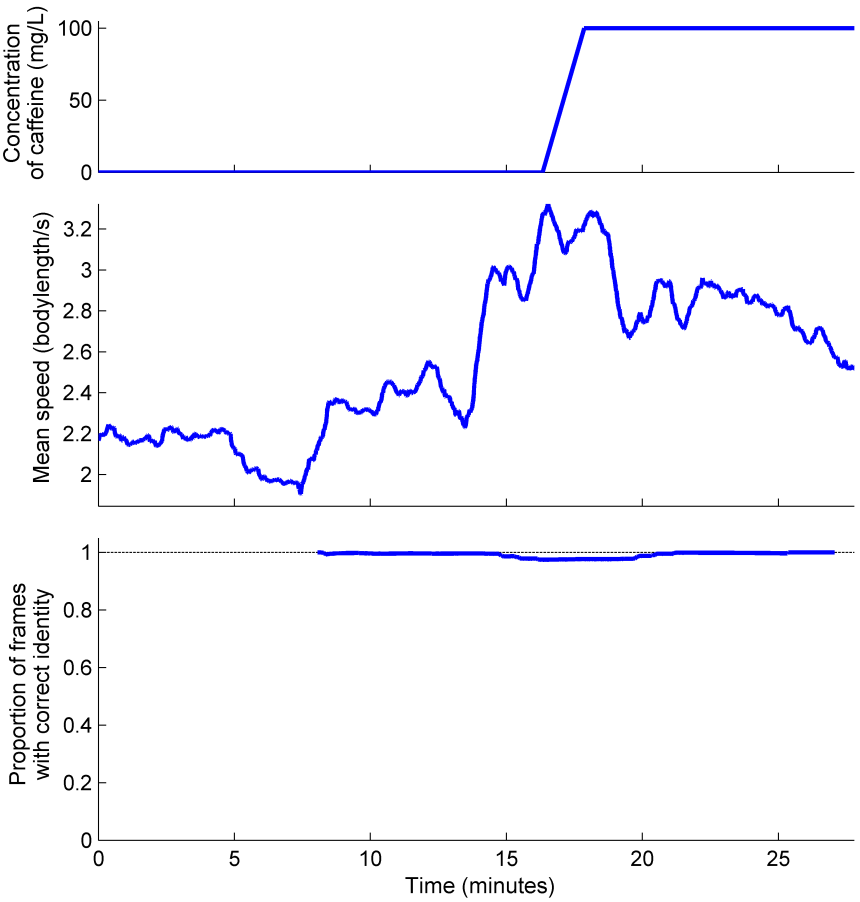
idTracker: Tracking individuals in a group by automatic identification of unmarked animals Supporting Text

Alfonso Pérez-Escudero, Julián Vicente-Page, Robert Hinz,
Sara Arganda, Gonzalo G. de Polavieja

Contents

Nature Methods: doi:10.1038/nmeth.2994

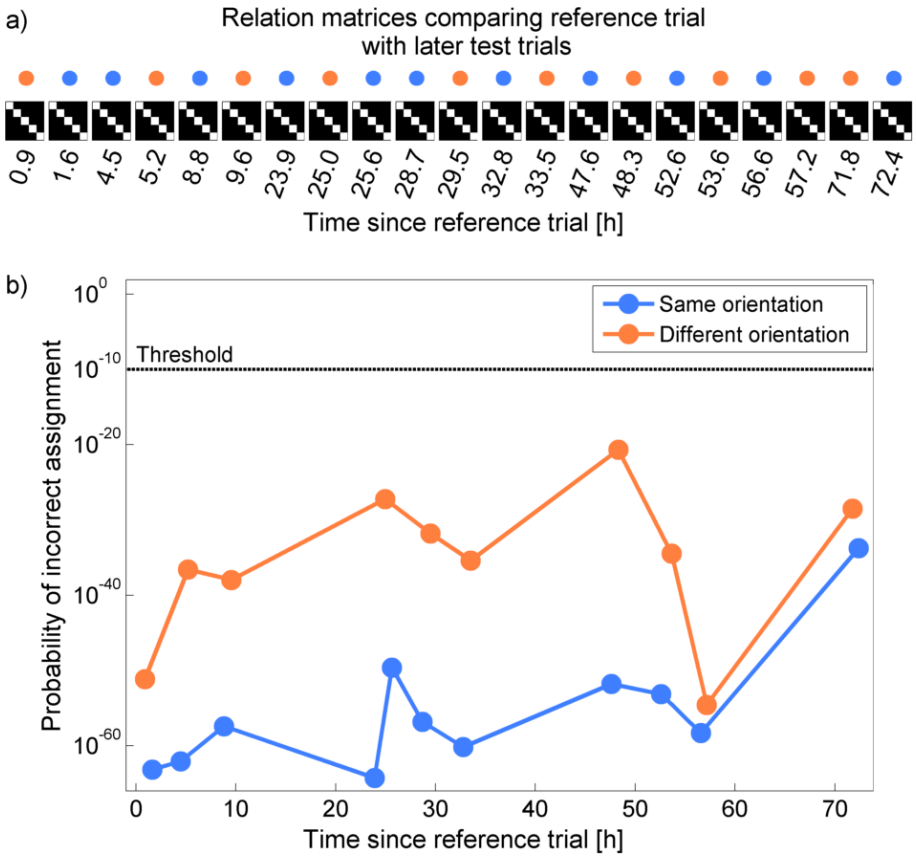
Supplementary Figure 1: Performance when behavior changes



Supplementary Figure 1: **Performance when behavior changes.** To test whether abrupt changes in behavior may affect the performance of the tracking system, we have recorded a video of 5 zebrafish in which at a certain moment caffeine is added to the water with a syringe up to a concentration of 100 mg/L. **Top.** Concentration of caffeine. **Middle.** Mean speed of the 5 zebrafish, showing an increase of activity during and after the addition of caffeine. **Bottom.** Proportion of correctly identified frames is maintained high during the experiment, with a slight decrease at peak velocities due to an increase of the number of crossings.

Nature Methods: doi:10.1038/nmeth.2994

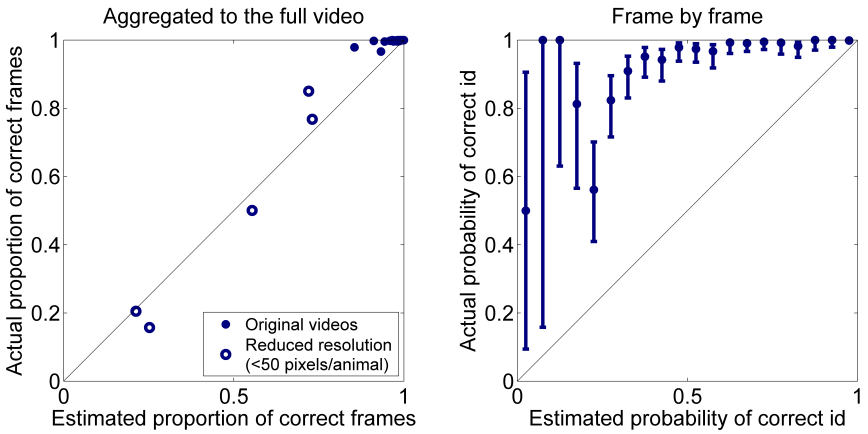
Supplementary Figure 2: Stability of the references during several days



Supplementary Figure 2: **Stability of the references during several days.** To validate the re-identification of the same individuals in different videos, we have recorded 5 zebrafish in a tank with dividers to keep them separated during 4 days. We have recorded videos three sessions a day (morning, noon and evening). To ensure that systematic differences in the illumination of the set-up do not help to identify the individuals, we record two videos in each session, rotating the set-up 180° each time. We extract 3000 frames for each fish from the first video of the first day, and use them as reference set. Then we extract 200 frames of each individual from each of the other videos, and use the reference set to identify them. We then follow the same procedure as we use to relate different videos, computing the probability P_3 that each individual of one video is the same as each individual of the other video (section 2.8 of Supplementary Note 2). **a)** 5×5 matrices representing the probability of assignment (P_3) of identities between trial 1 (y-axis) and in each of the 21 test trials during 4 days (x-axis). In a normal experiment we would use these probabilities to re-assign the identities of the fish in the second video. In this case we already know the identities, so the matrices are sorted in the way we know is correct. As expected, in all cases we find that the highest elements are in the diagonal. Dots indicate whether the set-up had the same orientation as in trial 1 (blue) or the opposite orientation (orange). **b)** Probability of incorrect assignment, calculated as the highest non-diagonal element of the assignment matrix. This probability is always many orders of magnitude lower than the threshold of 10^{-10} that we use when relating videos, and we do not detect any clear trend indicating deterioration with time.

Nature Methods: doi:10.1038/nmeth.2994

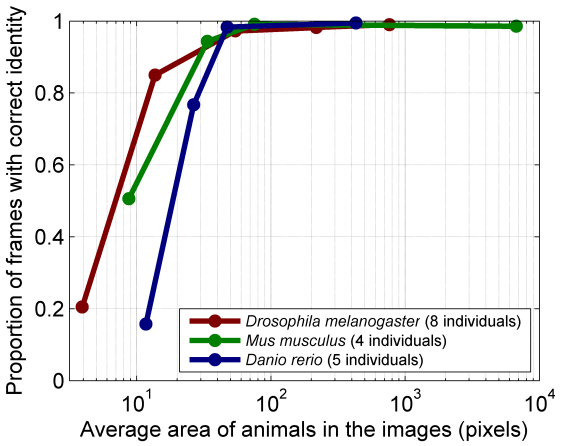
Supplementary Figure 3: Automatic estimation of tracking quality



Supplementary Figure 3: **Automatic estimation of tracking quality. Left.** Comparison of the automatic estimation of the proportion of correct frames (provider to the user at the end of the process) and the actual proportion computed with manual validation. The comparison is made using the original videos (filled circles) and a low-quality version of the same videos by resampling some of them so there are less than 50 pixels per animal (open circles). Videos of poor quality are detected with an estimated quality lower than 80%. For videos where the tracking works well, the estimation is conservative. **Right.** Estimated probability of correct identity for each frame (provided to the user together with the trajectories), compared to the actual probability of correct assignment (computed by grouping frames according to their estimated probability, and computing the proportion of correct frames in each category, according to our manual validation). Data pooled from all validated videos. Error bars show the 95% confidence interval. Again we find that idTracker's estimate is very conservative, with high-probability frames having indeed extremely high probability of being correct.

Nature Methods: doi:10.1038/nmeth.2994

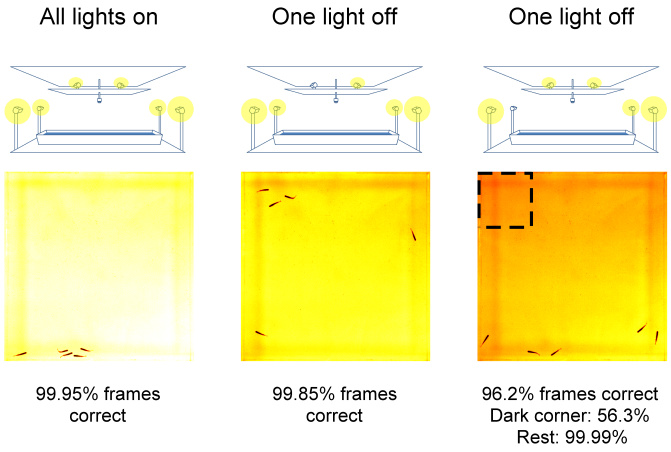
Supplementary Figure 4: Performance as a function of resolution



Supplementary Figure 4: **Performance as a function of resolution.** Proportion of frames that are identified correctly (compared with manual validation) as a function of the resolution of the video, measured by the number of pixels occupied by each individual in the images (averaged across individuals and for all frames of the video). We use a single video for each species (a video with 5 zebrafish, a video with 4 mice and a video with 8 fruitflies) and reduce the resolution by resampling the images. We resample by taking one every n pixels, so that noise is not reduced by the resampling. We find very little deterioration of the performance down to resolutions where each animal covers a surface of 50 pixels, and then a sudden drop to random assignments

Nature Methods: doi:10.1038/nmeth.2994

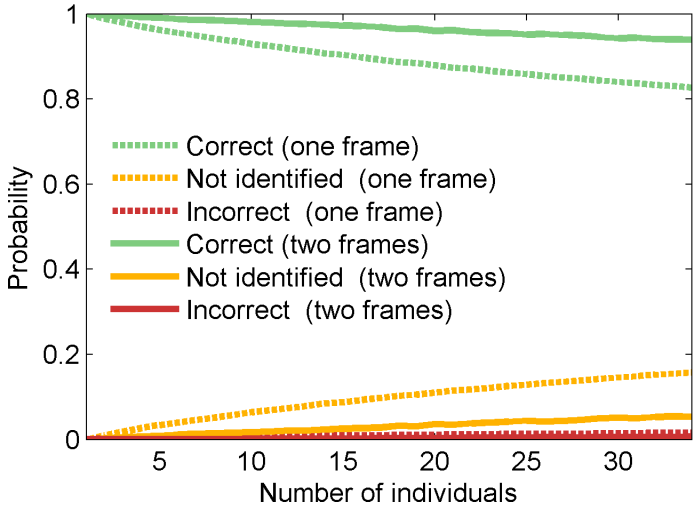
Supplementary Figure 5: Performance when changing illumination



Supplementary Figure 5: **Performance when changing illumination.** To test its robustness to inhomogeneities, we recorded a video of 5 zebrafish, and after 20 minutes we turned off one of the lights. After another 20 minutes we turned the light back on, and turned a different light off. The changes in overall level of illumination have little effect on the tracking, because each frame is normalized with its average intensity. The first two conditions have slight inhomogeneities in illumination, while the third condition (in which the off light is closer to the set-up) has the strongest inhomogneneity. **Left.** For symmetric illumination there are some inhomogeneities and the tracking has a high performance of 99.95% frames correct. **Middle.** Turning-off one of the top lights creates small inhomogeneities in illumination but performance is still very high at 99.85% frames correct. **Right.** Turning one of the main lights off creates a stronger inhomogeneity and performance lowers to 96.2% frames correct. Performance at the marked square dropped to 56.3% frames correct while for the rest was 99.99% frames correct.

