

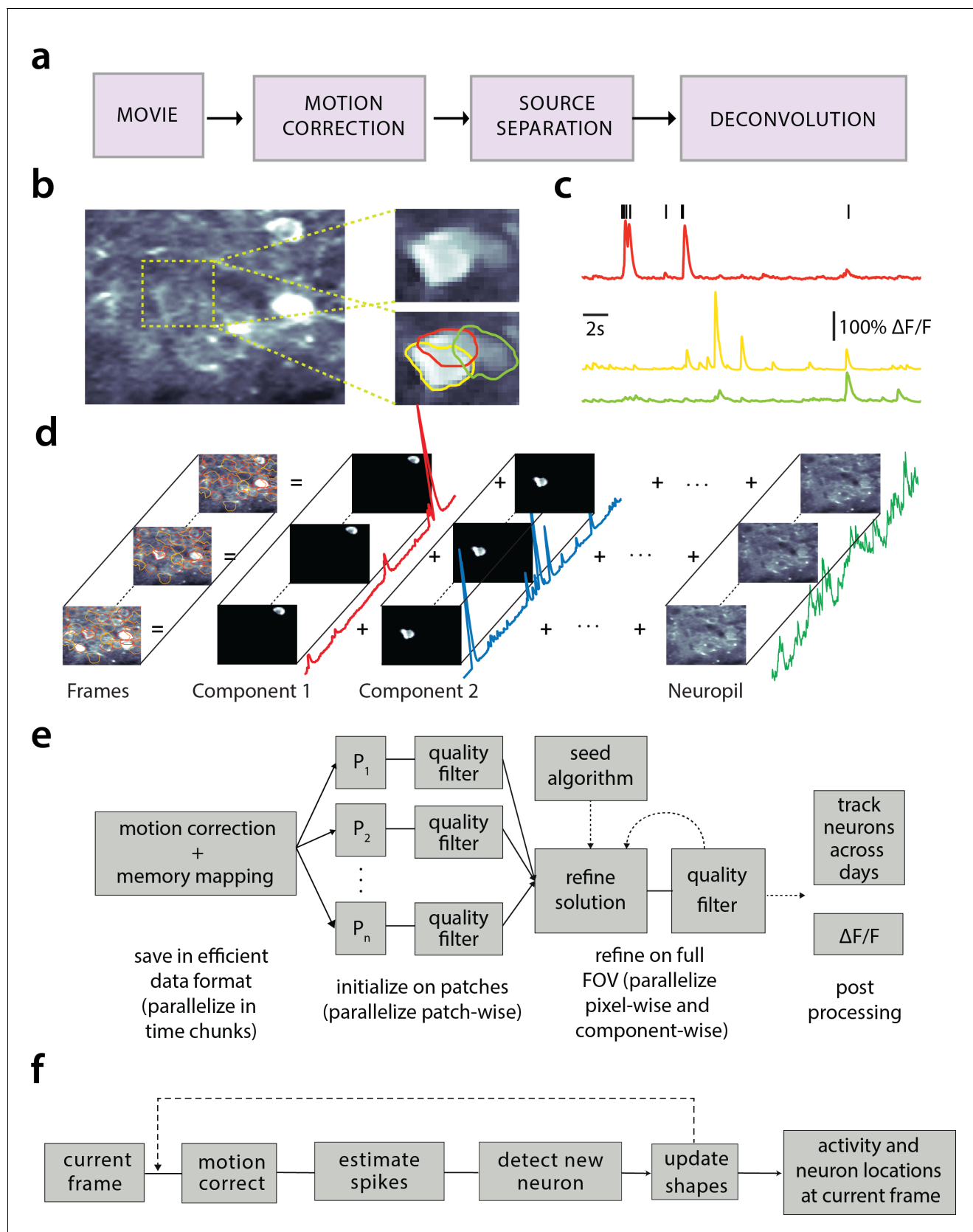


---

## Figures and figure supplements

CalmAn an open source tool for scalable calcium imaging data analysis

**Andrea Giovannucci *et al***

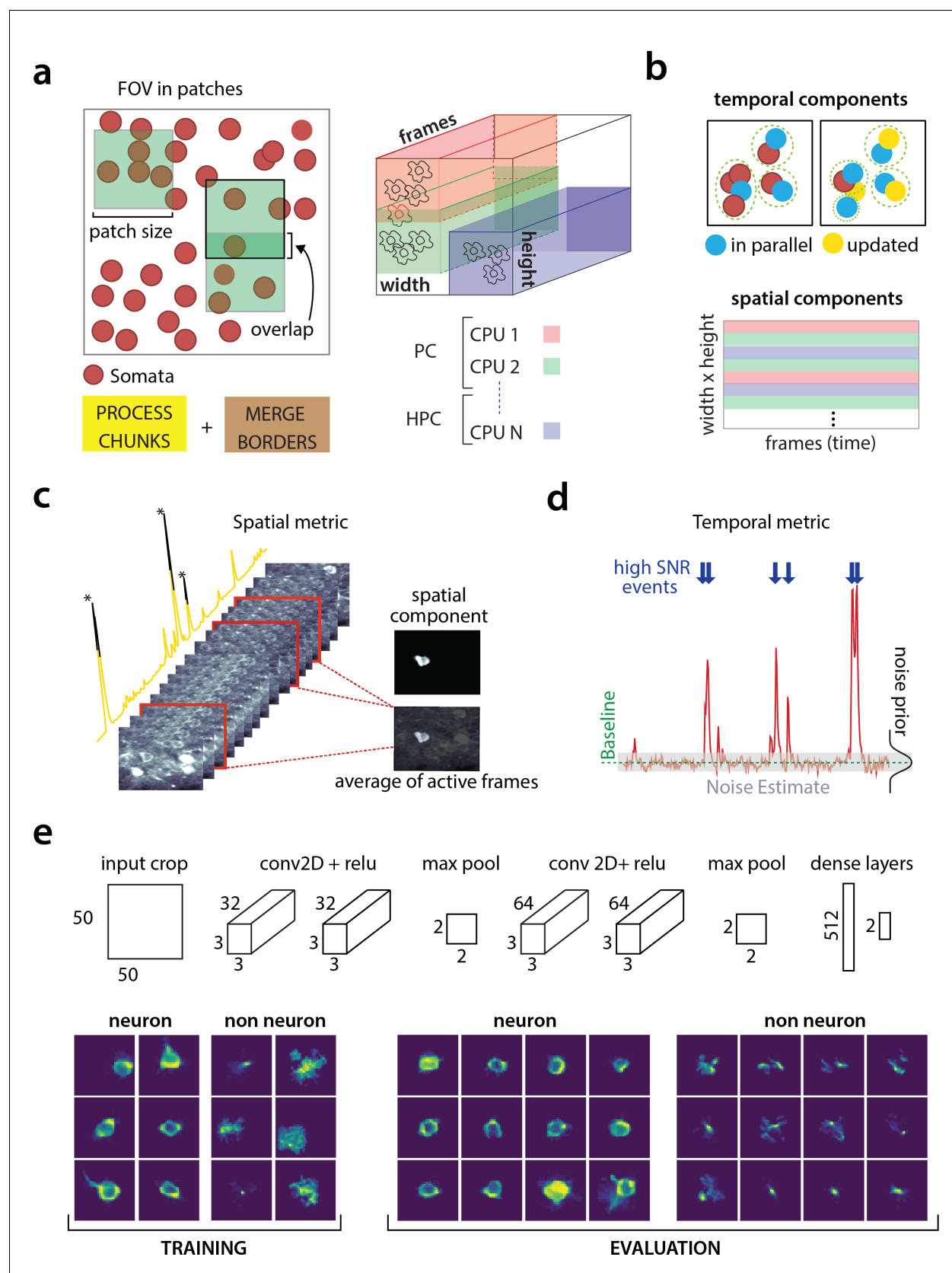


**Figure 1.** Processing pipeline of CALMAN for calcium imaging data. (a) The typical pre-processing steps include (i) correction for motion artifacts, (ii) extraction of the spatial footprints and fluorescence traces of the imaged components, and (iii) deconvolution of the neural activity from the Figure 1 continued on next page

*Figure 1 continued*

fluorescence traces. **(b)** Time average of 2000 frames from a two-photon microscopy dataset (left) and magnified illustration of three overlapping neurons (right), as detected by the CNMF algorithm. **(c)** Denoised temporal components of the three neurons in **(b)** as extracted by CNMF and matched by color (in relative fluorescence change,  $\Delta F/F$ ). **(d)** Intuitive depiction of CNMF. The algorithm represents the movie as the sum of spatially localized rank-one spatio-temporal components capturing neurons and processes, plus additional non-sparse low-rank terms for the background fluorescence and neuropil activity. **(e)** Flow-chart of the CALMAN BATCH processing pipeline. From left to right: Motion correction and generation of a memory efficient data format. Initial estimate of somatic locations in parallel over FOV patches using CNMF. Refinement and merging of extracted components via seeded CNMF. Removal of low quality components. Final domain dependent processing stages. **(f)** Flow-chart of the CALMAN ONLINE algorithm. After a brief mini-batch initialization phase, each frame is processed in a streaming fashion as it becomes available. From left to right: Correction for motion artifacts. Estimation of activity from existing neurons, identification and incorporation of new neurons. The spatial footprints of inferred neurons are also updated periodically (dashed lines).

