

1 *SongExplorer*: A deep learning workflow for discovery and segmentation of animal acoustic  
2 communication signals

3

4 Benjamin J. Arthur<sup>1</sup>, Yun Ding<sup>1,2</sup>, Medhini Sosale<sup>1,3</sup>, Faduma Khalif<sup>1,4</sup>, Elizabeth Kim<sup>1</sup>, Peter Waddell<sup>5</sup>,  
5 Srinivas C. Turaga<sup>1</sup>, David L. Stern<sup>1</sup>

6

7 <sup>1</sup> Janelia Research Campus of the Howard Hughes Medical Institute, 19700 Helix Drive, Ashburn, VA  
8 20147, USA

9 <sup>2</sup> Present Address: Department of Biology, 102 Leidy Laboratories, 433 S. University Ave., University of  
10 Pennsylvania, Philadelphia, PA 19104-4544. USA

11 <sup>3</sup> School of Fundamental Sciences, Massey University, Palmerston North, New Zealand

12

13 **Abstract**

14 Many animals produce distinct sounds or substrate-borne vibrations, but these signals have proved  
15 challenging to segment with automated algorithms. We have developed *SongExplorer*, a web-browser  
16 based interface wrapped around a deep-learning algorithm that supports an interactive workflow for (1)  
17 discovery of animal sounds, (2) manual annotation, (3) supervised training of a deep convolutional  
18 neural network, and (4) automated segmentation of recordings. Raw data can be explored by  
19 simultaneously examining song events, both individually and in the context of the entire recording,  
20 watching synced video, and listening to song. We provide a simple way to visualize many song events  
21 from large datasets within an interactive low-dimensional visualization, which facilitates detection and  
22 correction of incorrectly labelled song events. The machine learning model we implemented displays  
23 higher accuracy than existing heuristic algorithms and similar accuracy as two expert human annotators.  
24 We show that *SongExplorer* allows rapid detection of all song types from new species and of novel song  
25 types in previously well-studied species.

26

27 Key Words: Neural network, classifier, unsupervised discovery, *Drosophila*, animal sounds, song

28

## 29 Introduction

30 Animals produce diverse sounds (Kershenbaum et al., 2016), vibrations (Hill, 2006), and periodic  
31 electrical signals (Zakon et al., 2008) for many purposes, including as components of courtship, to sense  
32 their surroundings, and to localize prey. Quantitative study of these "sounds" is facilitated by  
33 automated segmentation. However, heuristic segmentation algorithms sometimes have low accuracy  
34 and fail to generalize across species (Arthur et al., 2013; Chesmore and Ohya, 2004; Coffey et al., 2019;  
35 Ivanenko et al., 2018; Koumura and Okanoya, 2016; LaRue et al., 2015; Parsons, 2001; Sattar et al.,  
36 2016). Song segmentation is particularly challenging for low signal-to-noise sounds, such as those  
37 produced by many insect species.

38 Deep neural network classifiers have been developed to study animal sounds (Chesmore and  
39 Ohya, 2004; Coffey et al., 2019; Ivanenko et al., 2018; Koumura and Okanoya, 2016; Parsons, 2001;  
40 Sattar et al., 2016; Steinfath et al., n.d.) and typically exhibit higher accuracy than heuristic algorithms. In  
41 addition, nonlinear dimensionality reduction techniques such as t-SNE (van der Maaten and Hinton,  
42 2008) and UMAP (McInnes et al., 2018) have proven useful for discovering structure in animal sound  
43 datasets (Clemens et al., 2018). To facilitate adoption of deep networks and dimensionality reduction  
44 techniques for the analysis of animal sounds, we developed *SongExplorer* to support interactive work-  
45 flows for discovery, annotation, and segmentation of animal sounds. We illustrate the utility of  
46 *SongExplorer* with examples from *Drosophila* courtship song (Ewing et al., 1968; Greenspan and  
47 Ferveur, 2000) because these low signal-to-noise songs have traditionally been challenging to segment  
48 (LaRue et al., 2015) and different species produce multiple distinct song types.

49

## 50 Results

51

52 *SongExplorer* provides a versatile interface for detecting and segmenting animal sounds

53

54 Each of the steps from discovery to automated segmentation of animal song events has  
55 traditionally required extensive manual investigation of songs and quantitative analysis, usually  
56 requiring use of multiple software tools. To democratize all steps of song analysis, we built a web  
57 browser-based program called *SongExplorer* that allows exploration of data, annotation of songs, and  
58 several analysis methods, including training of a deep learning neural network classifier, and  
59 quantitative assessment of classifier performance (Figure 1). The web interface presents users with  
60 three major domains in a single view (Figure 1A): the left side presents the data in multiple views; the  
61 middle section provides analysis “wizard” buttons, file handling, and parameter value windows; and the  
62 right side is a scrollable box containing extensive documentation and a tutorial.

