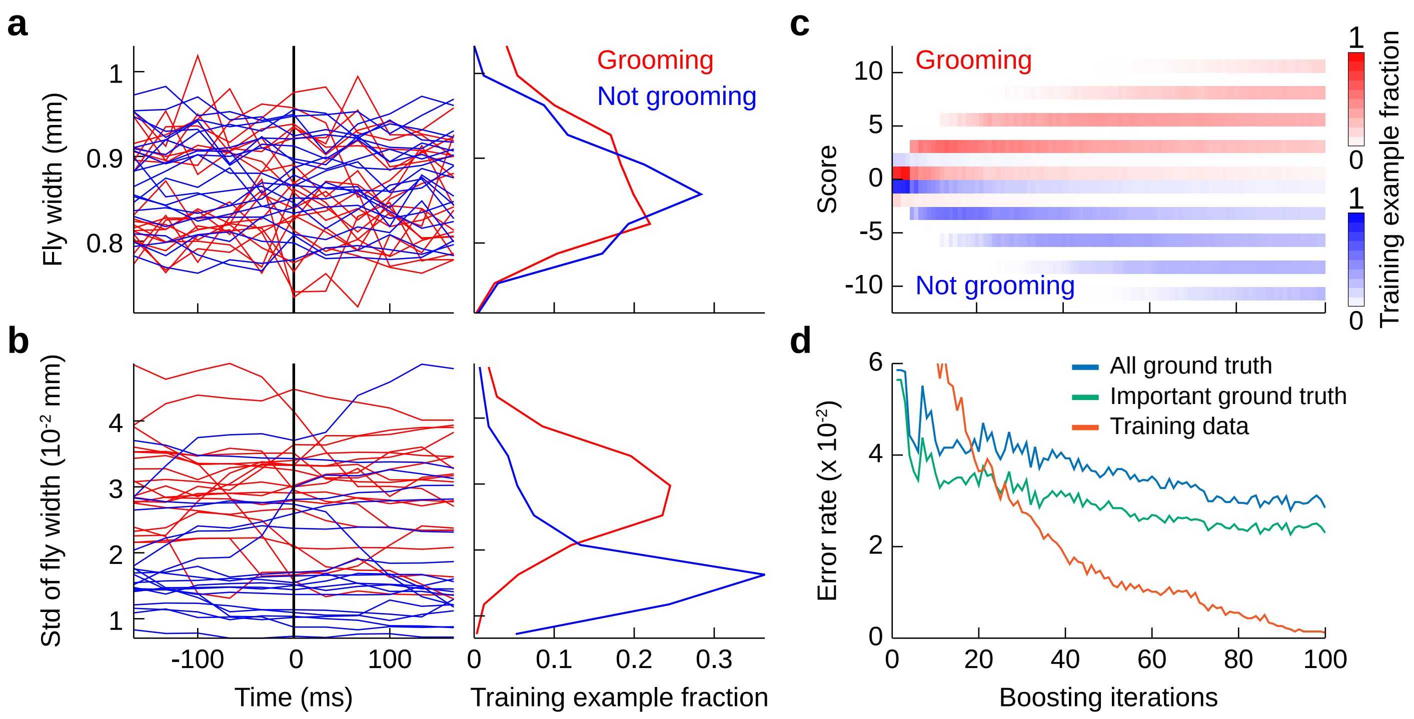
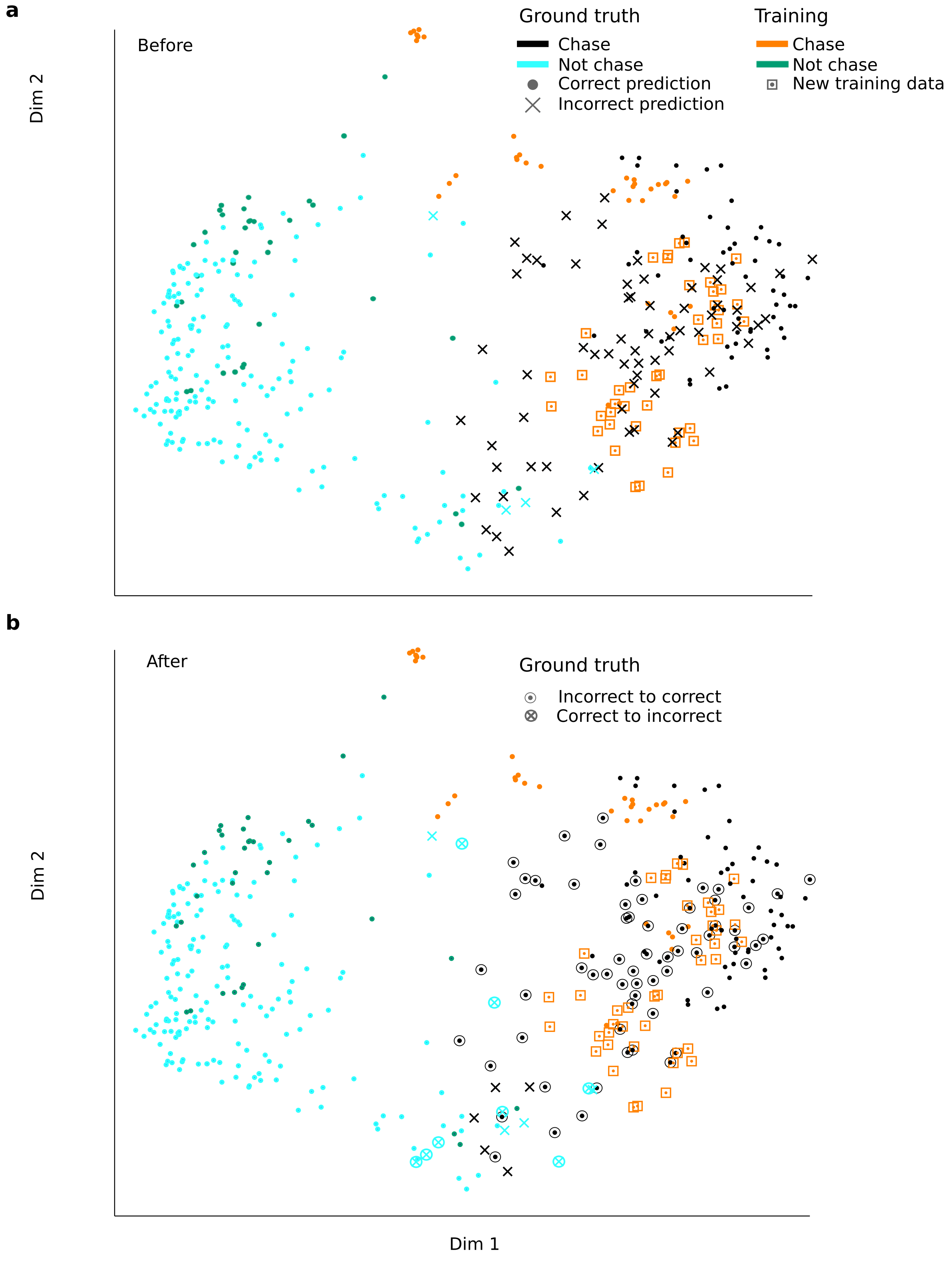
Supplementary Figure 1: Training a behavior detector.



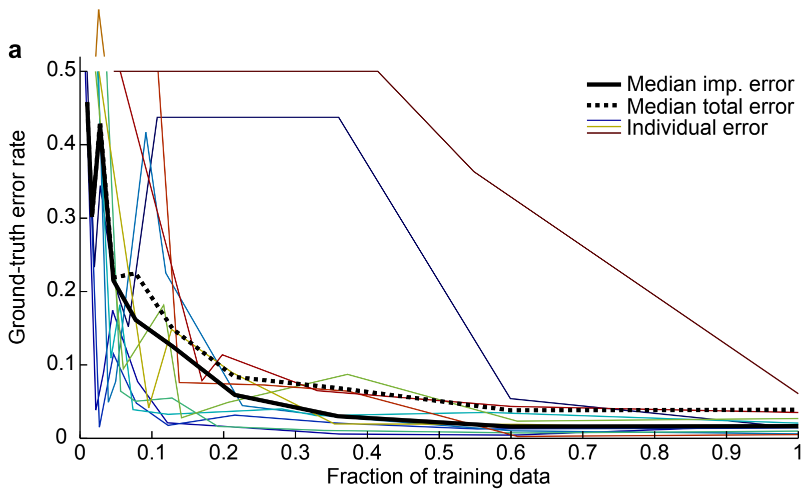
(**a**) Discriminative power of per-frame features. In the left panel each line represents the fly's width (y-axis) reported by the tracking system in a 333 ms (11 frame) interval (x-axis) surrounding sampled frames labeled while training our wing grooming detector. The color denotes the label for the center frame: red for wing grooming and blue for not wing grooming. In the right panel is the histogram of this per-frame feature (y-axis) for wing grooming and not wing grooming training frames. The y-axis indicates the fraction of frames with the given feature value. No single threshold of this per-frame feature can discriminate the two classes. (**b**) Discriminative power of window features. In the left panel each line represents the fly-width standard-deviation window feature for the intervals in a. This window feature is computed as the standard deviation of the width per-frame features in a 633 ms (21 frame) interval that starts at the current frame. The features are computed for all the frames by sliding the interval. In the right panel is the histogram of this window feature for wing grooming and not wing grooming training frames. This window feature can better discriminate the training examples than the per-frame feature in a. (**c**) Effect of increased boosting iterations on discriminability. We show the fraction (color saturation) of positive (red) and negative (blue) training examples as a function of the classifier's raw score (y-axis) after each iteration of boosting (x-axis). With each iteration, wing grooming and not wing grooming examples in the training set separate, moving farther away from the decision boundary (score = 0). (**d**) We plot the error rate (y-axis) after each iteration of boosting (x-axis) for the training data (orange), important ground-truth data (green) and all ground-truth data (blue) (see Methods section). All error rates decrease with increased iterations of boosting.

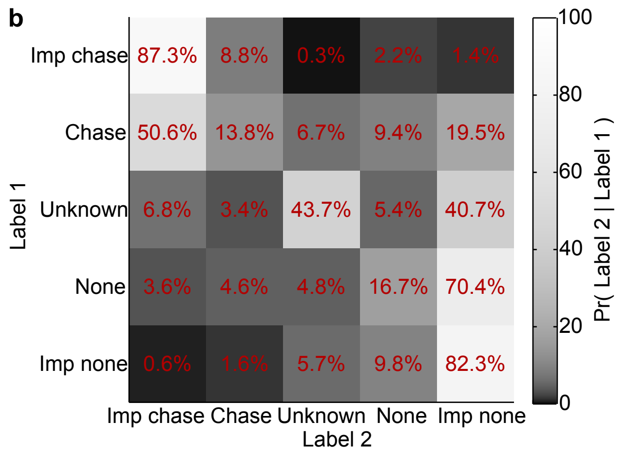
Supplementary Figure 2: Improving the classifier's performance by active learning.