DOI: <https://doi.org/10.7554/eLife.38173.003>



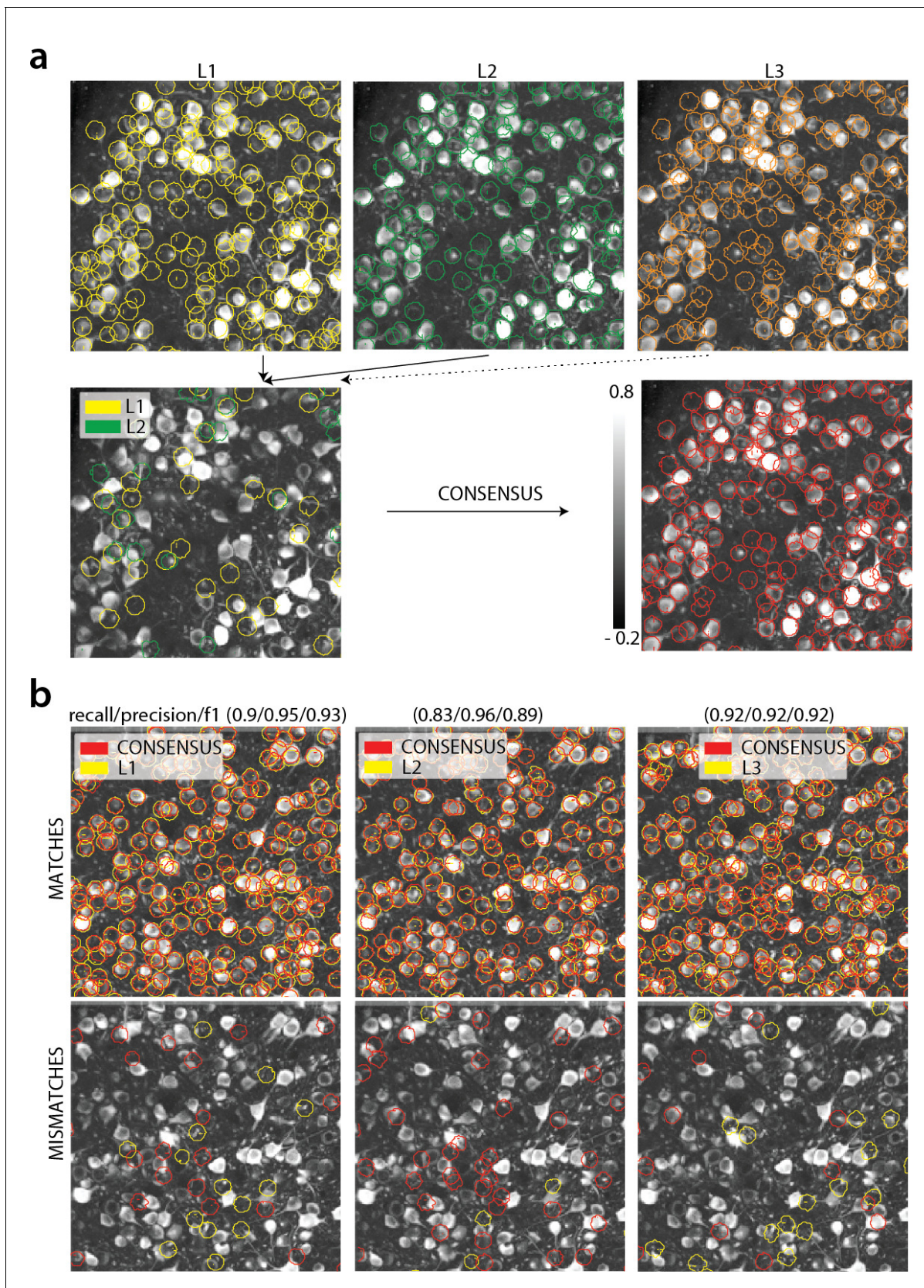
**Figure 2.** Parallelized processing and component quality assessment for CALMAN BATCH. (a) Illustration of the parallelization approach used by CALMAN BATCH for source extraction. The data movie is partitioned into overlapping sub-tensors, each of which is processed in an embarrassingly parallel fashion

Figure 2 continued on next page

*Figure 2 continued*

using CNMF, either on local cores or across several machines in a HPC. The results are then combined. **(b)** Refinement after combining the results can also be parallelized both in space and in time. Temporal traces of spatially non-overlapping components can be updated in parallel (top) and the contribution of the spatial footprints for each pixel can be computed in parallel (bottom). Parallelization in combination with memory mapping enable large scale processing with moderate computing infrastructure. **(c)** Quality assessment in space: The spatial footprint of each real component is correlated with the data averaged over time, after removal of all other activity. **(d)** Quality assessment in time: A high SNR is typically maintained over the course of a calcium transient. **(e)** CNN based assessment. *Top*: A 4-layer CNN based classifier is used to classify the spatial footprint of each component into neurons or not, see Materials and methods (*Classification through CNNs*) for a description. *Bottom*: Positive and negative examples for the CNN classifier, during training (left) and evaluation (right) phase. The CNN classifier can accurately classify shapes and generalizes across datasets from different brain areas.

DOI: <https://doi.org/10.7554/eLife.38173.004>

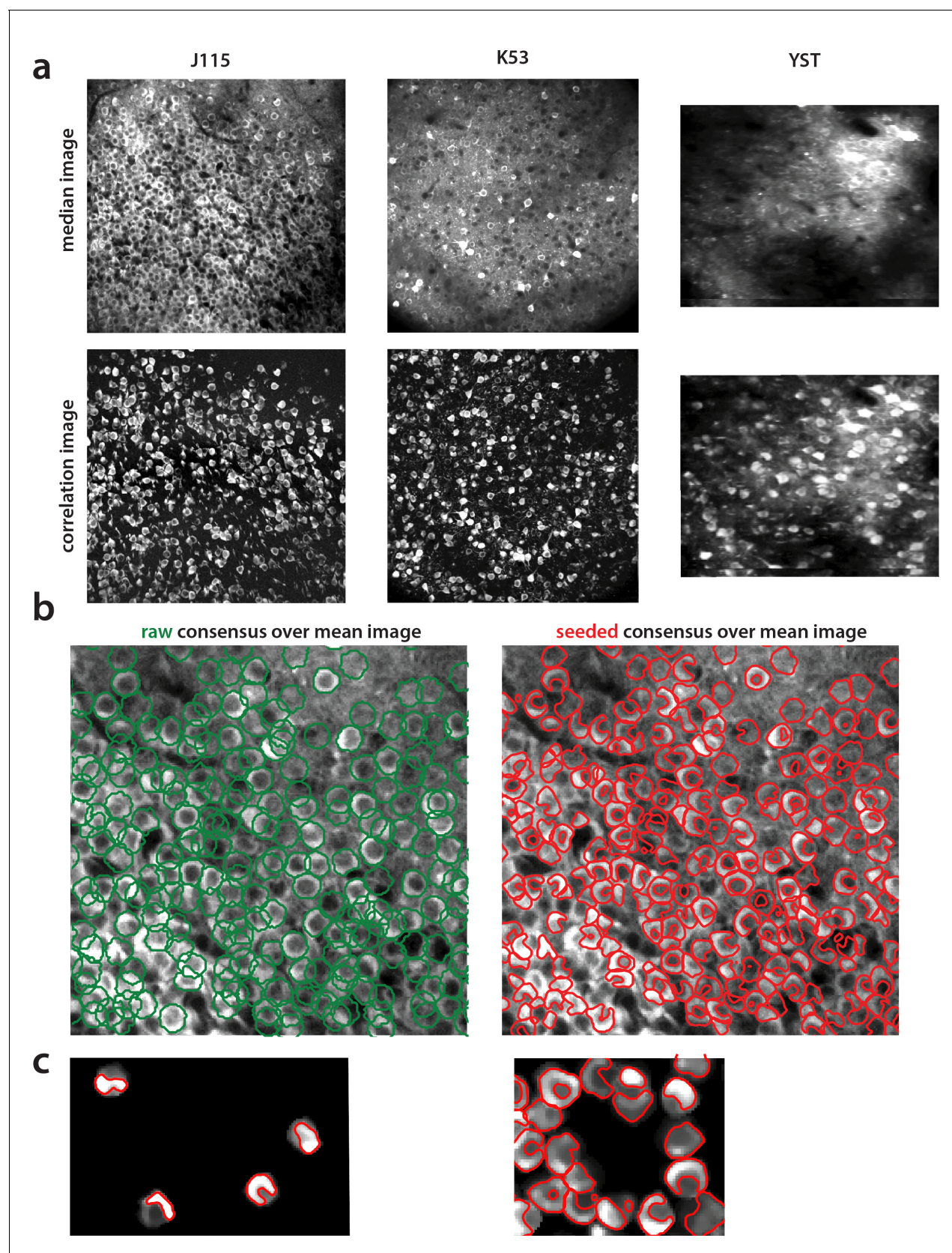


**Figure 3.** Consensus annotation generation. (a) Top: Individual manual annotations on the dataset K53 (only part of the FOV is shown) for labelers L1 (left), L2 (middle), L3 (right). Contour plots are plotted against the max-correlation image of the dataset. Bottom: Disagreements between L1 and L2. Figure 3 continued on next page

