and localization accuracy (*cf.* Fig. 1d), demonstrating the general applicability of our method beyond the iST domain. In relation to the training data requirements of the model, we observed that fine-tuning based on synthetic data can substantially reduce the annotation requirement for novel out-domain datasets, quickly approximating the accuracy of our benchmark training dataset composed of hundreds of annotated 512×512 images. Specifically, when we fine-tuned a Spotiflow model initially pre-trained on *synthetic-complex* with an incrementally increasing number of out-domain training images from the live-cell dataset *Terra*, we found that already as few as four training images sufficed to achieve good accuracy (F_1-score 0.738 *vs.* 0.174 when training from scratch, Fig. 1f, Supp. Fig. 6). This result underscores the efficiency of Spotiflow models in adapting to different modalities and out-of-distribution samples with minimal annotation, facilitating rapid adoption by end-users.

We next investigated the generalizablility of pre-trained models to the variability in sample types, which can encompass differing signal-to-noise ratios, unique artifacts, and distinct background features. We trained a Spotiflow model on HybISS-processed mouse embryonic brain sections and applied it on a variety of out-of-distribution HybISS samples originating from different tissues and probesets (mouse embryonic limb, frog tadpole developing limb, mouse gastruloid, radial glia progenitor cell cultures). Even though these images exhibit noticeably different structures with varying backgrounds and contrast compared to the training images, we found that the pre-trained model yielded qualitatively excellent transcript detection results without the need of any threshold-tuning (Fig. 2a,b and Supp. Fig. 7).

We then assessed the impact of the increased robustness and accuracy of Spotiflow for a full end-to-end iST experiment (*cf.* Methods) by using a starfish gene decoding pipeline where we swapped the spot detection component from the default LoG detector to Spotiflow. We processed sections of developing mouse brains at different timepoints, E12.5 (*cf.* Fig. 2c) and E13.5 (*cf.* Fig. 2d), using HybISS to spatially resolve 199 genes involved in neurodevelopment (*cf.* Methods). The resulting gene expression maps obtained with Spotiflow show a gene-dependent spatial pattern that is consistent with previous results (*cf.* Fig. 2c, Fig. 2d, Supp. Fig. 8). While for intensity-based methods (*e.g.* LoG/starfish) the quality of the obtained gene expression maps is highly sensitive to the used threshold and thus requires channel-specific threshold choices, Spotiflow is threshold agnostic and does not require any manual tuning (*cf.* Fig. 2d, Supp. Fig. 8). In addition, we found that in this end-to-end *iST* setting, Spotiflow is an order of magnitude more time and memory efficient than the default starfish pipeline, especially for large images (*cf.* Fig. 2e,f).

We hypothesized that the content-awareness of Spotiflow could be leveraged to solve tasks which are infeasible for classical spot detection methods. To explore this we examined whether Spotiflow could effectively differentiate between transcript-derived spots and spot-like patterned autofluorescence structures, such as those from lipofuscin [22, 23], that often render data collected from adult brain unusable. Applying Spotiflow to a HybISS-processed adult mouse brain section with a specific bootstrapping scheme (*cf.* Methods, Supp. Fig. 9) we achieved a 3x decrease in the number of autofluorescent spots detected in one channel (Supp. Fig. 9), thus substantially reducing the amount of noise in the expression maps obtained from iST. This is particularly notable considering that such a discrimination task is challenging even to experienced human annotators.

Finally, we demonstrate Spotiflow's flexibility to accurately detect spot-like structures in fluorescence microscopy images outside the iST domain. Concretely, we consider single-molecule detection and tracking of both telomeres and noncoding RNA molecules (*TERRA*) in live-cell time lapses of HeLa cells (*cf.* Methods, Fig. 2g). These images present different challenges compared to iST images, such as photobleaching causing the temporal decrease of image contrast, non-specific dot-like structures inside the cell nucleus, and unspecific signal that can lead to erroneous and unrealistic short tracks (Fig. 2g). After detecting spots with Spotiflow, we tracked them using TrackMate [24] (*cf.* Methods). We also detected spots with deepBlink and tracked them for comparison. For both telomeres and *TERRA*, the robustness of Spotiflow's detections at changing imaging conditions led to longer, more consistent (gap-free) tracks compared to deepBlink (*cf.* Fig. 2h, Fig. 2i, Supp. Fig. 10, Supp. Video 4), demonstrating the significant impact on the estimates of biological parameters which a more accurate molecule detection method is able to provide.

Training a Spotiflow model is fast (∼1h on a single GPU) and our implementation based on PyTorch[25] is an order-of-magnitude faster and over three times more memory efficient than commonly used methods especially for larger images, with *e.g.* prediction time of 80s for an image of size $32k \times 32k$ *vs.* 1000s for LoG/starfish (*cf.* Fig. 1e, Supp. Note 3.4). To facilitate the adoption by end-users, we provide extensive documentation, distribute Spotiflow as an easy-to-use napari[19] plugin, and provide several pre-trained models that can be used out-of-the-box for a variety of iST modalities (*cf.* Supp. Video 2).

Discussion

In summary, Spotiflow delivers high-quality detections across a variety of iST modalities and surpasses both commonly used and recently proposed methods. Notably we demonstrated on a diverse set of real benchmark datasets that assessments relying solely on simple synthetic data are insufficient to evaluate performance regarding real-world applications. Spotiflow generalizes well to out-of-distribution samples, does not need any manual tuning in end-to-end iST experiments, and is an order of magnitude more efficient than other methods to process whole samples. One limitation of Spotiflow is it being currently tailored for two-dimensional data. However, both multiscale Gaussian heatmaps and the stereographic flow can be naturally extended to n dimensions, and thus Spotiflow can be extended to detect spots in n-dimensional volumes, including 3D. Finally, our live-cell imaging experiments indicate Spotiflow's flexibility to various fluorescence microscopy modalities, and we foresee its broad utility to other imaging-based methods where localized structures need to be detected. We finally anticipate that the presented *stereographic flow* will impact other areas where prediction of dense vector fields has successfully been applied (*e.g*. cell segmentation [26, 27]).