Nature Methods: doi:10.1038/nmeth.2994

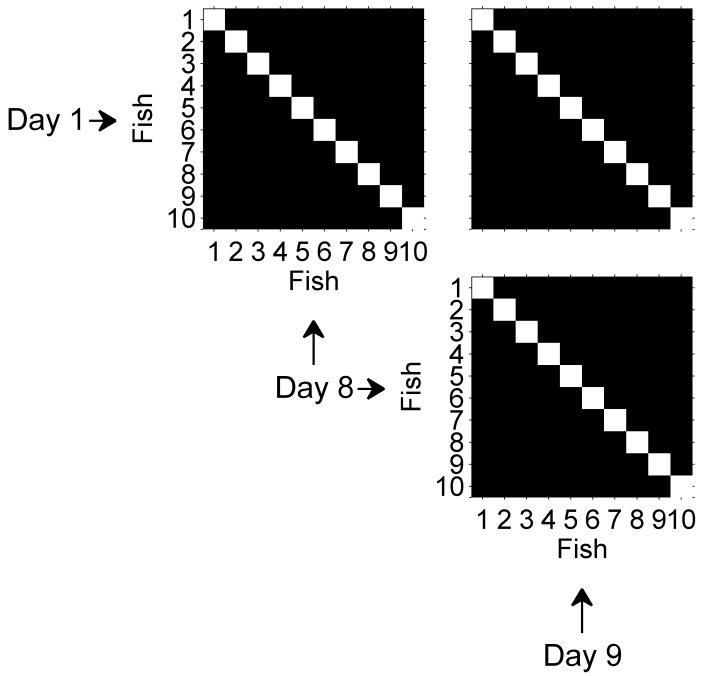
Supplementary Figure 6: Performance with high number of individuals



Supplementary Figure 6: **Performance with high number of individuals.** Proportion of frames that are identified correctly (green), incorrectly (red) or have uncertain identity (orange), averaged across all individuals and as a function of the number of individuals. Dashed lines correspond to identification using a single frame. These proportions are not the final proportions of correct frames that we would observe in a video, because aggregation of the information of all frames of each fragment greatly increases the probability of correct assignment. To illustrate this, thick lines correspond to identification using two independent frames, as a conservative estimate of the average size of fragments in a real video. We have tested the system in videos with up to 20 individuals, so for this test we pooled the data from 4 different videos (two videos of 4 individuals, two videos of 5 individuals and 2 videos of 8 individuals), making a total of 34 adult WIK zebrafish of the same age that belong to an in-bred population. For each individual we take from the reference images that were collected from the video and another 100 images that were not included in the references. To estimate the performance of the identification for a group of N individuals, we identify the 100 frames of one individual against a reference set that includes the reference images of the problem individual plus reference images of another $N-1$ individuals chosen randomly among the remaining 33 individuals. Dashed lines correspond simply to the proportion of these 100 frames with correct, incorrect and uncertain identity. For the solid lines, we take pairs of frames and assign the most likely identity unless there is a tie, in which case we classify them as uncertain. We repeat this procedure 100 times for each individual, and average all the results.

Nature Methods: doi:10.1038/nmeth.2994

Supplementary Figure 7: Redundancy of the relations across several videos



Supplementary Figure 7: **Redundancy of the relations across several videos.** The matrices show the probability (P_3) that each fish of one video is the same as one fish of another video (grayscale, white means 1 and black means 0) for the three videos of 10 medaka fish used in Figure 4 of the main text. The fish in the videos of days 8 and 9 have been reordered according to the matching with the fish in the video of day 1 (that is, to have the two top matrices with their highest elements on the diagonal). These orders of the individuals also make the matrix between days 8 and 9 diagonal, indicating that the assignments are consistent.

Nature Methods: doi:10.1038/nmeth.2994

Supplementary Table 1: **Results of validation.** To validate the system, we reviewed manually every crossing in a representative portion of each video. 'Time reviewed' indicates the length of the portion that is reviewed and used for validation. 'Revision time factor' indicates how many minutes were needed to review each minute of video (i.e. the product of 'Time reviewed' times 'Revision time factor' gives the time that it took to review the portion of the video). 'Area of animals' is the average area of the segmented blobs that are classified as individual animals. All percentages are computed as proportion of frames and as proportion of distances in the trajectories. We report the average proportion of the trajectories that corresponds to unoccluded images of the individuals (as opposed to crossings). % correct always refers to the total number of frames (or total distance) of each category (unoccluded portions, crossings, or the full video). 'Section describing set-up' indicates the section of this document that describes the set-up where each video was recorded.

Video	Length of video (min)	Frame rate (fps)	Area of animals (pixels)	Time reviewed (min)	Revision time factor	Used for dev.	Unoccluded portions				Crossings		Total	
							% of video		% correct		% correct		% correct	
							frames	dist.	frames	dist.	frames	dist.	frames	dist.
8 zebrafish	5,4	28,1	498	5,4		Yes	86,97	88	99,72	99,63	91,13	91,41	98,6	98,64
8 inbred WIK zebrafish	30,9	31,4	730	5,7	8,5	No	95,83	94,73	99,98	99,92	97,87	97,92	99,89	99,81
5 zebrafish (without cover)	30,3	32,2	404	5,2	7,7	Yes	90,81	93,63	99,58	99,75	95,74	95,17	99,23	99,46
2 juvenile nacre zebrafish (1)	16	24	553	16		No	99,08	99,14	100	100	99,29	99,25	99,99	99,99
2 juvenile nacre zebrafish (2)	15,9	24	419	15,9		No	98,97	99,13	99,99	99,99	94,27	91,18	99,93	99,92
2 juvenile nacre zebrafish (3)	15,9	24	497	15,9	0,4	No	99,59	99,66	100	100	98,4	98,39	99,99	99,99
2 juvenile nacre zebrafish (4)	15,9	24	340	15,9	0,3	No	99,47	99,52	100	99,99	97,52	94,5	99,98	99,97
8 female Drosophila (1)	31,6	28,6	768	2,3		Yes	94,23	98,43	99,94	99,98	100	100	99,94	99,98
8 female Drosophila (2)	21,4	28,6	761	13,4	2,4	No	97,27	99,46	100	100	100	100	100	100
6 Drosophila (4 fem, 2 mal)	60,3	42,5	964	3,6	10,6	No	95,96	98,45	99,98	99,97	98,05	97,82	99,91	99,94
2 agouti mice	31,4	49,2	12206	3,4		No	81,03	80,15	99,61	99,66	95,76	94,61	98,88	98,65
2 black mice (1)	31,4	49,2	11320	10,2	3,1	No	89,25	93,22	99,01	99,18	90,65	93,65	98,11	98,81
2 black mice (2)	31,3	49,2	11853	3,4	2,7	No	83,04	85,88	100	100	99,12	98,77	99,85	99,83
2 black mice (3)	31,4	49,2	13014	3,4	2,1	No	85,86	88,9	100	100	100	100	100	100
4 black mice (1)	32,8	24	9149	6,9	4,3	No	81,62	84,9	99,74	99,73	98,29	97,73	99,47	99,42
4 black mice (2)	40,9	25,1	6839	11	4,1	Yes	97,95	98,51	99,67	99,76	92,01	85,42	99,51	99,54
4 black mice (3)	76,2	25	5916	22,7	2	No	96,29	97,03	97,12	98,93	99,13	99,82	97,2	98,96
5 medaka fish	31,2	28,8	208	17,4	0,3	No	99,59	99,57	100	100	100	100	100	100
10 medaka fish	31,1	28,8	154	6	4,8	No	96,91	97,5	99,97	99,96	98,56	98,99	99,93	99,93
20 medaka fish	6,3	29,1	227	5,7	5,2	No	97,94	98,53	99,99	100	99,87	99,92	99,99	100
8 ants	32	28,6	765	2,9		Yes	78,31	95,84	96,88	99,81	93,56	89,69	96,16	99,39
5 ants, *Messor structor* (1)	34	25	759	9		No	95,82	97,81	100	100	100	100	100	100
5 ants, *Messor structor* (2)	41,7	25	626	6,7		No	96,69	98,84	99,99	100	96,12	94,34	99,86	99,93

Supplementary Note 1: User guide

NOTE: This user guide corresponds to version 1.1 of idTracker. See www.idtracker.es for the current version.

1 Quick start

This section describes the installation of the Windows stand-alone version of the program, and its use to track a video, such as the example video of five zebrafish that can be downloaded from www.idtracker.es. If you want to record your own video, first read Section 2.

1. **Install idTracker.** We strongly recommend to use Windows 7 (64 bits). idTracker also works under Windows XP (64 bits), but it does not recognize most video codecs. Download idTracker_setup.exe from www.idtracker.es, execute it and follow the instructions. In addition to idTracker, it will automatically install the Matlab Compiler Runtime (unless you already have it).

2. **Copy the video to an adequate location.** idTracker will generate a large amount of data (depending on the length of the video and the number of individuals; about 500 MB for the example video), so there must be free space in the disk. Also, idTracker needs to access big amounts of data frequently, so it will be extremely slow if the video is an external hard drive with a slow connection to the computer (e.g. USB 2.0). Place the video either in the internal harddrive, or in a external hard drive with fast connection (e.g. eSATA or USB 3.0).

3. **Execute idTracker.**

4. **Open file.** The first window lets you select one file from your computer. Select the file with the video. If your video is split in several files, select the first one (see Section 2 for the format of videos split in several files).

5. **Main screen.** A window like the one shown in **Supplementary Figure 8** will open. From now on, all references to components in this main window refer to the labels in **Supplementary Figure 8**

If you are using the example video provided in www.idtracker.es (5Zebrafish_nocover_22min.avi), you may jump directly to step 15 and skip the adjustment of all parameters, as default values work well for this video.

6. **Enter the number of individuals** in box 4. For the example video, enter 5.

7. **Choose type of contrast.** By default, the system looks for dark animals on bright background. Check the box 'Invert contrast' (9) to change to bright animals on dark background. For the example video, leave it unchecked.

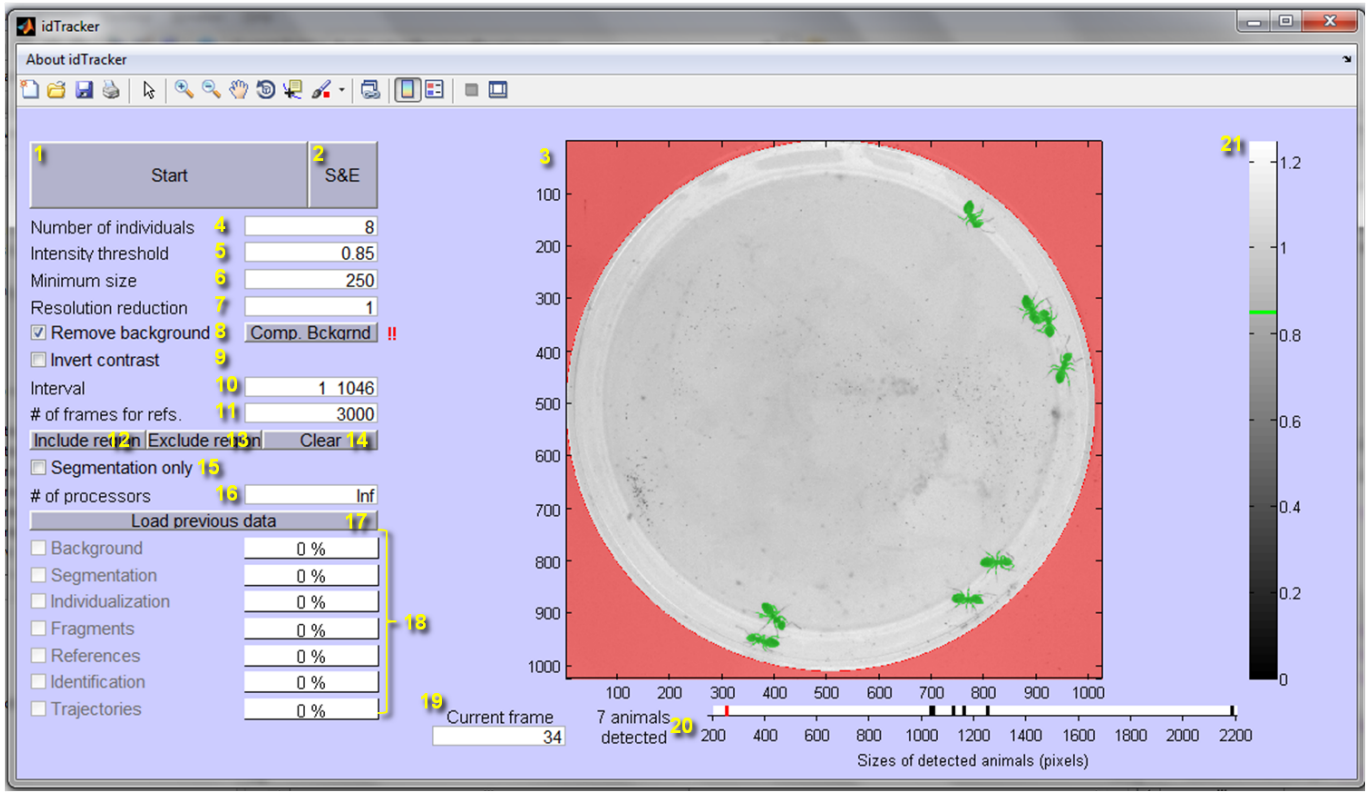
8. **Select Region Of Interest (ROI) and/or exclude regions.** If you want to select a Region of Interest, click on button 12 and follow the instructions. You can also choose a region to be excluded from the tracking with button 13. Click on button 14 to clear all previously defined ROI's or excluded regions. For the example video it is not necessary to select any ROI.