In both plots, training and ground-truth examples are embedded in a 2-dimensional space such that examples that are close in the embedded space are likely to have the same predictions when the training data set is slightly perturbed. Hence, such examples are similar for the classifier. The distance between examples was based on how correlated their predictions were from the decision stumps used by GentleBoost. (**a**) Before addition of new training examples. On chase (black) and not chase (light blue) ground-truth data points, we indicate whether the predictions of the first classifier are incorrect (crosses) or correct (dots). The chase (orange) and not chase (green) training data used to train the first classifier are indicated by dots. Training examples that will be added in the next round of training are indicated by a dot with square around it. (**b**) After addition of new training examples. On the same ground-truth data, we show the updated predictions of the second classifier after adding the new training examples. The ground-truth examples on which the predictions switched are highlighted by a circle around the points.

Supplementary Figure 3: Chase detectors trained by novice users.

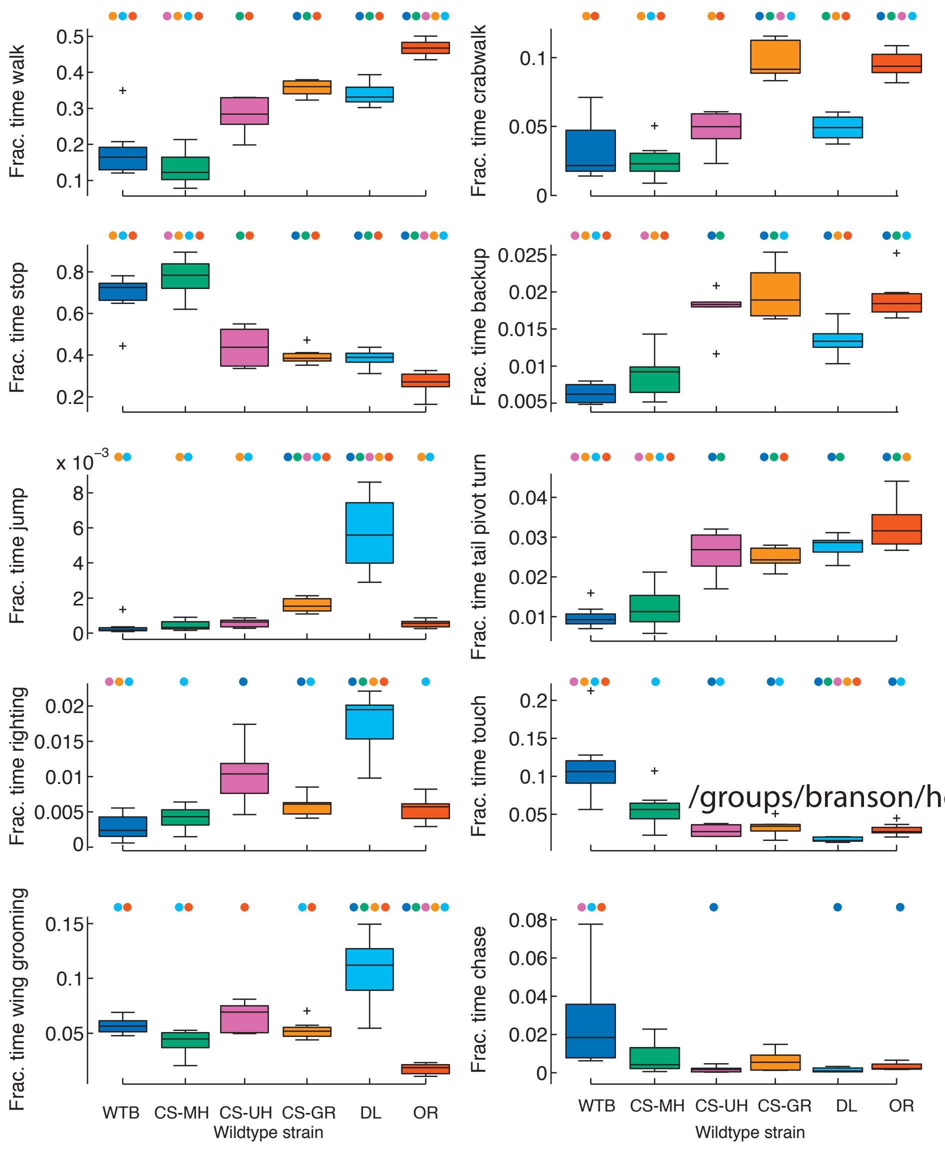




(**a**) Accuracy of detectors for each novice user. For each user, we trained behavior detectors on increasing amounts of data on a log scale. We evaluated each detector on 2,000 frames of ground-truth data labeled by the same user. The x-axis indicates the fraction of the user's training data. The y-axis indicates the detectors' error rates on ground-truth data. The median error rate over users is plotted in solid black (ground-truth data labeled as important) and dashed black (all ground-truth data labeled). In color, we plot the important error rate of each user's trained detector. While different users required different amounts of training data and used different labeling strategies, the interactive system allowed all users to create accurate classifiers, with final error rates ranging between 0.0% and 6.1%. (**b**) Inter-labeler confusion matrix. Entry row *i*, column *j* in this matrix indicates, if a user gives a frame a ground-truth label *i*, the average percentage of other users that give the frame a ground-truth label *j*. The color of each entry box also indicates this disagreement rate. The top-right corner thus indicates that if a user ground truthed a frame as "important chase", on average 1.4% of other users would ground truth the frame as "important none". This is a much lower disagreement rate than if the user ground truthed the frame as "chase", in which case on average 19.5% of other users would label the frame as "important none".

Supplementary Figure 4. Wildtype strains have discernibly different behavior profiles.

a

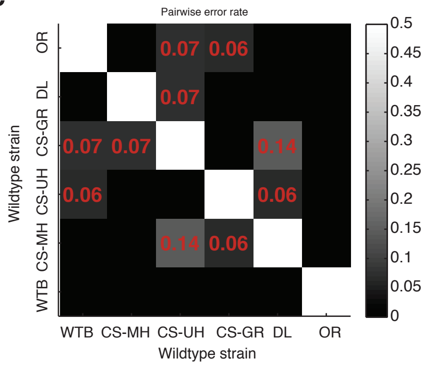


b

		CS-MH	CS-UH	CS-GR	DL	OR
Walk	WTB	0.16053	0.02930	**0.00062**	**0.00295**	**0.00008**
	CS-MH		**0.00133**	**0.00016**	**0.00016**	**0.00008**
	CS-UH			0.00799	0.04262	**0.00040**
	CS-GR				0.23450	**0.00008**
	DL					**0.00008**
Stop	WTB	0.10490	0.00466	**0.00031**	**0.00016**	**0.00008**
	CS-MH		**0.00067**	**0.00016**	**0.00016**	**0.00008**
	CS-UH			0.49084	0.28238	**0.00040**
	CS-GR				1.00000	**0.00008**
	DL					**0.00033**
Jump	WTB	0.16053	0.04262	**0.00187**	**0.00016**	0.02065
	CS-MH		0.22844	**0.00016**	**0.00016**	0.32127
	CS-UH			**0.00067**	**0.00067**	0.68891
	CS-GR				**0.00016**	**0.00008**
	DL					**0.00008**
Righting	WTB	0.13038	**0.00133**	**0.00295**	**0.00016**	0.00790
	CS-MH		0.00799	0.06496	**0.00016**	0.19959
	CS-UH			0.02930	0.01998	0.01758
	CS-GR				**0.00016**	0.48071
	DL					**0.00008**
Wing grooming	WTB	0.00466	0.28238	0.23450	**0.00187**	**0.00008**
	CS-MH		0.01265	0.04988	**0.00016**	**0.00058**
	CS-UH			0.10789	0.00799	**0.00040**
	CS-GR				**0.00062**	**0.00008**
	DL					**0.00008**
Crabwalk	WTB	0.87848	0.22844	**0.00016**	0.10490	**0.00008**
	CS-MH		0.01998	**0.00016**	**0.00187**	**0.00008**
	CS-UH			**0.00067**	0.85182	**0.00040**
	CS-GR				**0.00016**	0.81481
	DL					**0.00008**
Backup	WTB	0.04988	**0.00067**	**0.00016**	**0.00016**	**0.00008**
	CS-MH		**0.00133**	**0.00016**	0.00466	**0.00008**
	CS-UH			0.85182	0.02930	0.95465
	CS-GR				**0.00109**	0.96257
	DL					**0.00017**
Tail pivot turn	WTB	0.32821	**0.00067**	**0.00016**	**0.00016**	**0.00008**
	CS-MH		**0.00133**	**0.00031**	**0.00016**	**0.00008**
	CS-UH			0.49084	0.75458	0.06633
	CS-GR				0.03792	**0.00156**
	DL					0.04640
Touch	WTB	0.01476	**0.00067**	**0.00016**	**0.00016**	**0.00008**
	CS-MH		0.00799	0.01476	**0.00016**	0.00551
	CS-UH			0.41359	**0.00067**	0.86394
	CS-GR				**0.00109**	0.37038
	DL					**0.00033**
Chase	WTB	0.02067	**0.00067**	0.02067	**0.00016**	**0.00017**
	CS-MH		0.05927	0.95913	0.02067	0.32127
	CS-UH			0.10789	0.66201	0.27213
	CS-GR				0.01476	0.64142
	DL					0.02740