63 Quantitative study of animal sounds typically starts with supervised discovery of sounds. For  
64 species that produce loud and stereotyped sounds, like frogs and birds, it can be straightforward to  
65 identify individual types of songs. For other species, like many small insects, the initial step of identifying  
66 song types often requires examination of long recordings of audio and possibly video due to the sparse  
67 and quiet nature of their songs. Recent work has demonstrated that largely unsupervised clustering  
68 algorithms of song events can highlight distinct song types (Clemens et al., 2018). Therefore, to  
69 accelerate discovery, we provide methods for unsupervised clustering and visualization of song events.

70 The data view region on the left side of the browser window includes a box (Figure 1C) that  
71 displays data in a dimensionally reduced form, either as a UMAP or tSNE representation, in either two or  
72 three dimensions. Samples of sound wave forms or frequency spectra can be projected into these  
73 reduced dimensionality spaces (Clemens et al., 2018). However, we have found that the hidden layer

74 activations of a trained neural network provide more discrete representations of unique song types than  
75 do the original song events, which can facilitate identification of new song types (see later).

76 In *SongExplorer*, the reduced dimensionality space can be navigated rapidly with a modifiable  
77 “lens.” A sample of up to 50 sound events within the lens are represented in the adjacent window  
78 (Figure 1D). Raw sound traces are presented along with a spectrogram and a label indicating the song  
79 type. In the example shown in Figure 1, sounds were automatically detected (within *SongExplorer*) using  
80 thresholds either for high amplitude events (labelled “time”) or for events with a relatively strong signal  
81 in a subset of the spectrogram (labelled “frequency”). Clicking on each song event reveals the song  
82 event within a longer context of the recording in a separate window (Figure 1E). The context window  
83 (Figure 1E) can be navigated by zooming in or out and panning using buttons located below the context  
84 window. We have found that *Drosophila* song can sometimes be discriminated from other sounds most  
85 easily by listening to the song and watching associated video. Therefore, *SongExplorer* facilitates  
86 annotation by allowing the portion of the recording currently shown in Figure 1E to be played as audio  
87 and, if video data are available, for the video section for this region of the recording to be played.  
88 Individual song types can be named in boxes below the context window and labelled by using a  
89 computer mouse to double-click on events or to drag over ranges of continuous sounds. The number of  
90 annotated and automatically detected events are tracked below the context window.

91 The software is designed to encourage users to follow analysis pipelines that are enabled by  
92 “wizard” buttons (Figure 1B). For example, a user can explore a new dataset containing unlabeled data  
93 by selecting the “label sounds” wizard button, which enables five buttons that can be activated, from  
94 left to right, to perform the following steps: (1) automatically detect sounds above user-defined  
95 thresholds, (2) train a deep-learning classifier to recognize detected sound types, (3) calculate the  
96 classifier hidden-layer activations, (4) cluster these activation values, and (5) visualize the hidden layer  
97 activations using either UMAP or tSNE dimensionality reduction. Eight additional analysis pipelines are

98 provided as wizards, and individual analysis steps can be implemented independently of the wizard  
99 buttons. We include methods to correct false positive predictions by a trained classifier and to  
100 iteratively retrain the classifier with corrected values. This kind of iterative classifier training has proven  
101 powerful for training classifiers to recognize visual behaviors (Berg et al., 2019) and served as an  
102 inspiration for our approach. We provide video tutorials  
103 (<https://www.youtube.com/playlist?list=PLYXyXDkMwZip8x78RAyN6ee9NK42WBbKb>) to guide users  
104 through these analysis pipelines.

105  
106 *SongExplorer's* deep neural network displays higher accuracy than a heuristic classifier

107  
108 *SongExplorer* includes a deep neural network classifier that can be trained to automate the  
109 annotation of animal sounds. The deep network is configurable, and defaults to a simple 3-layer time-  
110 domain convolutional neural network that classifies each sample in a 5000 Hz acoustic waveform using a  
111 204.8 ms (1024 samples) window of context (Figure S1). We compared accuracy of this deep network  
112 versus a previously-described heuristic algorithm (Arthur et al., 2013) at classifying fly song. *D.*  
113 *melanogaster* males produce two distinctive types of courtship song, pulse song and sine song, and  
114 pulse song has traditionally been considered harder to classify than sine song. Kyriacou et al. (Kyriacou  
115 et al., 2017) performed a dense annotation of the pulse song events from 25 recordings of *D.*  
116 *melanogaster* courtship and a previously-described heuristic algorithm implemented on the same  
117 recordings, and with optimized parameters, displayed an F1 score (the harmonic mean of the precision  
118 and recall) of 87% (Stern et al., 2017).