9. **Background removal option.** Check the box 8 if you want to activate the background removal option. The background will be computed when the tracking starts. To compute it before (so that you can view the correct results in the visualization window 3) click on the 'Compute Bckgrnd' button 8. It will take a few seconds, the progress is shown in one of the progress bars 18. A red '!!' on the right of the 'Compute Bckgrnd' button means that the background is not updated (meaning that the segmentation presented in panel 3 is not updated, but has no effect in the final result of the tracking). For the example video, leave 'remove background' checked and click 'Compute Bckgrnd'.

10. **Enter the intensity threshold.** Use box 5 to change the intensity threshold for segmentation. The system considers that pixels with lower intensity than this threshold belong to the animals (or higher intensity, if "invert contrast" is checked). You can check the result for each threshold in the visualization window (panel 3. Green patches indicate the segmented blobs). You can use box 19 to view different frames. For the example video, threshold 0.8 works well.

11. **Enter the minimum size.** The program will reject blobs smaller than the minimum size entered in box 6. Watch several frames (use box 19 to change the frame), and choose a value that is that is clearly smaller than the sizes of the animals (the bar 20 on the bottom shows the sizes of the animals in the current frame. Each black vertical line corresponds to

Nature Methods: doi:10.1038/nmeth.2994



Supplementary Figure 8: Main interface of idTracker.

one animal, and the red line is the minimum size). For the example video, minimum size 250 works well.

12. **Choose resolution reduction.** If the size of the animals is very big compared to the image, the algorithm will take too long. If the sizes of the animals are bigger than 2000 pixels, input a number higher than 1 in box 7 (the number of pixels will be divided by n^2, where n is the number in box 7. Choose this number so that the resulting number of pixels of the blobs is below 2000). For the example video, leave a 1 in box 7.

13. **Choose an interval.** If you want to track only part of the video, enter the interval that you want to track in box 10 (in frames, use a space to separate the first frame and the last frame of the interval). For the example video, leave the full interval.

14. **Number of reference frames.** Box 11 shows the maximum number of reference frames that the program will use. Choose a lower number for increased speed, a higher number for increased accuracy. For the example video, leave it at 3000.

15. **Start tracking.** Click the 'start' button (1). In some computers, after pressing 'Start' the operating system's firewall will issue a warning that idTracker is trying to access the local network. This is caused by idTracker trying to connect to several processors of the computer simultaneously (it does not connect to the internet). In order to use more than one processor, grant access at this point. Once the tracking is started, you will not be able to change the parameters. Wait until the program finishes (a new window will appear indicating that the tracking has finished). It may take several hours, an in the meantime idTracker's window may not have a very smooth behavior (for example, if you minimize and maximize it again it is normal that it stays black for a long while).

16. **Get the results.** The results are in a folder called 'segm' located in the same folder as the video. The trajectories are in the Matlab file trajectories.mat

Nature Methods: doi:10.1038/nmeth.2994

(this file contains a 3D matrix called trajectories that contains the coordinates of each individual in each frame. The first dimension runs along frames, the second along individuals and the third along x, y). They are also in the file trajectories.txt in text format (in this file, each two columns correspond to the x, y coordinates of each individual.

2 Conditions for the video

idTracker needs the following conditions to work:

- **Background.** Ideally, there should be a uniform background, with a color that gives enough contrast with the animals. Different degrees of background structure are acceptable, and will be managed by the system in different ways:
  - Static or moving objects much smaller than the animals will not affect segmentation, because objects below a certain size are removed.
  - Static objects of any size can be removed in two alternative ways: They can be manually excluded when selecting the region of interest, or they can be automatically excluded by the background removal option. In either case, when an animal is in contact with one of these objects its segmentation is less consistent, and the segmented image is not used for identification. The animal can still be identified with images of the neighbouring frames (see section 2.10 of Supplementary Note 2), but the performance of the tracking may be lower for the portions of the video in which the animals overlap with the background objects.
  - Static or moving objects of any size that are present only in a small portion of the video may cause problems in the identifications of animals while present, but will not affect the rest of the video (see comments about robustness to perturbations in the main text).
- **Camera.** The system works on videos recorded with conventional cameras, including consumer handicams and even some modern photo cameras. The system works on grayscale images, so if the video is in color it will first transform it to grayscale. General requirements are the following.
  - Resolution of the video must be enough for identifications, meaning that the images of the animals must have a minimum size (in pixels). We have performed tracking successfully with as few as 150 pixels per animal (area of the segmented blobs, average across all individuals and full video), and in our experience 250 pixels per animal is enough for most species and conditions. See **Supplementary Table 1** for the resolution conditions of the videos used for validation.
  - Frame rate should be high enough so that the images of an animal moving at typical speed overlap in consecutive frames. We find that 25 fps is enough for all species used in the validation.
  - Exposure time should be short enough with respect to the typical speed of the animals, so that images are not blurred (a small proportion of burred images is acceptable, for example when animals turn very suddenly).
  - If the video is compressed, it should be high quality and high definition compression (for example we find good results with codecs AVHCD or HD mpeg4, which are typical in most modern consumer cameras). The system may sometimes work with medium-quality compression, but at the expense of having higher resolution (relative to body size) than would be required with uncompressed video.
- **Video format** must be compatible with Matlab, which can only access some of the codecs installed in the computer. Compatible formats may be different on different computers (especially if they have different operating systems). In Windows computers, we find that uncompressed avi or mpeg4-compressed avi files are usually readable. See www.idtracker.es for more information about converting videos to compatible formats. If the video is cut into several files, the last characters of the file names must be the number of the file, with no trailing zeros. The rest of the file name must be identical for all files (for example file1.avi, file2.avi, ... file9.avi, file10.avi ...).
- **Arena.** Ideally, the walls of the arena should be transparent or translucid, so that there are no strong shadows near the walls (sometimes opaque walls work fine when illumination is very diffuse). Reflections of

Nature Methods: doi:10.1038/nmeth.2994

the animals on the walls can be a problem, but usually they can be removed by selecting a tight Region of Interest that removes the walls. For aquatic animals with transparent walls, having water at both sides of the wall helps to soften the reflections. If a cover is needed to prevent the animals from escaping, it should be as transparent, and clean as possible. Care must be taken to prevent intense reflections from the cover (or from the water surface in the case of aquatic animals). See the description of the set-up in Methods for an example of how to prevent reflections.

- **The apparent size of each individual should not change very much during the video.** Therefore, the following conditions should be satisfied.
  - To prevent large changes of distance between camera and animals, the animals should move roughly in 2D. Animals that walk on a plane automatically obey this restriction. Swimming or flying animals should be restricted so that they move in a narrow region, which can be wide enough to allow crossings (this region can be as thick as 2 to 5% the distance between the camera and the animals).
  - The camera should point perpendicularly to the plane in which the animals move. This it to prevent perspective effects that change the apparent size of the animals when they change position.
- **Illumination** should be as uniform as possible. Indirect illumination usually gives the best results. Retroillumination is also an option, and we have successfully tracked retroilluminated videos (in our case there was also some ambient light, so the animals' skin patterns were still visible. When the patterns are not visible the identification is more difficult, but it may still work fine for some species and conditions).
- **Shadows** of the animals on the background may affect the segmentation, making the identifications more difficult. These shadows can be prevented by using a transparent base for the set-up, with the opaque background at a certain distance behind the animals (see for example the set-ups described in Methods). In this way, the shadows on the opaque background are very diffuse and do not affect segmentation.
- **Duration of video.** Collection of references requires a minimum length of the video in which all the animals are in view. This length depends on the number of animals, the species and their level of activity. As a reference, for groups up to 10 zebrafish usually 5 minutes is enough. 30 minutes is a safe value (we have never found a 30-minutes video in which the system fails to collect references). If the video is too short or the animals are not visible for a long time, the video can still be tracked using external references.

3 Description of the interface

The interface has the following controls (numbers correspond to those in **Supplementary Figure 8**).

1. **Start button.** When pressing this button the tracking process starts, and tracking parameters cannot be changed any more.
2. **Save & Exit button.** When pressing this button the program ends, but the tracking does not start. All tracking parameters are stored in file datosegm, so they can be used at a later time using button 17.
3. **Main visualization window.** This window shows a frame of the video (use box 19 to select the frame), with the current output of segmentation. It is updated whenever any parameter is changed. You can zoom in and out in this window with the tools provided in the toolbar. Red regions are excluded from the tracking, either because they are outside the region of interest (button 12) or because they have been excluded (button 13). Green regions are the blobs that have been segmented in the current frame.
4. **Number of individuals.** Edit this box to input the number of individuals that are present in the video. Not all individuals need to be visible at all times, this must be the total number of individuals.
5. **Threshold for segmentation.** Only pixels darker (i.e. with lower value) than this threshold will pass segmentation (see also point 9, 'invert contrast'). The value of the threshold is given as proportion of the average intensity of the whole frame.
6. **Minimum size.** Only blobs bigger than this size will be selected.
7. **Resolution reduction.** Resolution can be reduced to speed up the first stages of the algorithm. It is strongly recommended to reduce resolution if blobs are bigger than 5000 pixels. The width and height

Nature Methods: doi:10.1038/nmeth.2994

of each frame is divided by the number in this box (i.e. 1 means the original frame, 2 means that width and height will be reduced to one-half, so the total amount of pixels will be reduced by 4 times).

8. **Remove background.** Check this box to activate a routine that will prevent static objects from being confused with animals. When the checkbox is activated, the button 'Comp bckgrnd' appears. Push this button to compute the background with the current settings. Before the background is computed (or if any parameter that affects background calculation changes) a red '!!' sign will appear. This sign means that the output shown in the main visualization window (window 3) is not accurate because it has not been computed using the true background. But this sign only refers to visualization, if the button 'Comp bckgrnd' is not pressed (or any parameters change and it is not pressed again), the background will be computed after pressing 'Start' (button 1).

9. **Invert contrast.** By default, the program looks for dark animals on a bright background. When this box is checked the program looks for bright animals on dark background.

10. **Interval.** This box indicates the interval of video (in frames) that will be tracked. By default indicates the full video.

11. **Number of frames for references.** Maximum number of images of each individual that will be incorporated in the references. The program will acquire as many reference images as it can, up to this number. A lower number will make the process faster and will use less memory, at the expense of higher error rates (for many species and situations about 500 frames in the references can be used without appreciable increase in error rates, but if computation time and memory are not pressing constraints, we recommend to leave it at the default value of 3000).

12. **Region of Interest.** Press this button to select a region of interest, or to expand it once it has been created[1]. A new dialog will appear, giving choice between rectangular, circular or polygonal region. Once the selection is made, go to the main visualization window (window 3), and select as many points as necessary, depending on the type of region:

    - For rectangular region select 2 points, which will be opposite corners of the rectangle.
    - For circular region select 4 points. The border of the region will be the circle that best fits these 4 points.
    - For polygonal region select a minimum of 3 points, with no upper limit. Each point will be a corner of the polygon, and it is not necessary to finish on the same point as you started, a last side will be added between the last and the first points. Press ENTER once all corners have been selected. The polygon does not need to be convex.

13. **Exclude region.** Press this button to exclude a region. You can select a region with the same three geometries as for the region of interest. Excluding a region will not erase a previous region of interest, nor previous excluded regions. It is not necessary to have selected a region of interest to exclude a region.

14. **Clear region of interest and excluded regions.** Press this button to remove the region of interest and all excluded regions.

15. **Segmentation only.** If this box is checked, idTracker will exit after the segmentation step, leaving the tracking unfinished. The data can be recovered later using button 17.

16. **Number of processors.** IdTracker can use multiple processors to improve performance. The number of this box indicates how many processors idTracker will use ('Inf' means that idTracker will use all available processors). Using a lower number of processors leaves more resources for other programs while idTracker is running. Memory usage is lower when using fewer processors, so using fewer processors may be necessary in computers with low RAM memory, especially when tracking videos with many individuals.

17. **Load previous data.** This button will only be active if the program has been executed previously on the same video (even if the tracking process has not finished). When pressing it, all current parameters are erased and substituted by the parameters in the

[1] The behavior of this button changes once a ROI has been defined. At the beginning (when the button reads 'ROI') the region selected with this button will be the region of interest, and all pixels outside it will be excluded. But as soon as some ROI or excluded regions have been selected the behavior of this button changes to 'include region', and any region selected with this button will be added to the current ROI.

Nature Methods: doi:10.1038/nmeth.2994

previous run. Also, all intermediate steps that were completed in the previous run (background, segmentation, etc.) will be recovered and can be re-used.

18. **Process indicators.** These bars measure the process of each stage of the tracking. When a previous run of the tracking has been performed and the data have been loaded (using button 17), the user can choose whether to reuse different parts or not. Check the boxes to reuse parts, un-check them to recompute them (note that recomputing one part implies that all the parts that come afterwards will also need to be recomputed).

19. **Frame shown.** This box controls what frame is shown in the main visualization window (window 3).

20. **Summary of blobs detected.** It indicates how many blobs have been segmented in the current frame, and their sizes (black lines). The red line indicates the minimum size that has been specified in box 6.

21. **Colorbar.** It shows the grayscale used to show the frame in the main visualization window (window 3). The green line indicates the threshold for segmentation (introduced in box 5).