*Figure 3 continued*

(left), and consensus labels (right). In this example, consensus considerably reduced the number of initially selected neurons. **(b)** Matches (top) and mismatches (bottom) between each individual labeler and consensus annotation. Red contours on the mismatches panels denote false negative contours, that is components in the consensus not selected by the corresponding labeler, whereas yellow contours indicate false positive contours. Performance of each labeler is given in terms of precision/recall and  $F_1$  score and indicates an unexpected level of variability between individual labelers.

DOI: <https://doi.org/10.7554/eLife.38173.005>

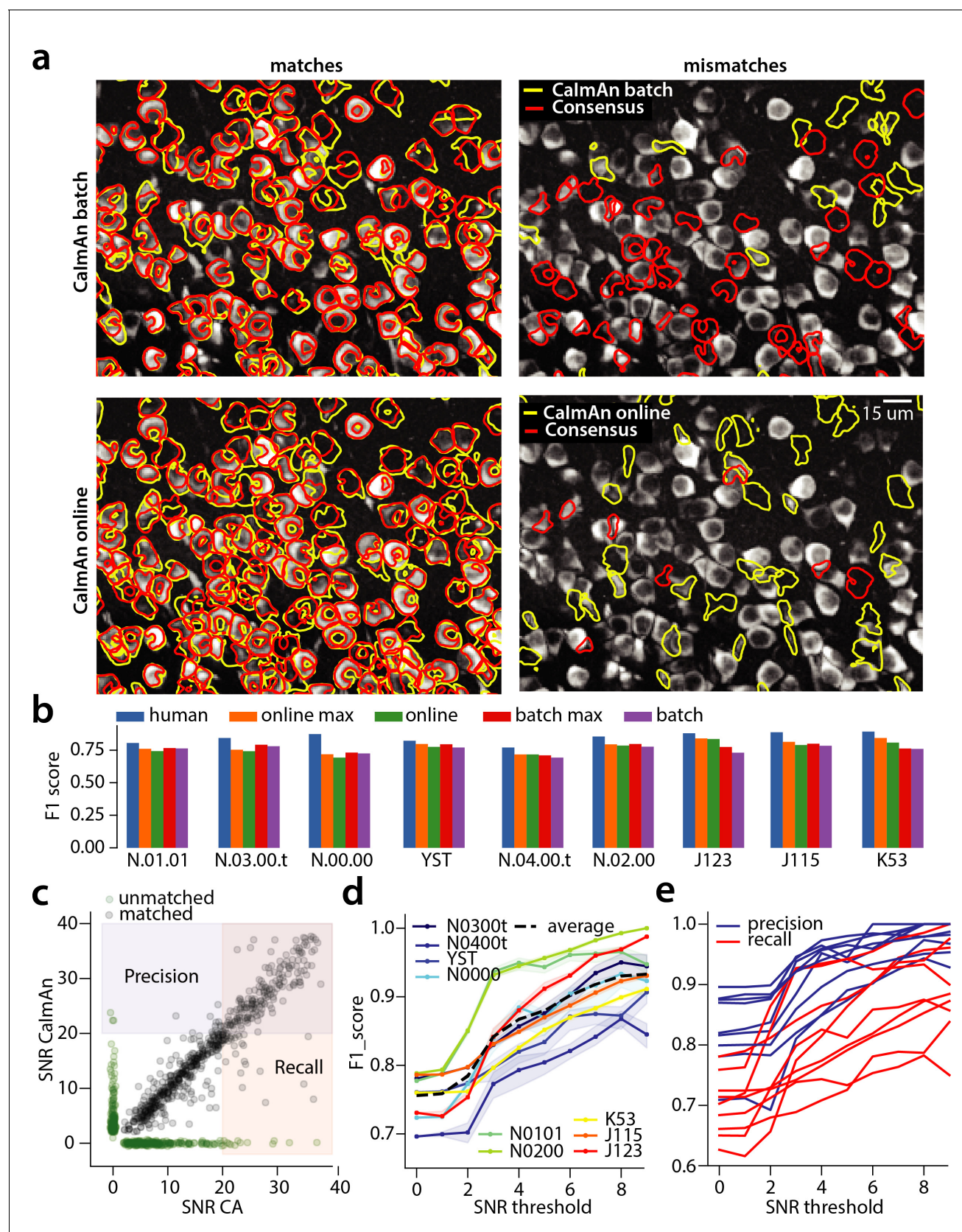


**Figure 3—figure supplement 1.** Construction of components obtained from consensus annotation. (a) Correlation image can efficiently display active neurons. Comparison of median across time (top) and max-correlation (bottom) image for annotated datasets J115 (left), K53 (middle) and YST (right).  
 Figure 3—figure supplement 1 continued on next page

*Figure 3—figure supplement 1 continued*

In all cases, the correlation image aids in manual annotation by providing an efficient way to remove neuropil contamination and visualize the footprints of active neurons. **(b)** Contour plots of manual annotations (left) vs spatial footprints obtained after running SeededInitialization (right), for dataset J115 overlaid against the mean image. Manual annotations are restricted to be of ellipsoid shape whereas pre-processing with SeededInitialization allows the spatial footprints to adapt to the footprint of each neuron in the FOV. **(c)** Thresholding of spatial footprints selects the most prominent part of each neuron for comparison against ground truth. Left. Four examples of non thresholded components overlaid to their corresponding contours. Right. Same as left, but including all neurons within a small region. Finding an optimal threshold to generate consistent binary masks can be challenging.

