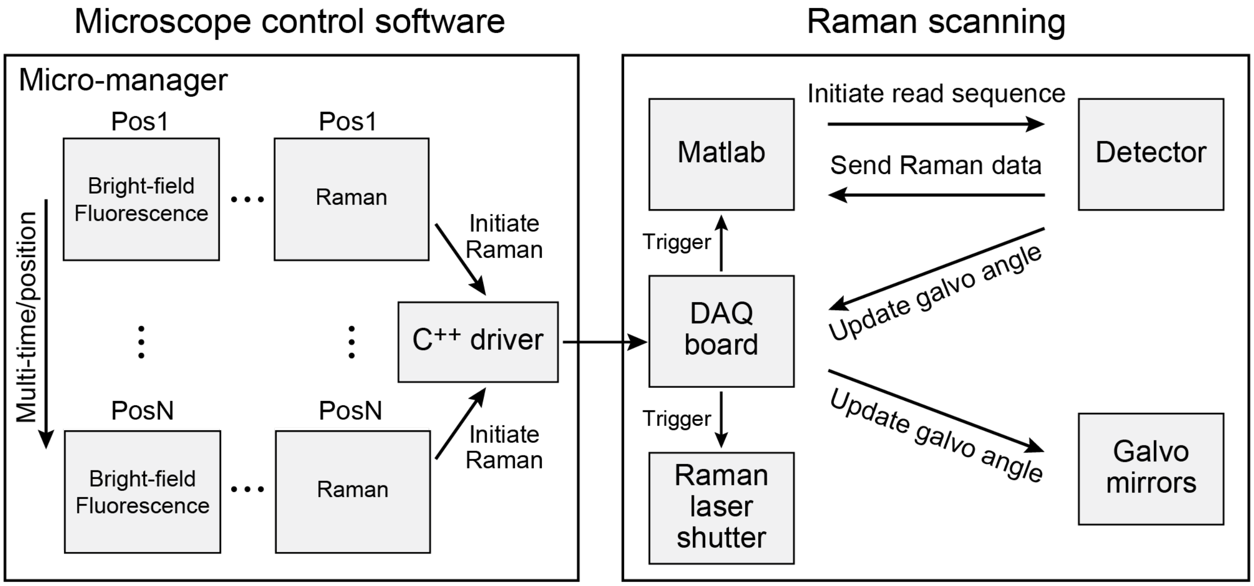
nature biotechnology

Article

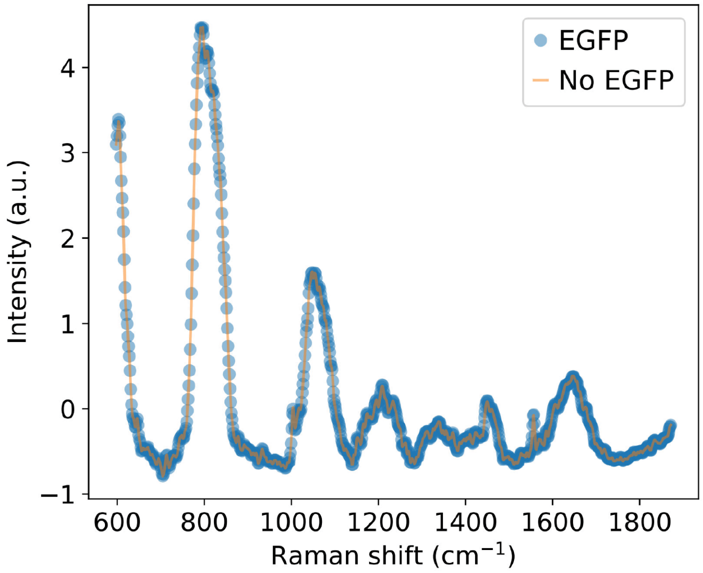
https://doi.org/10.1038/s41587-023-02082-2

Prediction of single-cell RNA expression profiles in live cells by Raman microscopy with Raman2RNA

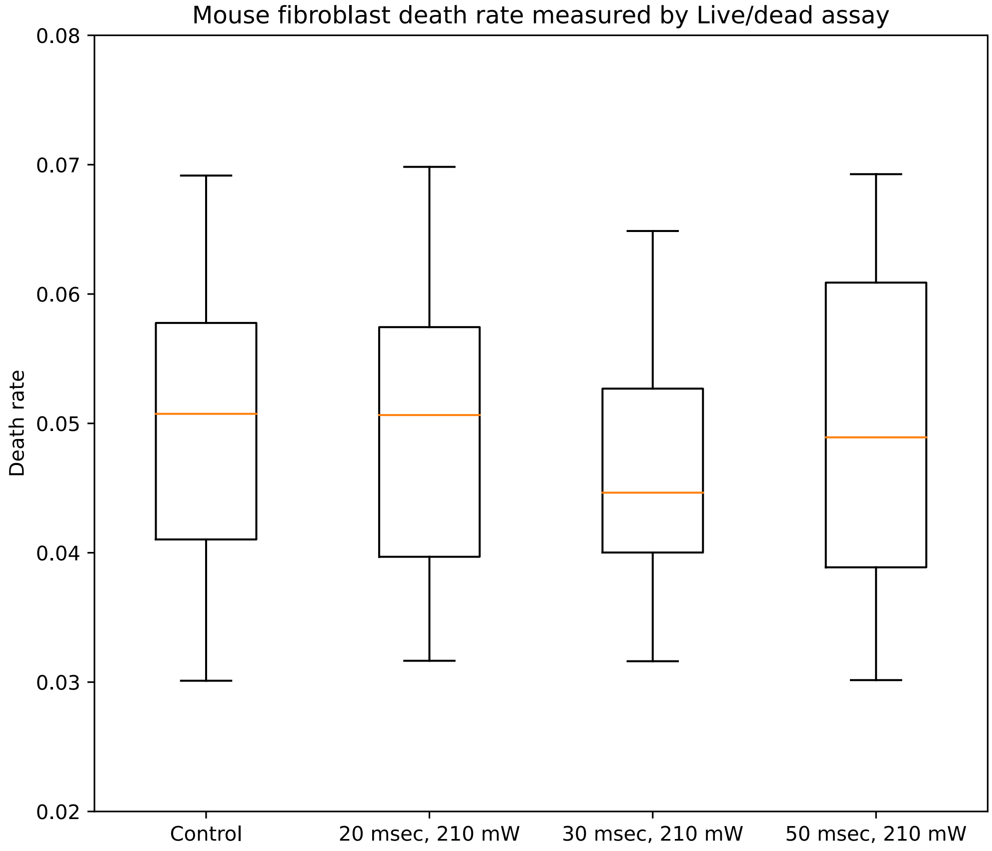
In the format provided by the authors and unedited



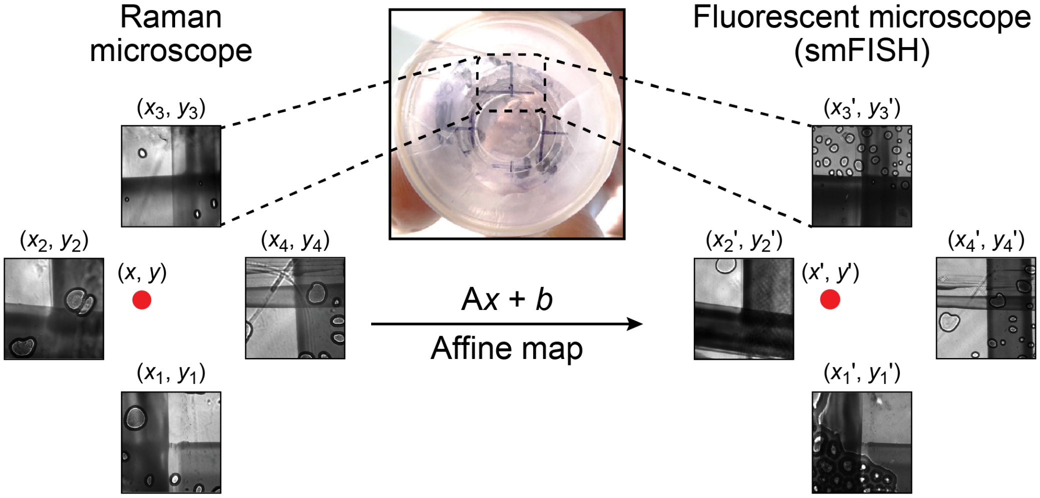
Supplementary Fig. 1 | Overview of high-throughput Raman imaging software used in the study. A general-purpose microscope control software Micro-manager and custom MATLAB script were combined to enable automated multi-modal measurements. Under Micro-manager, a Raman channel was registered as a 'dummy' channel along with brightfield and fluorescence channels. Micro-manager was responsible for changing the field of view (FOV) and imaging modality. During the Raman sequence, Micro-manager communicated with a digital acquisition (DAQ) board, through which a transistor-to-transistor logic (TTL) signal was generated to initiate the scanning sequence. Upon detection of the TTL signal, the MATLAB script controlled the Raman detector, laser shutter, and updated the galvo mirror angles through the DAQ board.