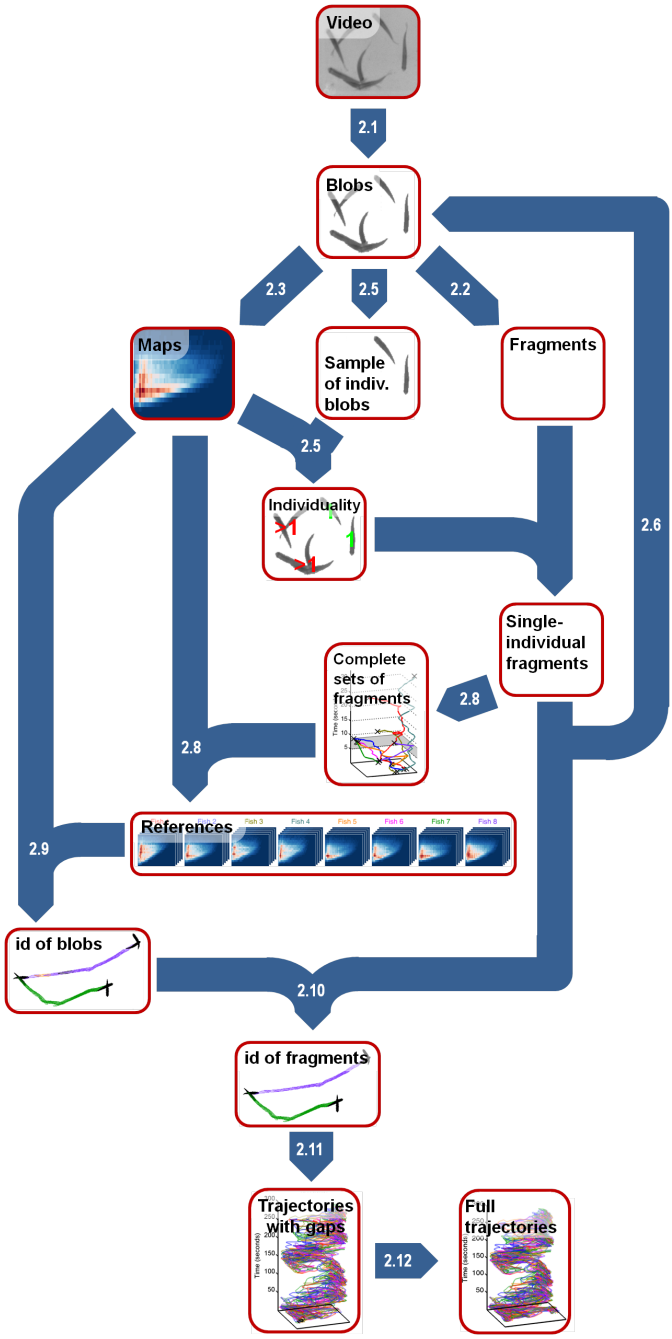
Supplementary Note 2: Description of the algorithm

This note presents a detailed description of the tracking algorithm. It is organized in three parts: section 1 presents a general overview of the algorithm, without any details; section 2 describes in detail each of the steps of the algorithm; and section 3 describes how to compute some of the quantities used to assign the identities.

1 Overview of the algorithm

Supplementary Figure 9 illustrates the workflow of the algorithm. The numbers on the arrows indicate the section where each step is described in detail, and we will also use them in this section to refer to different parts of the figure. All steps of the algorithm are completely automatic.

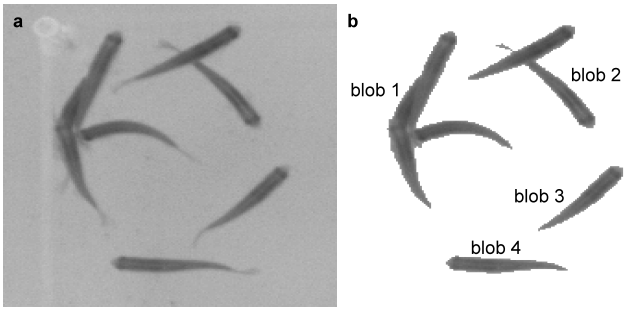
The first step is to segment each frame of the video to separate the animals from the background (2.1). Each set of pixels that belong to one or several overlapping individuals is called 'blob'. Then we perform a very simple tracking process, that identifies small portions of video where we can follow the same object with very high certainty (for example, when one animal is moving smoothly and well separated of all other individuals). In this way we can collect sets of blobs of consecutive frames that belong to the same individual or group of individuals (2.2). We refer to these sets as fragments of trajectories, or simply 'fragments'. In parallel, we extract a small sample of blobs that belong to single individuals (2.5). Also, we transform all blobs to obtain their contrast and intensity maps (2.3). Then, we compare (via their intensity and contrast maps) the sample of single individual blobs with all other blobs of the video. From this comparison, we classify each blob of the video as belonging to one single individual or to several overlapping individuals (2.5), and using this information we find the fragments that belong to single individuals (single-individual fragments). We now reconsider the segmentation, eroding part of the blobs that belong to several individuals, and keeping the eroded blobs when there is high certainty that the erosion separated the individuals successfully (2.6). The next step is to build a set of reference images for each individual (2.8). First, we look for short portions of the video where all individuals are separated. Each of these portions provides a 'complete set of fragments', (a set that contains one single individual fragment of each individual). We extract several complete sets of fragments from the video, and compare them (via the intensity and contrast maps of their images) to link



Supplementary Figure 9: Workflow of idTracker. White numbers on the arrows refer to the section that describes the process.

Nature Methods: doi:10.1038/nmeth.2994

them together. In this way we obtain a large number of reference images for each individual. Then we compare each blob of the video with the reference images, and obtain the identity of each blob (2.7). We aggregate the identities of all blobs of each fragment, to obtain the identity of each fragment with higher certainty (2.10). Finally, we use the identities of the single-individual fragments and the center-of-mass of the blobs to build the trajectory of each animal while unoccluded (2.11), estimating its position during the crossings (2.12).



Supplementary Figure 10: Example of segmentation. **(a)** Original image. **(b)** Result of segmentation.

2 Steps of the algorithm

This section describes the workflow of the tracking program.

Only three of the parameters discussed need to be provided by the user (number of animals, segmentation threshold and minimum size). The system can robustly track many different species in many different situations without further customization. See Supplementary Note 1 for a complete description of the user interface.

2.1 Segmentation

In this first step the system looks for sets of contiguous pixels ('blobs') that corresponds to animals, distinguishing them from the background (**Supplementary Figure 10**).

Illumination in a behavioral setup usually fluctuates (for example the intensity of incandescent lamps fluctuates due to the use of alternating current). To eliminate these fluctuations, each frame is normalized dividing the intensity of each pixel by the average of the intensities of all pixels of the frame.

We select pixels whose normalized intensity is below (or above) a given threshold, which can be set by the user. We only keep blobs that are larger than a certain minimum size, $a_{\min}$. This size depends on the species and the experimental conditions, being typically between 50 and 250 pixels, and can be changed by the user.

To deal with non-homogeneous static background, we have implemented an optional background removal step.[2] In this step, the program computes an average image along the whole video and flags the pixels that are above threshold in the average image. Then segmentation is done as described previously, but adding the following rules. Blobs that contain only flagged pixels are not recorded. If a blob contains less than 10% of flagged pixels, it is recorded 'as is'. If more than 10% of the pixels of a blob are flagged, then the flagged pixels are removed and the blob is reanalyzed. The removal of flagged pixels may have split the blob into several parts. Of these parts, those that are smaller than $a_{\min}/5$ are removed. If the remaining parts accumulate more than $a_{\min}$ pixels, the blob is recorded as a single unit, but flagging it as inadequate for identification (it can still be used for reconstruction of trajectories). If the remaining pixels are less than $a_{\min}$, the blob is discarded.

When a region of interest has been selected by the user, those blobs that are in contact with the edge of the region of interest are flagged as inadequate for identification (in case part of an individual is out of the region of interest).

2.2 Fragments of trajectories

The aim of this step is to find parts of the video in which one individual is moving without crossing with any other individual.

With 'fragment' we refer to a set of blobs of consecutive frames that with high probability correspond to the same individual or group of individuals. Two blobs of consecutive frames belong to the same fragment if they overlap with each other and do not overlap with any other blob of the two frames. If a blob does not meet the requirements to be included in an existing fragment, a new fragment is initialized. Therefore, each fragment exists for a given number of consecutive frames, and has one blob in each of these frames.

[2]The algorithm described here is different from the usual background subtraction in which segmentation is done by subtracting each frame from the background. Our version is less sensitive to small differences in contrast between the animals and the background, but has the advantage of providing a more consistent segmentation.

It often happens that a set of blobs in consecutive frames that actually belong to the same individual is cut into several fragments. This is not a problem, because they will be eventually identified as belonging to the same individual. The opposite type of mistake in which several different individuals are mixed in the same fragment would be more problematic, because it would produce an identification mistake in part of the fragment. That is why we have selected a very restrictive criterion for the formation of fragments, so that it is extremely unlikely to have two different individuals mixed in the same fragment.[3]

2.3 Transformation of images

All segmented images (except those flagged as unsuitable for identification) are transformed into their intensity and contrast maps, as described in the main text.

2.4 Comparison of images

In order to compare two images we subtract their maps element-by-element, and compute the mean of the absolute values of the differences. Therefore the difference between two images is summarized in two positive numbers, that we call 'intensity distance' between the two images (for the mean difference between the intensity maps) and 'contrast distance' between the two images (for the mean difference between the contrast maps). We use the term 'summed distance' for the sum of intensity and contrast distances.

2.5 Selection of blobs that belong to a single individual

In order to distinguish blobs that belong to one single individual from blobs that belong to several overlapping individuals and from noise, we use the following procedure.

First, the program extracts from the video a collection of images of single individuals. The program has no previous information about the aspect of an individual, so in order to recognize them it assumes that if a frame has the same number of blobs as the total number of individuals (which has been provided by the user), each blob belongs to one single individual. The program selects randomly 5 of these frames, and uses their blobs to build a collection of single-individual images.

The next step is to compute the typical variability among the images of single individuals. We first find the minimum summed distance[4] between each image of the collection of single-individual images and all the other images of the collection. We then compute the mean ($\mu_{\text{diffindiv}}$) and standard deviation ($\sigma_{\text{diffindiv}}$) of these minimum summed distances.

Finally, we classify all blobs of the video. For each blob, we compute its summed distance to all the single-individual images of the collection, and take the minimum. If this minimum summed distance is below $\mu_{\text{diffindiv}} + 3\sigma_{\text{diffindiv}}$ we classify the blob as belonging to a single unoccluded individual, and we flag it as adequate for identification.

This step can be omitted when external references are provided for the tracking (e.g. references that belong to another video of the same individuals). In this case, the program simply chooses a collection of one-individual images from the references.

2.6 Resegmentation

In some cases the animals touch briefly, and it is easy to separate them by eroding the image. A successful separation increases the length of the fragments of trajectories, improving the performance of later steps. We only resolve in this way the easiest crossings in order to ensure that no mistakes occur (we have never observed a mistake caused by this step in any of the validated videos).

We use a disk as structuring element for the erosion, so in our case erosion is equivalent to keeping only those pixels of a blob that are further from the periphery than a given distance d_{e}. We estimate an adequate distance d_{e} from the statistics of the same video: For each one-individual blob we compute the maximum of the distances between each pixel of the blob and the periphery of the blob. For each frame we take the minimum of these distances, to ensure that we select a distance that is adequate for the smallest individual. We define d_e as half the median of these distances.

We will resegment fragments classified as belonging to more than one individual (i.e. most of its blobs are classified as non-individual blobs). We will only consider a fragment if all the fragments with which it overlaps at both sides (i.e. before and after in time) have been classified as single-individual fragments. Then, we work starting from

[3] We have not performed a specific characterization of these probabilities. But we annotate an identification mistake when two individuals are mixed in the same fragment, so the probability of these events must be lower than the probability of wrong identification, which is very low (Supplementary Note 3)

[4] See section 2.4 for the definition of summed distance.

Nature Methods: doi:10.1038/nmeth.2994

both ends and towards the center. We erode the first blob of the fragment. If this erosion results in the separation of the blob in several sub-blobs and each sub-blob fulfills the requirements to belong to one of the neighbouring one-individual fragments, we add each new eroded blob to its overlapping one-individual fragment (flagging them as unsuitable for identification) and remove the original blob from the resegmented fragment. We then follow the same procedure with the last blob of the fragment, and iterate this procedure until the requirements are not fulfilled at any of the two sides or until the one-individual fragments at the two sides connect in the middle, in which case we merge them.

2.7 Identification of one image

Whenever we have one problem image and a set of reference images for each individual (all individuals must have the same number of reference images to prevent biases), we perform the identification as follows.

We compute the intensity distance between the problem image and all reference images. Then we select the reference image with lowest distance to the problem image, and record the individual to which it belongs. We repeat the process using contrast distances instead of intensity distances. If the recorded individual is the same for both of them, we take it as the identity of the problem image. If they do not match, the problem image is left as ambiguous (not identified).

2.8 Collection of reference images

Now the program extracts from the video a collection of reference images of each individual. The procedure is as follows.

If there are N individuals in the video and there is a set of N one-individual fragments[5] that coexist in at least one frame, then each of these fragments must belong to a different individual and all individuals are present in the set of fragments. For this reason, we call this set of one-individual fragments 'complete set of fragments' (for brevity, in the remaining of this section we will call it simply 'set'). The time for which all fragments of a set coexist is usually short because when any two individuals cross their fragments end. But all other fragments continue beyond that point, their length being typically much longer than the coexistence period (Figure 3a of main text). Because the set includes the full fragments, even a short coexistence time can provide the algorithm with a large amount of images of each individual.

When an animal is not moving or is moving very slowly, its images in two consecutive frames will be very similar, and the second one adds very little new information. Therefore, the system does not use all the images of each fragment, but only images that differ significantly from each other. Specifically, the system skips new images until less than 80 % of the animal's area overlaps with the last image that was included. The system will only use sets whose shortest fragment has at least 15 images that fulfill this criterion.

Usually a single of these complete sets of fragments does not contain enough images of each individual to build the complete references (reliable identification may require hundreds or thousands of images for each individual), so the program extracts several complete sets of fragments from the video. The problem now is how to relate these sets of fragments to find which fragment of each set corresponds to the same individual. To do this, for each pair of complete sets of fragments, the program uses the one with the highest number of images to identify the images of the other in the way described in section 2.7[6]. We use these identities to compute the probability that each fragment of one set belongs to the same individual as each fragment of the other set. We refer to this probability as P_2 (see section3 for the method to compute P_2). For example, **Supplementary Figure 11**a shows the probabilities P_2 that relate three complete sets of fragments in a video with eight individuals. The element (i, j) of any of the matrices in **Supplementary Figure 11**a can be interpreted as the probability that fragment i of one of the sets belongs to the same individual as fragment j in the other set.