Acknowledgements

We thank members of the Weigert and La Manno labs as well as Lars Borm (KU Leuven) for their feedback and discussions of the project. We would also like to thank the EPFL BioImaging & Optics Core Facility (BIOP) and the EPFL Histology Core Facility for their assistance in imaging and sample preparation. M.W. was supported by the ELISIR program of the EPFL School of Life Sciences and by generous funding from CARIGEST SA.

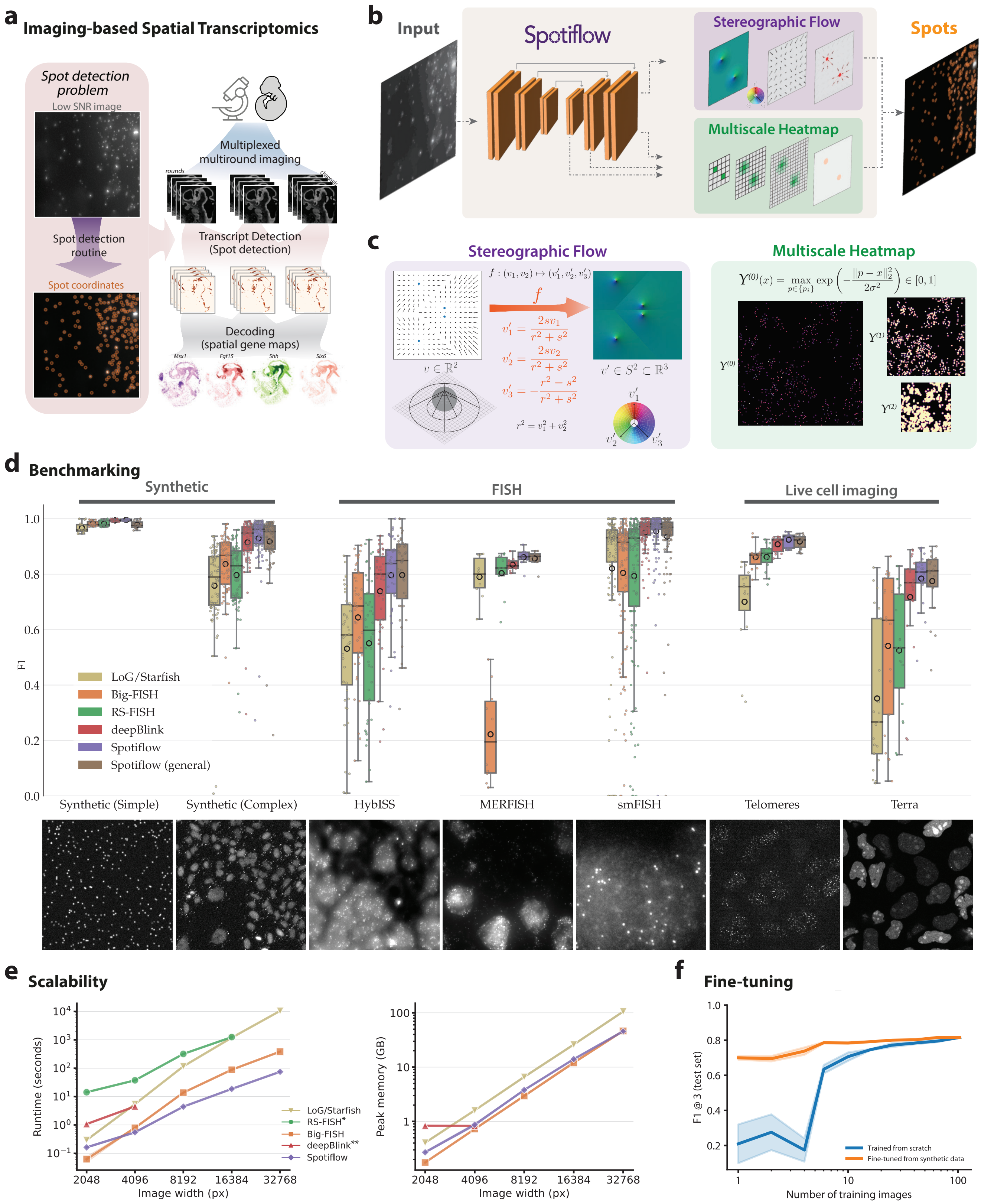

Figure 1: Fast, scalable, and accurate fluorescence spot detection with Spotiflow. a) Depiction of a common processing pipeline for imaging-based spatial transcriptomics (iST) data, in which spot detection is a critical step. **b)** Spotiflow is trained to detect spots from microscopy images via two different synergic tasks, *multiscale heatmap regression* and *stereographic flow regression*.**c)** The ground truth objects to be regressed are computed from point annotations $\{p_k\}$. First, a full-resolution Gaussian heatmap $Y^{(0)}$ is obtained by generating isotropic Gaussian distributions of variance σ^2 centered at spot locations. This Gaussian heatmap is further processed to obtain lower resolution versions, yielding *multiscale heatmaps* $Y^{(l)}$, which are all regressed. Second, a local vector field $V = \{v_{ij}\}$ is built by placing a vector directed to the closest spot center at every pixel of the image. We obtain the *stereographic flow* $V' = \{v_{ij}'\}$ by computing, position-wise, the inverse stereographic projection f of the local vector field. **d)** Benchmarking of spot detection methods on different datasets, grouped by their modality (*Synthetic, FISH, Live cell imaging*). Shown is the distribution of F_1 scores per image in the test set of every dataset (higher is better, *cf.* Supp. Note 3.2). Each method was trained and tested individually in each dataset except Spotiflow *(general)*, which was trained on all datasets. A sample training image is depicted under each dataset. **e)** Runtime (top) and memory (bottom) assessment for different methods at different image sizes. Parameters of each method were calibrated so that the amount of detections were in the same order of magnitude. * RS-FISH not shown for sizes >32k due to Java size-related limitations. RS-FISH memory was not profiled as the implementation is not in Python. ** deepBlink could not be run for sizes > 4k due to GPU memory limitations. **f)** F_1 score on live-cell dataset *Terra* after fine-tuning a Spotiflow model pre-trained on *synthetic-complex* with an incrementally increasing number of out-domain training images from *Terra*.