DOI: <https://doi.org/10.7554/eLife.38173.006>

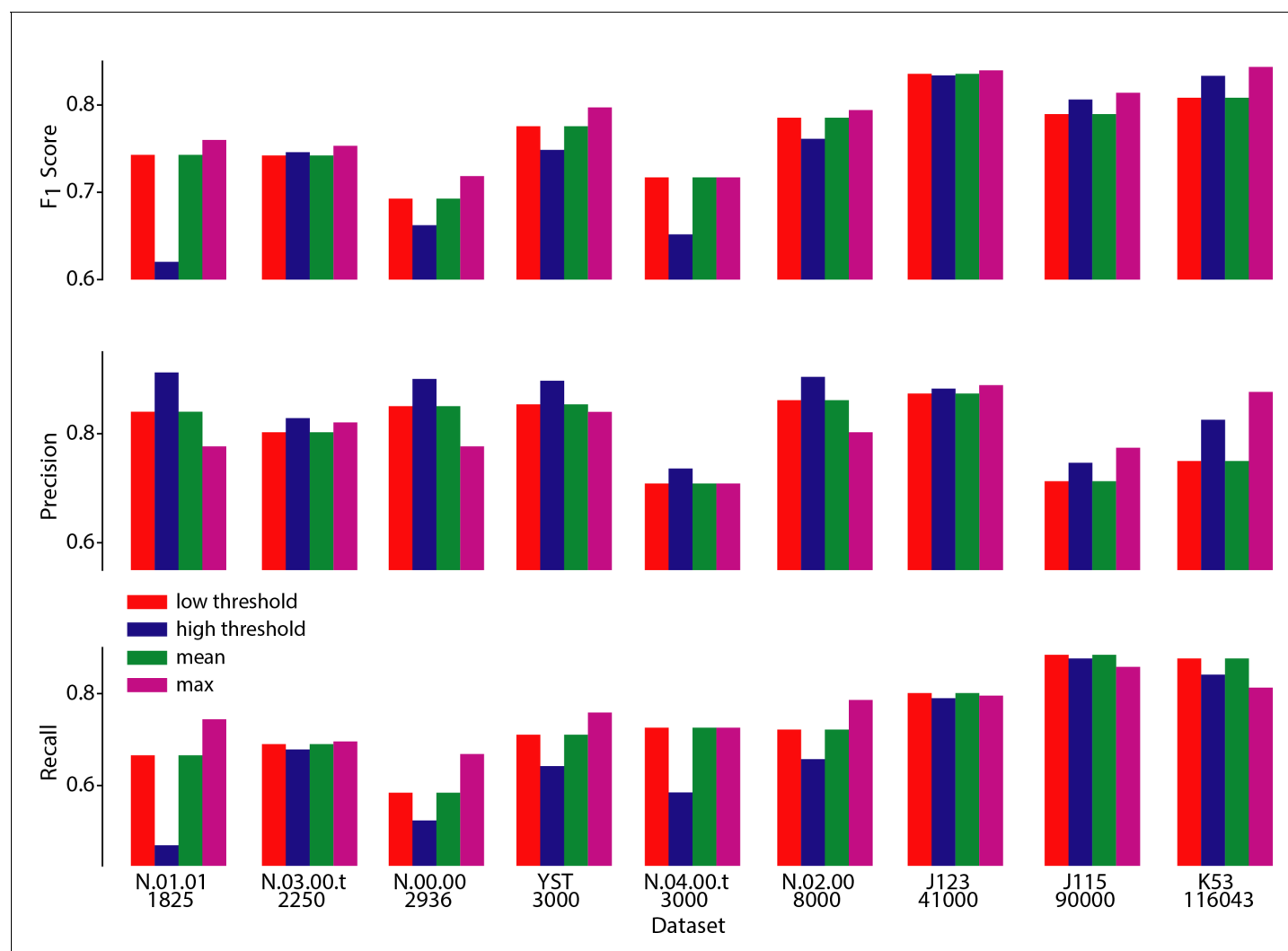


**Figure 4.** Evaluation of CALMAN performance against manually annotated data. (a) Comparison of CALMAN BATCH (top) and CALMAN ONLINE (bottom) when benchmarked against consensus annotation for dataset K53. For a portion of the FOV, correlation image overlaid with matches (left panels, red: Figure 4 continued on next page

*Figure 4 continued*

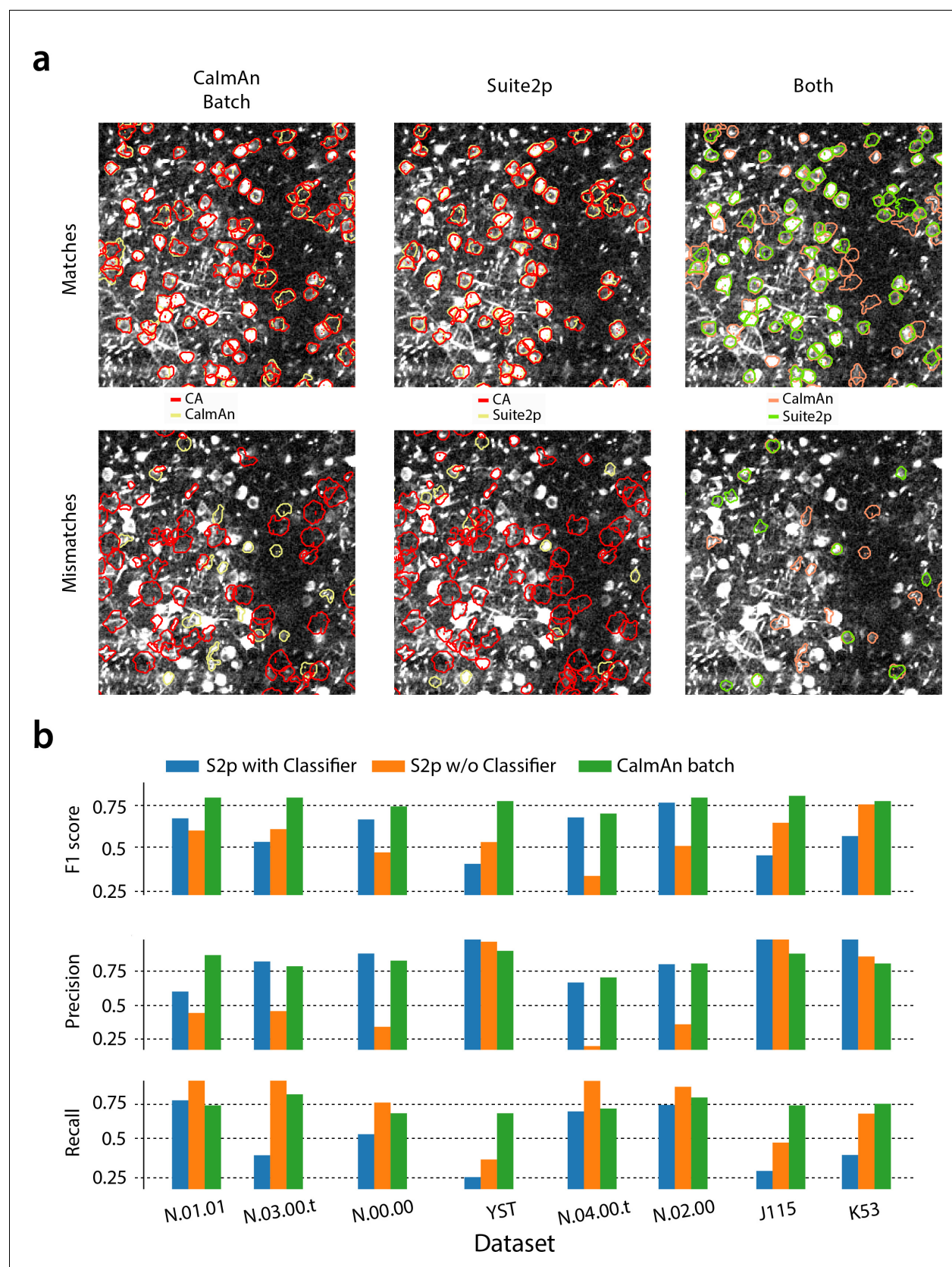
consensus, yellow: CALMAN) and mismatches (right panels, red: false negatives, yellow: false positives). **(b)** Performance of CALMAN BATCH, and CALMAN ONLINE vs average human performance (blue). For each algorithm the results with both the same parameters for each dataset and with the optimized per dataset parameters are shown. CALMAN BATCH and CALMAN ONLINE reach near-human accuracy for neuron detection. Complete results with precision and recall for each dataset are given in **Table 1**. **(c–e)** Performance of CALMAN BATCH increases with peak SNR. **(c)** Example of scatter plot between SNRs of matched traces between CALMAN BATCH and consensus annotation for dataset K53. False negative/positive pairs are plotted in green along the x- and y-axes respectively, perturbed as a point cloud to illustrate the density. Most false positive/negative predictions occur at low SNR values. Shaded areas represent thresholds above which components are considered for matching (blue for CALMAN BATCH and red for consensus selected components) **(d)**  $F_1$  score and upper/lower bounds of CALMAN BATCH for all datasets as a function of various peak SNR thresholds. Performance of CALMAN BATCH increases significantly for neurons with high peak SNR traces (see text for definition of metrics and the bounds). **(e)** Precision and recall of CALMAN BATCH as a function of peak SNR for all datasets. The same trend is observed for both precision and recall.

DOI: <https://doi.org/10.7554/eLife.38173.008>



**Figure 4—figure supplement 1.** Performance of CALMAN ONLINE over different choices of parameters. Performance of CALMAN ONLINE over different choices of three parameters:  $\theta = (\text{TraceSNR threshold, CNN threshold, \# of candidate components})$  scores (top), Precision (middle) and Recall (bottom) are shown for all labeled datasets for four different cases: low threshold/large number setting (red) where  $\theta = (1.2, 0.65, 10)$ , high threshold/low number setting where  $\theta = (2, 0.6, 5)$ , and setting that maximizes performance averaged over all datasets (green)  $\theta = (1.2, 0.65, 10)$ . The maximum  $F_1$  score (and corresponding precision/recall) for each dataset is also shown (magenta). Lower threshold settings are more desirable for shorter datasets (N.03.00.t, N.04.00.t, N.00.00, N.01.01, YST) because they achieve high recall rates without a big penalty in precision. On the contrary, higher threshold settings are more desirable for longer datasets (J115, J123, K53) because they achieve high precision without a big penalty in recall.