Because of the relatively low number of images available in each complete set of fragments, these identifications are not very reliable. But this is compensated by having many complete sets of fragments and comparing all with all (**Supplementary Figure 11**), because the redundancy of these comparisons improves the certainty of the relations. For example, in **Supplementary Figure 11**a the relation between individuals 3 and 4 of set A and individuals 1 and 2 of set B are uncertain (i.e. the

[5] A one-individual fragment is a fragment (as defined in section 2.10) whose blobs are classified as one-individual blobs (as described in section 2.7).

[6] As described in section 2.7, all individuals must have the same number of reference images. Therefore, when a complete set of fragments is used as reference to identify an image, the number of reference images for each individual will be limited by the fragment that has fewer images.

Nature Methods: doi:10.1038/nmeth.2994

probabilities are not close to 0 or 1, but have intermediate values). But because both individual 3 of set A and individual 1 of set B are assigned to individual 1 in set C, they must be the same individual. Likewise, we can relate individual 4 of set A and individual 2 of set B, because they are both assigned with high reliability to individual 3 of set C.

The actual algorithm used to make use of the redundancies is as follows. We choose one complete set of fragments as reference, and re-sort all other sets of fragments so that the first individual of each set is the one that corresponds with highest probability to the first individual of the reference set, the second with the second, etc. In the case of **Supplementary Figure 11**, if we choose set A as reference, we would re-sort sets B and C such that all the matrices in the first row would have their highest elements in the diagonal (**Supplementary Figure 11b**). Then, if the relations between sets B and C are consistent with their relations with set A, the matrix in the second row of **Supplementary Figure 11b** will automatically have its highest elements in the diagonal. After re-sorting in this way, let $P_2^{i,j,l,m}$ be the probability that the i-th individual of set l is the same as the j-th individual of set m (as computed in Equation S4). Then, the probability that individual i of fragment l is correctly assigned in all other fragments can be estimated[7] as

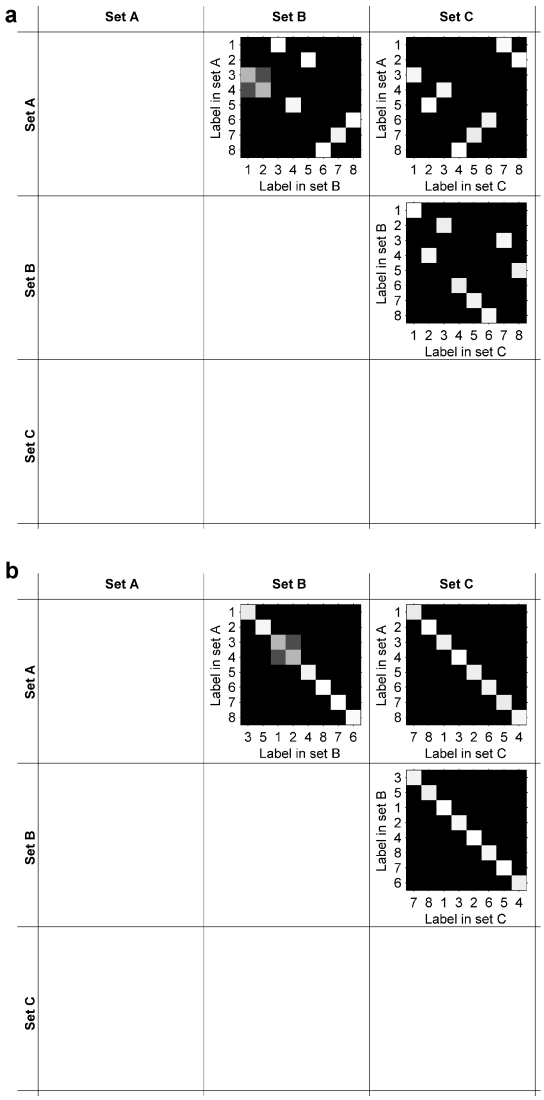
$$P_3^{i,l} = \frac{\prod_{m \neq l} P_2^{i,i,l,m}}{\sum_{j=1}^{I} \prod_{m \neq l} P_2^{i,j,l,m}}. \quad \text{(S1)}$$

Then, we characterize each set of fragments by its most uncertain individual, taking the minimum of these probabilities as

$$P_4^l = \min_{i=1\ldots I} P_3^{i,l}. \quad \text{(S2)}$$

When the relations between sets of fragments are consistent P_4 converges very fast to 1, but it tends to 0 for a set that is not sorted in a consistent way with the majority of other sets. Therefore, a low value of P_4^l means that set l does not match with the rest of the sets, at least with its current order of fragments. In order to prevent the mismatching sets to affect the relations among the other sets,

[7]The quantities that we compute to estimate probabilities have all the properties of true probabilities, having values between 0 and 1, and being normalized. But because they are computed using crude approximations and heuristic rules, they are not proper probabilities (they do not actually measure the probability of assignment of a given fragment, they only correlate with the actual probability). We should rather call them 'probability-like measures', but for simplicity we will refer to them just as probabilities.



Supplementary Figure 11: **Illustration of the process to relate three different complete sets of fragments.** Each matrix shows the probability (P_2, see section 2.10) of assignment of the fragments in one set with the fragments of another set (grayscale, black = 0, white = 1). **(a)** shows the original matrices. **b** shows the matrices after re-ordering sets B and C so that they match with set A. This re-ordering means that the two matrices in the top row must have the stronger elements in the diagonal. But the matrix in the second row will only have the strongest elements in the diagonal if all relations among the three matrices are consistent.

Nature Methods: doi:10.1038/nmeth.2994

we remove from the calculation all sets with $P_4 < 0.5$, and the values of P_3 and P_4 are recomputed for all remaining sets of fragments. Sometimes a mismatching set can be recovered by re-sorting it. This is the case when the problematic set and the reference set were not very reliable (and therefore its re-ordering is incorrect), but the problematic set can be reliably related to other sets. To recover the problematic set when this is the case, we try different orderings of its fragments and re-compute P_3 and P_4 for all sets. If for one re-ordering of the excluded set its value of P_4 rises above 0.5, we add it back to the general calculation.

Once all probabilities are computed with the most consistent orderings, we only select those sets that can be related with very high certainty. In particular, we select sets for which $1 - P_4 < 10^{-5}$. The system obtains in this way as many images of each individual as possible, up to a maximum of 3000 images of each individual, which we have found are enough even in the hardest conditions. However, for most videos we have observed that 300 reference images are typically enough, and in some cases even 50.

The images of each individual collected in this way will be used as reference images in the next steps of the algorithm.

2.9 Identification of one-individual blobs

At this step, each one-individual blob in each frame is identified following the procedure described in section 2.7. Each blob is identified independently of all other blobs, so one fragment of trajectory may contain frames identified as different individuals, and some non-identified frames.

2.10 Identification of fragments

Now we find the most likely assignment of each fragment as a whole using the identity of each blob obtained in the previous step. We do this in three steps: First, we find the probability that each fragment belongs to each individual, taking into account the identities of its images. We call this probability P_1 (section 3.1 describes how to compute P_1). Second, we re-compute the probability that each fragment belongs to each individual, but this time taking into account not only the identities of its images, but also the identities of other fragments that coexist with it (this is probability P_2, as described in section 3.2). Third, we use the probability P_2 to assign each fragment to its most likely individual in the following way.

First, we take the fragment assigned with most certainty of the whole video, that is, the fragment for which the ratio between the probability (P_2) of the most likely individual and the probability (P_2) of the second-most likely individual is highest. We assign the most likely individual to this fragment, and we record its corresponding probability P_2 as a measure of the certainty of the assignment. In order to ensure that no two co-existing fragments are assigned to the same individual, we modify the probabilities P_1 of the assigned fragment, setting it to 1 for the assigned individual and to 0 for the rest of individuals. We then re-compute the probabilities P_2 of all remaining fragments. Because of the change in P_1 that we have just effected, all fragments that co-exist with the fragment that we have assigned will have $P_2 = 0$ for the individual that has been already assigned. We then repeat the process, looking among the remaining fragments for the fragment that can be assigned with highest certainty, and assign it. We iterate until all fragments have been assigned. It may happen that at the end of this process some fragments have equal probability of belonging to several individuals. These fragments will be left unidentified.

By construction, after this step all frames of each fragment are assigned to the same individual, and no two blobs of the same frame can be assigned to the same individual.

2.11 Building of the trajectories

The program builds the trajectory of each individual as the succession of the centers of all the blobs that have been assigned to it. We compute the center of a blob as the center of mass of the eroded blob, using the same erosion parameters as in section 2.6. This definition has two advantages with respect to just computing the center of mass of the whole blob. First, for resegmented blobs (section 2.6) and for some of the blobs during crossings (section 2.12) we can only compute the center using the eroded blob (because the original one belonged to more than one individual). Using the same criterion in the rest of the frames increases the consistecy of the trajectories. Second, in some cases this criterion provides a center that is closer to the point a human would consider the center of the animal. For example when zebrafish turn they become C-shaped. The center of mass of the 'C' often falls outside the body of the animal, while the center of the resegmented blob is in most cases inside the body.

Only blobs belonging to one-individual fragments are assigned to individuals, so the trajectories have gaps when several individuals cross.

Nature Methods: doi:10.1038/nmeth.2994

2.12 Estimation of each individual's position during the crossings

Until now we have used no dynamical information in the crossings to identify the individuals. For this reason some of the brief identity switches may occur between individuals that are very far apart in the set-up. While the proportion of mistaken identities is very small, some of these mistakes are so easy to correct that it is worth to do it before filling the gaps in the trajectories.

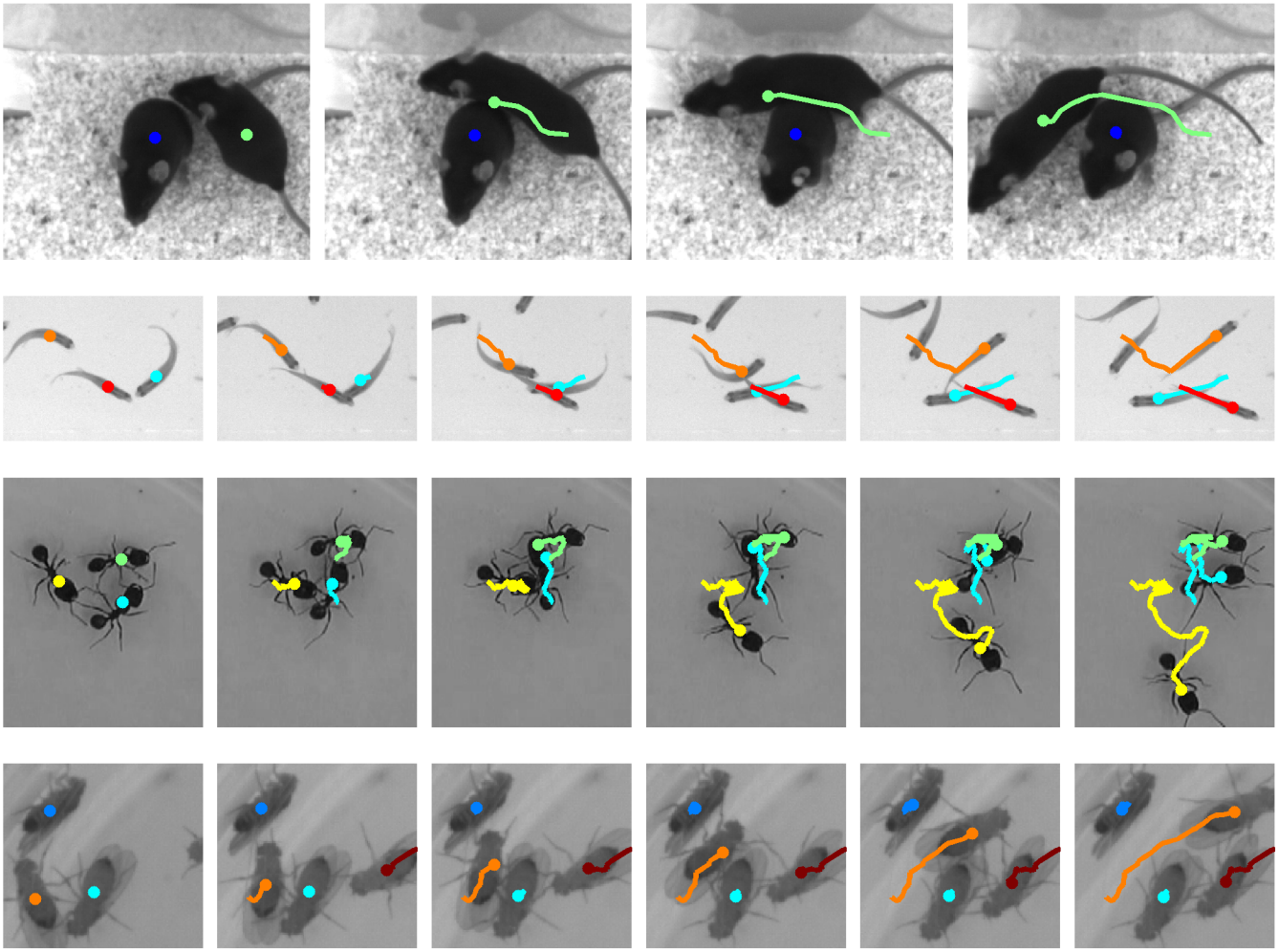
First we extract from the trajectories the speed of each individual in all frames in which it is unoccluded, calculate the typical maximum velocity $v_{\max}$ as the 99-percentile of the speeds. We will suspect wrong identity whenever connecting two fragments of the same individual requires a speed higher than $2v_{\max}$. We will work iteratively, using the identities of the most certain fragments (we will say that these fragments have 'fixed' identities) to estimate what identities of the less certain fragments are more compatible with the dynamics. First, we fix the identities of all fragments whose estimated probability of being correct is higher than 0.9. Then we find the possible identities for each fragment whose identity has not been fixed. We say that identity A is possible for fragment f when (1) no other fragment that coexists with f has identity A and has been fixed, (2) the speeds needed to connect the first frame of fragment f to the last frame of the previous fragment fixed to identity A is lower than $2v_{\max}$ and (3) the same as condition 2 with the last frame of fragment f and the first frame of the next fragment fixed to identity A. When a fragment has only one possible identity, we fix it into this identity. For fragments that have several possible identities, we look for the one that minimizes the distances between its extreme frames and those of the previous and next fragments. If this identity is the same as the one originally assigned to the fragment, we fix it. We iterate this procedure until no more fragments can be fixed. Then remove the identity of fragments with several possible identities whose distance to the previous and next neighbours would be minimized by changing its original identity and whose estimated probability of correct identity is lower than 0.7. These fragments are left without identity, and might recover it in the next step. Then we update the trajectories to reflect the changes in the identities.