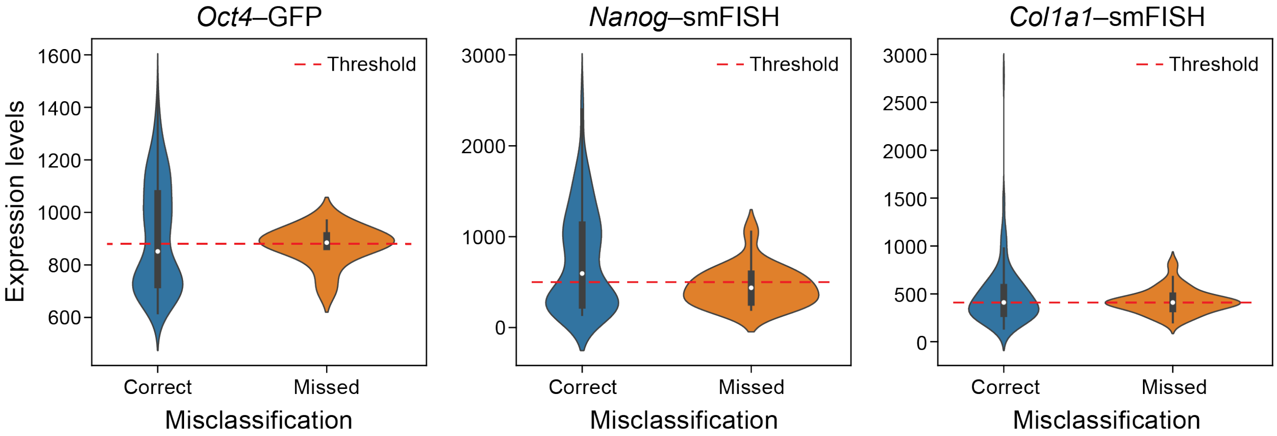
Supplementary Fig. 2 | GFP does not interfere in Raman spectra measurement. Raman spectra of culture media with (blue) and without (orange) GFP at physiological concentration.



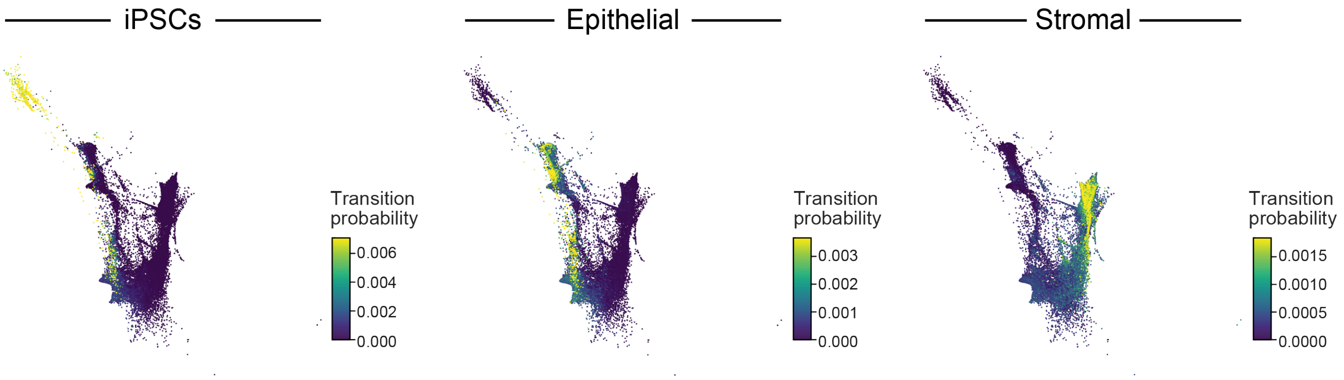
Supplementary Fig. 3 | Raman imaging does not induce substantial photo-toxicity. Proportion of dead cells (y axis) by live/dead staining at 3h after exposure to Raman laser under conditions used in this study (typically 20 ms at 210 mW). Statistics over 10 fields of view (FOV), 513 cells total. On average, each cell contained ~50 pixels of Raman measurements. Error bars: maximum (minimum) value at top (bottom) of whisker, first (third) quantile at boundary of top (bottom) of box, and median value at red line inside box.



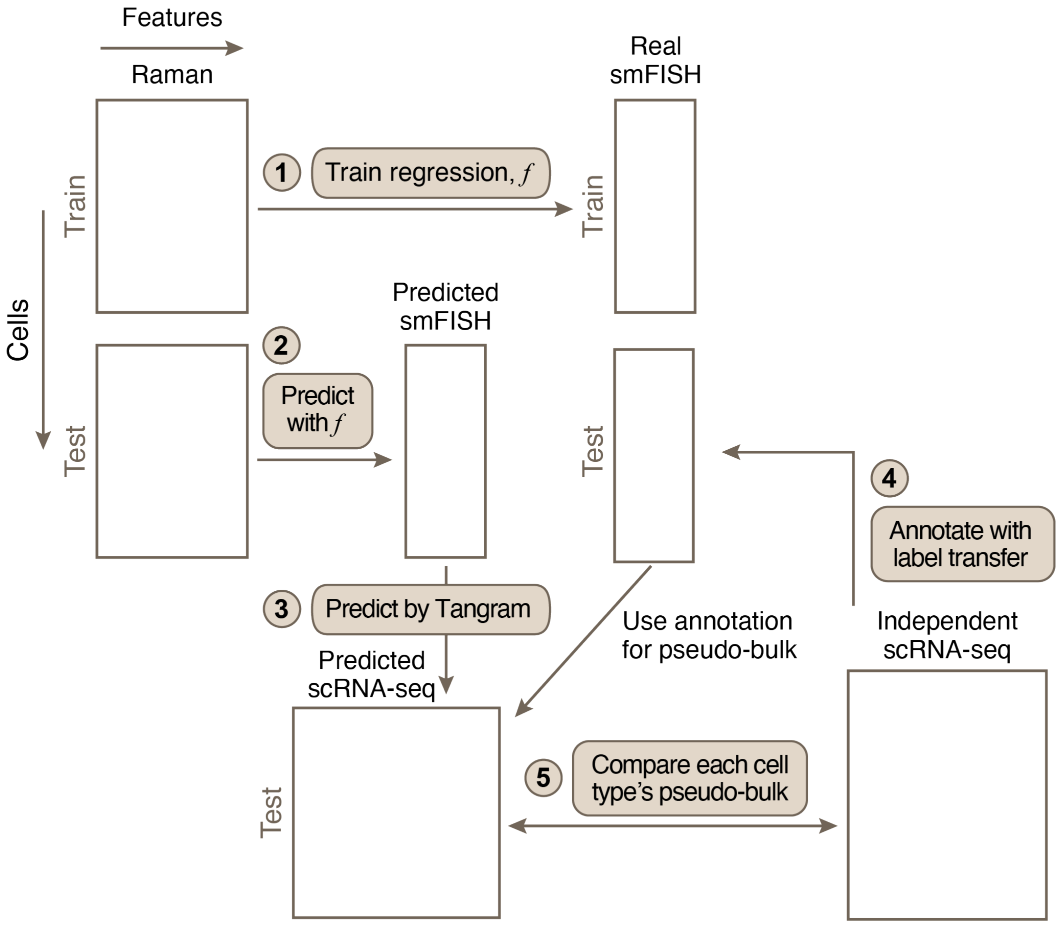
Supplementary Fig. 4 | Image registration between the Raman and smFISH microscope using control points. Control points were inscribed under Petri dishes with permanent markers and the coordinates were measured prior to any data acquisition. After Raman measurement and smFISH processing, samples were placed back to the microscope and control point coordinates were remeasured. Then, affine mapping was used to update the FOV coordinates to locate the exact same cells.