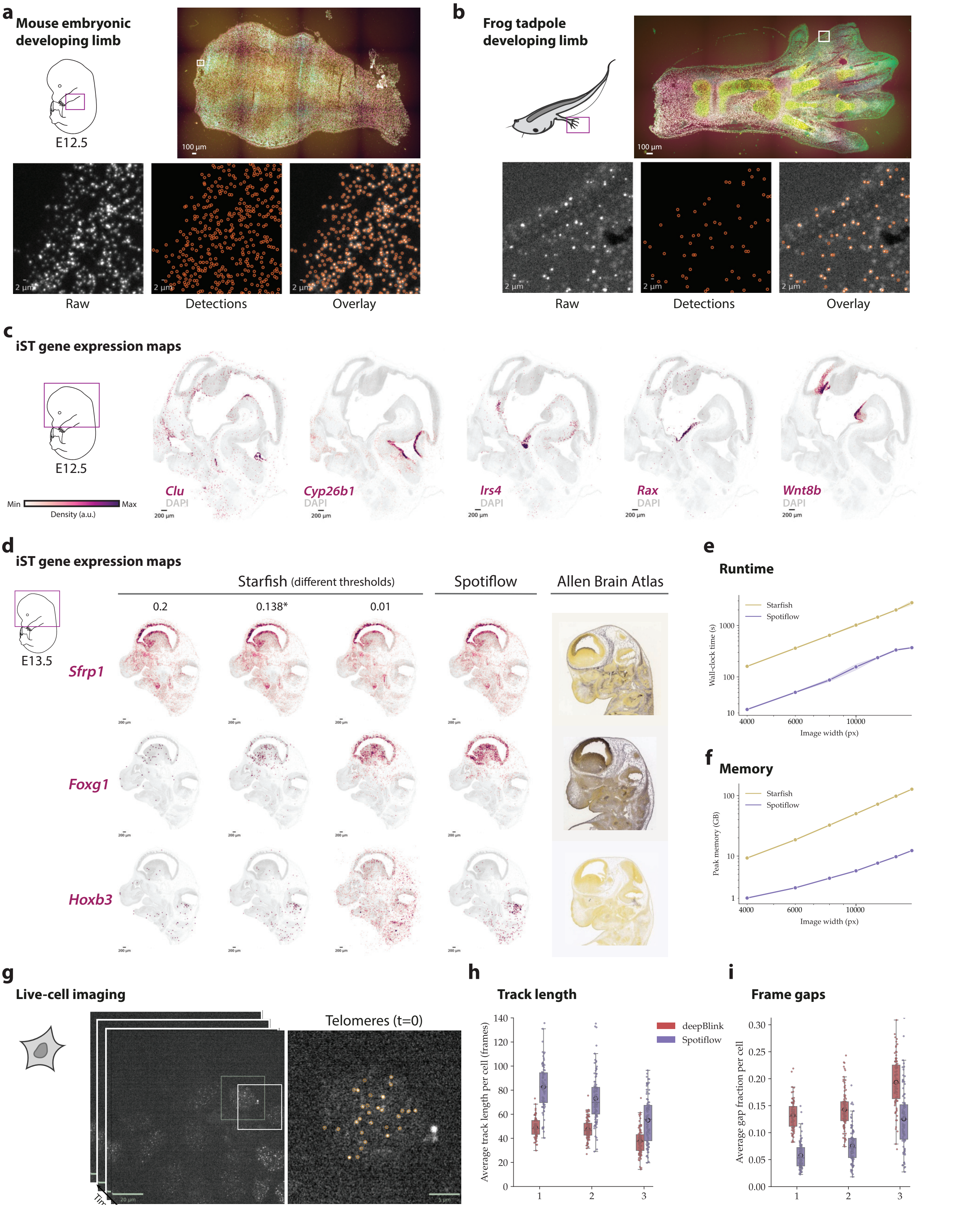

Figure 2: Application of Spotiflow in a variety of biological settings. a, b) Predictions of a pretrained Spotiflow model on two out-of-distribution samples, a mouse embryonic developing limb (a) and a frog tadpole developing limb (b). The Spotiflow model was trained on the *HybISS* dataset, consisting only of images of embryonic mouse brain embryos. **c)** Gene expression maps based on Spotiflow of an E12.5 mouse embryo brain processed using HybISS. Five different genes (*Clu, Cyp26b1, Irs4, Rax, Wnt8b*) involved in neurodevelopment overlaid on the DAPI channel are displayed. The Spotiflow model used was trained on the *HybISS* dataset. **d)** Comparison of gene expression maps based on Spotiflow *vs.* the default LoG Starfish Starfish [12] detector of an E12.5 mouse embryo brain processed using HybISS. Depicted are results for three different genes (*Sfrp, Foxg1, Hoxb3*). The Starfish detector is run at three different thresholds (0.2, 0.01 and 0.138, the latter being the optimal on the *HybISS* training dataset) as well as the Spotiflow model trained on *HybISS* (using the default threshold). The last column contains an ISH reference of similar sections from the Allen Brain Atlas (ISH) for the three depicted genes. **e, f)** Runtime (e) and memory (f) assessment of both methods in an end-to-end setting. Depicted are wall-clock time (e) and peak CPU memory usage (f). **g)** Live-cell acquisition of HeLa cells with labeled Telomeres (orange). **h)** Quantification of Telomere track length on three different experiments using deepBlink and Spotiflow to detect spots per frame which are then tracked using TrackMate [24]. Telomeres are expected to be stable throughout the movie, thus longer tracks are expected. **i)** Quantification of number of frames where a track does not contain any detected spot (gap fractions). Smaller gap fractions indicate more stable detections.

Methods

Spotiflow

Architecture overview

Given an input image and the corresponding spot center annotations, a U-Net [20] is trained to predict two different sets of outputs which encode the location of spots in the image: first *multiscale probability heatmaps*, and second, the *stereographic flow* (*cf.* Fig. 1b, Supp. Fig. 1). During training, the overall loss function optimized is

$$\mathcal{L} = -\left(1 + \lambda \mathbb{1}_{\text{Spot}}\right)\left(\mathcal{L}_{heat} + \mathcal{L}_{flow}\right) \qquad (1)$$

where $\mathcal{L}_{heat}$ is the multiscale heatmap loss (see below), $\mathcal{L}_{flow}$ the stereographic flow loss (see below) and $\mathbb{1}_{\text{Spot}}$ is a pixel-wise indicator function which takes the value 1 if the pixel is very close to a spot location (closer than some cutoff distance ε) and 0 otherwise. $\lambda \in \mathbb{R}$ is used to increase the loss contribution near spot centers (*cf.* Supp. Note 3.1). We set $\lambda = 10$, $\varepsilon = 5$px when training all our models.