We define a gap in the trajectories as the portion between the two nearest frames where all individuals have an identity. We will fill these gaps starting from the edges and moving towards the center. We start in the first frame of the gap, and we erode all the blobs keeping only pixels that are further away from the periphery than a distance d_{e} (defined in section 2.6). For each individual that is not assigned in the current frame, we consider only the eroded blobs that overlap with the blob where the individual was located in the previous frame. If no eroded blobs overlap with it, we consider all of them. We start from an initial position computed by simple interpolation between the position in the previous frame and the position at the other side of the gap. If this position falls inside an eroded blob, we assign the individual to it. Otherwise, we move the position to the nearest eroded blob, as long as one of the following three conditions is fulfilled: (1) The distance between the interpolated position and the eroded blob is lower than the threshold $2v_{\max}$, (2) the distance between the eroded blob and the position of the individual in the previous frame is lower than $2v_{\max}$, or (3) the eroded blob overlaps with the blob assigned to the individual in the previous frame. Once the positions of all individuals have been estimated in one frame, we check whether any of the original blobs contains the estimated position of one single individual. If this is the case we assign its identity to the whole fragment of trajectory corresponding to the blob. It may happen that an eroded blob contains one single individual, but that the original blob contains also other individuals (if the erosion divided the original blob in several eroded blobs). In this case we send the estimated to the center of the eroded blob and propagate the position of this individual to as many frames as possible, using on the segmented blobs the same criterion that we used to create the fragments (section 2.2). Then we move to the frame in the opposite side of the gap, and iterate until the full gap is closed.

In order to improve the quality of the estimation in the frames where erosion does not separate the individuals, after closing a gap for the first time we remove all the estimated positions except those of individuals that are alone in an eroded blob. In this way the original gap is divided into smaller gaps. We repeat the procedure described above on each of these smaller gaps, working from the sides towards the center.

This procedure has the advantage of being applicable to any species and having no parameters that must be tuned by the used. They provide an estimate of the position that falls inside the body of the animal in 96.5% of the cases (see **Supplementary Table 1**). **Supplementary Figure 12** shows some examples of the estimated trajectories.

Nature Methods: doi:10.1038/nmeth.2994



Supplementary Figure 12: **Illustration of the estimation of position during crossings.** Several examples showing the estimated trajectories during a crossing.

Nature Methods: doi:10.1038/nmeth.2994

3 Computation of the identification probabilities (P_1 and P_2)

3.1 Identification of one individual from several images (calculation of P_1)

Consider several images that belong to the same individual (problem individual). We want to compute the probability[8] of each possible identity for the problem individual.

The first step is to identify each of the problem images as described in section 2.7. After this step each image is either assigned to one of the possible identities, or not assigned (if the identification was ambiguous, or the image was flagged as not valid for identification).

Let F be the total number frames, and let f_i be the number of frames that have been identified as belonging to the i-th individual, with f_0 corresponding to the frames that could not be identified. We have that $\sum_{i=0}^{I} f_i = F$, where I is the total number of individuals present in the video. We make the conservative assumption that the probability to assign one image to the correct individual is twice as large as the probability to assign the image to any of the incorrect individuals.[9] Then, assuming that all individuals are equally likely *a priori*, and that images in the fragment are independent of each other, the probability that the fragment belongs to the k-th individual is[10]

$$P_1(\text{indiv. } k) = \frac{2^{f_k}}{\sum_{i=1}^{I} 2^{f_i}}. \qquad \text{(S3)}$$

However, the assumption that the images in a fragment are independent of each other is unrealistic: when animals move slowly, two consecutive frames can be almost identical, giving almost the same information. In the process of collecting references (section 2.8) this problem is solved by only taking frames that are different from each other, so we can use directly Equation S3. But when making the final identifications of the video (section 2.10) we often need to identify very short fragments, and in this case we need to use all frames of the fragment. To take dependencies between neighbouring frames into account, we weight each frame according to how much unique information it contains. We do this by still using Equation S3, but re-computing the values of f_i so that each frame contributes to f_i with one unit if it provides unique information, and contributes almost nothing if it contains mostly duplicated information. To measure the similarity between images (and therefore duplication of information), we use the overlap between consecutive blobs. Thus, we will assign to each blob a weight that is inversely proportional to the amount of overlapping with blobs of other frames of the fragment. The algorithm works as follows: For each pixel of one blob identified as the i-th individual, we add $1/(pb)$ units to f_i, where p is the number of pixels of the blob, and b is the number of blobs of the fragment that overlap with that same pixel. Note that p and b are not constants, and in general will be different for each blob and each pixel. But we omit subscripts to keep notation simple. Thus, for example one blob that does not overlap with any other blob will contribute one unit (b will be 1 for all its pixels, so its total contribution will be $p(1/p)$. In the other extreme, if we have for example 7 identical blobs that overlap completely, all of them together will contribute one unit, because p will be equal for the 7 blobs and b will be 7 for all their pixels, so the total contribution will be $7p/(7p)$. For intermediate cases, the contribution will be inversely proportional to the overlaps among blobs.

3.2 Identification of one individual, including information from other individuals (calculation of P_2)

In this section we deal with the case in which we have several images that belong to the same individual, and one or several other sets of images that belong to different individuals. This will be the case, for example, when several individuals are moving separately in different parts of the set-up. In this case we can use the information extracted from the images of the other individuals to improve the

[8] See footnote 7.

[9] In general some wrong individuals will have higher probabilities than others, but we neglect these differences.

[10] Equation S3 is derived as follows. Let α be the probability of identifying a frame as one of the wrong individuals. Then, by hypothesis, 2α is the probability of identifying the frame as the correct individual. Then, if k is the correct individual, the probability of obtaining $\{f_j\}_{j=1}^{I}$ independent frames identified as each of the $j = 1 \dots I$ individuals is

$$P(\{f_j\}_{j=1}^{I}|k) = \Omega(2\alpha)^{f_k} \prod_{\substack{i=1 \\ i\neq k}}^{F} \alpha^{f_i} = \Omega 2^{f_k} \alpha^{(F-f_0)},$$

where Ω is a combinatorial term that we do not need to compute, because it will cancel out. According to Bayes' theorem, the probability that k is the correct individual, given the values of $\{f_j\}_{i=j}^{I}$ is

$$P(k|\{f_j\}_{j=1}^{I}) = \frac{P(\{f_j\}_{j=1}^{I}|k)P(k)}{\sum_{i=1}^{I} P(\{f_j\}_{j=1}^{I}|i)P(i)}.$$

Because we assume that all individuals are equally likely *a priori* the terms $P(k)$ and $P(i)$ cancel out. Then, substituting the first equation into the second and simplifying, we obtain Equation S3.

identification of the problem individual.

In particular, we will compute the probability that the problem individual is individual k, *and* none of the other individuals is individual k. Let us denote by s the set of images whose probability we are computing, and let $c_1, c_2 \dots c_C$ be all the other sets of images, that we know belong to different individuals. We compute the probability that fragment s belongs to individual k is as

$$P_2(s,k) = \frac{P_1(s,k)\prod_{j=1}^{C}(1-P_1(c_j,k))}{\sum_{i=1}^{I} P_1(s,i)\prod_{j=1}^{C}(1-P_1(c_j,i))}, \qquad \text{(S4)}$$

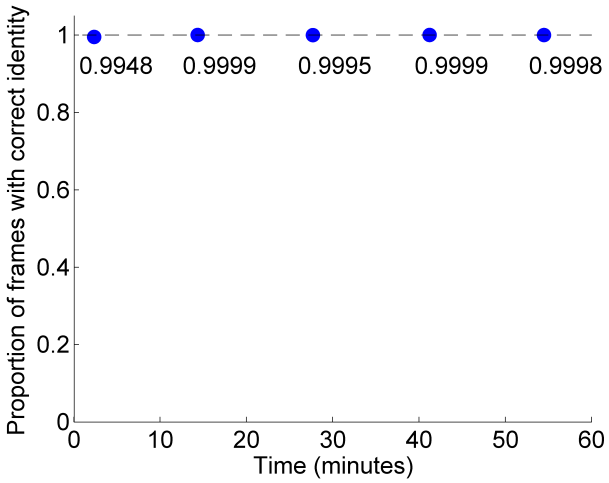
where $P_1(s,k)$ is the probability that set s belongs to individual k only taking into account its own images (computed using Equation S3 as described in the previous section).

Supplementary Note 3: Validation

1 Procedure

To validate the system we recorded several videos of different species (**Supplementary Table 1**). Besides the differences inherent to recording each species, we have tested three different types of illumination: hallogen floodlights pointing to the ceiling, hallogen flodlights pointing inwards and small hallogen lamps pointing laterally (see Methods). Some of the videos were used also during development of the software but we do not observe better performance for these cases (see **Supplementary Table 1**). We track each video with idTracker, adjusting only the parameters available to the user via the graphical interface.

The process of validation of each video is as follows. First, we ran idTracker on the whole video. Then we reviewed every crossing of a portion of the video. For this we use the visualization tool integrated in the software, that shows the video together with the results of the tracking. To validate the performance of identification in unoccluded frames, whenever we detected a fragment with mistaken identity we corrected it typing the correct identity. To validate the estimation of the positions of the individuals during the crossings, we annotated all instances when the estimated position of one animal falls outside its body, recording the frame and the animal for which it happens. Reviewing the videos is extremely time-consuming, so we only validated a portion of each video (any portion of the video is representative of the whole sequence because the performance of idTracker does not deteriorate with time, **Supplementary Figure 13**).



Supplementary Figure 13: **Tracking performance is constant in time.** Proportion of unoccluded frames with correct identity (calculated from manual validation) computed using 3000 frames at five different times during a one hour long experiment of five zebrafish.

2 Performance of the system

2.1 Performance of the system for unoccluded frames

Here we present the results only for those fragments of the video in which an animal moves separated from the rest, without crossing with any other animal.

Supplementary Figures 14-18 give an analysis of the validation results for each video. These figures contain several boxes for each video: Box (a) gives a representative frame of the video; (b) gives a histogram of the sizes of the individuals; (c) gives the correctly labeled trajectories in color, and the incorrect portions in black (in most cases black portions are too small to be seen).

To identify an animal, the system uses all the frames between two consecutive crossings (all the frames of a fragment, see section 2.10 of Supplementary Note 2). Mistakes are less likely when the animal travels a longer distance between consecutive crossings, so we analyzed the probability of mistake as a function of this distance (**Supplementary Figures 14-18**, box d). We compute the distance traveled from the trajectories, smoothing them with a moving average of 10 frames. Each blob that corresponds to a single individual may be in one of these three states: Identified with the correct identity (green), not identified (orange) or identified with the wrong identity (red). In all cases we find that the probability of mistake is low even for the shortest fragments, and decreases fast as the length of fragments increases (most mistakes occur in fragments shorter than 1 body-length). Short fragments make up a very small proportion of the video, so overall proportion of mistakes is extremely low. On average 99.8 % of the unoccluded trajectories are correctly identified (**Supplementary Table 1**). We compute percentages as number of correct over total number (including correct, incorrect and non-identified), and taking into account only unoccluded portion of the video.