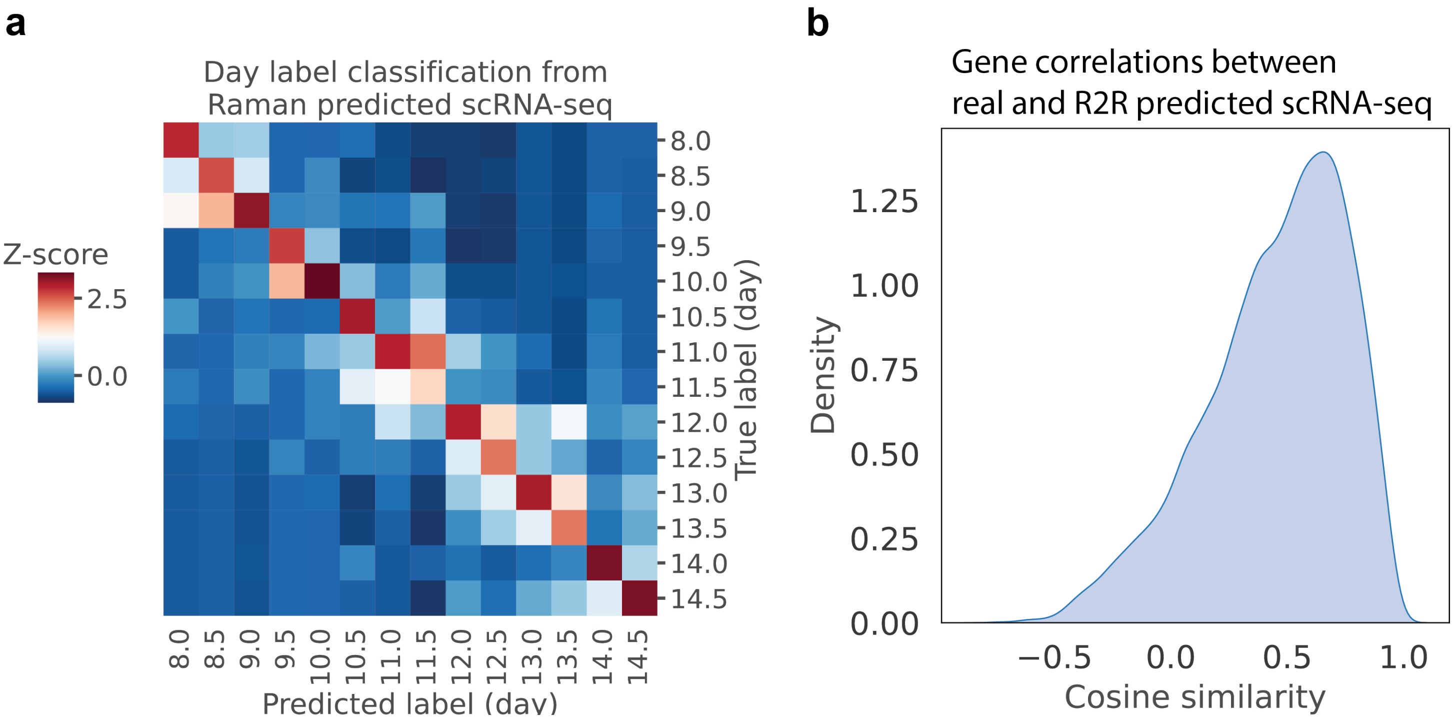
Supplementary Fig. 5 | Misclassification of genes in the cell mixture classification experiment occurs when the ground truth smFISH is near the expression threshold. Distribution of measured smFISH expression level (y axis) for cells correctly (blue) or incorrectly (orange) classified by their Raman spectra for the expression of that gene across n=217 cells. Horizontal line: an example threshold used for the logistic regression classifier. Violin plots: interquantile ranges at thick gray bar, 1.5x interquantile ranges at thin gray bar, and median values at white dot in the gray bars.



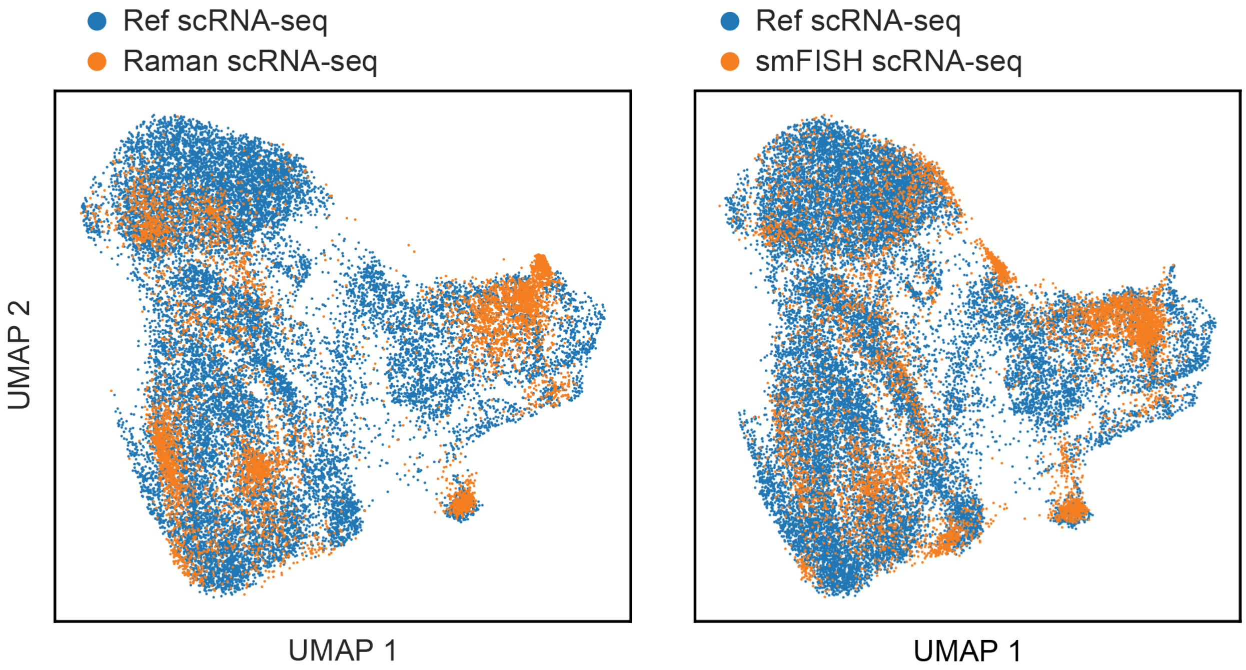
Supplementary Fig. 6 | Cell transition probabilities inferred by Waddington-OT from scRNA-seq during reprogramming. Force-directed layout embedding (FLE) of scRNA-seq profiles (dots) from days 8 to 14.5 of reprogramming (dots) colored by the transition probability of each cell as inferred by Waddington-OT to be an ancestor of iPSCs (left), epithelial cells (middle) or stromal cells (right) at day 14.5.



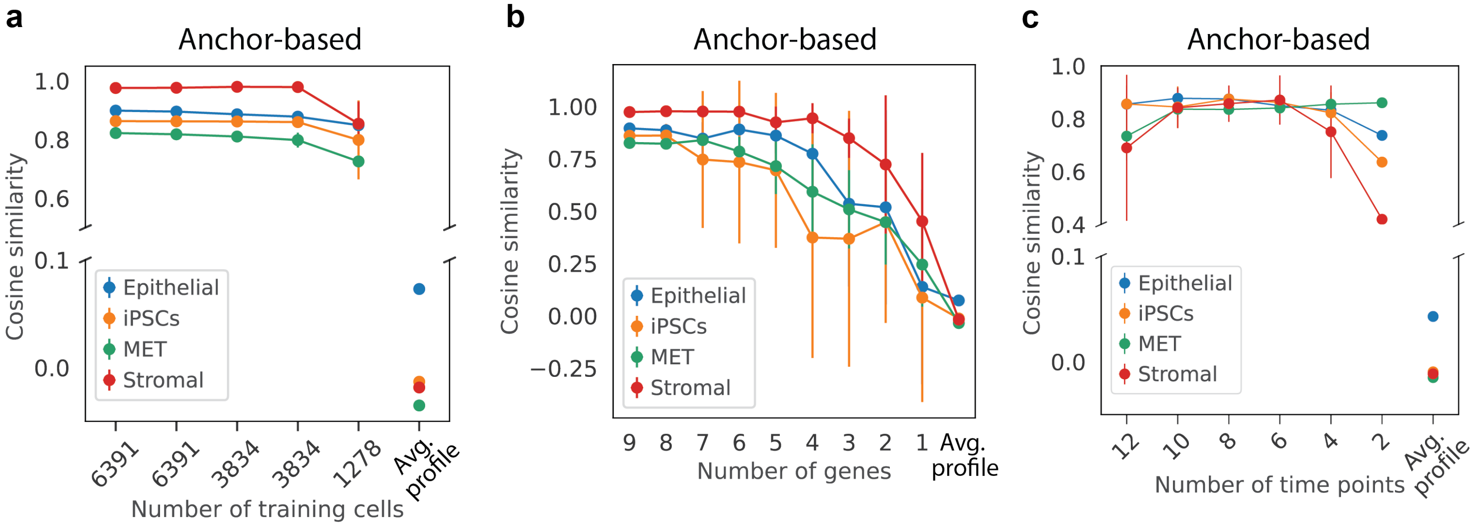
Supplementary Fig. 7 | Validation scheme for anchor-based Raman2RNA. Left: Paired Raman/smFISH profiles are split to a training and a test set. The training Raman/smFISH partition were used to learn a Raman to smFISH model (step 1). smFISH profiles were predicted from the Raman images in the test set with the learned model (step 2), and then scRNA-seq profiles were predicted by Tangram from the predicted smFISH (step 3). In parallel, to validate the predictions, scRNA-seq labels from an independent dataset were transferred (step 4), to the real smFISH of test cells. Finally, the transferred labels were used to calculate pseudo-bulk of both predicted and real scRNA-seq for comparison.