Multiscale heatmap regression

Let $X \in \mathbb{R}^{w \times h}$ denote the input image and $\{p_k\}$ with $p_k \in \mathbb{R}^2$ denote the ground truth spot center annotations. We first build a full resolution probability heatmap $Y \in \mathbb{R}^{w \times h}$ by generating a Gaussian distribution of variance σ^2 centered at every spot so that the probability map exponentially decays around the annotated centre.

$$Y^{(0)}(x) = \max_{p \in \{p_k\}} \exp\left(-\frac{\|p - x\|_2^2}{2\sigma^2}\right) \in [0, 1] \qquad (2)$$

Note that instead of summing the individual Gaussian distributions, we take the maximum value at each pixel to create sharp boundaries between spots.

We further generate the heatmaps at L different resolution levels, where level $l = 0$ denotes the highest resolution and $l = L - 1$ the lowest. In order to generate a heatmap at resolution level l ($Y^{(l)}$) from $l - 1$ ($Y^{(l-1)}$), we apply max pooling (with a downsampling factor of 2) to $Y^{(l)}$ and then process the result with a Gaussian filter of variance σ_d^2 with a scaling prefactor of $2\pi\sigma_d^2$, which effectively increases the variance of distributions and ensures the dynamic range of the heatmap is in the interval $[0, 1]$ (*cf.* Supp. Fig. 1).

The U-Net backbone is then trained to regress all heatmaps $Y = \{Y^{(l)}\}_{l=0}^{L-1}$ at the different scales (*multiscale heatmap regression*, *cf.* Fig. 1c, Supp. Fig. 1). We achieve this by adding a loss term at different stages in the decoder whose size correspond directly to the target to be regressed. More specifically, let $D^{(i)}$, $i \in [1, L]$, denote the feature maps at the output of the i-th decoder stage in the U-Net. We process $D^{(i)}$ with a lightweight convolutional module to compute the prediction $\hat{Y}^{(L-i)}$ (*cf.* Supp. Fig. 1). A pixel-wise loss term is then computed between the ground truth heatmap $Y^{(l)}$ and the prediction $\hat{Y}^{(l)}$ at every resolution level l with the binary cross-entropy loss. We then aggregate this terms in the overall objective function for the multiscale heatmap, $\mathcal{L}_{heat}$:

$$\mathcal{L}_{heat}(Y, \hat{Y}) = \sum_{l=0}^{L-1} \frac{1}{2^l}\left[\hat{Y}^{(l)} \log\left(Y^{(l)}\right) + \left(1 - \hat{Y}^{(l)}\right) \log\left(1 - Y^{(l)}\right)\right] \qquad (3)$$

Stereographic flow

For each pixel $(i, j) \in \mathbb{Z}^2$ of the image X, we first define a *local vector field* $V = \{v_{ij}\} = \{(v_x, v_y) \in \mathbb{R}^2\}$ given by the vector from the pixel to the nearest ground truth spot (*cf.* Fig. 1b, Supp. Fig. 2). To induce stability and improve modelling at points far from spot locations, we make use of a scaled *inverse stereographic projection* $f : (v_x, v_y) \in \mathbb{R}^2 \to (v'_x, v'_y, v'_z) \in S^2$ defined as

$$v'_x = \frac{2sv_x}{r^2 + s^2}, \quad v'_y = \frac{2sv_y}{r^2 + s^2}, \quad v'_z = -\frac{r^2 - s^2}{r^2 + s^2} \qquad (4)$$

$$\text{with } r^2 = v_x^2 + v_y^2 \quad \text{and} \quad {v'}_x^2 + {v'}_y^2 + {v'}_z^2 = 1$$

where $s \in \mathbb{R}^+$ is a fixed length scale (we set $s = 1$). We define the *stereographic flow* $V' = \{v'_{ij}\}$ as the result of applying f to each component of the local vector field v_{ij}. Effectively, we represent each element of the local vector field as a point on the unit 3D sphere S^2 (note that this generalizes to arbitrary dimensions). In particular, f maps the zero vector $(0, 0)$ to the north pole $(0, 0, 1)$ and all vectors with infinite length ("points at infinity") to the south pole $(0, 0, -1)$. The stereographic flow is computed using an extra lightweight convolutional module operating at the highest resolution (*cf.* Supp. Fig. 1). The corresponding loss function is a pixel-wise weighted L_1 loss $\mathcal{L}_{flow}$ between the ground truth stereographic flow $V' = \{v'_{ij}\}$ and the prediction $\hat{V}' = \{\hat{v'}_{ij}\}$:

$$\mathcal{L}_{flow}(V', \hat{V}') = \sum_{i,j} ||v'_{ij} - \hat{v'}_{ij}||_1 \qquad (5)$$

Note that, by construction, v'_{ij} and $\hat{v'_{ij}}$ lie on the three-dimensional unit sphere S^2 yielding a bounded target to be regressed.

Let $S'^2 = S^2 \setminus \{(0, 0, -1)\}$ denote the set of all points in the unit sphere S^2 but the south pole. The stereographic flow can be analytically inverted position-wise by applying the *stereographic projection* $f^{-1} : (v'_x, v'_y, v'_z) \in S'^2 \to (v_x, v_y) \in \mathbb{R}^2$:

$$v_x = \frac{sv'_x}{1 + v'_z}, \quad v_y = \frac{sv'_y}{1 + v'_z} \qquad (6)$$

We note that despite the stereographic projection being undefined at the south pole $(0, 0, -1)$, in practice we only invert the stereographic flow at positions that are close to a spot, which are embedded far from the south pole.

Inference

To retrieve the spot centers from the two outputs of the network (*i.e.* multiscale heatmaps and stereographic flow), we first detect all local maxima in the highest resolution predicted heatmap $\hat{Y}^{(0)}$. These local maxima are filtered so than only those above a specific threshold (*probability threshold* $t \in [0, 1]$) are kept. This threshold is optimized right after training on the validation data and thus does not need to be set explicitly during inference. This procedure results in a set of points $\{(x_k, y_k)\}$ where $x_k, y_k \in \mathbb{Z}$.