/groups/branson/home/robiea/Code_versioned/FlyBowlAnal

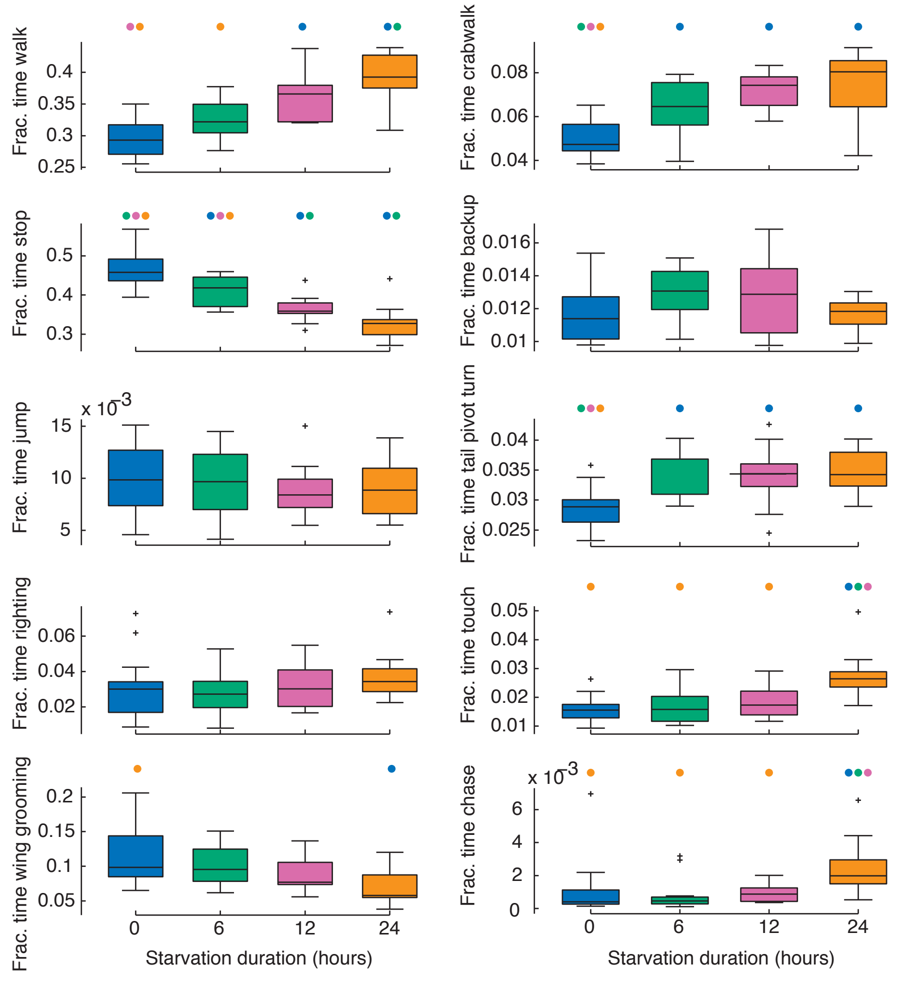
c



(**a**) Box plots show the distribution of the fraction of time spent performing each behavior: walk, stop, jump, righting, wing grooming, crabwalks, backup, tail-pivot turn, touch and chase for each wildtype strain: WTB (dark blue), CS-MH (green), CS-UH (pink), CS-GR (light orange), DL (light blue) and OR (dark orange). The fraction of time for each trial was calculated as the number of frames (over all flies) detected as performing the behavior divided by the total number of frames for all flies. The box plots display the medians, the 25th and the 75th percentiles of the data distributions and the outliers, defined as outside the whisker range of ± 2.7 sigma and are plotted as '+'. 20 flies were used in each trial for wildtype strains: WTB (number of trials n = 8), CS-MH (n = 8), CS-UH (n=6), CS-GR (n=8), DL (n = 8) and OR (n=9). Many of the data distributions were not normal (Kolmogorov-Smirnov test) thus non-parametric methods were used. The statistical comparisons were done using the two-tailed Mann-Whitney U-test and an α level of 0.05 with Bonferroni correction for multiple comparisons. All pairs with statistically significant differences are shown graphically: above each strain's box plot are colored dot(s) to indicate which of the other fly strain(s) are significantly different. (**b**) The actual p values for each comparison with statisitcally signifcant values at the adjusted α level, 0.003, in bold. (**c**) For each pair of strains (x-axis and y-axis), we evaluate the error in predicting the strain of a trial based on its behavioral statistics. The error is estimated using leave-one-out cross validation and learning the predictor using logistic regression from the remaining trials (see **Supplementary Note**). The intensity of the box indicates the pair-wise error. For pairs of strains that had non-zero error, the error rate is shown in red text.

Supplementary Figure 5. Behavioral effects of starvation duration in adult flies.

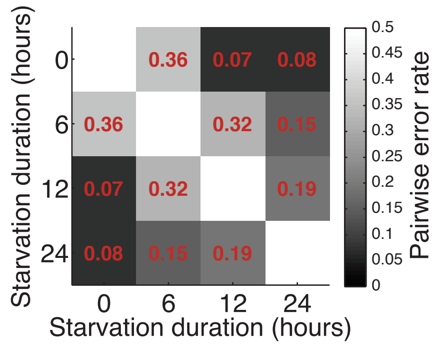
a



b

		6 h	12 h	24 h
Walk	0 h	0.02294	**0.00006**	**0.00004**
	6 h		0.00943	**0.00035**
	12 h			0.03724
Stop	0 h	**0.00353**	**0.00002**	**0.00004**
	6 h		**0.00304**	**0.00024**
	12 h			0.00939
Jump	0 h	0.59722	0.32321	0.45579
	6 h		0.66247	0.55419
	12 h			0.85714
Righting	0 h	0.83619	0.66247	0.20762
	6 h		0.30122	0.05376
	12 h			0.36807
Wing grooming	0 h	0.56573	0.11292	**0.00693**
	6 h		0.14780	0.00939
	12 h			0.07598
Crabwalk	0 h	**0.00719**	**0.00002**	**0.00091**
	6 h		0.11292	0.04219
	12 h			0.20762
Backup	0 h	0.05654	0.24133	0.69968
	6 h		0.80049	0.01677
	12 h			0.11671
Tail pivot turn	0 h	**0.00165**	**0.00408**	**0.00091**
	6 h		0.94505	0.62511
	12 h			0.85714
Touch	0 h	0.76520	0.30122	**0.00016**
	6 h		0.39531	**0.00109**
	12 h			**0.00076**
Chase	0 h	0.87224	0.10281	**0.00310**
	6 h		0.05651	**0.00109**
	12 h			**0.00130**

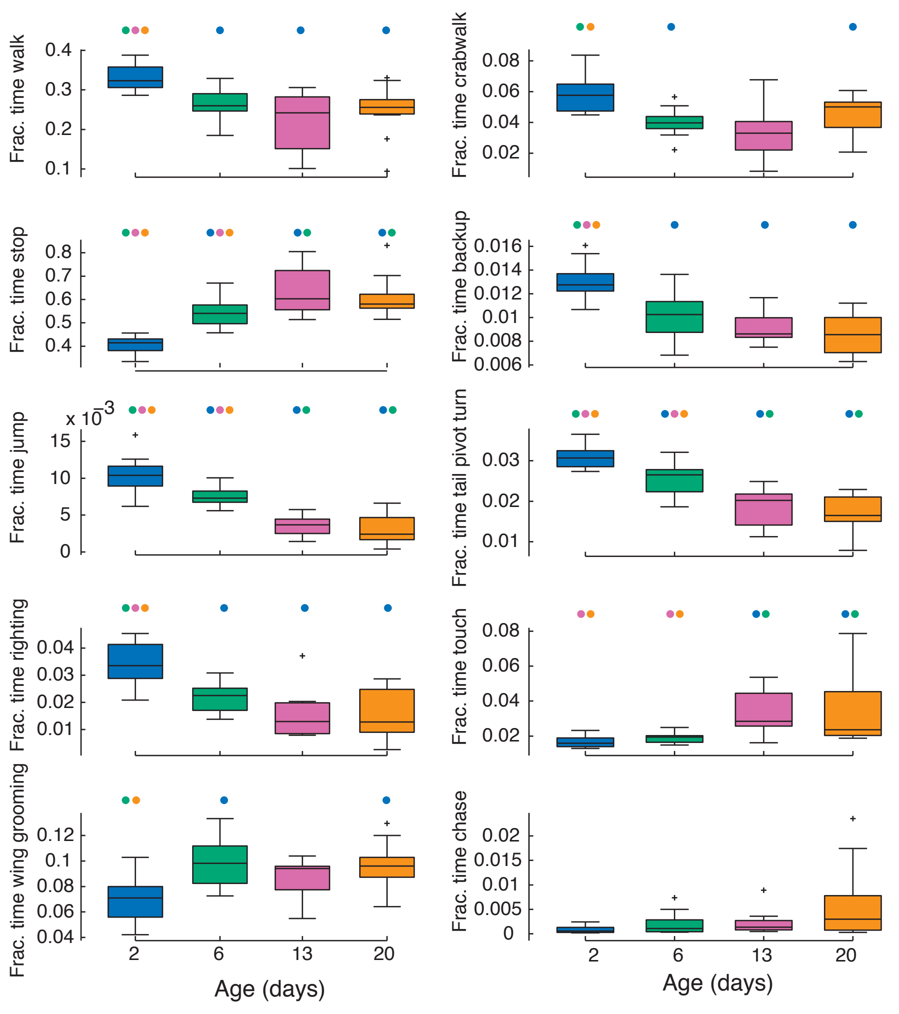
c



(**a**) Box plots show the distribution of the fraction of time spent performing each behavior: walk, stop, jump, righting, wing grooming, crabwalks, backup, tail-pivot turn, touch and chase for each starvation duration group: 0 hours starved (dark blue), 6 hours starved (green), 12 hours starved (pink), 24 hours starved (light orange). The data was calculated and presented as in Supplementary Figure 4. For 0 hours starved number of trials is n = 14, for 6 hours starved n = 14, for 12 hours starved n = 14 and for 24 hours starved n = 12. (**b**) The actual p values for each comparison with statisitcally signifcant values at the adjusted α level, 0.008, in bold. (**c**) The pair-wise error in predicting the starvation duration (x-axis and y-axis), based on logistic regression of their behavioral statistics, calculated and presented as in **Supplementary Figure 4**.

Supplementary Figure 6. Behavioral effects of age in adult flies.