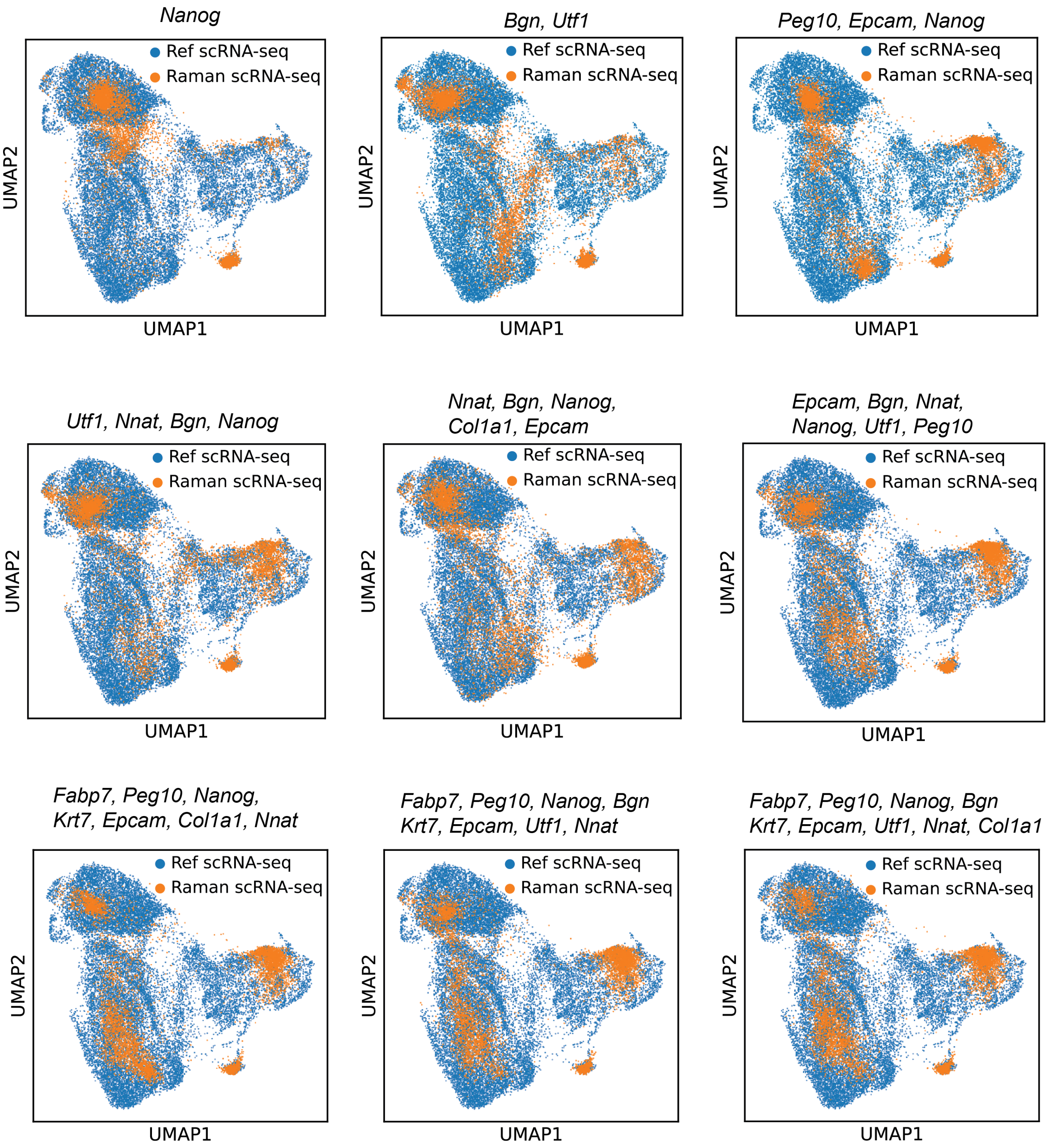
Supplementary Fig. 8 | Raman predicted scRNA-seq reflects temporal information at 0.5-day resolution across the transcriptome. (a) Number of cells (color bar, Z score) with R2R generated profiles from each real time point ("true label", rows) and each classified time point (columns) as predicted by a Catboost classifier trained with day labels as ground truth with 50/50 train/test split ratio and applied to R2R-generated profiles on the test cells. **(b)** Distribution of cosine similarities of each gene's pseudobulk profile across all time points and cell types (MET, stromal, epithelial, iPSC; of present with at least 50 cells per time point) in real scRNA-seq and R2R inferred profiles.



Supplementary Fig. 9 | RNA profiles predicted directly from 9 anchor smFISH measurements lead to reduced variance compared to scRNA-seq. UMAP co-embedding of cells from scRNA-seq (blue) and Raman (orange) experiments, with the latter based on either the Raman-predicted RNA profiles (left) or only smFISH-predicted RNA profiles (right).



Supplementary Fig. 10 | Similarity of R2R-generated and scRNA-seq profiles decreases with fewer cells, anchors, and time points. Cosine similarity (x axis) between 'pseudo-bulk' RNA profiles from real scRNA-seq of different cell types (colors) and the corresponding R2R-generated profiles on the test set, when down sampling the number of cells used in training (**a**, x axis), the number of anchor genes (**b**, x axis), or the number of time points (**c**, x axis). "Avg": baseline calculated as the profile generated by the average of the four pseudo-bulk (real) RNA profiles. In **c**, the start and end time points (days 8 and 14.5) were always included and other time points were randomly subsampled (days 8.5~14) in the training. Then, the excluded time points were used as the test set. Error bars: standard deviation of 10 random trials, and mean values at center.