These points are then refined using the stereographic flow to achieve subpixel precision by adding the corresponding predicted vector at every position. Specifically, let $\hat{V}$ denote the pixel-wise stereographic projection of the predicted stereographic flow $\hat{V}'$, so that the vector at position ij $\hat{v}_{ij} \in \mathbb{R}^2$. We generate the final set of points $\{p_k\}$, which correspond to the spot centers, as $\{p_k = (x_k, y_k) + \hat{v}_{x_k y_k}\}$, where $\{(x_k, y_k)\}$ are the local maxima extracted from the full resolution heatmap and $p_k \in \mathbb{R}^2$, thus allowing the prediction of non-integer (subpixel-precise) spot centers.

Spot detection benchmarking

Datasets (synthetic)

The dataset *synthetic-simple* was generated by randomly generating spot locations and placing Gaussian distributions with $\sigma = 1.5$ and varying intensity on a blank image, after which Poisson and Gaussian noise is added. The dataset *synthetic-complex* was generated similarly but instead of Gaussian spots we simulated realistically aberrated point-spread functions (PSFs) using the approach described in [28] and added fluorescence DAPI background, Gaussian noise, Perlin noise and Poisson noise at different levels yielding images with different SNRs across the dataset. Different densities (*i.e.* number of spots) were used to mimic different sparsity of real data.

Datasets (real)

We gathered the datasets *HybISS*, *Terra* and *Telomeres* by randomly cropping square tiles of width 512 and/or 1024 from different acquistions (see below). The dataset *MERFISH* was compiled by using raw images from [29] which we hand-annotated with *napari* [19]. In order to speed up the annotation process, we used initial solutions obtained from LoG and/or other Spotiflow models that we iteratively refined by adding, removing and/or moving the proposed spot centers. Different contrasts were considered when annotating to take into account potential uneven illumination. The annotated *smFISH* dataset was used as released in [17].

Dataset preprocessing

Images were preprocessed equally for each method by normalizing them using percentile-based normalization:

$$I^{norm}(x, y) = \frac{I(x, y) - I_{p_{min}}}{I_{p_{max}} - I_{p_{min}}}$$

where I_p denotes the p-th percentile of the image intensity. We set $p_{min} \in \{1, 3\}$ and $p_{max} = 99.8$ throughout our experiments.

Parameter tuning

Parameters specific to LoG (intensity threshold) and Big-FISH (variance of the filters) were optimized on the training split of each dataset. We did not optimize the intensity threshold on Big-FISH as the software has a custom threshold optimization procedure which works on an image-by-image basis. For RS-FISH, we optimized its parameters on the test split (thus overestimating its performance) due to the high computational load required and the large number of parameters that can be tuned (*cf.* Supp. Note 3.3). Learning methods (deepBlink and Spotiflow) were trained on the training split using their default configuration without performing any hyperparameter tuning (*cf.* Supp. Note 3.3). All reported scores are on the test split of the datasets.

Spotiflow (general) model

In order to assess the potential capacity of Spotiflow models, we trained the *general* model on a dataset gathered by merging all real datasets (*HybISS*, *MERFISH*, *smFISH*, *Telomeres*, *Terra*) as well as the dataset *synthetic-complex*.

Metrics

To compute overall detection metrics for each image, we first uniquely match ground truth $\{p_i\}$ and predicted spots $\{\hat{p}_j\}$ according to their spatial proximity via hungarian matching [30]. We then define a spatial cutoff $c \in \mathbb{R}$ and count a matched pair $(p, \hat{p})$ as *true positive* (TP) if their Euclidean distance $d(p, \hat{p}) \leq c$, a predicted spot $\hat{p}$ as *false positive* (FP) if there was no matched ground truth spot, and a ground truth spot p as *false negative* (FN) if there was no matched predicted spot. We then define the following metrics for each image

$$F_1[c] = \frac{|TP|}{|TP| + \frac{1}{2}(|FP| + |FN|)} \tag{7}$$

$$AP[c] = \frac{|TP|}{|TP| + |FP| + |FN|} \tag{8}$$

$$\tag{9}$$

We also report the F_1AuC [17] (*cf.* Supp. Note 3.2), which takes into account different spatial cutoffs:

$$F_1AuC_{[c_L;c_H]} = \frac{1}{c_H - c_L}\frac{\sum_{k=L}^{H-1} F_{1[c_k]} + F_{1[c_{k+1}]}}{2}\Delta \quad \text{with } L < H,\ c_L < c_H \tag{10}$$

where Δ is a constant defined as $c_{k+1} - c_k$ for any $k \in [L, H)$. Finally, we adapt the Panoptic Quality segmentation metric [31], which incorporates the spatial localization accuracy, to the spot detection task. We refer to it as the *Panoptic Localization Quality* (PLQ):

$$LA_{[c]} = \frac{1}{|TP|}\sum_{(p,\hat{p})\in TP} 1 - \frac{\max(d(p,\hat{p}), c)}{c} \tag{11}$$

$$PLQ_{[c]} = LA_{[c]} \cdot F_{1[c]} \tag{12}$$

where LA is the localization accuracy (*cf.* Supp. Note 3.2). We report results at $c = 3$, $c_L = 1$ and $c_H = 5$.

Scalability assessment

In order to assess the scalability of different methods (*cf.* Fig. 1e, Supp. Note 3.4), we used images of different sizes which were all obtained by consecutively expanding a center crop from a full HybISS cycle (see below). Given the dependency of intensity-based methods on the number of spots, their parameters were set so that the number of detections were in the same order of magnitude across all methods. Results were obtained using the Python-based profiling tool Scalene for all methods but RS-FISH, whose results have been obtained by the Unix `time` command. Intensity-based methods were run on an AMD Ryzen Threadripper PRO 5965WX 24-Cores CPU with 256GB of memory. Learning methods (deepBlink and Spotiflow) were run on an NVIDIA GeForce RTX 4090 GPU (24GB).

Spatial transcriptomics experiments

Tissue collection and preparation

All animal procedures were in accordance with the Swiss Federal Veterinary Office guidelines and as authorized by the Cantonal Veterinary Authorities and the Cantonal Commission for Animal Experimentation under the following licenses: cantonal animal license number VD3651 and national animal license number 33167 for mouse samples as well as cantonal animal license number VD3652c and national animal license number 33237 for frog tadpole samples.