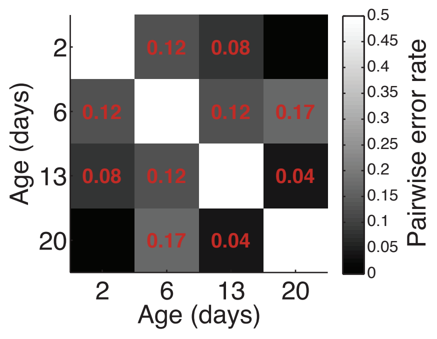
a



b

		6 days	13 days	20 days
Walk	2 days	**0.00048**	**0.00010**	**0.00059**
	6 days		0.15721	0.54437
	13 days			0.47049
Stop	2 days	**0.00004**	**0.00004**	**0.00004**
	6 days		0.02258	0.03509
	13 days			0.75083
Jump	2 days	**0.00295**	**0.00004**	**0.00005**
	6 days		**0.00006**	**0.00008**
	13 days			0.34078
Righting	2 days	**0.00165**	**0.00025**	**0.00031**
	6 days		0.01202	0.04040
	13 days			0.97697
Wing grooming	2 days	**0.00165**	0.01414	**0.00244**
	6 days		0.23658	0.83986
	13 days			0.26024
Crabwalk	2 days	**0.00038**	**0.00165**	0.03038
	6 days		0.12602	0.31232
	13 days			0.05310
Backup	2 days	**0.00511**	**0.00012**	**0.00008**
	6 days		0.19393	0.04639
	13 days			0.43573
Tail pivot turn	2 days	**0.00165**	**0.00004**	**0.00004**
	6 days		**0.00295**	**0.00073**
	13 days			0.54437
Touch	2 days	0.05310	**0.00048**	**0.00038**
	6 days		**0.00244**	**0.00427**
	13 days			0.66501
Chase	2 days	0.12602	0.02258	0.01019
	6 days		0.43573	0.17485
	13 days			0.31232

c



(**a**) Box plots show the distribution of the fraction of time spent performing each behavior: walk, stop, jump, righting, wing grooming, crabwalks, backup, tail-pivot turn, touch and chase for each age group: 2 days post ecolsion (dpe) (dark blue), 6 dpe (green), 13 dpe (pink), 20 dpe (light orange). The data was calculated and presented as in **Supplementary Figure 4**. For age group 2 dpe the number of trials was n = 12, for 6 dpe n = 12, for 13 dpe n = 12 and for 20 dpe n = 12. (**b**) The actual p values for each comparison with statisitcally signifcant values at the adjusted α level, 0.008, in bold. (**c**) The pair-wise error in predicting the age of the flies (x-axis and y-axis), based on logistic regression of their behavioral statistics, calculated and presented as in **Supplementary Figure 4**.

Supplementary Table 1. Training data sets

Animal	Behavior	Frames labeled			Videos used		
		Total	Positive	Negative	Wildtype strain	Genetic control	GAL4
Fly	Walk	4165	2517	1648	0	3	2
Fly	Stop	4095	1312	2783	0	1	2
Fly	Jump	10340	756	9584	1	2	5
Fly	Righting	25731	2350	23381	0	2	4
Fly	Wing grooming	11407	2964	8443	2	3	0
Fly	Crabwalk	2295	729	1566	0	3	1
Fly	Backup	3268	199	3069	0	3	0
Fly	Tail-pivot turn	6611	1154	5457	0	2	2
Fly	Touch	5582	864	4718	3	3	1
Fly	Chase	8354	2527	5827	0	2	3
Fly	Center-pivot turn	2315	361	1954	0	1	7
Fly	Wing flick	2514	247	2267	0	1	5
Fly	Wing extension	1387	422	965	0	1	2
Fly	Attempted copulation	7559	2716	4843	0	2	6
Fly	Copulation	4352	1500	2852	0	0	4
Larvae	Crawling	5735	2554	3181	3	0	0
Larvae	Head cast	3277	1063	2214	3	0	0
Mice	Following	4838	1786	3052	1	0	N/A
Mice	Walking	3962	2681	1281	1	0	N/A

Supplementary Table 2. Ground-truth error rates and labeled frames for multispecies behavior detectors

		Error rates						Frames labeled				
Animal	Behavior	average important error rate	total error rate	false negative rate	false positve rate	false negative rate	false positive rate	positive labels imortant	negative labels imortant	total positive labels	total negative labels	Total
Fly	Backups	0.057	0.108	0.111	0.002	0.213	0.004	135	5331	268	5538	5806
Fly	Crabwalks	0.025	0.069	0.037	0.012	0.120	0.019	454	3260	749	3714	4463
Fly	Touches	0.025	0.050	0.050	0.000	0.098	0.001	201	4655	326	5025	5351
Fly	Wing Grooming	0.029	0.034	0.027	0.031	0.029	0.038	1169	4218	1188	4362	5550
Fly	Chase	0.006	0.040	0.012	0.001	0.072	0.009	857	4506	1221	4543	5764
Fly	PivotTail	0.022	0.054	0.039	0.005	0.083	0.024	51	5576	204	5689	5893
Fly	Stops	0.003	0.013	0.006	0.001	0.012	0.014	2345	2568	2455	3128	5583
Fly	Walks	0.018	0.071	0.036	0.004	0.103	0.038	1327	3754	2014	3901	5915
Fly	Jumps	0.000	0.018	0.000	0.001	0.034	0.002	45	5584	89	5661	5750
Fly	Righting	0.008	0.072	0.014	0.002	0.138	0.006	293	5224	392	5250	5642
Fly	Copulation	0.022	0.025	0.044	0.000	0.050	0.000	2984	2832	3032	2832	5864
Fly	Attempted Copulation	0.012	0.051	0.024	0.000	0.098	0.004	367	4813	469	4830	5299
Fly	Wing Flick	0.043	0.122	0.085	0.000	0.237	0.007	141	5653	266	5688	5954
Fly	Wing Extension	0.011	0.024	0.015	0.008	0.029	0.019	1708	3490	2000	3530	5530
Fly	PivotCenter	0.009	0.085	0.017	0.001	0.161	0.009	293	5267	602	5306	5908
Larvae	Crawling	0.059	0.100	0.065	0.053	0.110	0.089	3456	7558	4137	7898	12035
Larvae	Head cast	0.045	0.132	0.084	0.006	0.246	0.017	693	10893	986	11026	12012
Mice	Following	0.037	0.037	0.045	0.029	0.045	0.029	2235	17780	2235	17780	20015
Mice	Walking	0.050	0.050	0.042	0.058	0.042	0.058	5150	7073	5150	7073	12223

Supplementary Table 3. Ground-truth error rates and labeled frames for detector performance on GAL4 lines

		Error rates						Frames labeled				
Video	Behavior	average imortant error rate	total error rate	important false negative rate	important false positive rate	total false negative rate	total false positive rate	positive labels important	negative labels important	total positive labels	total negative labels	total
1	chase	0.000	0.104	0.000	0.000	0.204	0.003	61.000	1461.000	240.000	1744.000	1984.000
1	jump	0.001	0.001	N/A	0.001	N/A	0.001	0.000	1879.000	0.000	1894.000	1894.000
1	righting	0.005	0.007	0.000	0.010	0.000	0.014	7.000	1874.000	10.000	1976.000	1986.000
2	chase	0.030	0.131	0.061	0.000	0.252	0.009	66.000	1459.000	206.000	1760.000	1966.000
2	jump	0.000	0.251	N/A	0.000	0.500	0.003	0.000	1536.000	2.000	1736.000	1738.000
2	righting	0.026	0.127	0.051	0.002	0.238	0.017	177.000	1185.000	598.000	1285.000	1883.000
3	chase	0.000	0.061	0.000	0.000	0.121	0.001	228.000	1327.000	364.000	1609.000	1973.000
3	jump	0.000	0.002	0.000	0.001	0.000	0.003	8.000	1612.000	18.000	1668.000	1686.000
3	righting	0.005	0.030	0.008	0.001	0.040	0.019	126.000	1671.000	149.000	1816.000	1965.000
4	chase	0.000	0.380	N/A	0.000	0.755	0.005	0.000	1554.000	53.000	1925.000	1978.000
4	jump	0.001	0.072	0.000	0.001	0.143	0.001	4.000	1844.000	14.000	1910.000	1924.000
4	righting	0.001	0.010	0.000	0.002	0.000	0.019	67.000	1861.000	72.000	1920.000	1992.000
5	chase	0.007	0.048	0.014	0.000	0.096	0.001	147.000	1629.000	240.000	1749.000	1989.000
5	jump	0.001	0.003	0.000	0.002	0.000	0.005	11.000	1777.000	16.000	1983.000	1999.000
5	righting	0.004	0.103	0.005	0.002	0.199	0.008	200.000	1328.000	292.000	1577.000	1869.000
6	chase	0.000	0.126	0.000	0.000	0.238	0.015	8.000	1744.000	63.000	1917.000	1980.000
6	jump	0.001	0.038	0.000	0.001	0.074	0.002	15.000	1529.000	27.000	1573.000	1600.000
6	righting	0.000	0.001	0.000	0.000	0.000	0.003	69.000	1819.000	70.000	1930.000	2000.000
7	chase	0.000	0.069	N/A	0.000	0.125	0.014	0.000	1613.000	8.000	1990.000	1998.000
7	jump	0.000	0.046	0.000	0.000	0.091	0.001	11.000	1885.000	22.000	1972.000	1994.000
7	righting	0.000	0.024	0.000	0.000	0.041	0.008	127.000	1418.000	148.000	1848.000	1996.000
8	chase	0.000	0.094	0.000	0.000	0.159	0.030	276.000	839.000	573.000	1348.000	1921.000
8	jump	0.001	0.009	0.000	0.001	0.015	0.004	30.000	1890.000	69.000	1920.000	1989.000
8	righting	0.002	0.046	0.000	0.003	0.065	0.028	110.000	986.000	155.000	1234.000	1389.000
9	chase	0.000	0.091	0.000	0.000	0.179	0.004	166.000	1410.000	246.000	1683.000	1929.000
9	jump	0.000	0.011	0.000	0.000	0.020	0.002	23.000	1847.000	49.000	1917.000	1966.000
9	righting	0.001	0.015	0.000	0.001	0.024	0.006	146.000	1501.000	165.000	1785.000	1950.000
10	chase	0.000	0.022	0.000	0.000	0.028	0.017	305.000	1394.000	399.000	1577.000	1976.000
10	jump	0.001	0.015	0.000	0.002	0.023	0.006	46.000	1844.000	86.000	1896.000	1982.000
10	righting	0.023	0.143	0.045	0.001	0.277	0.009	67.000	1735.000	101.000	1864.000	1965.000
11	chase	0.000	0.029	0.000	0.000	0.057	0.001	313.000	1393.000	406.000	1575.000	1981.000
12	chase	0.008	0.064	0.015	0.001	0.119	0.009	331.000	1288.000	471.000	1501.000	1972.000
13	jump	0.020	0.058	0.039	0.000	0.114	0.003	77.000	1730.000	141.000	1843.000	1984.000
14	jump	0.025	0.080	0.047	0.002	0.152	0.009	150.000	1332.000	330.000	1642.000	1972.000
15	righting	0.013	0.118	0.025	0.002	0.228	0.009	162.000	1471.000	285.000	1643.000	1928.000
16	righting	0.035	0.081	0.071	0.000	0.161	0.001	467.000	1119.000	700.000	1291.000	1991.000

Supplementary Note

1 Animals

1.1 Flies

Drosophila melanogaster were reared in standard vials on dextrose-based medium. Flies for the *TrpA* activation were reared at 21.8°C and 60% relative humidity (RH) on a 16/8 light/dark cycle; these flies were starved for 24 hours on agar medium prior to experiments. Wildtype strains were reared at 25°C and 65% RH on a 12/12 light/dark cycle; these flies were starved for 4 hours prior to experiments. The flies for the age and starvation experiments were reared at 22°C and 55% RH on a 16/8 light/dark cycle. For the age experiments, flies were transferred onto fresh food vials each week until use and were not starved before the experiment. Flies for the starvation duration experiments were transferred onto agar starvation media 24, 12, or 6 hours before the experiments while unstarved flies were used directly out of food vials. All flies were sorted into groups of 10 males and 10 females under cold anesthesia at least 24 hours prior to starvation. All experiments were performed in the evening of their circadian period; *TrpA* activation experiments at 29°C and 50% RH and wildtype, age, and starvation experiments at 25°C and 60% RH.