DOI: <https://doi.org/10.7554/eLife.38173.009>

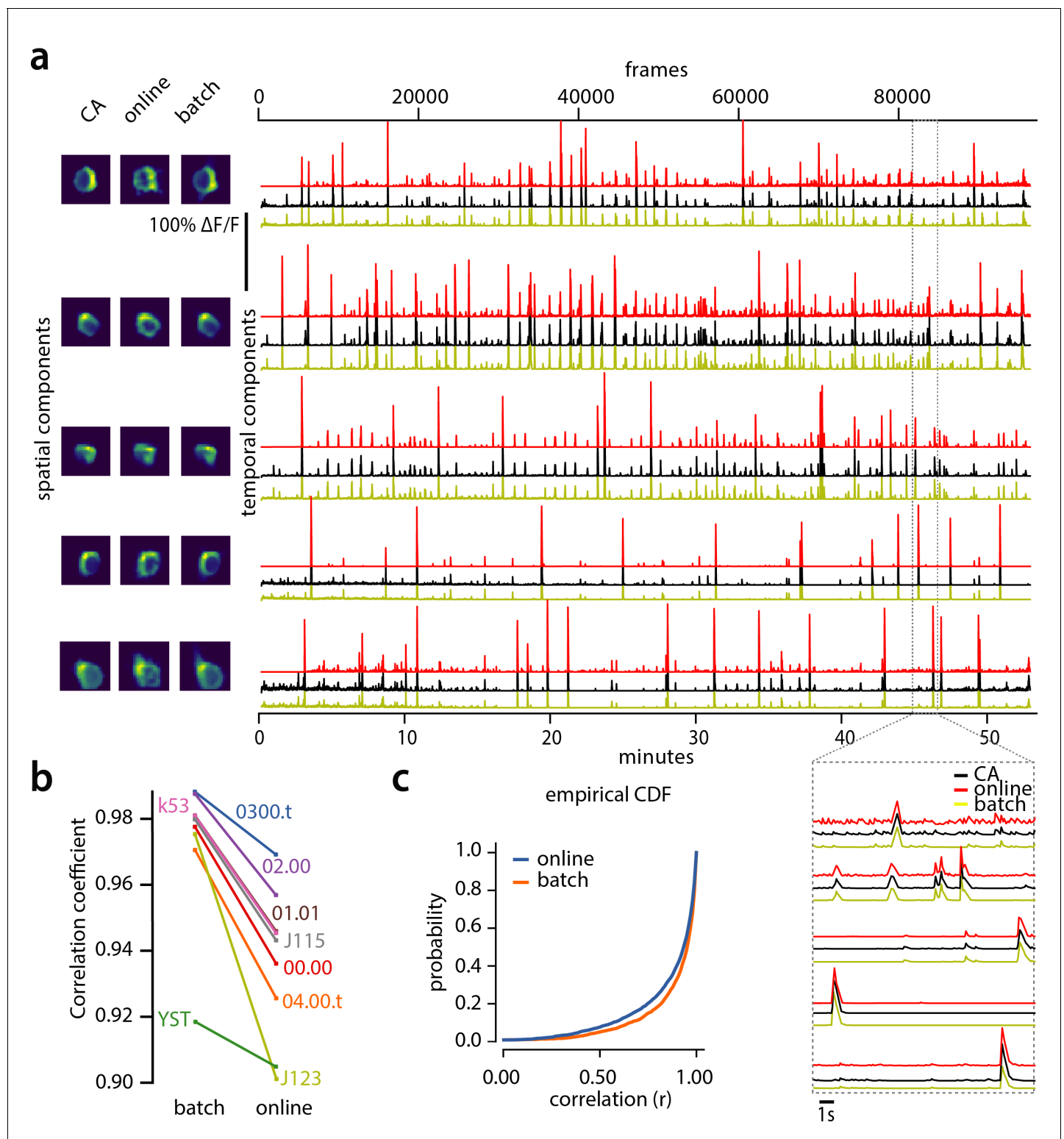


**Figure 4—figure supplement 2.** Calman BATCH outperforms the Suite2p algorithm in all datasets when benchmarked against the consensus annotation. (a) Contour plots of selected components against consensus annotation (CA) for (left) and Suite2p with the use of a classifier (middle) and direct Figure 4—figure supplement 2 continued on next page

*Figure 4—figure supplement 2 continued*

comparison between the algorithms (right) for the test dataset N.00.00. identifies better components with a weak footprint in the summary correlation image. **(b)** Performance metrics  $F_1$  score (top), precision (middle) and recall (bottom), for Suite2p (with and the without the use of the classifier) and for the eight test datasets. consistently outperforms Suite2p, which can have significant variations between precision and recall. See Materials and methods (*Comparison with Suite2p*) for more details on the comparison.

DOI: <https://doi.org/10.7554/eLife.38173.010>



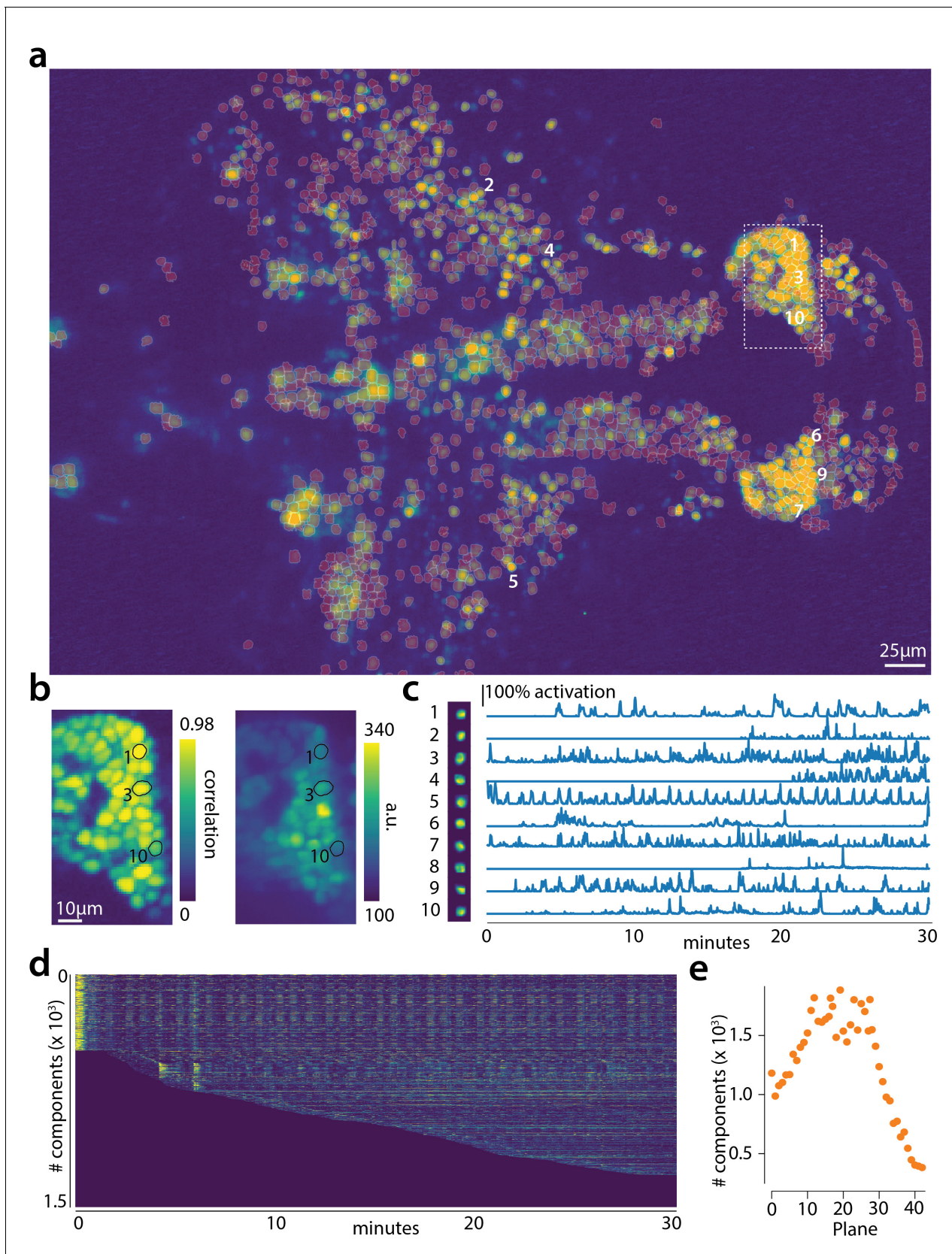
**Figure 5.** Evaluation of CALMAN extracted traces against traces derived from consensus annotation. (a) Examples of shapes (left) and traces (right) are shown for five matched components extracted from dataset K53 for consensus annotation (CA, black), CALMAN BATCH (yellow) and CALMAN ONLINE (red) algorithms. The dashed gray portion of the traces is also shown magnified (bottom-right). Consensus spatial footprints and traces were obtained by seeding CALMAN with the consensus binary masks. The traces extracted from both versions of CALMAN match closely the consensus traces. (b) Slope graph for the average correlation coefficient for matches between consensus and CALMAN BATCH, and between consensus and CALMAN ONLINE. Batch processing produces traces that match more closely the traces extracted from the consensus labels. (c) Empirical cumulative distribution functions of

*Figure 5 continued on next page*

*Figure 5 continued*

correlation coefficients aggregated over all the tested datasets. Both distributions exhibit a sharp derivative close to 1 (last bin), with the batch approach giving better results.