Mouse embryos and frog tadpole samples

Mouse embryos at E12.5 and E13.5 were collected from wild-type CD1 pregnant mice dissecting out from the uterine horn in ice-cold PBS. Nieuwkoop and Faber (FB) stage 58 frog tadpole samples were collected in PBS. Immediately after collection, fresh tissues were cryopreserved in optimal cutting temperature (OCT) and stored at -80 °C until sectioning. Tissues were sectioned with a cryostat (Leica CM3050 S) at 10 μm, placed on SuperFrost Plus microscope slides, and stored at -80 °C until HybISS processing.

Mouse gastruloid generation

Gastruloid generation was performed as previously described in [32]. Briefly, mouse embryonic stem cells (mESCs) (EmbryoMax 129/SVEV, gifted by Denis Duboule Lab) were cultured in gelatinized tissue culture dishes with 2i LIF DMEM medium consisting of DMEM + GlutaMAX (Gibco 61965-026) supplemented with 10% mES-certified FBS (Gibco 16141-079), non-essential amino acids (Gibco 11140-035), sodium pyruvate (Gibco 11360-039), beta-mercaptoethanol (Gibco 31350-010), penicillin/streptomycin (Gibco 15140-122), 100 ng ml-1 mouse LIF (EPFL Protein Facility), 3 μM CHIR99021 (Calbiochem: 361559) and 1 μM PD0325901 (Selleckchem S1036). Cells were passaged every 2-3 days and maintained in a humidified incubator (5% CO2, 37°C). mESCs were collected after trypsin treatment, washed, and resuspended in prewarmed N2B27 medium (50% DMEM/F12 (Gibco 31331-028), 50% Neurobasal medium supplemented (Gibco 21103-049) with 0.5x N2 (Gibco 17502-048), 0. 5x B27 (Gibco 17504-044), non-essential amino acids (Gibco 11140-035), sodium pyruvate (Gibco 11360-039), beta-mercaptoethanol (Gibco 31350-010), 0.5x Glutamax (Gibco 35050-061) and penicillin/streptomycin (Gibco 15140-122). A total of 300 cells were seeded in 40 μl of N2B27 medium in each well of a 96-well plate with a rounded bottom and low adherence (Thermo Fisher, 174925). Forty-eight hours after aggregation, 150 μl of N2B27 medium supplemented with 3 μM CHIR99021 was added to each well. A total of 150 μl of medium was replaced every 24 h. Gastruloids were collected and flash-frozen 120 h after aggregation.

Radial glia progenitor culture

E11.5 mouse brains were collected from wild-type CD1 pregnant mice in ice-cold EBSS (14155-048, Life-technologies). Meninges were removed using fine-tipped forceps under a dissection stereomicroscope (Nikon SMZ18). Then, brains were fragmented into small pieces, transferred to a 50 ml plastic tube, and digested for 30-45 min at 37 °C in 5 ml of solution containing 1mM CaCl2, 1 mM MgCl2, 100 U/ml of DNAse I (LS02058 Worthington, Lake Wood NJ), and 20 U/ml of previously activated papain (Sigma L2020). After that, the cell suspension was briefly decanted, transferred into a 15-ml plastic tube, and centrifuged at 300 rcf for 5 min at 4°C. Then, the cells were resuspended in 3ml of EBSS, and the suspension was transferred into a 15 ml plastic tube containing 3ml of papain inactivating solution containing 50% FBS/50% EBSS. Cells were centrifuged at

300 rcf for 5 min at 4°C and then resuspended in culture media consisting of Neurobasal medium (21103049, Life Technologies) supplemented with L-glutamine (Gibco; cat. no. 25030- 123), B27 (Gibco; cat. no. 17504-044), Gentamicin (15750037, Life Technologies), and 20 ng/ml of Epidermal growth factor (EFG, PeproTech catalog no. AF-100-15). Finally, cells were seeded on 8 well chamber slides (80841, IBIDI) and incubated in a humidified 37°C incubator at 5% CO2 for 48 hours until HybISS processing.

In-situ sequencing by HybISS

To process all samples (apart from the adult mouse brain, see below) HybISS [9] was performed as published at protocols.io [33]. For embryonic mice, target genes were selected based on marker genes with expression within the target regions at the target developmental stages. Samples were imaged either on a Leica DMi8 epifluorescence microscope equipped with LED light source (Lumencor SPECTRA X, nIR, 90-10172), sCMOS camera (Leica DFC9000 GTC, 11547007) and 20x objective (HC PC APO, NA 0.8, air) yielding a pixel size of 0.34μm, or on a epifluorescence microscope Nikon Eclipse 90i equipped with LED light source (Lumencor SPECTRA X, nIR, 90-10172), CMOS camera (Nikons DS Qi2) and 20x objective (CFI PLAN PC, NA 0.75, air) yielding a pixel size of 0.15μm. In both microscopes, samples were imaged with 10% overlap between tiles to cover the entire tissue and between 8 and 12 z planes were acquired with 1 µm spacing among them. A full experiment results in a multicycle, multichannel image stack (5 cycles and DAPI + 4 HybISS signal channels) image stack.

To process the adult mouse brain used for the autofluorescence removal experiment (*cf.* Supp. Fig. 9), HybISS was performed on fresh frozen 6 weeks old mouse brains 10μm sections, using a Phi29 enzyme (NxGen F83900-1). Images were acquired using a 20X 0.8NA objective on a Zeiss AxioImager Z1 Wiedfield microscope with a PCO.edge 4.2bi camera. The microscope was controlled via MicroManager. Exposure times were, in order of acquisition: 3ms for DAPI (Zeiss filter set 49: G365, FT 395, BP445/50), 450ms for 750nm (filters: Alluxa ultra 740.5-35 OD6, 766, 801.5-50 OD6), 300ms for 650nm (chroma BP 640/30, FT ZT640rdc, ET680/40), 300ms for 550nm (filters BP546/12, LP T560lpxr, ET590-50), and 200ms for 488nm (chroma filters BP450-490, T510, ET 525/36). 130 tiles of 2048x2028 pixels with a 10% overlap were acquired to cover the entire tissue, with a pixel size of 0.3225μm. Each tile was a z-stack of 11 planes with a 0.8μm step size. Rounds of probe hybridization, imaging and stripping were performed with a modified Labsat microfluidics device from Lunaphore, allowing us to place the stainer chamber under the microscope. A quenching buffer (Lunaphore BU08) was used to reduce autofluorescence before bridge probe hybridization and an imaging buffer (Lunaphore BU09) was used during imaging.