1.2 Larvae

The larvae were reared at a density of 200 eggs per food plate (60 mm dish filled with standard fly medium) at 25°C in constant light. Each experiment consisted of ten mid-third instar larvae placed on a test plate (60 mm petri dish) and recorded for 50 minutes at 30 frames per second using a digital camera (Sony DCR-HC52). The test plates contained 10% sucrose, 20% yeast, 1% agar medium and all the experiments were conducted in a climate-controlled room at 25C and 50-60% RH.

1.3 Mice

Male and female C57Bl/6J mice (Jackson Labs), aged 6-16 weeks, were used. Prior to recording, the two female mice (sisters) and the two male mice (brothers) were housed in separate cages and maintained on a 12/12 light/dark cycle. The estrus state of the females was monitored and when both were determined to be in proestrus the experiments were started. During recording all four mice were housed in a .6 × .6 × .6 m polycarbonate population cage and were filmed at 30 fps for five days. Bedding was a 25%/75% mix of corn cob and alpha-dri (Shepherd). Shelters for the mice were custom-made square tunnels made of IR-transparent acrylic (cylindrical tunnels distorted the image of the mice within the tunnel, degrading tracking performance). Mice always had ad libitum access to food and water.

2 Fly wing tracking

For two of the fly behaviors (double wing flick and wing extension), after tracking the bodies of the flies using Ctrax, we performed a second step of tracking to fit the positions of the wings using custom MATLAB (Mathworks) software. Our wing tracking software assigns both a fly identity and a label of body, wing, or other to each foreground pixel based on its proximity to the trajectories output by Ctrax and the background subtracted image intensity. In our video, wing pixels are usually lighter than body pixels and darker than leg pixels. The tracker fits either one or two wings to the blob of pixels identified as wing for a given fly by histogramming the angle to the pixel in the fly's coordinate system, and searching for sufficiently large peaks.

3 Larva tracking

The tracking software worked by segmenting pixels into foreground and background pixels using background subtraction. Connected components of foreground pixels were detected, and smoothed outlines and 11-point skeletons were fit to each connected component. The head-tail ambiguity was resolved using the motion of the larva during its entire trajectory. Using heuristics based on size and motion, connected components that corresponded to multiple, touching larvae were identified and ignored. Identities were assigned between consecutive frames based on overlapping area.

4 Behavior descriptions

The walk, crabwalk, chase, backup, jump, stop, touch, copulation, and wing extension behaviors were previously defined [1, 2]. The two types of pivot turns are similar to the sharp turns previously described as "large and rapid change in orientation" [1]. The 'tail pivot' is restricted to those sharp turns where the center of rotation was near the tip of the abdomen and there was minimal forward movement. Conversely, the 'center pivot' is those sharp turns where the center of rotation was close to the centroid of the fly and there was little to no translational movement.

While the flies are stopped they groom their wings as well as other body parts. 'Wing grooming' occurred when "one or both of the metathoracic legs are rubbed over the top or the underside of the wing(s)" as previously described [3]. During 'wing grooming' their wings were often moved away from their centerline and deflected downward by their legs.

We observed that flies when supine sometimes performing righting maneuvers consisting of high velocity rolling to and fro before managing to turn over, we termed this behavior 'righting' after the righting reflex (reflexes to return the body to a normal orientation) common to most animals. In normal flies this behavior mainly occurs after a jump in the enclosed space of our chamber when the flies end up upside-down or when they fall over attempting to climb onto the ceiling (made slick by an application of Sigmacote to keep flies off).

The behavioral elements of *Drosophila* courtship have been extensively studied. Here, in addition to improving the state-of-the-art for automatic detectors of chase, copulation and wing extension, we have also built the first automatic detectors for additional elements of the courtship: attempted copulation and double wing flick. These behaviors are similar to 'mounting' and 'wing scissoring' [4]. However, 'double wing flick' as defined here does not require repeated flicks and can be performed by animals of either sex.

The two larva behaviors were defined previously [5, 6], as were the two mouse behaviors [7].

5 Training behavior classifiers

To train the classifiers with JAABA, we began by labeling a small number of bouts in which we were certain the animal was performing the behavior and a couple of nearby bouts in which we were certain the animal was not performing the behavior. We specifically avoided labeling frames near the starts and ends of bouts when it was difficult to determine the exact frame on which the behavior transition occurred. We trained the initial classifier after just these few bouts. We then scanned this classifier's predictions, and labeled frames in which the classifier's predictions were incorrect (**Supplementary Video 1**). We prioritized labeling egregious errors to ensure that the classifier would not make this type of error. We frequently retrained the classifier, after labeling just a handful of new bouts. As the classifier began to make fewer mistakes, we also added labels for 'easy', canonical examples on which the classifier's confidence was low. When looking at the classifier's predictions, we frequently switched which animal we were labeling and (for flies and larvae) which video we were labeling in, to get a better understanding of the types of generalization errors the classifier would make. After the classifier was correctly classifying all the 'easy' frames, we added labels on the more difficult frames that we were still certain of the desired label, again, retraining often. We examined the cross-validation error on occasion to quantify our progress.

For the majority of the adult fly behaviors, we began by labeling behaviors in the *pBDPGAL4U* control flies. We also trained on *GAL4* lines observed to have different phenotypic properties, e.g. lines that had increased social phenotypes or lines that jumped more. The center-pivot turn, wing extension, copulation, attempted copulation, and double wing flick behaviors did not occur frequently in our control flies, thus we trained these using mainly hand-picked *GAL4* lines. For the chase, righting, jump, stop, walk, tail-pivot turn, after initial training on control data, we ran the resulting classifiers on a large set of the *GAL4* lines, then examined the detections on lines predicted to perform the given behavior the most. This helped us find failure modes of the initial classifier on frames dissimilar to all frames in the initial training data. For example, we found a failure mode in our initial chase classifier for a fly that was standing behind another fly but not moving for extended periods of time. The JAABA software allowed for quick incorporation of this new training data and generation of an updated classifier. We similarly added new training data after examining the classifiers' performance on the different wildtype strains. This was necessary in particular for wing grooming, as the initial classifier was dependent on the size of the flies, and some of the wildtype strains were larger.

For larvae, we trained the classifiers on two videos from each of the three drosophilid species. For mice, we trained the classifiers on one 30-minute video. **Supplementary Table 1** describes the training data set sizes in detail.

6 Boosting Algorithm

GentleBoost (Algorithm 6.1), Adaboost [8, 9] and other boosting learning algorithms learn an accurate classifier by combining many weak rules. The input to the boosting algorithm is a training set $(x_1, y_1), \ldots, (x_n, y_n)$ where each x_i belongs to a *feature space* X, and each *label* y_i is in a label set Y. For our system $Y = \{-1, +1\}$ as our detector classifies each frame as an animal performing a behavior or not. The boosting algorithm adds the weak rules repeatedly in a series of rounds $t = 1, \ldots, T$. Boosting combines these received weak rules into an accurate predictor by keeping a distribution or a set of weights over the training set. The weight of this distribution on training example i on round t is denoted by $D_t(i)$. Initially, all the instances are given equal weight. But with each round the weights of incorrectly classified examples are increased so that the weak learner is forced to concentrate on the hard examples.

Boosting is able to combine the weak rules into an accurate predictor if at each step the weak rule is better than random prediction. In other words, the weak rule $h_t : X \rightarrow \{-1, +1\}$ received by the boosting algorithm should have an error ϵ_t with respect to D_t less than 0.5.

$$\epsilon_t = Pr_{i \sim D_t}[h_t(x_i) \neq y_i] = \sum_{i : h_t(x_i) \neq y_i} D_t(i) < 0.5$$

Once the weak rule h_t has been received, boosting selects a parameter α_t that assigns

```
Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$
For $t = 1, \ldots, T$:

  • Find weak $h_t : X \to \{-1, +1\}$ with least error

        $\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$

  • Choose $\alpha_t = 1 - 2\epsilon_t$

  • Update:

        $\text{margin}_i = y_i \times \sum_{t'=1}^{t} \alpha_{t'} h_{t'}(x_i)$
        $D_{t+1}(i) = \frac{1}{1 + \exp(\text{margin}_i)} \times \frac{1}{Z_t}$

    where $Z_t$ is a normalization factor that makes $D_{t+1}$ sum to 1.

Output the final hypothesis:

    $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$
```

Algorithm 6.1: Gentleboost algorithm.

weight to the rule based on the "advantage" $(1 - 2\epsilon_t)$ of that weak rule . The distribution D_t is next updated using the rule shown in Figure 6.1. As a result of this update, weights of instances misclassified by h_t are increased and weights of correctly classified instances are decreased. Thus, the weight tends to concentrate on "hard" examples.

The *final classifier* H is a weighted majority vote of the T weak rules where α_t is the weight assigned to h_t. The final rule puts more trust into weak rules with lower error as the weight of a weak rule α_t is more if the error ϵ_t is small.

For any example, the raw score $\sum_{t'} \alpha_{t'} h_{t'}(x_i)$ is the signed version of example's distance to the decision boundary. To get a binary prediction the score is thresholded at zero.