DOI: <https://doi.org/10.7554/eLife.38173.012>

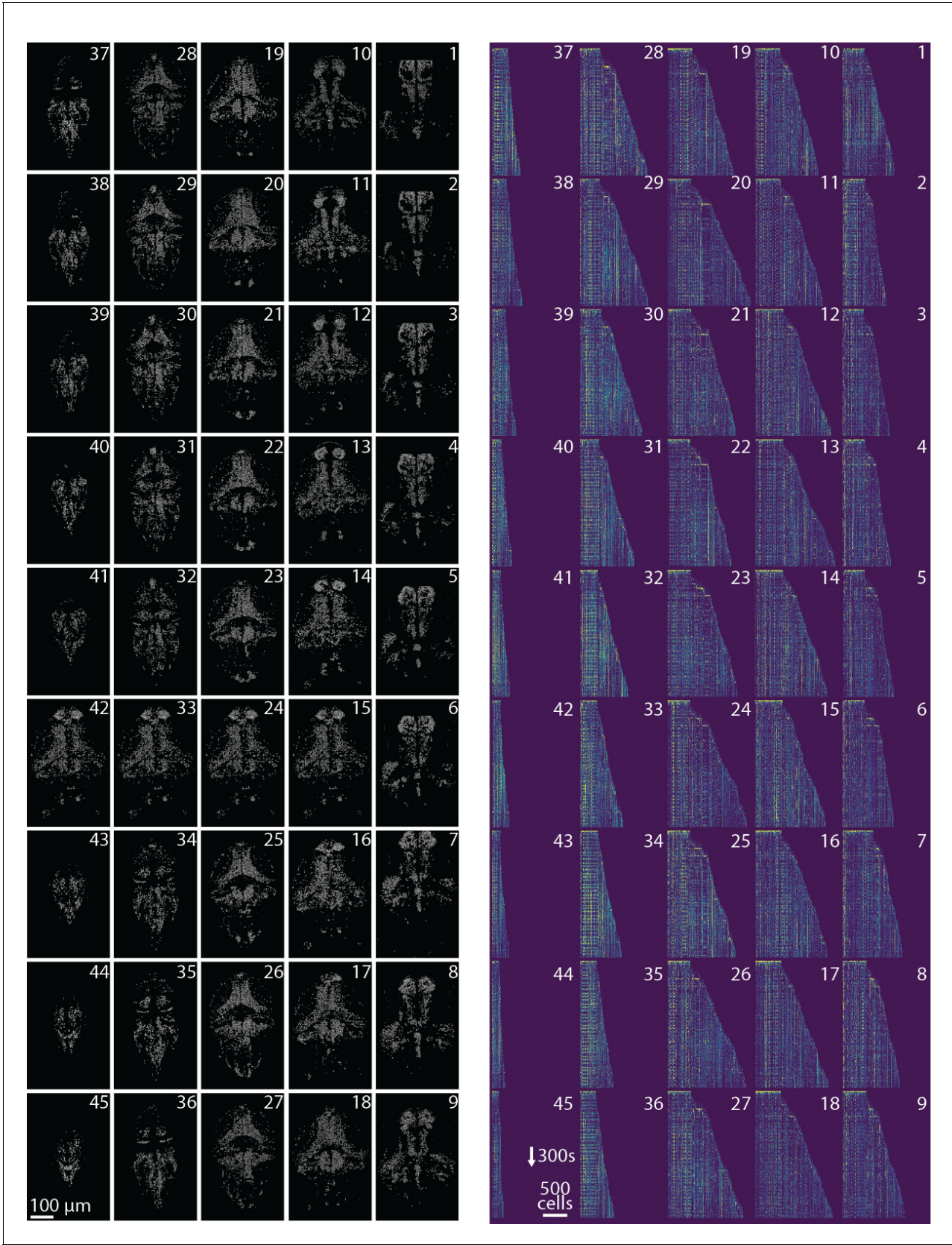


**Figure 6.** Online analysis of a 30 min long whole brain recording of the zebrafish brain. (a) Correlation image overlaid with the spatial components (in red) found by the algorithm (portion of plane 11 out of 45 planes in total). (b) Correlation image (left) and mean image (right) for the dashed region in Figure 6 continued on next page

*Figure 6 continued*

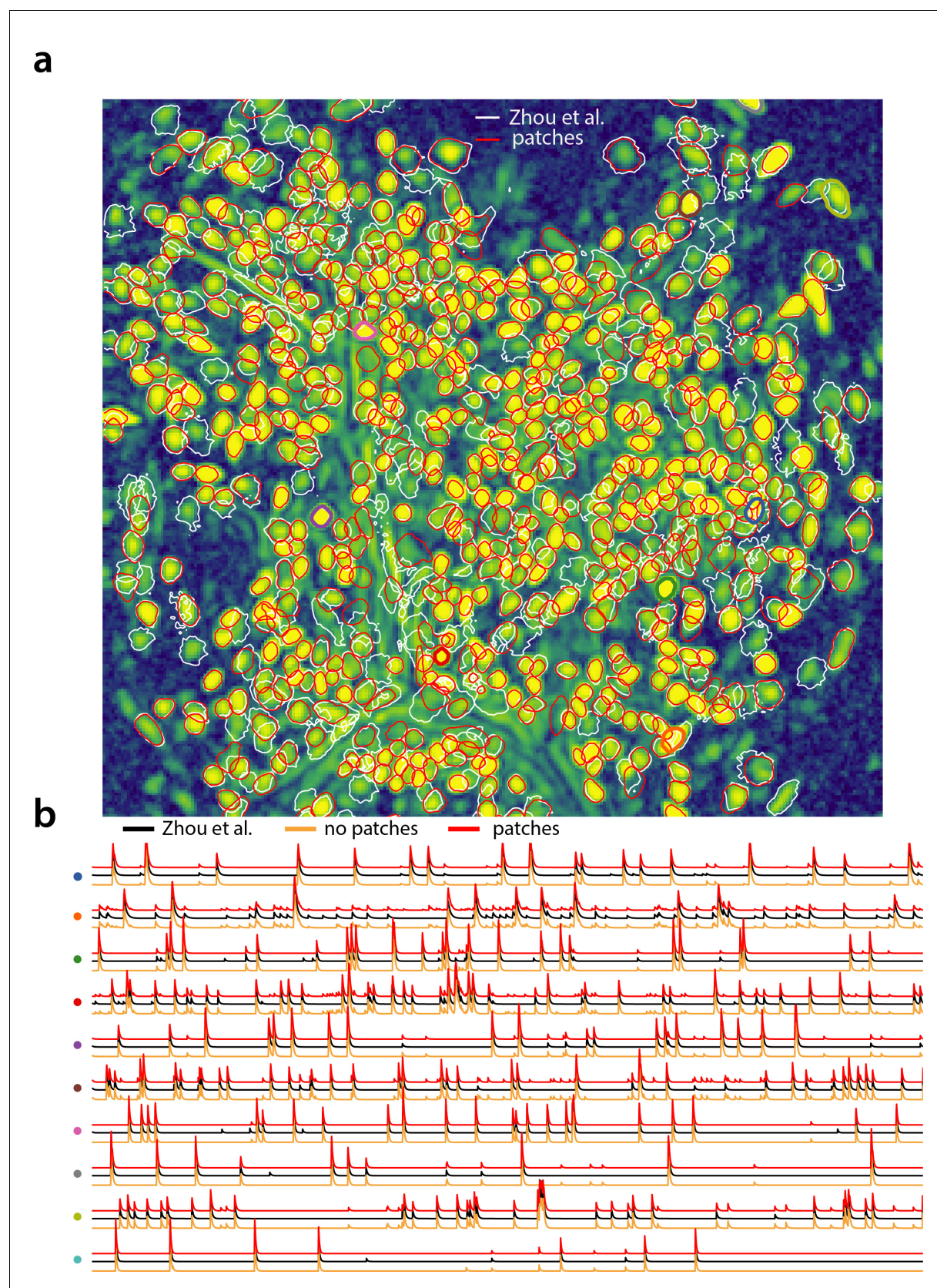
panel (a) with superimposed the contours of the neurons marked in (a). (c) Spatial (left) and temporal (right) components associated to the ten example neurons marked in panel (a). (d) Temporal traces for all the neurons found in the FOV in (a); the initialization on the first 200 frames contained 500 neurons (present since time 0). (e) Number of neurons found per plane (See also **Figure 6—figure supplement 1** for a summary of the results from all planes).