Nanog

Bgn, Utf1

Peg10, Epcam, Nanog

Utf1, Nnat, Bgn, Nanog

Nnat, Bgn, Nanog, Col1a1, Epcam

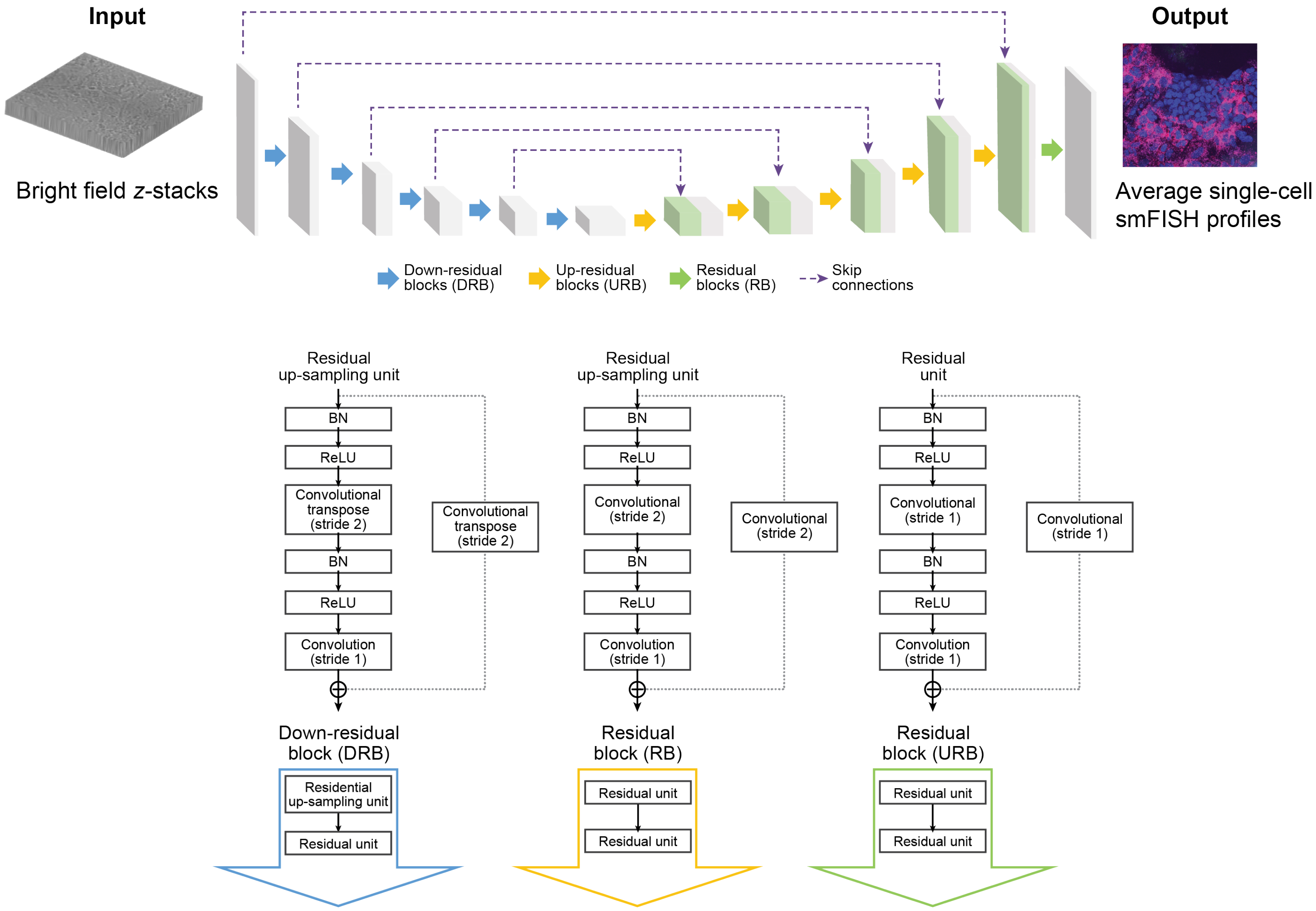
Epcam, Bgn, Nnat, Nanog, Utf1, Peg10

Fabp7, Peg10, Nanog, Krt7, Epcam, Col1a1, Nnat

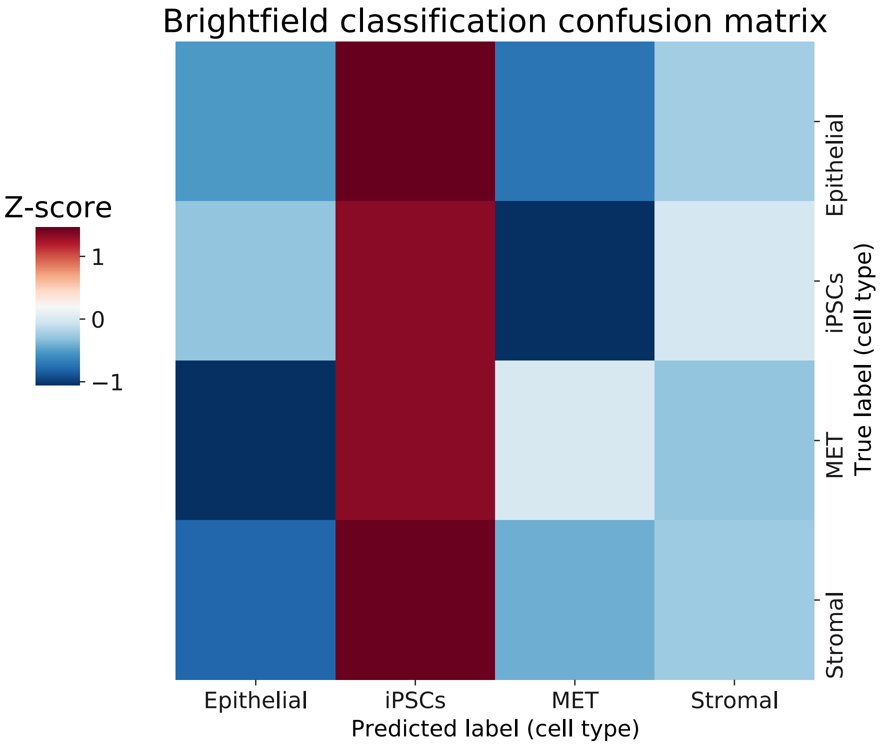
Fabp7, Peg10, Nanog, Bgn Krt7, Epcam, Utf1, Nnat

Fabp7, Peg10, Nanog, Bgn Krt7, Epcam, Utf1, Nnat, Col1a1

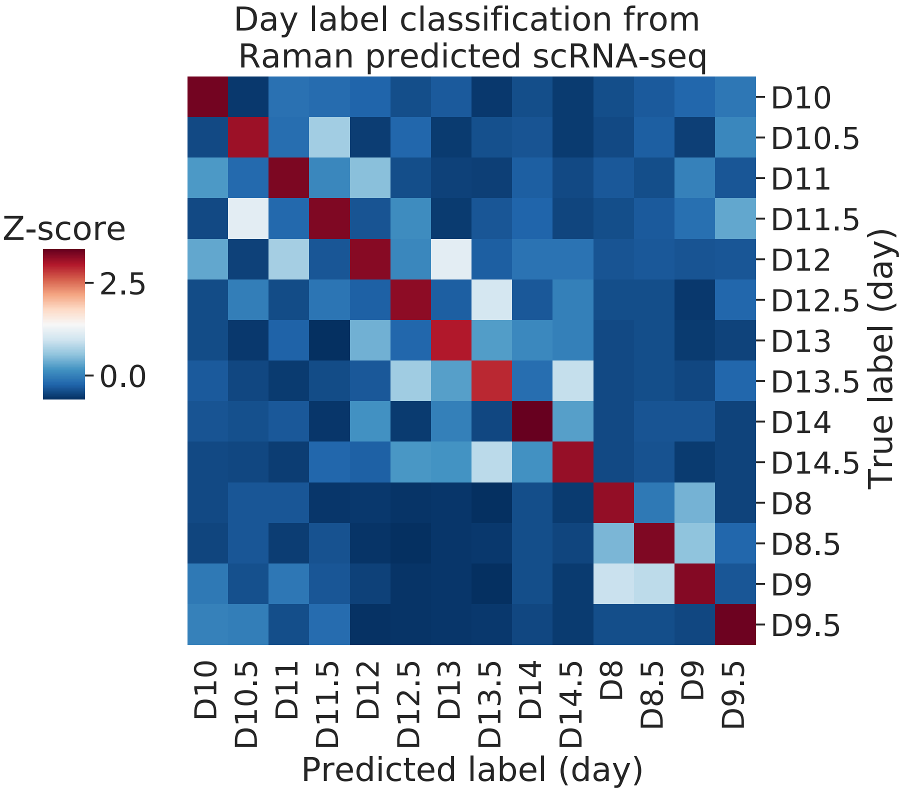
Supplementary Fig. 11 | Impact of the number of anchor genes on the quality of co-embedding real and R2R-generated single cell profiles. UMAP embedding of real scRNA-seq profiles (blue dot) and the projection to this embedding of R2R-generated profiles (orange dots) from models using increasing numbers of marker genes (from top left to bottom right, anchors labeled on top).