7 Improving the learning algorithm's speed

At each iteration of GentleBoost, we have to find a decision stump that has the least error on the wighted training examples and this is the slowest step of the algorithm. We modified the way we searched for the best decision stump to make the algorithm faster. First, instead of searching over all possible threshold points for all features to find the best decision stump, we sample 30 candidate threshold points at roughly every 3^{rd} percentile of the feature values in the training data. We then picked the decision stump with the largest advantage over the features and their candidate threshold points. Second, we randomly selected one third of the features every 10 iterations of boosting and found the best weak rule over only these selected features. Third, if the user added relatively few training examples to the previous training set, instead of training a completely new classifier, we updated the previous classifier. The updated classifier uses the decision thresholds of the previous classifier but updates their weights. After updating the weights of the previous classifier, we added 20 more decision stumps by running 20 more iterations of boosting.

8 Visualizing the classifier's confidence

To show the classifier's confidence in its predictions, we show the raw scores computed by the classifier found. These scores are thresholded at 0 to obtain the final predictions of whether the animal is or is not performing the behavior. The classifiers scores are scaled between -1 and 1 by dividing scores by the 80th percentile of the absolute values of the raw scores on the training data and thresholding at -1 and 1.

9 Cross-Validation

Users can obtain feedback on how well their current behavior classifiers will generalize to novel data by examining the cross-validation error through the JAABA interface. In cross validation, we randomly split the training data into 7 subsets. We train a classifier on 6/7 of these subsets of training data, then examine the error rate on the remaining subset of training data. We do this for each subset of the training data, resulting in an estimate of the generalization error. When grouping together frames into subsets, we keep frames in the same behavioral bout in the same subset. This is because consecutive frames of the same bout are highly dependent. Placing these related frames in both the training and test set, would result in an underestimate of the generalization error. We also split the positive and negative bouts independently to keep the proportion of positive and negative bouts consistent.

10 Ground truthing classifier performance

In many of the fly behaviors we examined, frames in which the animal is performing the behavior constitute a small fraction of the video. If we select frames to ground truth in an unbiased manner, accurately estimating false negative rates would require a huge amount of labeled ground truth data. To remedy this, for adult flies, we chose examples to label randomly, but increased the weight of intervals containing predicted positive frames. The weight of a given interval is the sum of the weights of its constituent frames, and the weights of individual frames are set so that the total weight of predicted positive and predicted negative examples is the same.

For the ground-truth results on *GAL4* lines shown in **Figure 3** we labeled 2,000 frames in each video. The lines used in videos 1-5 were selected randomly from the 5th, 25th, 50th, 75th and 95th percentiles of the mean velocity range of the 14,524 videos of the neural activation screen. Videos 6-10 were similarly selected from the range of mean distance to closest fly values. Videos 11-16 were a sample videos for lines selected from the top two percent of the 2,144 lines based on the line-level average for mean fraction of time spent performing the given behaviors: chasing (11, 12), jumping (13, 14) and righting (15, 16). Details of the ground-truth data sets are given in **Supplementary Table 3**.

11 Logistic Regression

To evaluate the discriminative capability of the combined behavior measurements to differentiate the comparable wildtype strains and age and starvation conditions, we trained logistic regression classifiers. We used logistic regression with lasso regularization to construct pairwise classifiers [10]. The optimal regularization coefficient λ was found using leave-one-out cross validation. The reported total accuracy is $1-$ averaged error across all pairwise comparisons.

12 Window features

For a given per-frame feature, we compute window features to describe the data around each frame.

12.1 Window feature functions

The list below details the window feature functions that were used to compute the window features. For each window feature we used windows of size 3, 11 and 21 frames for flies and mice and 3, 31 and 61 frames for larvae. For each window size, the window was placed so that the frame was at the beginning, center or the end of the frame.

- mean: Mean of the per-frame feature within the window. The smallest window had a size 1 instead of 3 so that the classifier could use the per-frames feature's current value as a feature.
- min: Minimum of the per-frame feature within the window.
- max: Maximum of the per-frame feature within the window.
- std: Standard deviation of the per-frame feature within the window.
- harmonic: The convolution of the data in the current window with a cosine function. For each window size, we used 3 cosine functions with time period of 0.5, 1 and 1.5 times the window size.
- change: The difference between the average value of the given per-frame feature in a small window at the end and the average value of a given per-frame feature in a small window at the start of the window. We used two small windows of size 3 and 7 frames.
- diff_neighbor_mean: The difference between the current frame's per-frame feature and the mean of the per-frame feature in the given window around the current frame.
- diff_neighbor_min: The difference between the current frame's per-frame feature and the minimum of the per-frame feature in the given window around the current frame.
- diff_neighbor_max: The difference between the current frame's per-frame feature and the maximum of the per-frame feature in the given window around the current frame.
- zscore_neighbors: The current frame's per-frame feature normalized by the mean and standard deviation of the per-frame feature in the given window around the current frame (subtract the mean then divide by the standard deviation).
- hist: This window feature requires as a parameter the edges of the histogram bins for the given per-frame feature. For each bin of the histogram (and each radius and offset) you will get a new window feature, which is the fraction of per-frame data within the given window which falls within the given bin. This window feature was only used to detect righting in flies. For each per-frame feature, the histogram bins were the 5, 15, 30, 50, 70, 85 and 95 percentile of the per-frame feature values found over all flies in the experiments that was used to train to the detector.

To train the righting detector for flies, we used a more extensive set of window features. For all window features, we used windows of size of 3, 7, 11, 17 and 21 frames. Along with the windows that were placed so that the current frame was at the beginning, center or end of the window we also the placed the windows at two new positions so that the current frame was at the center of the first and second half of the window.

12.2 Window feature transformations

For some per-frame features, we also transform the features by taking the absolute value, relative value and/or flipping all the values by the sign of the current per-frame value. Absolute-transformed features are used for per-frame features like change in orientation where the sign, indicating turning left or turning right, is often not discriminatory. Flip-transformed features are used in similar cases to absolute-transformed features, but allow the sign to be preserved within a window. Relative-transformed features are used to normalize features by per-animal statistics. For example, some populations of or individual animals will walk slower than the normal animal. In such cases, the speed at which the animal is perceived to walk or chase is much slower compared to normal animals. Converting feature values into relative values enables the classifier to detect behaviors for such cases, and generalize better to types of animals outside of the training set. We transform a feature value into a relative value by finding the fraction of feature values over the whole trajectory that are smaller than the current value. For example, if the feature value of the current frame is the median then its relative transformed value is 50. If 10% of the features values over the trajectory are smaller than the current value then the relative value is 10. In other words, the relative value of a feature value is its percentile value within its trajectory.

For the absolute and flip transformations, we do this by either taking the transformation of the per-frame data first, then computing the window function on the transformed data, or we do this by computing the window function first, then computing the transformation on the output of the window data. Whether we do the transformation first or not depends on the window feature. We do the transformation first for min, max, hist, diff_neighbor_min, diff_neighbor_max. We do the transformation after for mean, prctile, change, harmonic, diff_neighbor_mean, zscore_neighbors. We do not do the transformation for std. For the relative transformation, we always do the transformation first.

13 Fly and mouse per-frame features

The fly and mouse per-frame features are computed from an ellipse model that is fitted to the animal by the tracking software. The output of such a tracker is the x and y location of the centroid of the fitted ellipse, the orientation (theta) of the animal and the length of the major ($4 \times$ a_mm) and minor ($4 \times$ b_mm) axis length of the ellipse fitted to the animal. For both flies and mice, the “head” and “tail” refer to the two ends of the fitted ellipse where the front and the back are disambiguated by the orientation output of the tracker.

The per-frame features are computed to have real units (millimeters and seconds) instead of pixels and frame numbers so that they are easy to use across videos captured at different resolutions.

For each per-frame feature we also list the transformations that we used when we trained the classifier.

13.1 Appearance features

These set of features capture the appearance like size and eccentricity of the animal based on the ellipse fitted to the animal.

- a_mm: Quarter major axis length (mm). This feature was not used for mice. Transformation: none, abs.
- area: Area of the ellipse (mm^2). Transformations: none, relative.
- b_mm: Quarter minor axis length (mm). Transformations: none, abs.
- darea: Change in area from frame t to t+1 (mm/s). Transformations: none, abs.
- da: Change in quarter major axis length from frame t to t+1 (mm/s). Transformations: none, abs.
- db: Change in quarter minor axis length from frame t to t+1 (mm/s). Transformations: none, abs.
- decc : Change in the eccentricity of the ellipse from frame t to t+1 (1/s). Transformations: none, abs.
- ecc : Eccentricity of the ellipse (no units). Transformations: none, abs.

13.2 Locomotion features

These set of features capture the locomotive properties of the animal and describe aspects such as the speed, change in orientation, location of center of rotation.

- absdtheta: Angular speed (rad/s) |dtheta|. Transformations: relative.
- absdv_cor: Sideways speed of the animal's center of rotation (defined by corfrac_maj and corfrac_min) (mm/s). Transformations: relative.

- corfrac_maj: Projection of the center of rotation on the animal's major axis (no units). This is a measure of the point on the animal that translates least from one frame to the next. It is 0 at the center of the animal, 1 at the forward tip of the major axis, and -1 and the backward tip of the major axis. Transformations: none, abs.

- corfrac_min: Projection of the center of rotation on the animal's minor axis (no units). This is a measure of the point on the animal that translates least from one frame to the next. It is 0 at the center of the animal, 1 at the right tip of the minor axis, and -1 and the backward tip of the minor axis. Transformations: flip, abs.

- dphi: Change in the velocity direction (rad/s). Transformations: none, abs.

- dtheta: Angular velocity (rad/s). Transformations: flip, abs.

- du_cor: Sideways velocity of the animal's center of rotation (mm/s). This is the projection of the change in the position of the center of rotation on the animal onto the direction orthogonal to the animal's orientation. Transformations: none, abs and relative.

- du_ctr: Forward velocity of the animal's center (mm/s). This is the projection of the animal's velocity on its orientation direction. Transformations: none, abs and relative.

- du_tail: Forward velocity of the backmost point on the animal (mm/s). Transformations: none, abs and relative.

- dv_cor: Sideways velocity of the animal's center of rotation (mm/s). This is the projection of the change in the position of the center of rotation on the animal onto the direction orthogonal to the animal's orientation. Transformations: flip and abs.

- dv_ctr: Sideways velocity of the animal's center (mm/s). This is the projection of the animal's velocity on the direction orthogonal to the orientation. Transformations: flip and abs.

- dv_tail: Sideways velocity of the backmost point on the animal (mm/s). Transformations: flip and abs.