DOI: <https://doi.org/10.7554/eLife.38173.013>



**Figure 6—figure supplement 1.** Spatial and temporal components for all planes. Profile of spatial (left) and temporal (right) components found in each plane of the whole brain zebrafish recording. (Left) Components are extracted with CALMAN ONLINE and then max-thresholded. (Right) See Results section for a complete discussion. .

DOI: <https://doi.org/10.7554/eLife.38173.014>

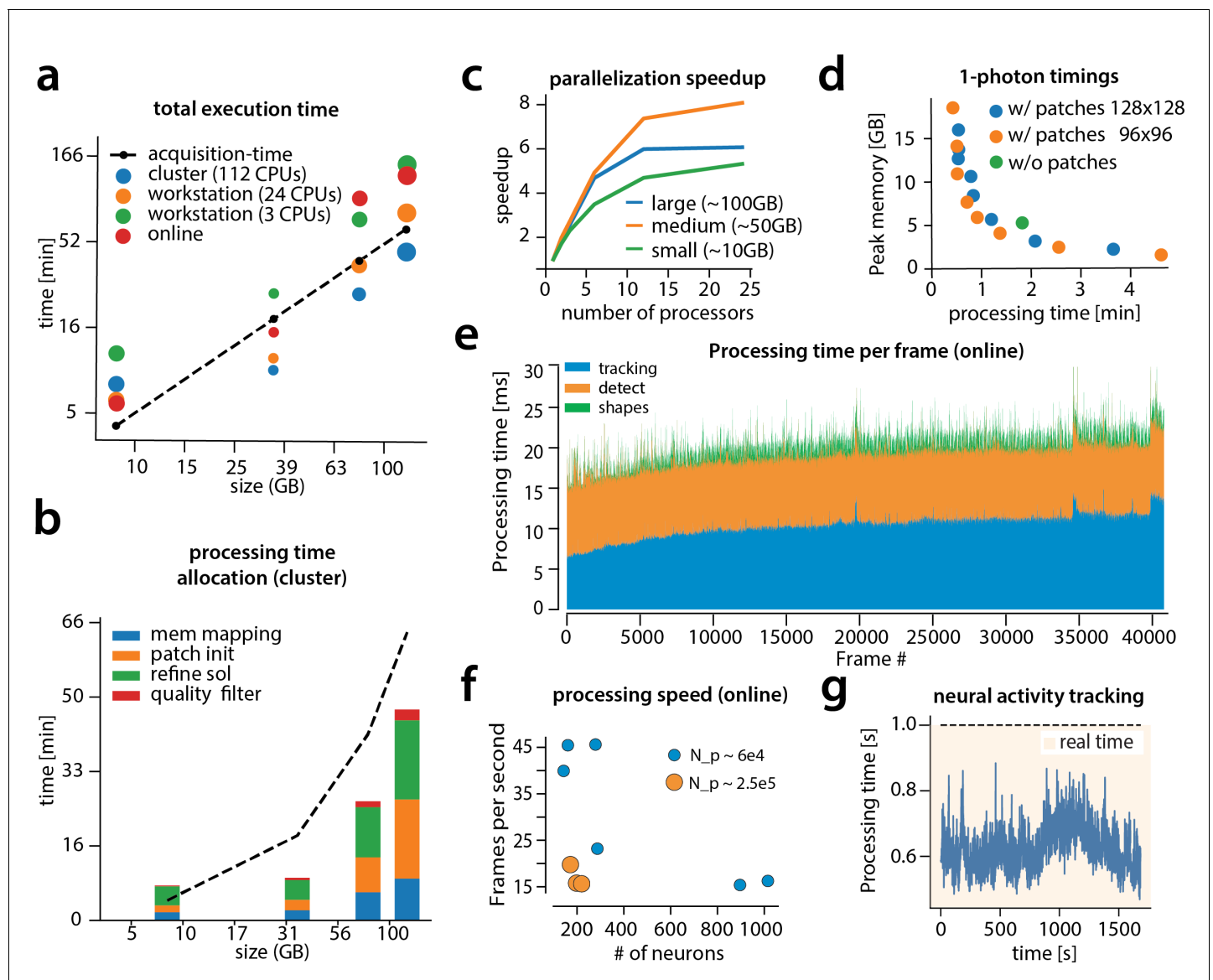


**Figure 7.** Analyzing microendoscopic 1 p data with the CNMF-E algorithm using CALMAN BATCH . (a) Contour plots of all neurons detected by the CNMF-E (white) implementation of **Zhou et al. (2018)** and CALMAN BATCH (red) using patches. Colors match the example traces shown in (b), which *Figure 7 continued on next page*

Figure 7 continued

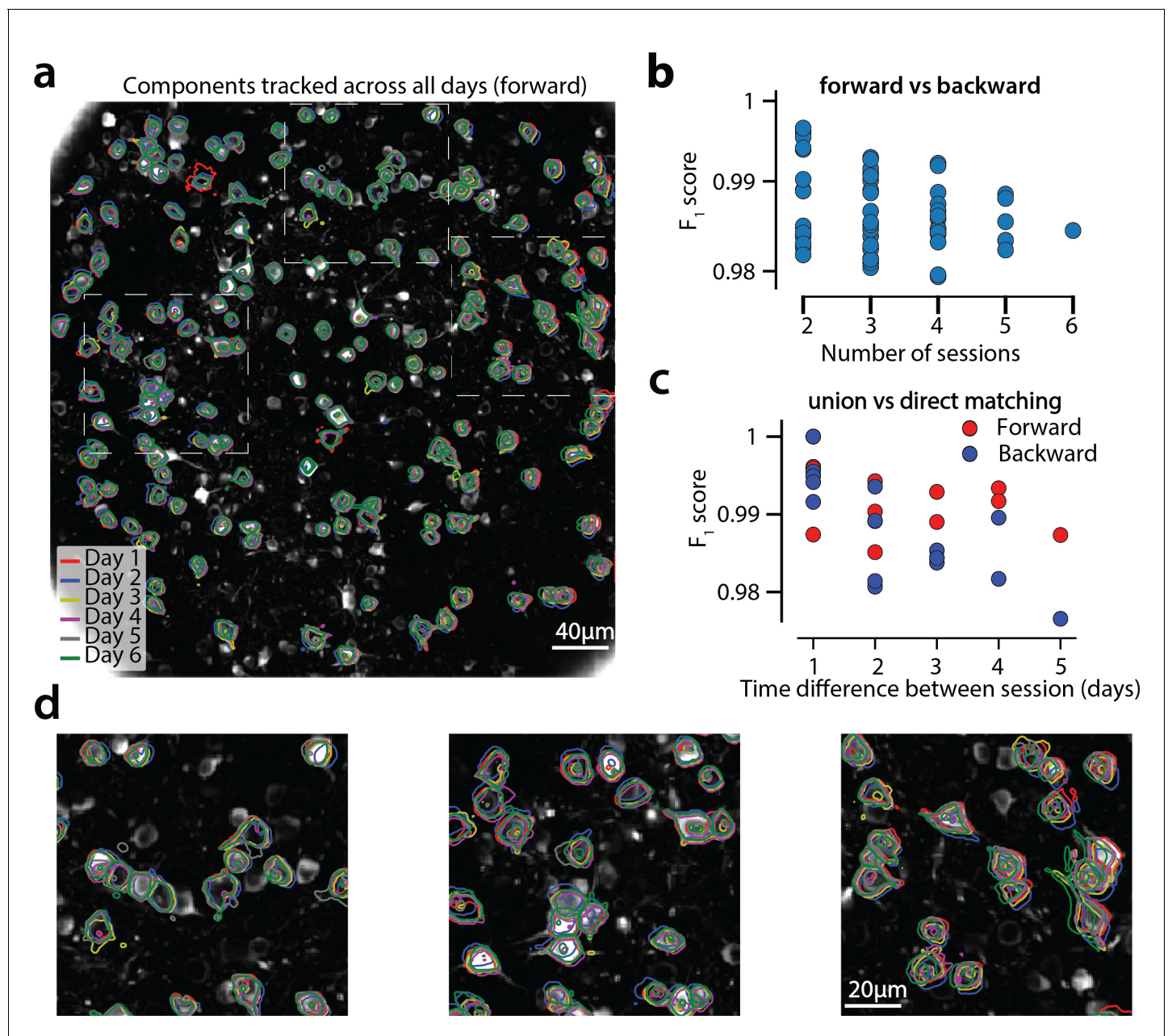
illustrate the temporal components of 10 example neurons detected by both implementations CALMAN BATCH . reproduces with reasonable fidelity the results of **Zhou et al. (2018)**.