Nature Methods: doi:10.1038/nmeth.2994

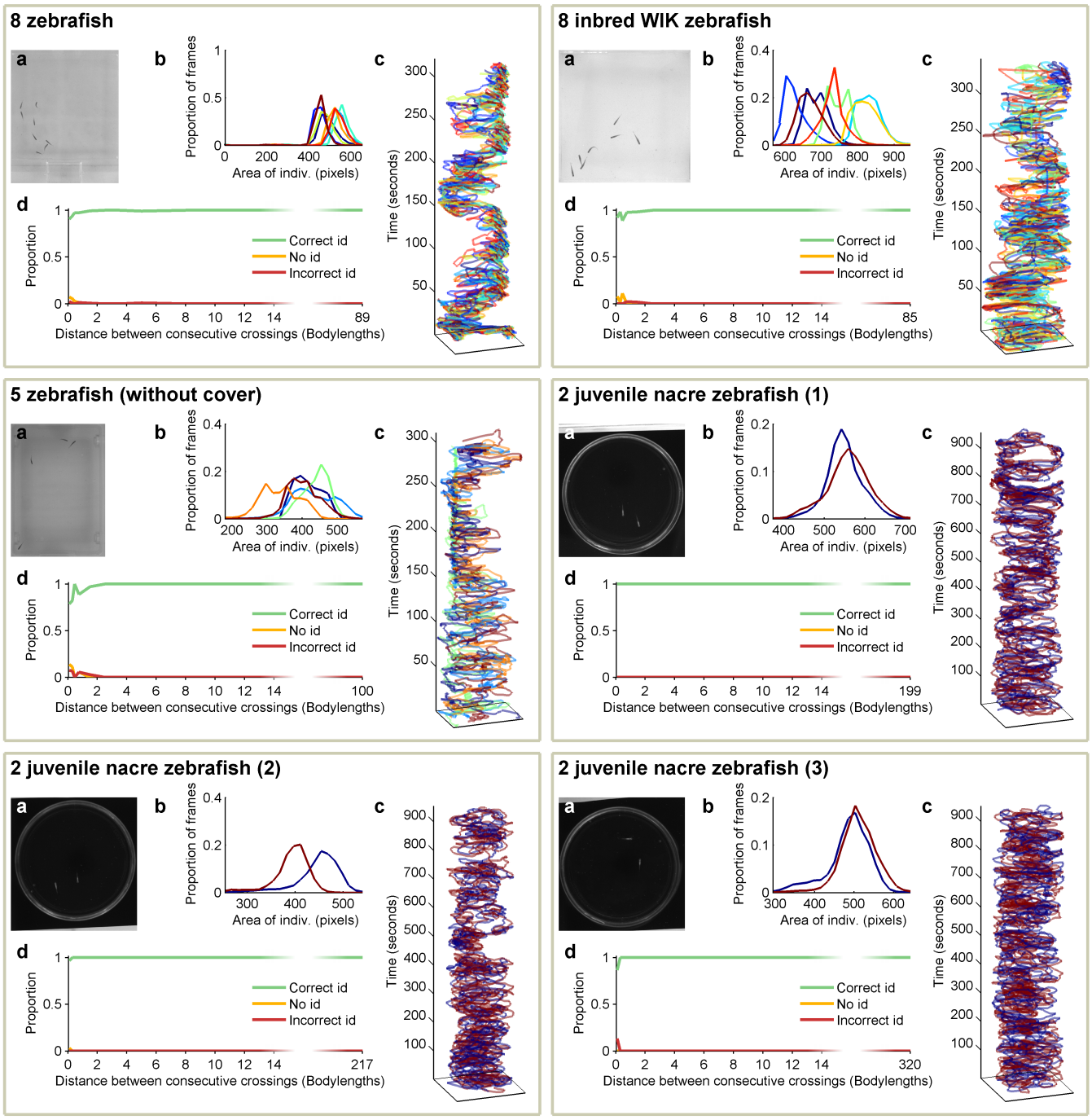
2.2 Performance of the system during the crossings

The estimated position of each during the crossings fall inside the body of the animal on average 96.5% of the times (**Supplementary Table 1**).

2.3 Overall performance

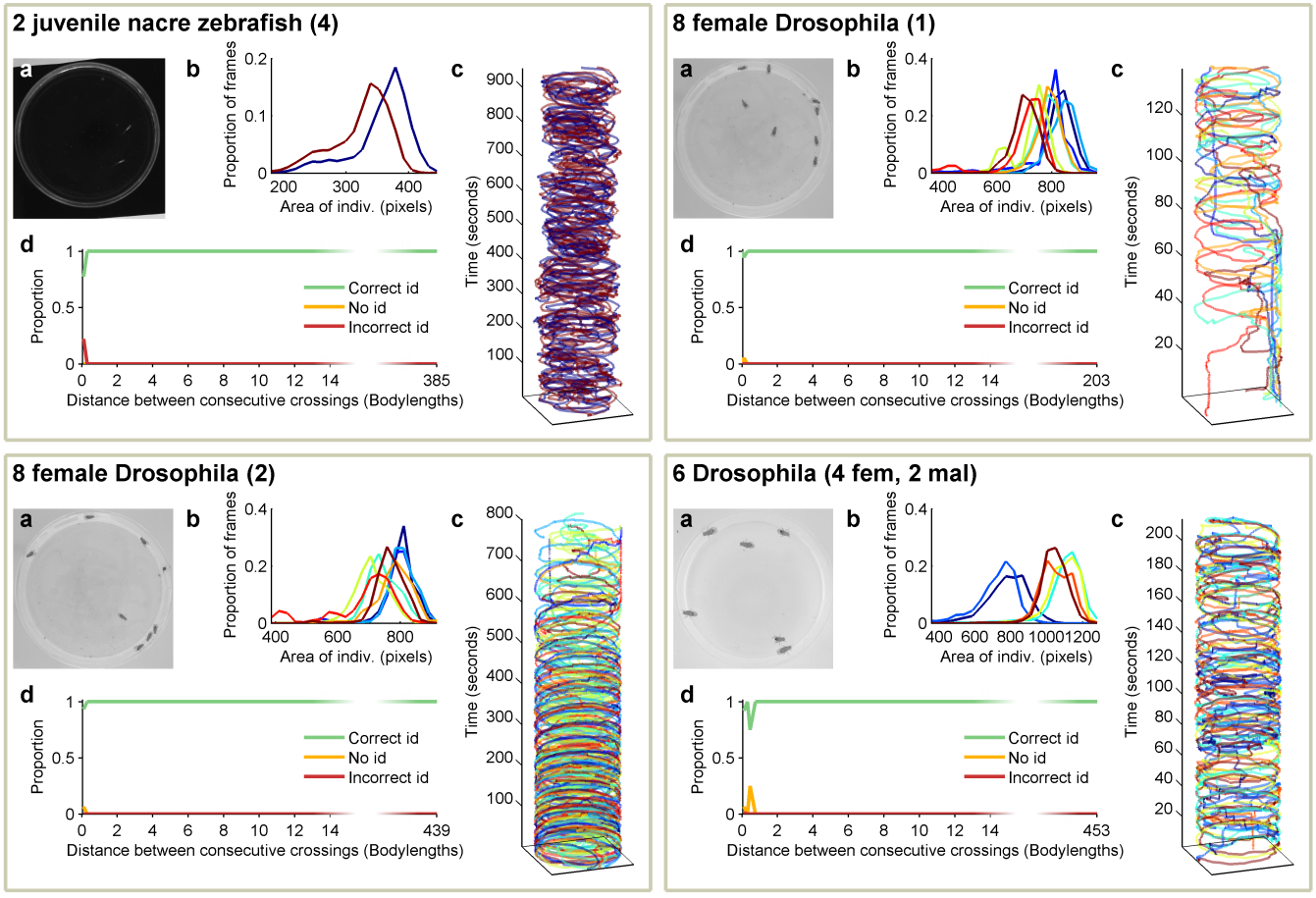
We find a very good overall performance, with 99.8% of correct identities on average (**Supplementary Table 1**).

Nature Methods: doi:10.1038/nmeth.2994



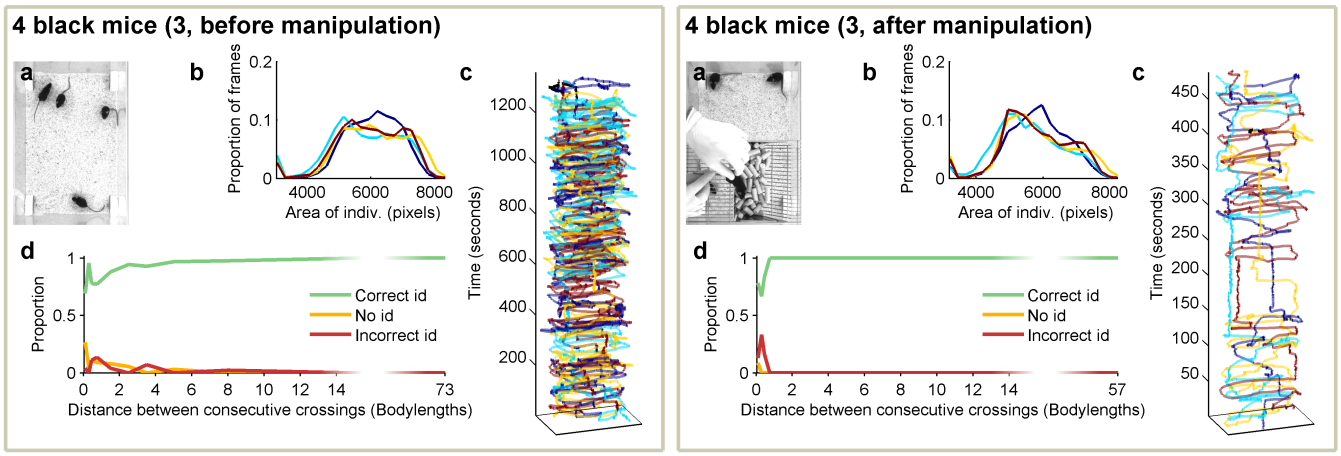
Supplementary Figure 14: **Results of validation for zebrafish (1/2).** For each video: **(a)** Example frame. **(b)** Distribution of sizes (area of the segmented blobs) of each individual along the validated portion of the video. **(c)** Trajectories (x and y axes correspond to the sides of the set-up, z axis shows time). Colors represent correct identities, and black represents wrong identities. **(d)** Probability of outcome for single-individual fragments of different lengths. Green for correct identity, orange for non-identified fragment, and red for wrong identity. The x-axis extends up to the longest fragment of the validated portion.

Nature Methods: doi:10.1038/nmeth.2994



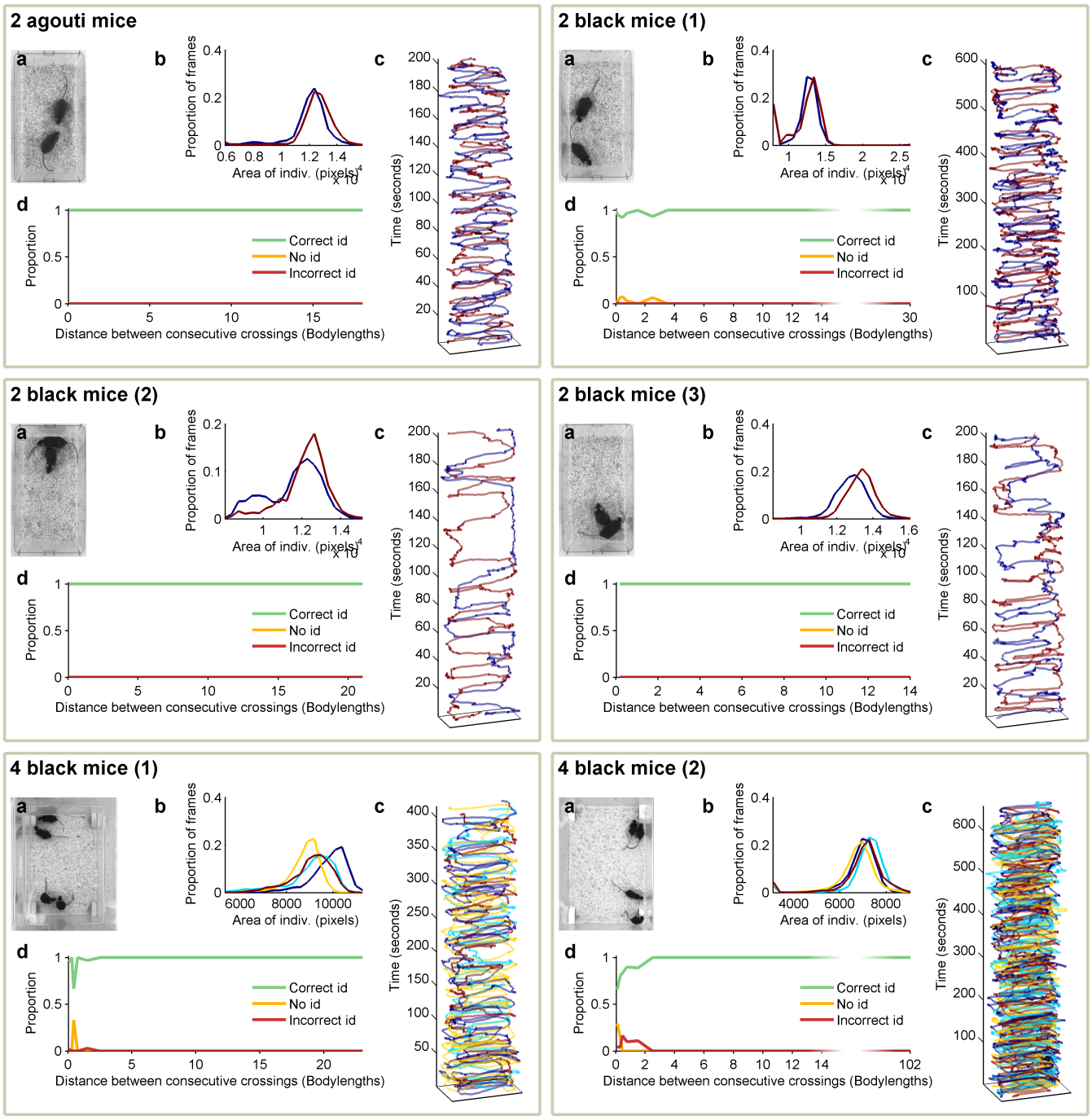
Supplementary Figure 15: **Results of validation for zebrafish (2/2) and *Drosophila*.** For each video: **(a)** Example frame. **(b)** Distribution of sizes (area of the segmented blobs) of each individual along the validated portion of the video. **(c)** Trajectories (x and y axes correspond to the sides of the set-up, z axis shows time). Colors represent correct identities, and black represents wrong identities. **(d)** Probability of outcome for single-individual fragments of different lengths. Green for correct identity, orange for non-identified fragment, and red for wrong identity. The x-axis extends up to the longest fragment of the validated portion.