Image processing

Projection, stitching and alignment

Two yield 2D images the acquired stacks were reduced using either maximum intensity projection (MIP) or a custom implementation of extended depth-of-field (EDF, [34]). Tiled acquisitions were stitched together into a mosaic image with Ashlar [35], which uses a variant of phase correlation [36] to compute the offset between the different tiles at subpixel precision [37] in a simultaneous fashion. Only the DAPI channel was used to retrieve the stitching coordinates, and different cycles in the same experiment were stitched independently. After obtaining the mosaics for all cycles in an experiment, we registered them with *wsireg* [38], which uses *elastix* [39, 40] as a backend. We allow for rigid-body alignment as well as non-linear warping, which we found did not aggresively deform the sample and was able to align fine details properly. We only used the DAPI channel for inter-cycle registration.

Spot detection (Spotiflow)

All results with Spotiflow were obtained using a Spotiflow network trained on the HybISS dataset (*cf.* Fig. 1b, Fig. 1c, Supp. Fig. 4, Supp. Note 2) to detect diffraction-limited spots independently across cycles and channels. The probability threshold used was optimized from the validation data of the *HybISS* dataset.

Spot detection (LoG/starfish)

We ran LoG using starfish's implementation, which is based on scikit-image [41], and used different intensity thresholds including the 'optimal' one (0.138) which was computed from the training data of the *HybISS* dataset. To ensure a fair comparison, we detected transcripts independently across cycles and channels as done for Spotiflow.

Gene decoding

To extract gene expression maps the detected transcripts were assigned a gene using starfish's implementation of an intensity-based nearest-neighbor decoder. When decoding spots detected via Spotiflow, we fed the decoder the spot probabilities output by the network instead of their raw intensity. Gene expression heatmaps were obtained by performing Gaussian kernel density estimation (KDE) with variance $\sigma^2 = 5$ on the gene signal. The heatmaps were clipped to $(0, 1)$ after applying percentile-based normalization with $p_{min} = 0$, $p_{max} = 99.9$.

Time/memory benchmarking

To compare the time and memory efficiency of starfish and Spotiflow in an end-to-end setting (*cf.* Fig. 2e, Fig. 2f), we used Unix's *time* module. We detected spots on an E12.5 mouse embryonic brain using LoG on the maximum intensity projection of the input along the cycle and channel dimensions, as done in previous studies [9, 42]. The spot intensities were then traced back along the non-projected input to retrieve the intensity of spots at the different cycles and channels. For Spotiflow, the detections were done independently on each cycle and channel as it is computationally affordable for larger tile sizes.

Zero-shot autofluorescence removal

We first built a dataset from a single experiment consisting of a regular *HybISS* acquistion preceded by imaging an only-autofluorescence cycle. After registering both images, we then detected spots (using the Spotiflow model trained on the *HybISS* dataset) in one channel (corresponding to 750nm) of the autofluorescence cycle and the same channel of the first HybISS cycle. We generated the non-autofluorescent ground-truth by substracting all the autofluorescent detections that matched (at spatial cutoff $c = 3$) to a detection in the HybISS channel. We generated three spatially-disjoint splits from this experiment (training, validation and test). We finally fine-tuned the Spotiflow model pre-trained on *HybISS* on the generated training dataset to predict the non-autofluorescent spots. Quantification is reported on the test split (*cf.* Supp. Fig. 9).

Live-cell imaging experiments

Tissue collection and preparation

HeLa cells expressing endogenously tagged Halo-TRF1 were labelled with Janelia Fluor 646 Halo ligand (Promega) in order to mark telomeres. To visualize TERRA, ectopically expressed 15q-TERRA species were tagged with PP7 stem-loop structures that were bound by phage coat protein fused to GFP (PCP-GFP). Live cells were imaged using a Nikon Confocal Spinning Disk microscope equipped with two Photometrics Prime 95B cameras and sCMOS Grayscale Chips. Imaging was performed with a 100x objective in an equilibrated incubation chamber at 37°C and 5% CO2. Images were acquired as multi-channel single planes at a rate of 20 frames per second (50 ms exposure, 200 frames per movie).

Movie processing

Movies were first spatially cropped so that each crop contains only one cell. In each crop, spots were detected independently per frame and channel, where one channel contains the Telomeres marker and the other the TERRA marker, using the deepBlink and Spotiflow models trained on the *Telomeres* and *Terra* datasets. The optimized probability threshold on each dataset was used for Spotiflow. For deepBlink, the probability threshold was set to the default (0.5). Single particle tracking was performed using Trackmate [24] with a spot radius of 0.15 μm and simple LAP tracking with the following parameters for Telomeres/TERRA: linking maximum distance 0.22/0.60 μm, gap-closing maximum distance 0.44/1.0 μm and gap-closing maximum frame gap 10 frames.

References

[1] I. Smal et al. “Quantitative comparison of spot detection methods in fluorescence microscopy”. In: *IEEE transactions on medical imaging* 29.2 (2009), pp. 282–301.

[2] K. Štěpka et al. “Performance and sensitivity evaluation of 3D spot detection methods in confocal microscopy”. In: *Cytometry Part A* 87.8 (2015), pp. 759–772.

[3] M. A. Mabaso, D. J. Withey, and B. Twala. “Spot detection methods in fluorescence microscopy imaging: a review”. In: *Image Analysis and Stereology* 37.3 (2018), p. 173.

[4] D. Sage et al. “Super-resolution fight club: assessment of 2D and 3D single-molecule localization microscopy software”. In: *Nature Methods* 16.5 (2019), pp. 387–395.

[5] M. Steinfath et al. “Automated image analysis for array hybridization experiments ”. In: *Bioinformatics* 17.7 (2001), pp. 634–641. eprint: `https://academic.oup.com/bioinformatics/article-pdf/17/7/634/48837019/bioinformatics\_17\_7\_634.pdf`.

[6] A. W. Dowsey, M. J. Dunn, and G.-Z. Yang. “The role of bioinformatics in two-dimensional gel electrophoresis”. In: *PROTEOMICS* 3.8 (2003), pp. 1567–1596. eprint: `https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/pmic.200300459`.