- flipdv_cor: Sideways velocity of the animal's center of rotation, sign-normalized so that if the animal's orientation is turning right, then flipdv_cor is positive if the animal's center of rotation is also translating to the right (dv_cor $\times$ signdtheta) (mm/s). Transformations: relative.

- phisideways: Difference between velocity direction and the animal's orientation. Transformations: none.

- velmag: Speed of the center of rotation (mm/s). Transformations: none and relative
- velmag_ctr: Speed of the fitted ellipse's center(mm/s). Transformations: none and relative.
- velmag_nose: Speed of the animal's nose (mm/s). Transformations: none and relative.
- velmag_tail: Speed of the animal's tail (mm/s). Transformations: none and relative.
- yaw: Difference between velocity direction and orientation (rad). Transformations: flip and abs.

13.3 Social features

These set of features capture the social properties of the animal like the distance to the closest animals, speed towards the closest animal.

- absanglefrom1to2_nose2ell: Absolute difference between direction to closest animal based on dnose2ell and current animal's orientation (rad). Transformations: none.
- absphidiff_anglesub: Absolute difference in velocity direction between current animal and closest animal based on anglesub (rad). Transformations: none.
- absphidiff_nose2ell: Absolute difference in velocity direction between current animal and closest animal based on dnose2ell (rad) . Transformations: none.
- absthetadiff_anglesub: Absolute difference in orientation between current animal and closest animal based on anglesub (rad). Transformations: none.
- absthetadiff_nose2ell: Absolute difference in orientation between this animal and closest animal based on dnose2ell (rad). Transformations: none.
- anglefrom1to2_anglesub: Angle to closest (based on angle subtended) animal's centroid in current animal's coordinate system. Metric that encodes the position of the closest animal relative to the current animal. Transformations: flip, abs.
- anglefrom1to2_nose2ell: Angle to closest (based on distance from nose to ellipse) animal's centroid in current animal's coordinate system. Metric that encodes the position of the closest animal relative to the current animal. Transformations: flip, abs.
- angleonclosestfly: Angle of the current animal's centroid in the closest (based on distance from nose to ellipse) animal's coordinate system. Metric that encodes the position of the closest animal relative to the current animal. Transformations: flip, abs.

- anglesub: Maximum total angle of animal's field of view (fov) occluded by another animal (rad). The parameter fov that can be set, and for our classifier's we set it to π radians. Transformations: none.
- danglesub: Change in maximum total angle of animal's view occluded by another animal (rad/s). Transformations: none, abs.
- dcenter: Minimum distance from this animal's center to other animal's center. Transformations: none, relative.
- ddcenter: Change in minimum distance between this animal's center and other flies' centers (mm/s). Transformations: none, abs.
- dell2nose: Minimum distance from any point of this animal's ellipse to the nose of other flies. Transformations: none, relative.
- dnose2tail: Minimum distance from any point of this animal's nose to the tail of other flies. Transformations: none, relative.
- dnose2ell: Minimum distance from any point of this animal's nose to the ellipse of other flies. Transformations: none, relative.
- dnose2ell_angle_30tomin30: Minimum distance from this animal's nose to the ellipse of other flies. The distance to flies that lie within the cone of $-30°$ to $30°$ are multplied by a factor greater than 1 dependent on the angle. This feature is used to find distance to flies that are close but not infront of the animal. Transformations: none.
- dnose2ell_angle_min20to20: Minimum distance from this animal's nose to the ellipse of other flies. The distance to flies that lie outside the $-20°$ to $20°$ cone in front of the animal are multplied by a factor greater than 1 depending on the angle. This feature is used to find distance to flies that are close and infront of the animal. Transformations: none.
- dnose2ell_angle_min30to30: Minimum distance from this animal's nose to the ellipse of other flies. The distance to flies that lie outside the $-30°$ to $30°$ cone in front of the animal are multplied by a factor greater than 1 depending on the angle. This feature is used to find distance to flies that are close and infront of the animal. Transformations: none.
- magveldiff_anglesub: Magnitude of difference in velocity of this animal and velocity of closest animal based on anglesub (mm/s). Transformations: none, relative.
- magveldiff_nose2ell: Magnitude of difference in velocity of this animal and velocity of closest animal based on dnose2ell (mm/s). Transformations: none, relative.

- nflies_close: Number of flies within 2 body lengths (4a). Transformations: none.
- veltoward_anglesub: Velocity of this animal in the direction towards the closest animal (closest animal being defined based on anglesub) (mm/s). Transformations: none, relative.
- veltoward_nose2ell: Velocity of this animal in the direction towards the closest animal (closest animal being defined based on dnose2ell) (mm/s). Transformations: none, relative.

13.4 Identity features

These set of features identify the nearest animal. The nearest animal is defined based on different criteria like the distance between the centers, distance between the nose of the current animal and any point on the ellipse of the other animal. These are mostly used as intermediate features to compute various social per-frame features.

- closestfly_anglesub: Identity of closest animal, based on anglesub, which is the total angle of animal's field of view (for) occluded by the other animal (rad). The parameter fov that can be set, and for our classifier's we set it to π radians.. Transformations: none.
- closestfly_center : Identity of closest animal, based on dcenter. Transformations: none.
- closestfly_ell2nose : Identity of closest animal, based on dell2nose. Transformations: none.
- closestfly_nose2ell : Identity of closest animal, based on dnose2ell. Transformations: none.

13.5 Arena features

These set of features capture the animal in relation to the arena. These features were not used for mice. For flies, the arena was circular and the radius and the center of the arena were specified as parameters.

- angle2wall: Angle to closest point on the arena wall from animal's center, relative to the animal's orientation (rad). Transformations: flip,abs.
- arena_r: Distance of animal's center from arena's center. Transformations: none.
- dangle2wall: Change in the angle to closest point on the arena wall to animal's center, relative to the animal's orientation (rad). Transformations: flip, abs.

- ddist2wall: Change in the distance to arena wall (mm/s). Transformations: none.
- dist2wall: Distance to the arena wall from the animal's center (mm). Transformations: none, relative.

13.6 Position features

These set of features capture the animals location in the coordinate system of the arena. These features were not used for mice.

- arena_angle: Animal's angular position in the arena measured as angle from x-axis. Transformations: none.
- arena_r: Distance of animal's center from arena's center. Transformations: none.
- dphi: Change in velocity direction (rad). Transformations: none, abs.
- phi: Velocity direction (rad). Transformations: none.

13.7 Score features

These features capture the scores output from different behavior detector. These features were not used for mice.

- scores_jump: Scores given by the jump classifier. Transformations: none.
- scores_pivot_tail: Scores given by the pivot tail classifier. Transformations: none.

13.8 Wing Appearance

These features are based on wing tracking and describe the appearance of the wings. In each frame, wing tracking detects both the wings, finds the angle of the wings and the area of the wings. Parts of the wings that overlap with the fly's body are not counted in the wing area. If a wing is not detected it's area and angle are set 0.

- angle_biggest_wing: Angle of the bigger wing. The bigger wing is decided based on the detected area of the wings. Transformations: none.
- angle_smallest_wing: Angle of the smaller wing. The smaller wing is decided based on the detected area of the wings. Transformations: none.
- area_inmost_wing: Area of the wing that is closer to the body. Transformations: none, relative.
- area_outmost_wing: Area of the wing that is further away from the body. Transformations: none, relative.

- mean_wing_area: Mean of the areas of the wings. Transformations: none, relative.
- mean_wing_angle: Mean of the angles of the wings. Transformations: none.
- wing_angle_diff: Angle between the right wing and the left wing. Transformations: none.
- wing_angle_imbalance: Difference of the right wing angle and the left wing angle. Transformations: none.
- nwingsdetected: Number of wings detected by the wing tracker. It can be either 0, 1 or 2. Transformations: none.
- dnwingsdetected: Change in nwingsdetected. Transformations: none, abs.
- min_wing_area: Minimum of the wing areas. Transformations: none, relative.
- max_wing_area: Maximum of the wing areas. Transformations: none, relative.
- min_wing_angle: Minimum of the wing angles. Transformations: none.
- max_wing_angle: Maximum of the wing angles. Transformations: none.

13.9 Wing Movement

These features are based on wing tracking and describe the movement of the wings.

- dangle_biggest_wing: Change in the angle of the bigger wing. The bigger wing is decided based on the detected area of the wings. Transformations: none, abs, flip.
- dangle_smallest_wing: Change in the angle of the smaller wing. The smaller wing is decided based on the detected area of the wings. Transformations: none, abs, flip.
- darea_inmost_wing: Change in the area of the wing that is closer to the body. Transformations: none, abs, relative.
- darea_outmost_wing: Change in the area of the wing that is more away from the body. Transformations: none, abs, relative.
- dmax_wing_area: Change in the area of the larger wing. Transformations: none, abs, relative.
- dmin_wing_area: Change in the are of the smaller wing. Transformations: none, abs, relative.
- dmax_wing_angle: Change in the angle of the larger wing angle. Transformations: none, abs.

- dmin_wing_angle: Change in the angle of the smaller the wing angle. Transformations: none, abs.
- dwing_angle_diff: change in wing_angle_diff. Transformations: none, abs.
- dwing_angle_imbalance: change in wing_angle_imbalance. Transformations: none, abs,.
- max_absdwing_area: Maximum of the largest absolute change in the wing areas. Transformations: none, relative.
- min_absdwing_area: Minimum of the largest absolute change in the wing areas. Transformations: none, relative.
- max_absdwing_angle: Maximum of the largest absolute change in the wing angles. Transformations: none.
- min_absdwing_angle: Minimum of the largest absolute change in the wing angles. Transformations: none.
- max_dwing_angle_in: Change in the angle of the wing that moves in the most. Transformations: none.
- min_dwing_angle_in: Change in the angle of the wing that moves in the least. Transformations: none.
- max_dwing_angle_out: Change in the angle of the wing that moves out the most. Transformations: none.
- min_dwing_angle_out: Change in the angle of the wing that moves out the least. Transformations: none.

14 Larvae per-frame features

All the orientations have been calculated assuming the origin of coordinates in the center of the arena. For each we fit a spline which we call animal's spine and the per-frame features are defined based on this spine. Head is defined as the first point in the spine, center as the midpoint in the spine and tail as the last point in the spine. The inflection point is a point on the spine that is furthest away from the segment that connects the head and the tail. 'Tailsm' and 'headsm' correspond to the points in the contour with the highest curvature near the tail and the head respectively. The difference between consecutive frames is computed for the perframe features marked with an asterisk (*). These new perframe features are defined as: d[name of the perframe feature], the units of these variables are [units of the per-frame feature]/s. Also, note that vx_cm and vy_cm, are equivalent to dx_cm and dy_cm.