Nature Methods: doi:10.1038/nmeth.2994



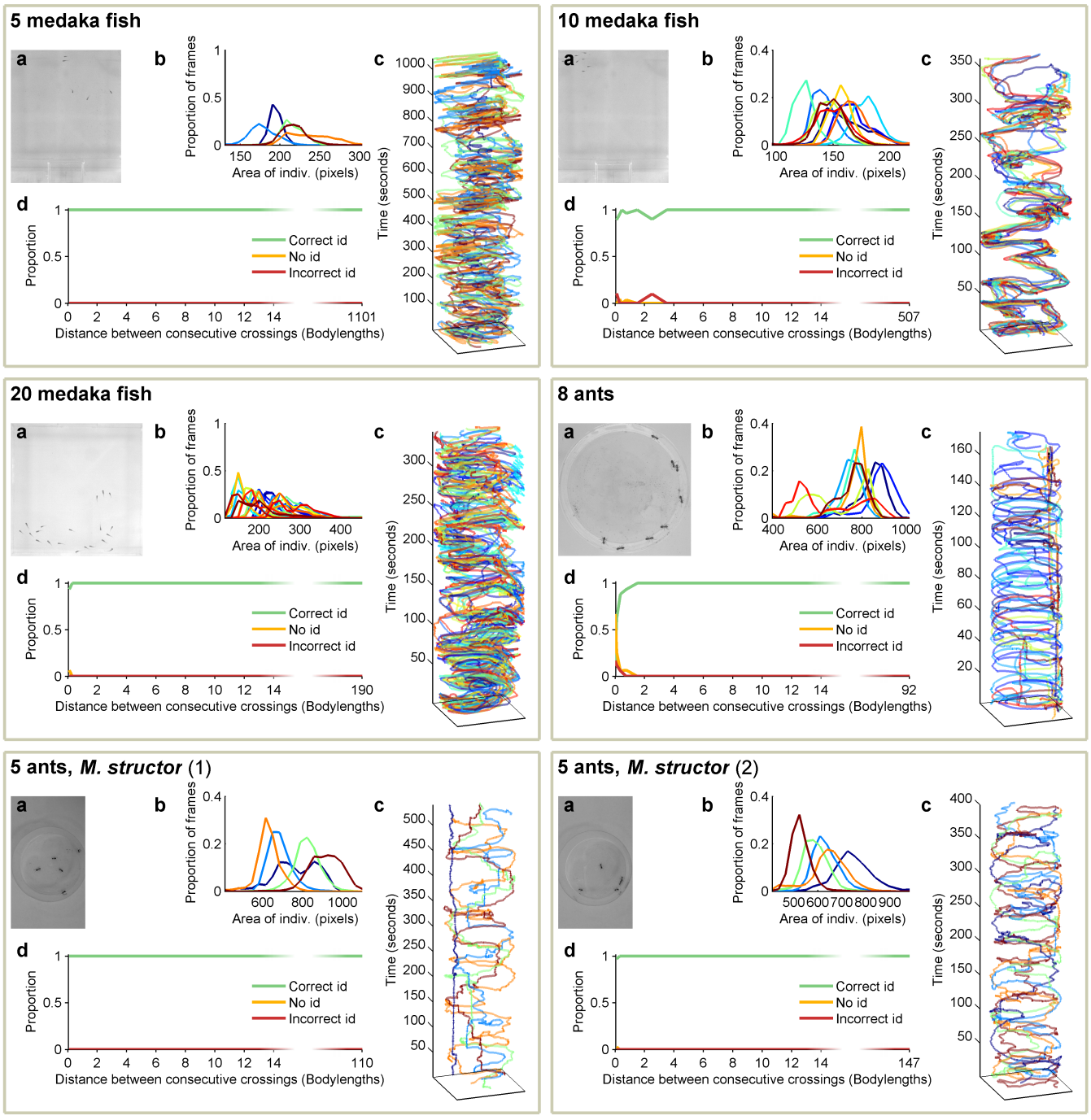
Supplementary Figure 16: **Results of validation for mice before and after manipulation.** We recorded a 1-hour video of 4 mice (typical frame in box a on the left). 20 minutes after the start of the video we placed a platform on the set-up, captured the four mice one by one, and caressed them against the direction of fur for a few seconds (box a on the right). This manipulation lasted about 3 minutes, and afterwards we removed the platform and kept recording for the remaining time. We tracked the whole video without any special setting regarding the time of the manipulation. idTracker succesfully built the references in spite of the disturbance, and has a good performance both before and after (left and right portions of the figure, respectively) the manipulation. We checked that the identities are consistent before and after the manipulation, using small marks on the tails of the mice (the marks are too small and light to be segmented, so they do not affect the tracking). For each video: **(b)** Distribution of sizes (area of the segmented blobs) of each individual along the validated portion of the video. **(c)** Trajectories (x and y axes correspond to the sides of the set-up, z axis shows time). Colors represent correct identities, and black represents wrong identities. **(d)** Probability of outcome for single-individual fragments of different lengths. Green for correct identity, orange for non-identified fragment, and red for wrong identity. The x-axis extends up to the longest validated fragment.

Nature Methods: doi:10.1038/nmeth.2994



Supplementary Figure 17: **Results of validation for mice.** For each video: **(a)** Example frame. **(b)** Distribution of sizes (area of the segmented blobs) of each individual along the validated portion of the video. **(c)** Trajectories (x and y axes correspond to the sides of the set-up, z axis shows time). Colors represent correct identities, and black represents wrong identities. **(d)** Probability of outcome for single-individual fragments of different lengths. Green for correct identity, orange for non-identified fragment, and red for wrong identity. The x-axis extends up to the longest fragment of the validated portion.

Nature Methods: doi:10.1038/nmeth.2994



Supplementary Figure 18: **Results of validation for medaka and ants.** For each video: **(a)** Example frame. **(b)** Distribution of sizes (area of the segmented blobs) of each individual along the validated portion of the video. **(c)** Trajectories (x and y axes correspond to the sides of the set-up, z axis shows time). Colors represent correct identities, and black represents wrong identities. **(d)** Probability of outcome for single-individual fragments of different lengths. Green for correct identity, orange for non-identified fragment, and red for wrong identity. The x-axis extends up to the longest fragment of the validated portion.

Nature Methods: doi:10.1038/nmeth.2994

3 Test with other systems.

We have checked that other tracking systems are unable to produce comparable results without human supervision. We have run the system in ref. 8 on one video of 5 zebrafish (without cover), one video of 8 fruitflies and one video of 2 mice. The system solves a high portion of the crossings correctly (especially for the case of flies), but in the end some identity swaps occur and the final proportion of correct frames is compatible with random assignments (around 25% for zebrafish, 10% for flies and 50% for mice). We have also tested the system in ref. 11 on two of our videos of two mice. Again we find a good performance in the resolution of crossings, but identity swaps occur eventually, and without manual intervention the final proportion of correct frames is around 50%, with the system completely losing track of both animals in the most severe cases. In contrast, our system achieves 99.46%, 99.98% and 99.3% trajectories correct for these videos (**Supplementary Table 1**).

Nature Methods: doi:10.1038/nmeth.2994

Supplementary Note 4: Contribution of each pixel to identification

Our identification algorithm is based on a global transformation of the image, and therefore in principle it does not make use of specific local features of the animals. However, the differences between individuals might actually reside in local features. If this was the case, we should expect that only the pixels of the image that form part of the relevant feature really contribute to the identification, while the rest simply add noise.

In order to measure the contribution of each pixel to the identification of a problem image, we take the following steps: First we measure the contribution to identification of each element of the intensity and contrast maps of the image (Figure 3e-f of main text). Then, for each pixel of the image, we find the overall contribution of all the elements of the intensity map related to it, and likewise for the contrast map. The exact procedure is as follows.

Step 1. Contribution of each element of the intensity map. We take a video of I individuals that has been tracked with idTracker. We select a blob that is identified with high certainty as belonging to individual a. Let $p_{l,m}$ be the elements of the intensity map of the problem image. Then, for each individual we select the reference image which is nearest (in terms of its intensity distance[11]) to the problem image. Let $r^i_{l,m}$ be the elements of the intensity map of the reference image of the i-th individual that is nearest to the problem image. For each element of the intensity maps, we compare the difference between the problem map and the reference map of the correct individual (a) with the differences between the problem map and the reference maps of the incorrect individuals. In particular, we compute a matrix of contributions (C) whose elements are

$$C_{l,m} = \sum_{i=1}^{I} \left(\left| p_{l,m} - r^i_{l,m} \right| - \left| p_{l,m} - r^a_{l,m} \right| \right). \qquad \text{(S5)}$$

Higher values of $C_{l,m}$ mean that the corresponding element of the intensity map has a higher contribution to the correct identification of the problem image.

Step 2. Contribution of each pixel of the image Each element of the intensity map is computed using many pairs pixels of the image. In turn, each pixel of the image contributes to many elements of the intensity map. For this reason, to compute the influence of one pixel of the image in the identification, we sum all the elements $C_{l,m}$ to which it contributes. In particular, we compute the contribution of pixel k as

$$D'_k = \sum_{f=1}^{F} C_{l(k,f),m(k,f)}, \qquad \text{(S6)}$$

where F is the total number of pixels, and $l(k,f)$ and $m(k,f)$ are the indices of the elements of the intensity map that correspond to the pair of pixels k, f.

For visualization purposes we normalize the contribution so that the maximum is 1, by defining

$$D^{\text{intensity}}_k = \frac{D'_k}{\max\limits_{f=1\ldots F} D'_f}. \qquad \text{(S7)}$$

We follow the same process for the contrast maps, getting the relative contribution of each pixel to identification through the contrast map, D^{contrast}.

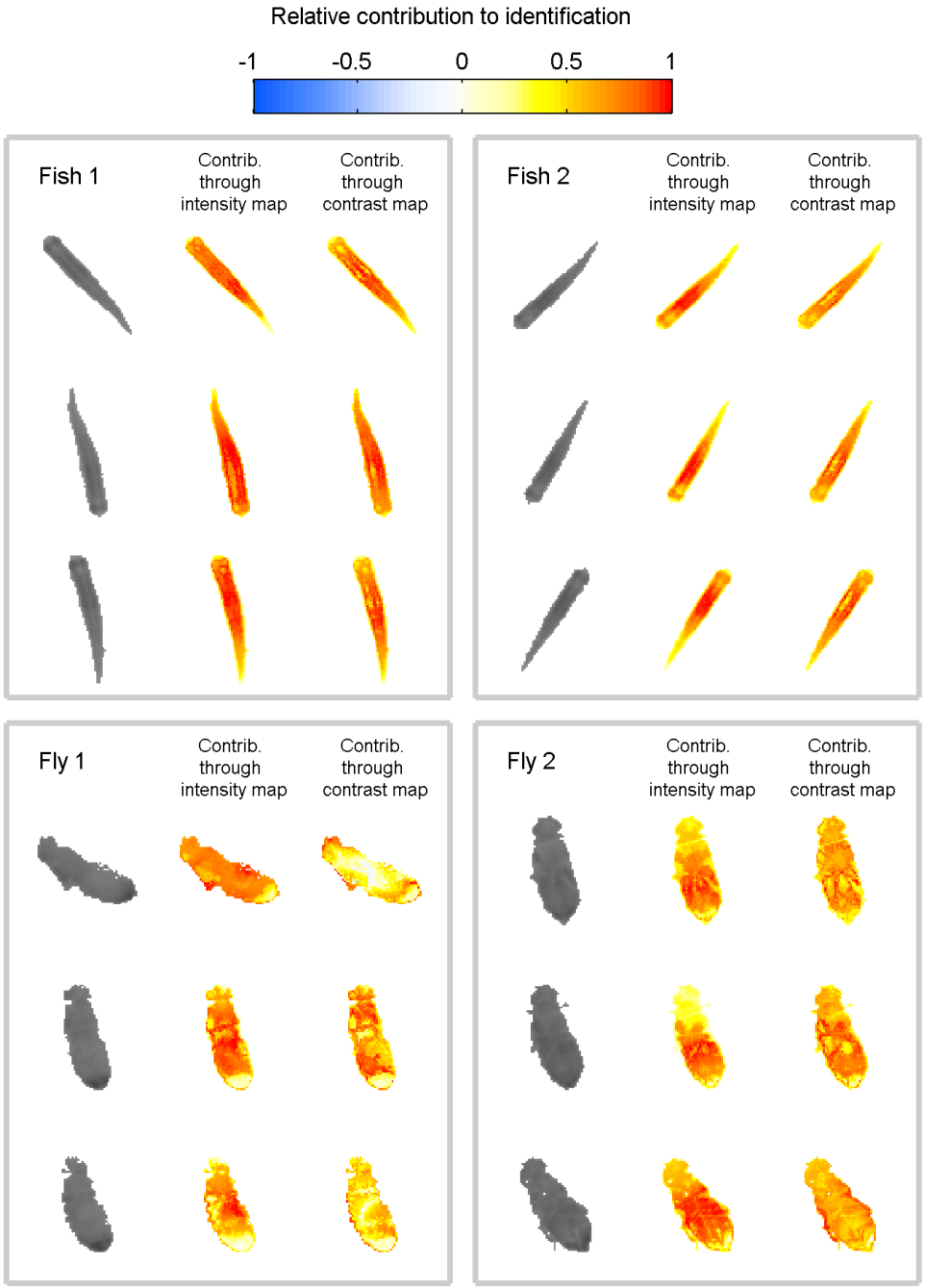
Supplementary Figure 19 shows $D^{\text{intensity}}$ and D^{contrast} for some example images of zebrafish and fruitflies. We do not find evidence of any local features that are responsible for the identification, for the three following reasons:

- Most pixels contribute positively to identification (hot colors in **Supplementary Figure 19**). Very few pixels have negative contriution, and in all cases are near-zero (white in **Supplementary Figure 19**).
- There are no clear maxima in small regions that can be associated to local features.
- The pixels with maximum contribution are not always the same for the intensity and contrast maps.
- The pixels with maximum contribution are in different parts of the body for different individuals, and even for different images of the same individual.

We conclude that idTracker is distinguishing the individuals using information that is distributed over the whole image, rather than exploiting local features.

[11] Defined in Section 2.4 of Supplementary Note 2

Nature Methods: doi:10.1038/nmeth.2994



Supplementary Figure 19: **Contribution of each pixel of the image to the identification of the individual.** Positive values indicate that the net effect of the pixel is to contribute to the correct identification, and negative values indicate that the net effect is to contribute to wrong identification. Each gray box corresponds to one individual. For each individual, each row corresponds to a different frame of the same video. **Left column:** Segmented blob. **Middle column:** Contribution of each pixel to identification through the intensity map ($D^{\text{intensity}}$). **Right column:** Contribution of each pixel to identification through the contrast map (D^{contrast}).

Nature Methods: doi:10.1038/nmeth.2994