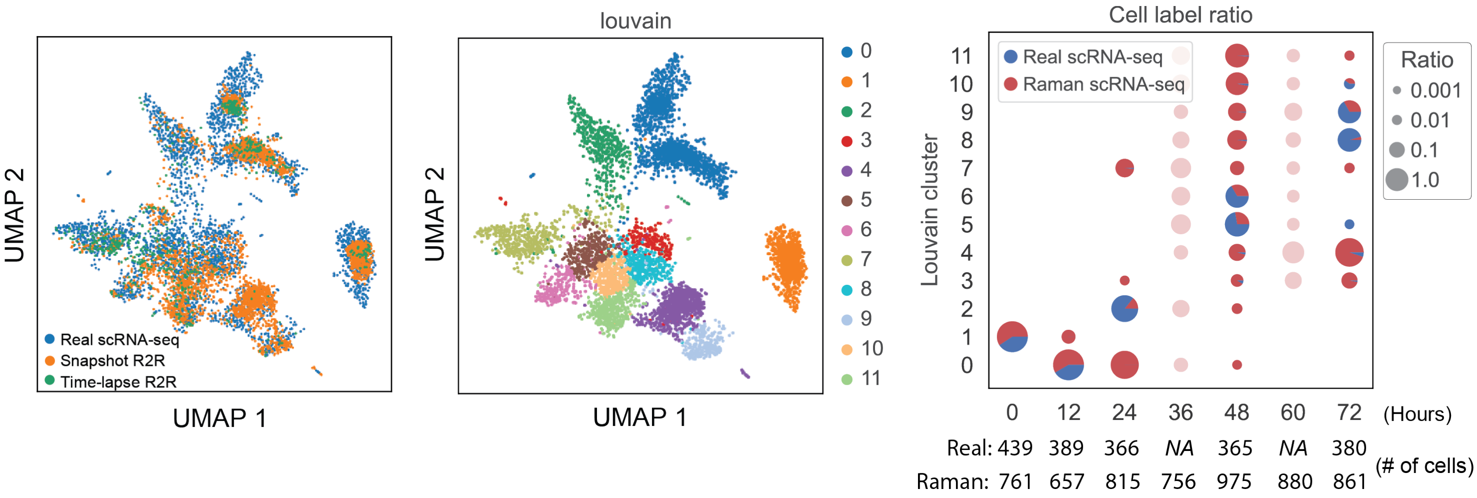
Supplementary Fig. 12 | U-net with residual connection for regression of mean single-cell smFISH profiles from brightfield z-stacks.



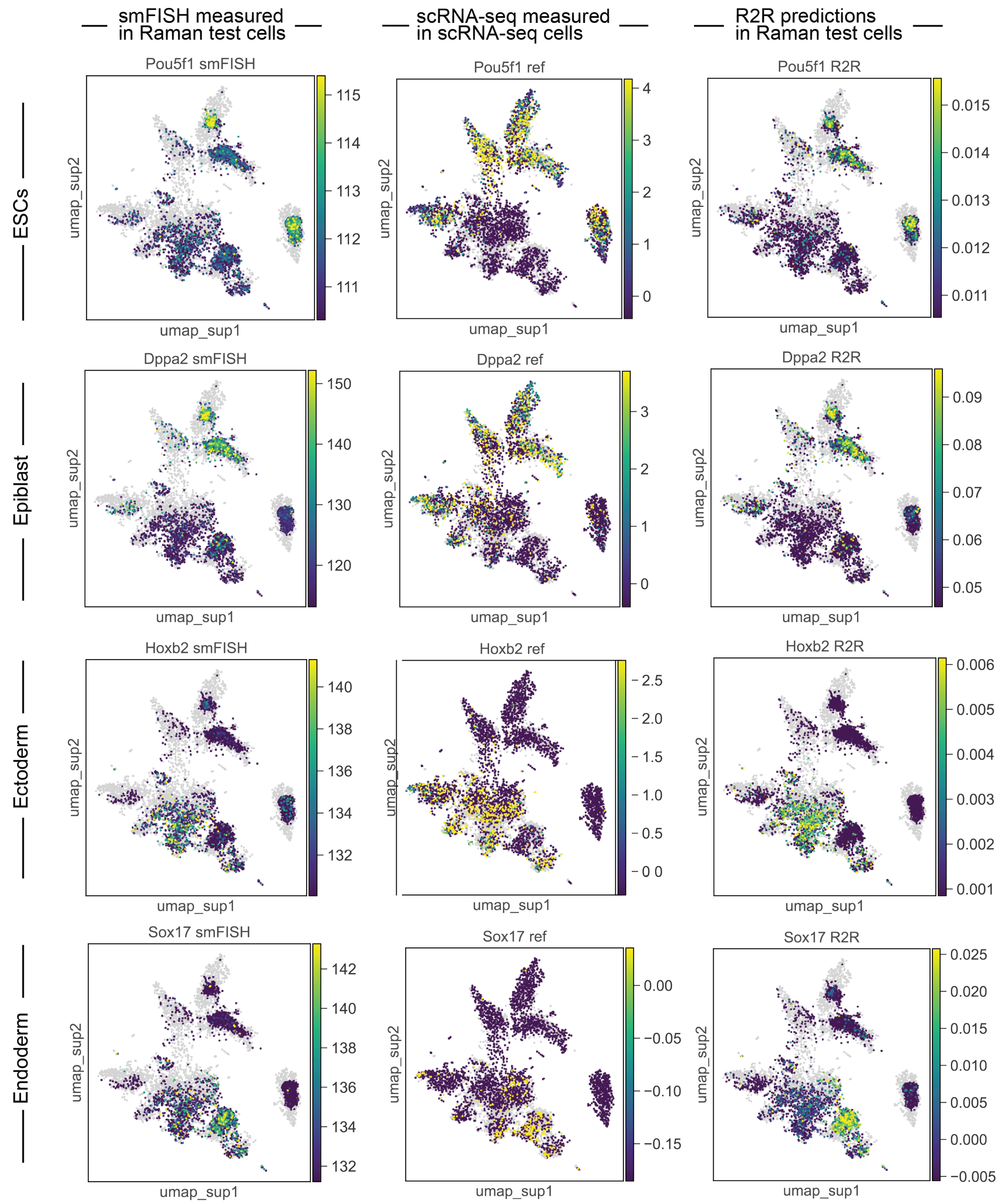
Supplementary Fig. 13 | Low accuracy of CNN-based cell type classification from brightfield images in mouse iPSC reprogramming. Number of cells (color bar) from each real cell type ("true label" based on smFISH, rows) and each classified cell type (columns) based on a CNN model trained with brightfield images and cell type labels as ground truth with 50/50 train/test split ratio.



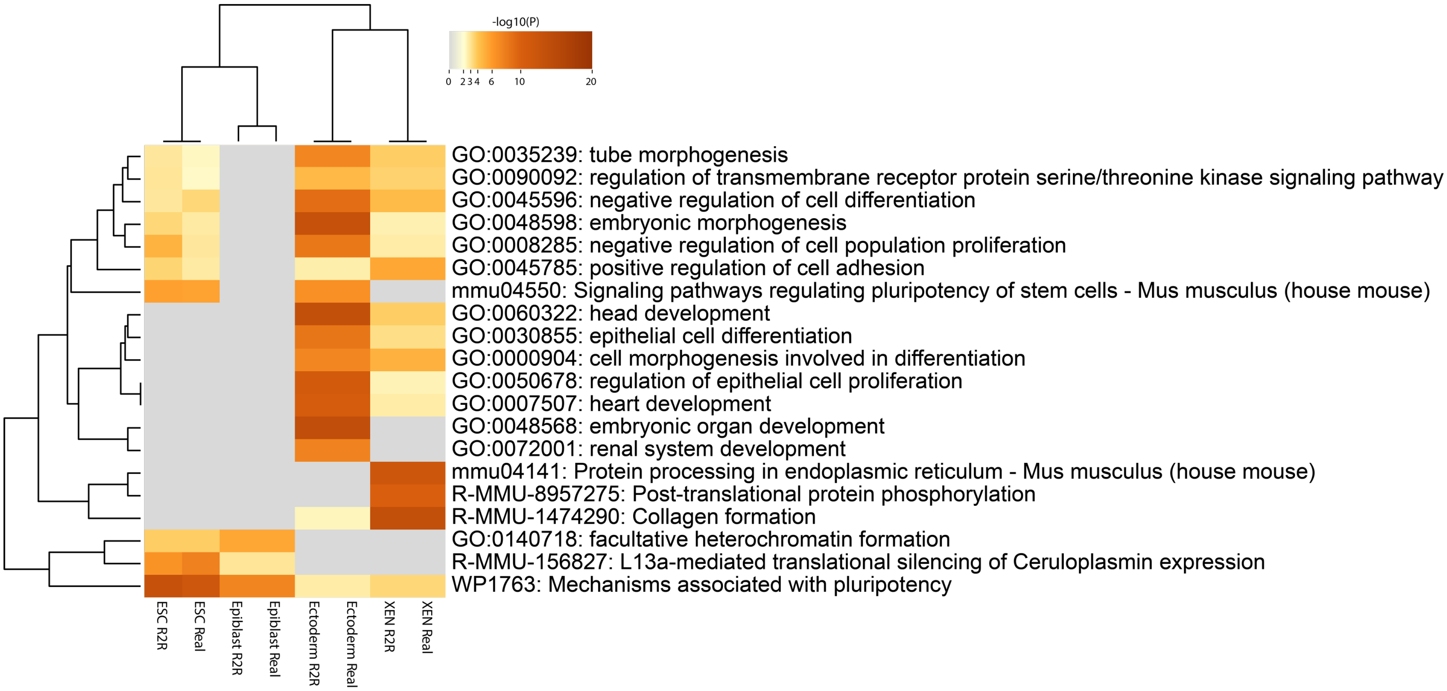
Supplementary Fig. 14 | Anchor-free predictions of scRNA-seq reflect temporal information at 0.5-day resolution. Number of cells (color bar, Z score) with anchor-free R2R generated profiles from each real time point ("true label", rows) and each classified time point (columns) as predicted by a Catboost classifier trained with day labels as ground truth with 50/50 train/test split ratio and applied to R2R-generated profiles on the test cells.