[7] L. Moses and L. Pachter. “Museum of spatial transcriptomics”. In: *Nature Methods* 19.5 (2022). Number: 5 Publisher: Nature Publishing Group, pp. 534–546.

[8] J. R. Moffitt et al. “Molecular, spatial, and functional single-cell profiling of the hypothalamic preoptic region”. In: *Science* (2018). Publisher: American Association for the Advancement of Science.

[9] D. Gyllborg et al. “Hybridization-based in situ sequencing (HybISS) for spatially resolved transcriptomics in human and mouse brain tissue”. eng. In: *Nucleic Acids Research* 48.19 (2020), e112.

[10] S. Shah et al. “In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus”. In: *Neuron* 92.2 (2016), pp. 342–357.

[11] S. Codeluppi et al. “Spatial organization of the somatosensory cortex revealed by osmFISH”. en. In: *Nature Methods* 15.11 (2018). Number: 11 Publisher: Nature Publishing Group, pp. 932–935.

[12] S. Axelrod et al. “Starfish: Open source image based transcriptomics and proteomics tools”. In: *The Journal of Open Source Software* 6 (2018), p. 2440.

[13] A. Imbert et al. “FISH-quant v2: a scalable and modular tool for smFISH image analysis”. In: *RNA* 28.6 (2022), pp. 786–795.

[14] E. Bahry et al. “RS-FISH: precise, interactive, fast, and scalable FISH spot detection”. In: *Nature Methods* (2022). Publisher: Nature Publishing Group, pp. 1–5.

[15] K. R. Maynard et al. “dotdotdot: an automated approach to quantify multiplex single molecule fluorescent in situ hybridization (smFISH) images in complex tissues”. In: *Nucleic Acids Research* 48.11 (2020), e66.

[16] P. R. Gudla et al. “SpotLearn: Convolutional Neural Network for Detection of Fluorescence In Situ Hybridization (FISH) Signals in High-Throughput Imaging Approaches”. In: *Cold Spring Harbor Symposia on Quantitative Biology* 82 (2017). Publisher: Cold Spring Harbor Laboratory Press, pp. 57–70.

[17] B. T. Eichenberger et al. “deepBlink: threshold-independent detection and localization of diffraction-limited spots”. In: *Nucleic Acids Research* 49.13 (2021). Publisher: Oxford Academic, pp. 7292–7297.

[18] T. Wollmann et al. “Detnet: Deep Neural Network For Particle Detection In Fluorescence Microscopy Images”. In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). 2019, pp. 517–520.

[19] N. Sofroniew et al. *napari: a multi-dimensional image viewer for Python*. 2022.

[20] O. Ronneberger, P. Fischer, and T. Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[21] W. Xie, J. A. Noble, and A. Zisserman. “Microscopy cell counting and detection with fully convolutional regression networks”. In: *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization* 6.3 (2018), pp. 283–292.

[22] M. L. Katz and W. Robison. “What is lipofuscin? Defining characteristics and differentiation from other autofluorescent lysosomal storage bodies”. In: *Archives of Gerontology and Geriatrics* 34.3 (2002), pp. 169–184.

[23] J. Yang et al. “Quenching autofluorescence in tissue immunofluorescence”. In: *Wellcome Open Research* 2 (2017), p. 79.

[24] J.-Y. Tinevez et al. “TrackMate: An open and extensible platform for single-particle tracking”. In: *Methods* 115 (2017). Image Processing for Biologists, pp. 80–90.

[25] A. Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[26] U. Schmidt et al. “Cell detection with star-convex polygons”. In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*. Springer. 2018, pp. 265–273.

[27] C. Stringer et al. “Cellpose: a generalist algorithm for cellular segmentation”. In: *Nature Methods* 18.1 (2020), pp. 100–106.

[28] D. Saha et al. “Practical sensorless aberration estimation for 3D microscopy with deep learning”. In: *Opt. Express* 28.20 (2020), pp. 29044–29053.

[29] M. Zhang et al. “Spatially resolved cell atlas of the mouse primary motor cortex by MERFISH”. In: *Nature* 598.7879 (2021), pp. 137–143.

[30] P. Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272.

[31] A. Kirillov et al. “Panoptic segmentation”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 9404–9413.

[32] L. Beccari et al. “Multi-axial self-organization properties of mouse embryonic stem cells into gastruloids”. In: *Nature* 562.7726 (2018), pp. 272–276.

[33] D. Gyllborg and M. Nilsson. “HybISS: Hybridization-based In Situ Sequencing v1”. In: (2020).

[34] B. Forster et al. “Extended Depth-of-Focus for Multi-Channel Microscopy Images: A Complex Wavelet Approach”. In: *Proceedings of the Second IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI’04)*. Arlington VA, USA, 2004, pp. 660–663.

[35] J. L. Muhlich et al. “Stitching and registering highly multiplexed whole-slide images of tissues and tumors using ASHLAR”. In: *Bioinformatics* 38.19 (2022), pp. 4613–4621. eprint: `https://academic.oup.com/bioinformatics/article-pdf/38/19/4613/49885528/btac544.pdf`.

[36] C. Kuglin and D. Hines. “The phase correlation image alignment method”. In: *Proceeding of IEEE International Conference on Cybernetics and Society* (1975), pp. 163–165.

[37] M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup. “Efficient subpixel image registration algorithms”. In: *Opt. Lett.* 33.2 (2008), pp. 156–158.

[38] H. Patterson. *Multi-modal whole slide image registration (wsireg)*. Version 0.3.8. 2022.

[39] S. Klein et al. “elastix: A Toolbox for Intensity-Based Medical Image Registration”. In: *IEEE Transactions on Medical Imaging* 29.1 (2010), pp. 196–205.

[40] D. Shamonin et al. “Fast Parallel Image Registration on CPU and GPU for Diagnostic Classification of Alzheimer’s Disease”. In: *Frontiers in Neuroinformatics* 7 (2014).

[41] S. van der Walt et al. “scikit-image: image processing in Python”. In: *PeerJ* 2 (2014), e453.

[42] G. La Manno et al. “Molecular architecture of the developing mouse brain”. In: *Nature* 596.7870 (2021), pp. 92–96.