DOI: <https://doi.org/10.7554/eLife.38173.018>



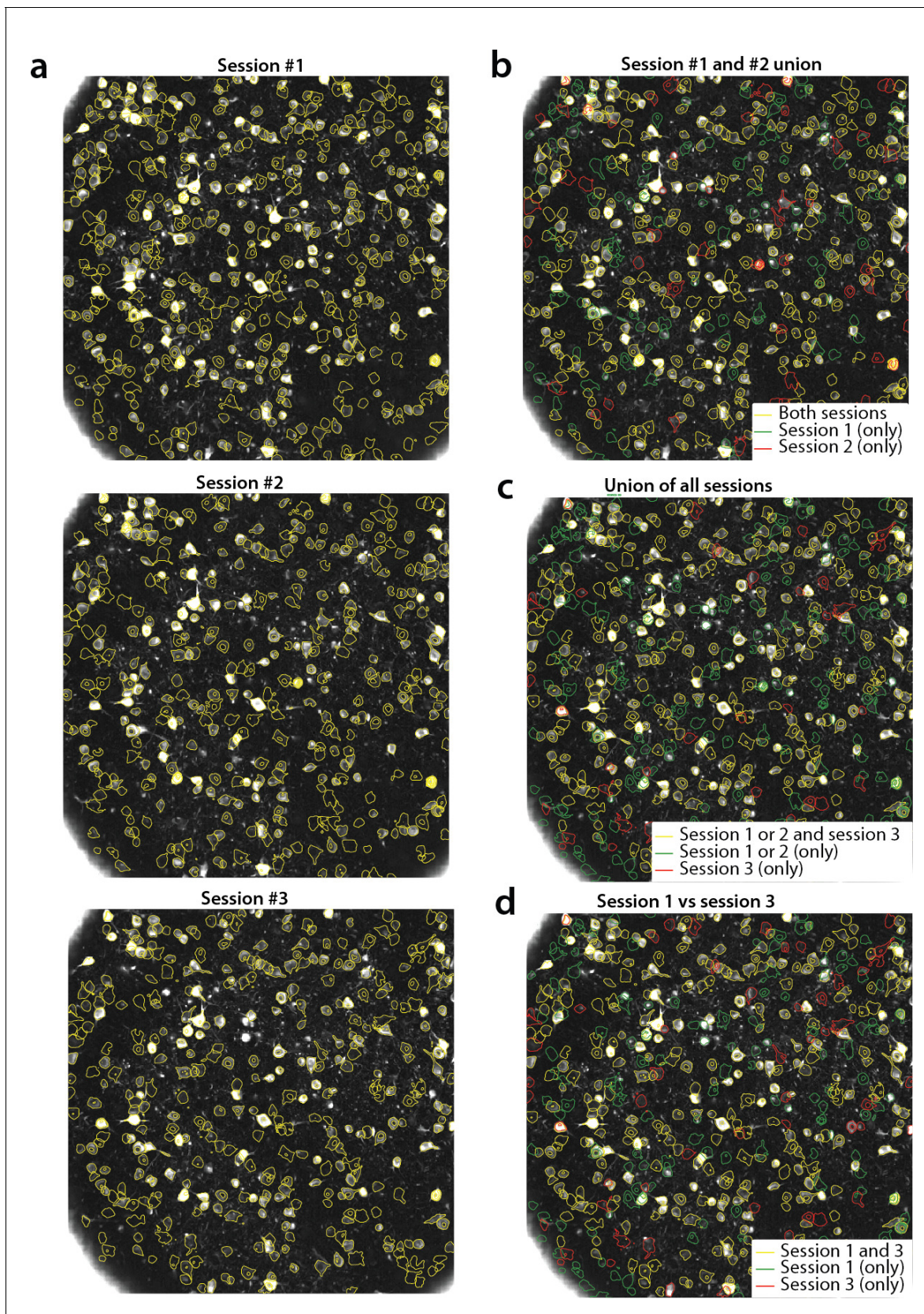
**Figure 8.** Time performance of CALMAN BATCH and CALMAN ONLINE for four of the analyzed datasets (small, medium, large and very large). (a) Log-log plot of total processing time as a function of data size for CALMAN BATCH two-photon datasets using different processing infrastructures: (i) a desktop with three allocated CPUs (green), (ii) a desktop with 24 CPUs allocated (orange), and (iii) a HPC where 112 CPUs are allocated (blue). The results indicate a near linear scaling of the processing time with the size of dataset, with additional dependence on the number of found neurons (size of each point). Large datasets (>100 GB) can be seamlessly processed with moderately sized desktops or laptops, but access to a HPC enables processing with speeds faster than the acquisition time (considered 30 Hz for a  $512 \times 512$  FOV here). However, for smaller datasets the advantages of adopting a cluster vanishes, because of the inherent overhead. The results of CALMAN ONLINE using the laptop, using the 'strict' parameter setting (**Figure 4—figure supplement 1**), are also plotted in red indicating near real-time processing speed. (b) Break down of processing time for CALMAN BATCH (excluding motion correction). Processing with CNMF in patches and refinement takes most of the time for CALMAN BATCH. (c) Computational gains for CALMAN BATCH due to parallelization for three datasets with different sizes. The parallelization gains are computed by using the same 24 CPU workstation and utilizing a different number of CPUs for each run. The different parts of the algorithm exhibit the same qualitative characteristics (data not shown). (d) Cost analysis of CNMF-E implementation for processing a 6000 frames long 1p dataset. Processing in patches in parallel induces a time/memory tradeoff and can lead to speed gains (patch size in legend). (e) Computational cost per frame for analyzing dataset J123 with CALMAN ONLINE. Tracking existing activity and detecting new neurons are the most expensive steps, whereas updating spatial footprints can be efficiently distributed among all frames. (f) Processing speed of CALMAN ONLINE for all annotated datasets. Overall speed depends on the number of detected neurons and the size of the FOV ( $N_p$  stands for number of pixels). Spatial downsampling can speed up processing. (g) Cost of neural activity online tracking for the whole brain zebrafish dataset (maximum time over all planes per volume). Tracking can be done in real-time using parallel processing.

DOI: <https://doi.org/10.7554/eLife.38173.019>



**Figure 9.** Components registered across six different sessions (days). (a) Contour plots of neurons that were detected to be active in all six imaging sessions overlaid on the correlation image of the sixth imaging session. Each color corresponds to a different session. (b) Stability of multiday registration method. Comparisons of forward and backward registrations in terms of  $F_1$  scores for all possible subsets of sessions. The comparisons agree to a very high level, indicating the stability of the proposed approach. (c) Comparison (in terms of  $F_1$  score) of pair-wise alignments using readouts from the union vs direct alignment. The comparison is performed for both the forward and the backwards alignment. For all pairs of sessions the alignment using the proposed method gives very similar results compared to direct pairwise alignment. (d) Magnified version of the tracked neurons corresponding to the squares marked in panel (a). Neurons in different parts of the FOV exhibit different shift patterns over the course of multiple days, but can nevertheless be tracked accurately by the proposed multiday registration method.

DOI: <https://doi.org/10.7554/eLife.38173.020>



**Figure 9—figure supplement 1.** Tracking neurons across days; step-by-step description of multi session registration. (a) Correlation image overlaid to contour plots of the neurons identified by in day 1 (top, 453 neurons), 2 (middle, 393 neurons) and 3 (bottom, 375 neurons). (b) Result of the pairwise registration of session 1 and 2. (c) Result of the pairwise registration of session 1 and 3. (d) Result of the pairwise registration of session 2 and 3. (e) Result of the pairwise registration of session 1 and 2. (f) Result of the pairwise registration of session 1 and 3. *Figure 9—figure supplement 1 continued on next page*

*Figure 9—figure supplement 1 continued*

registration between session 1 and 2. The union of distinct active components consists of the components that were active in i) both sessions (yellow - where only the components of session two are displayed), ii) only in session 2 (green), and iii) only in session 1 (red), aligned to the FOV of session 2. (c) At the next step the union of sessions 1 and 2 is registered with the results of session three to produce the union of all distinct components aligned to the FOV of session 3. (d) Registration of non-consecutive sessions (session 1 vs session 3) without pairwise registration. Keeping track of which session each component was active in enables efficient and stable comparisons.

DOI: <https://doi.org/10.7554/eLife.38173.021>