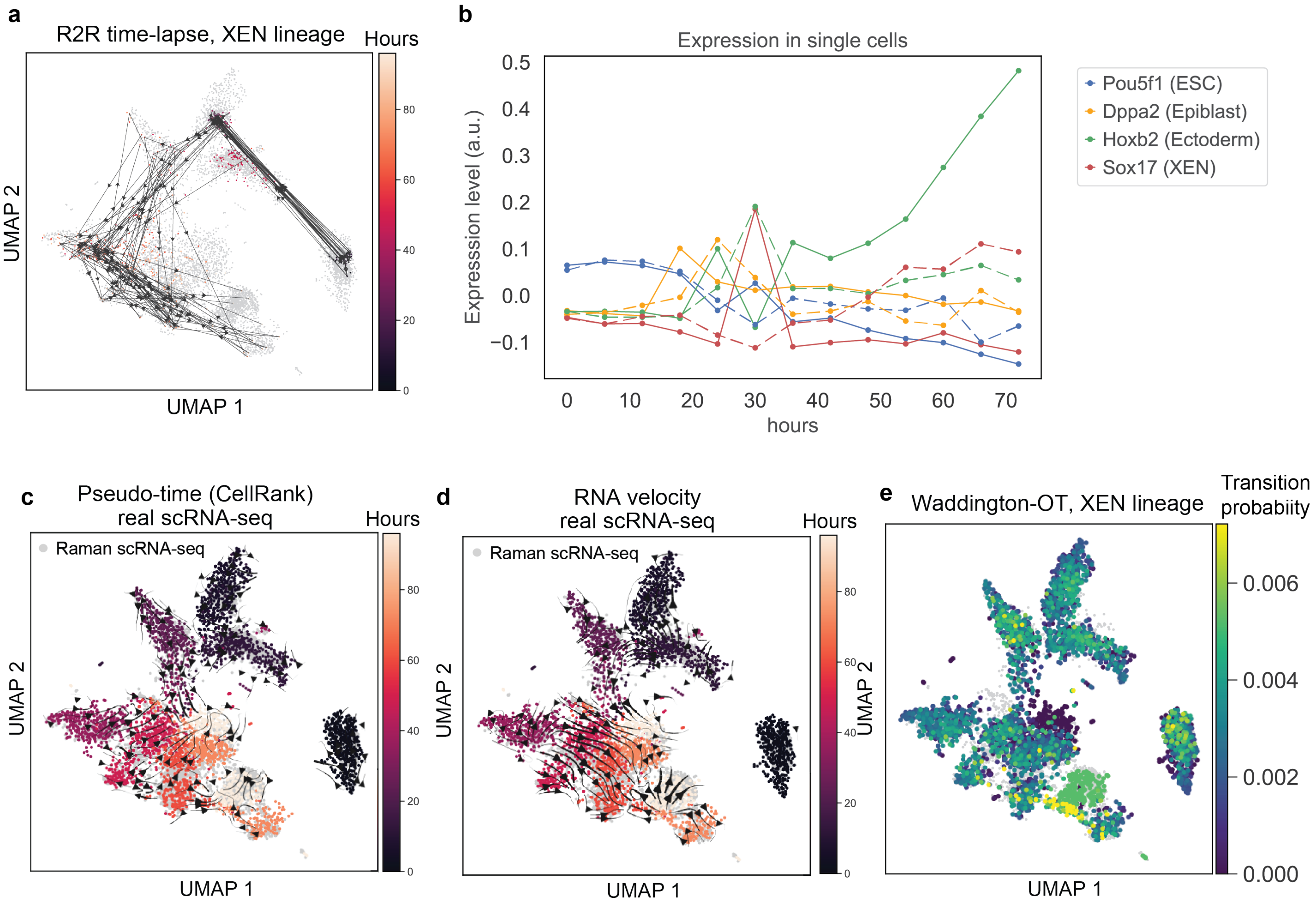
Supplementary Fig. 15. Temporal correspondence of real and R2R generated scRNA-seq profiles in mESC differentiation. Left: UMAP co-embedding of real and R2R predicted profiles (dots) colored by source of cell (same as **Fig. 4b**). Middle: UMAP co-embedding colored by Louvain cluster membership. Right: the proportion of cells (pie chart) from real (blue) or R2R (red) in each cluster (rows) at each timepoint (columns). The relative size of each cluster (out of all cells in one timepoint) is reflected by the circle size (legend). For time points 36h and 60h there are only Raman measurements.



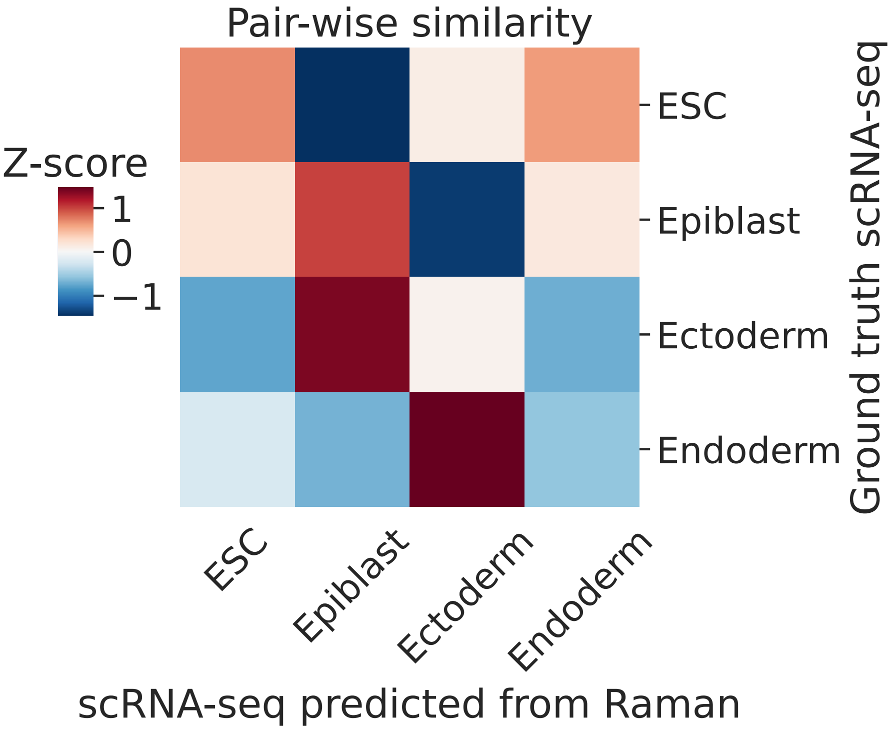
Supplementary Fig. 16 | R2R-generated profiles and real scRNA-seq co-embed well based on measured expression of key markers in smFISH and scRNA-seq. UMAP co-embedding of R2R-generated RNA profiles (on both snapshot and live-imaged cells) and measured scRNA-seq profiles (dots) colored by expression of key marker genes (rows) as determined by smFISH measurements of the test cells (left), real scRNA-seq (middle, for cells with scRNA-seq profiles), or R2R predictions (right).



Supplementary Fig. 17 | R2R profiles of mESC differentiation capture biological processes and pathways similar to ground truth scRNA-seq. Enrichment[1] (-log$_{10}$(P-value), hypergeometric test) for functional gene sets (rows) in the top 100 differentially expressed genes of each cell subset (columns) defined in real or R2R predicted scRNA-seq.