14.1 Position

These set of features capture the animals location in the coordinate system of the arena.

- tailheadang*: Orientation of the vector from the tail to the head (rad). Transformations: none
- inflheadsmang: Orientation of the vector from the inflection point to the headsm (rad). Transformations: none
- tailsminflang: Orientation of the vector from the tailsm to the inflection point (rad). Transformations: none
- tailsmheadsmang: Orientation of the vector from the tailsm to the headsm (rad). Transformations: none
- tailsmcentralang: Orientation of the vector from the tailsm to the center (rad). Transformations: none
- centralheadsmang: Orientation of the vector from the center to the headsm (rad). Transformations: none
- ytailsm_mm*: y coordinate of the position of the tailsm (mm). Transformations: none
- xtailsm_mm*: x coordinate of the position of the tailsm (mm). Transformations: none
- yheadsm_mm*: y coordinate of the position of the headsm (mm). Transformations: none
- xheadsm_mm*: x coordinate of the position of the headsm (mm). Transformations: none
- inflheadang*: Orientation of the vector from the inflection point to the head (rad). Transformations: none
- tailinflang*: Orientation of the vector from the tail to the inflection point (rad). Transformations: none
- yinflection_mm*: y coordinate of the position of the inflection point (mm). Transformations: none
- xinflection_mm*: x coordinate of the position of the inflection point (mm). Transformations: none

- centralheadang*: Orientation of the vector from the center to the head (rad). Transformations: none
- tailcentralang*: Orientation of the vector from the tail to the center (rad). Transformations: none
- xtail_mm*: x coordinate of the position of the tail (mm). Transformations: none
- ycentral_mm*: y coordinate of the position of the center (mm). Transformations: none
- xcentral_mm*: x coordinate of the position of the center (mm). Transformations: none
- yhead_mm*: y coordinate of the position of the head (mm). Transformations: none
- xhead_mm*: x coordinate of the position of the head (mm). Transformations: none
- y_cm: y coordinate of the position center of mass (mm). Transformations: none
- x_cm: x coordinate of the position of center of mass (mm). Transformations: none

14.2 Appearance

These set of features capture the animals appearance.

- sumspinerelanglect*: Sum of the angles subtended by consecutive segments of the spine from tail to center (rad). Transformations: none,abs, relative
- ratioarealength*: Ratio between area and spinelength (mm). Transformations: none, relative
- theta*: Orientation of the major axis of an ellipse fitted to the image (rad). Transformations: none
- a_mm*: Quarter major axis of an ellipse fitted to the image (mm). Transformations: none, relative
- b_mm*: Quarter minor axis of an ellipse fitted to the image (mm). Transformations: none, relative
- ecc*: Eccentricity of an ellipse fitted to the image calculated as the ratio between a_mm and b_mm (no units). Transformations: none, relative
- spinelength*: Total length of the spine (mm). Transformations: none, relative
- area*: Total area of the larvae (mm^2). Transformations: none, relative

- angihsmvsangtsmi: Angle subtended between vectors defined for inflheadsmang and tailsminflang (rad). Transformations: abs, flip
- angihvsangti*: Angle subtended between vectors defined inflheadang and tailinflang (rad). Transformations: abs, flip
- inflheadmag*: Distance from the inflection point to the head (mm). Transformations: none, relative
- tailinflmag*: Distance from the tail to the inflection point (mm). Transformations: none, relative
- sumspinerelanglehc*: Sum of the angle subtended by consecutive segments of the spine from center to head (rad). Transformations: none, abs and relative
- sumspinerelangle*: Sum of the angle subtended by consecutive segments of the spine from tail to head (rad). Transformations: none, abs and relative
- angchvsangtc*: Angle subtended between centralhedang and tailcentralang (rad). Transformations: abs, flip
- centralheadmag*: Distance from the center to the head (mm). Transformations: none, relative
- tailcentralmag*: Distance from the tail to the center (mm). Transformations: none, relative
- tailheadmag*: Distance from the tail to the head (mm). Transformations: none, relative

14.3 Locomotion

These set of features locomotive features of the animals.

- veltsmhsm: Speed of the center of mass in the tailsmheadsmang direction (mm/s). Transformations: none, relative
- veltsmc: Speed of the center of mass in the tailsmcentralang direction (mm/s). Transformations: none, relative
- velchsm: Speed of the center of mass in the centralheadsmang direction (mm/s). Transformations: none, relative
- veltailperth: Speed of the tail in the perpendicular direction to the tailheadang direction (mm/ s). Transformations: none, relative

- velcentralperth: Speed of the center in the perpendicular direction to the tailheadang direction (mm/s). Transformations: none, relative
- velheadperth: Speed of the head in the perpendicular direction to the tailheadang direction (mm/s). Transformations: none, relative
- veltailth: Speed of the tail in the tailheadang direction (mm/s). Transformations: none, relative
- velcentralth: Speed of the center in the tailheadang direction (mm/s). Transformations: none, relative
- velheadth: Speed of the head in the tailheadang direction (mm/s). Transformations: none, relative
- veltailpertc: Speed of the tail in the perpendicular direction to the tailcentralang direction (mm/s). Transformations: none, relative
- velcentralpertc: Speed of the center in the perpendicular direction to the tailcentralang direction (mm/s). Transformations: none, relative
- velheadpertc: Speed of the head in the perpendicular direction to the tailcentralang direction (mm/s). Transformations: none, relative
- veltailtc: Speed of the tail in the tailcentralang direction (mm/s). Transformations: none, relative
- velcentraltc: Speed of the center in the tailcentralang direction (mm/s). Transformations: none,relative
- velheadtc: Speed of the head in the tailcentralang direction (mm/s). Transformations: none, relative
- veltailperch: Speed of the tail in the perpendicular direction to the centralheadang direction (mm/s). Transformations: none, relative
- velcentralperch: Speed of the center in the perpendicular direction to the centralheadang direction (mm/s). Transformations: none, relative
- velheadperch: Speed of the head in the perpendicular direction to the centralheadang direction (mm/s). Transformations: none, relative
- veltailch: Speed of the tail in the centralheadang direction (mm/s). Transformations: none,relative
- velcentralch: Speed of the center in the centralheadang direction (mm/s). Transformations: none, relative

- velheadch: Speed of the head in the centralheadang direction (mm/s). Transformations: none, relative

- velmagtail: Speed of the tail (mm). Transformations: none, relative

- velmagcentral: Speed of the center (mm). Transformations: none, relative

- velmaghead: Speed of the head (mm). Transformations: none, relative

- velperth: Speed of the center of mass in the perpendicular direction to tailhedang direction(mm/s). Transformations: abs, flip

- velth: Speed of the center of mass in the tailhedang direction (mm/s). Transformations: none, relative

- velpertc: Speed of the center of mass in the perpendicular direction to tailcentralang direction(mm/s). Transformations: abs, flip

- veltc: Speed of the center of mass in the tailcentralang direction (mm/s). Transformations: none, relative

- velperch: Speed of the center of mass in the perpendicular direction to centralhedang direction(mm/s). Transformations: abs, flip

- velch: Speed of the center of mass in the centralhedang direction (mm/s). Transformations: none, relative

14.4 Locomotion_global

These set of features locomotive features of the animals in the global coordinate system of the arena.

- velangtail: Orientation of the velocity vector of the tail (rad). Transformations: none, abs, relative

- velangcentral: Orientation of the velocity vector of the center (rad). Transformations: none, abs, relative

- velanghead: Orientation of the velocity vector of the head (rad). Transformations: none, abs, relative

- velang: Orientation of the velocity vector of the center of mass (rad). Transformations: none

- vy_cm: Speed of the center of mass in the y direction (mm/s). Transformations: none

- vx_cm: Speed of the center of mass in the x direction (mm/s). Transformations: none

15 Features used to train detectors

The per-frame features used for different behaviors are listed in the following list.

- Fly Walk: Locomotion
- Fly Crabwalk: Locomotion
- Fly Chase: Locomotion, social
- Fly Backup: Locomotion, appearance
- Fly Jump: Locomotion, appearance
- Fly Stop: Locomotion
- Fly Wing grooming: Locomotion, appearance
- Fly Tail-pivot: Locomotion
- Fly Head-pivot: Locomotion, appearance, social and tail-pivot scores
- Fly Righting: Locomotion, appearance and jump scores
- Fly Touch: Locomotion, appearance and social
- Fly Copulation: Locomotion, appearance and social
- Fly Attempted Copulation: Locomotion, appearance and social
- Fly Wing Extension: Locomotion, appearance, social, wing appearance and wing movement
- Fly Wing Flick: Locomotion, appearance, social, wing appearance and wing movement
- Larvae Head Cast: Locomotion and appearance
- Larvae Crawl: Locomotion and appearance
- Mice Following: Locomotion, social and appearance
- Mice Walking: Locomotion and appearance

References

[1] Branson, K., Robie, A. A., Bender, J., Perona, P. & Dickinson, M. High-throughput ethomics in large groups of *Drosophila*. *Nature Methods* **6**, 451–7 (2009).

[2] Dankert, H., Wang, L., Hoopfer, E. D., Anderson, D. J. & Perona, P. Automated monitoring and analysis of social behavior in *Drosophila*. *Nature Methods* **6**, 297–303 (2009).

[3] Dawkins, R. & Dawkins, M. Hierarchical organization and postural facilitation – rules for grooming in flies. *Animal Behaviour* **24**, 739–55 (1976).

[4] Spieth, H. T. Courtship behavior in *Drosophila*. *Annual Review of Entomology* **19**, 385–405 (1974).

[5] Luo, L. *et al.* Navigational decision making in *Drosophila* thermotaxis. *Journal of Neuroscience* **30**, 4261–72 (2010).

[6] Gomez-Marin, A., Stephens, G. J. & Louis, M. Active sampling and decision making in *Drosophila* chemotaxis. *Nature Communications* **2**, 441 (2011).

[7] de Chaumont, F. *et al.* Computerized video analysis of social interactions in mice. *Nature Methods* **9**, 410–7 (2012).

[8] Freund, Y. & Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* **55**, 119–139 (1997).

[9] Friedman, J., Hastie, T. & Tibshirani, R. Additive logistic regression: a statistical view of boosting. *Annals of Statistics* **28**, 2000 (1998).

[10] Friedman, J., Hastie, T. & Tibshirani, R. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1 (2010).