Supplementary Fig. 18. R2R allows live tracking of cell profiles to resolve differentiation relations that are challenging to capture with computational trajectory inference methods applied to scRNA-seq. (**a, c-e**) Co-embedding of R2R and scRNA-seq profiles (as in **Fig. 4g**), where R2R-generated time-lapse profiles (**a**) or real scRNA-Seq (c-e) are colored by time point (color bar) and connected respectively by either live cell tracking of their underlying cells as directly tracked by brightfield time-lapse images and cell tracking of segmented nuclei (**a**) or by pseudo-time trajectory inference (**c**) or generalized RNA velocity (**e**), from CellRank[2] with default parameters) or inferred transition probabilities from Waddington-OT for each cell profile to be an ancestor of XEN cells at day 4 (**e**). (**b**) R2R predicted expression level (y axis) of marker genes for each lineage (color) at each time point (x axis) post RA induction along XEN-like (solid) and ectoderm-like (dashed) trajectories, tracked for individual representative cells for each lineage.



Supplementary Fig. 19. R2R predictions between biological systems. Pair-wise cosine similarity (color bar) between measured (rows) and predicted (column) cell type profiles where an anchor-free model trained on the iPSC reprogramming system was applied to Raman spectra from the mESC system.

References

1. Zhou, Y. *et al.* Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. *Nat. Commun.* **10**, 1523 (2019).

2. Lange, M. *et al.* CellRank for directed single-cell fate mapping. *Nat. Methods* **19**, 159–170 (2022).

A how-to guide for setting up multi-modal Raman microscope

Overview of Hardware and Software

Our Raman microscope is built on a standard, inverted fluorescence microscope with custom code to carry out a multi-modal, -time, and -positional confocal Raman microscopy measurement. Although building a Raman microscope is reasonably straightforward and well-documented (see **Methods** and Ref. 1, except make sure to use a short-pass dichroic filter to allow for a dual fluorescent/Raman capable microscope and dual axis galvo mirror for XY scanning), to the best of our knowledge **no previously existing open-source software is able to support all multi-modal/time/position measurement** on a galvo mirror-based confocal Raman microscope, which is necessary for live-cell time-lapse imaging. This is because Raman microscopes inherently obtain high-dimensional **hyperspectral images**, whereas standard commercial or open-source microscope control software are designed for 2D or, at most, 4D (XYZ and time) images with specific hardware requirements.

To avoid the need for creating such software for Raman from scratch, we took a hybrid approach – we carried out the traditional fluorescence imaging based on µManager[2], an open-source microscope control software, and combined it with a custom MATLAB script to carry out the galvo scanning, laser shutter control, and camera readout (**Fig. H1** and **Methods**). To facilitate inter-software communication between µManager and MATLAB, we used a digital acquisition (DAQ) board to send trigger signals from µManager to MATLAB so the Raman acquisition sequence begins, and the camera/galvo mirror/laser shutter can be controlled through MATLAB. Overall, the basic fluorescence and multi-time/positional control and measurements are carried out by µManager, while Raman imaging is tricked in µManager as a 'demo' image, where in fact, a trigger is sent to the DAQ board and Raman imaging commences. We describe below in detail the required optics and software components and how to assemble them.

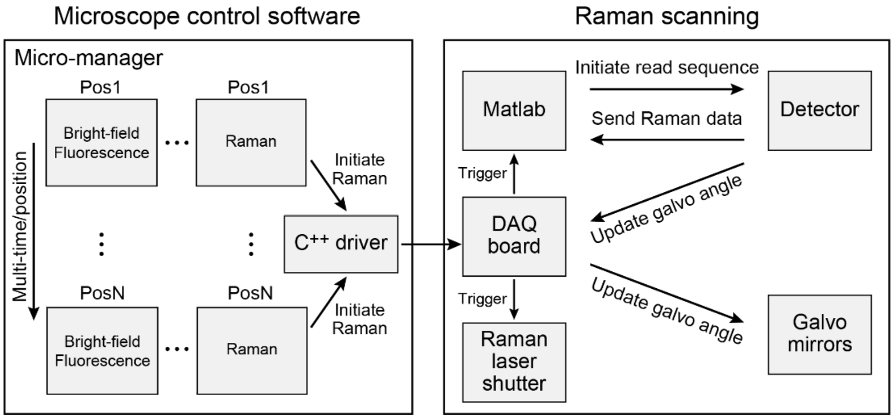


Fig. H1: Structure of microscope control software

Optics requirement

- Nd:YVO4 laser (Spectra Physics, Millennia)

- Continuous-wave (CW) Ti:Sapphire laser (Spectra Physics, model no. 3900S)
- Optical table, size > 1,200 mm × 3,000 mm × 457 mm
- Fluorescence microscope (Olympus IX83 equipped with Orca Flash 4.0 v2)
- Stable stage (Prior, Motorized stage H117)
- Digital acquisition Board (National Instruments, PCIe-6351)
- Spectrograph (Holospec f/1.8i 785 nm model)
- Cooled CCD camera (Princeton Instruments, cat. no. PIXIS100BR)
- sCMOS camera for fluorescence (Hamamatsu Photonics, Orca Flash 4.0 v2)

Software requirement

- Windows 10 64 bit, >16GB RAM, >2 PCIe slots, 1TB storage
- µManager 2.0 (https://micro-manager.org/)
- MATLAB 2020 or newer
- MATLAB wrapper for NI-DAQmx library (https://github.com/tenss/MATLAB_DAQmx_examples)
- Light Field 6 or newer (Princeton Instruments)
- NI MAX 21.0 (National Instruments)
- Custom MATLAB scripts (`initialize_start.m`, `interrupt.m`, `destructor.m`, and `experimentDataReady.m`)

Setting up the hardware circuits, software, and procedure

1. Install all software and connect components to the computer according to vendor recommendations.
2. Connect the laser shutter to a digital output port (DO1), the two galvo mirror control wires to analog output ports (AO1/2), and the camera readout output port to a digital input port `PFI0` on the DAQ board. The `PFI0` port must be used here in order to replace the internal clock of the DAQ board with the camera readout trigger.
3. Connect a digital output port (DO2) to an analog input port (AI1) on the DAQ board. DO2 will be controlled by µManager, and AI1 will be monitored by the MATLAB script to decide whether or not to begin a Raman imaging sequence.
4. Open µManager and load the settings file (raman_fluorescence.cfg) during startup. This settings file includes settings for an Olympus IX83 microscope equipped with six fluorescence channels and additional settings for Raman imaging (channel denoted as `Raman`).
5. Mount the sample on the microscope. Be sure to use a quartz glass substrate as standard coverslips produce auto-fluorescence signals, overwhelming the Raman spectra.
6. In the Multi-Dimensional Acquisition (MDA) panel, choose all imaging channels (including brightfield, fluorescence, and Raman) and the number of time points, XY positions, and z-stacks.
7. Set exposure times for each channel. **IMPORTANT**: Give some margin to the Raman channel exposure time. During this Raman channel measurement in µManager, the MATLAB script is initiated, and scanning is

carried out. If the µManager Raman channel exposure time is shorter than the actual measurement carried out in the script, MATLAB will crash.

8. Open MATLAB and move to the directory where all custom scripts are present.
9. Rewrite all parameters in `initialize_start.m` such as pixel dwell time, scanning range, scanning step, location/name to save measurements, and Raman peak to show a 2D image after every successful measurement.
10. Run `initialize_start.m`. `initialize_start.m` will open Light Field, the Raman camera control software, and will begin monitoring the DAQ board analog input port AI1, which continuously monitors any signal coming from µManager.
11. After the message "monitoring trigger" appears, start the µManager MDA measurement. This will start measuring all channels (brightfield, fluorescence, and Raman).
12. After a successful Raman imaging, MATLAB will output a 2D plot of the Raman imaging focusing on a single peak.
13. **If you need to interrupt a measurement**: First, stop the MDA sequence in µManager. Then run `interrupt.m` in MATLAB followed by `destructor.m`. This releases and closes any objects related to Light Field or the DAQ board software. Restart MATLAB and redo from 9.
14. **To finish measurement and close software**: Assuming that the MDA sequence has been completed in µManager, run `destructor.m` and close both MATLAB and µManager**.**

References

1. Palonpon, A. F. *et al.* Raman and SERS microscopy for molecular imaging of live cells. *Nat. Protoc.* **8**, 677–692 (2013).
2. Edelstein, A., Amodaj, N., Hoover, K., Vale, R. & Stuurman, N. Computer control of microscopes using µManager. *Curr. Protoc. Mol. Biol.* **Chapter 14**, Unit14.20 (2010).