119 To examine accuracy of the deep-learning neural network, we aligned pulse annotations to the  
120 nearest peak within five milliseconds and labelled all other points in time as “other” or “ambient”

121 depending on whether a time-domain threshold was exceeded or not, respectively. We withheld five of  
122 the 25 recordings for validation and used the remaining 20 recordings to train the classifier. The  
123 classifier returned the probability that a pulse is centered on every time point in the recording (Figure  
124 2A). To allow comparison with the heuristic algorithm, we implemented a discrete-valued ethogram by  
125 calculating a threshold based on the ratio of precision to recall; precision (also called the positive  
126 predictive value) is the fraction of true positives amongst all positives (true and false positives) and recall  
127 (sometimes called sensitivity) is the fraction of true positives detected amongst all real events (true  
128 positives plus false negatives). Using a precision to recall ratio of one (Figure S2), the trained network  
129 displayed an F1 of 94% (Figure 2B) for dense predictions made on the five withheld recordings, a  
130 considerable improvement over the heuristic algorithm. Within *SongExplorer*, users can select a lower or  
131 higher precision to recall ratio for thresholding to detect more of the relevant events (amongst more  
132 false positives) or mostly (but fewer) true positives, respectively. To explore the effect of sample size on  
133 classifier accuracy, we subsampled from the full dataset and found that 100 labelled pulse events  
134 produced accuracy very similar to the full dataset (Figure S3A). We also explored several parameters of  
135 the neural network and found that performance was largely insensitive to most hyperparameters  
136 (Figure S3B).

137

138 A trained classifier predicts pulse events approximately as well as an “average” human

139

140 To determine how well the trained neural-network classifier could generalize to predict pulse  
141 events in new recordings of *D. melanogaster* song, we made new recordings of *D. melanogaster*  
142 courtship song using chambers that were smaller than the chambers employed for the recordings  
143 described above. This is a challenging test case, because the noise characteristics differed systematically

144 between the two sets of recordings (Figure S4). Two human experts independently performed dense  
145 annotations of pulse events for 23 randomly-chosen one-minute segments of these recordings without  
146 prior discussion of how they would label events. Different humans often disagree on annotation of low  
147 signal-to-noise events and, indeed, Person 1 labelled more events than Person 2. Both annotators and  
148 the classifier agreed on most pulse events (Figure 2C). Overall, the classifier and the two humans  
149 displayed similar levels of unlabeled events, suggesting that the classifier, even when trained on  
150 different sets of recordings, performed approximately as well as the “average” human.

151

### 152 A common deep classifier architecture generalizes to many song types

153

154         Given the high accuracy of the classifier at detecting pulse song, we asked whether a classifier  
155 could be trained to accurately predict many song types across multiple species. We performed sparse  
156 annotations of all definable courtship song types from nine additional species, systematically labelled  
157 inter-pulse intervals between labelled pulse events, and trained a classifier to recognize each of the 37  
158 song types (Figure 3A). Since these samples were labelled sparsely, we cannot examine accuracy as we  
159 did above for dense annotations. Instead, we examined the likelihood that an event was labeled  
160 correctly, given that there was an annotated song event at a particular time, and present the results as a  
161 confusion matrix (Figure 3B). The classifier assigned most events with greater than 90% accuracy,  
162 suggesting that the neural network architecture that we employed can be used to classify song from  
163 many *Drosophila* species.

164         The classifier discriminated with high accuracy similar song types within a species and very  
165 similar song types from closely related species. For example, *D. persimilis* produces two different pulse  
166 types (Figure 3D-E) and the classifier accurately discriminated between these subtypes (Figure 3B).

167 Surprisingly, the classifier also accurately distinguished between the very similar pulse events of the  
168 sister species *D. simulans* and *D. mauritiana* recorded on the same set of microphones, which, in our  
169 experience, cannot be discriminated by humans (Figure 4F-K). The classifier employs a context window  
170 of 204.8 ms surrounding each event, and the classifier may therefore have used information about the  
171 diverged inter-pulse intervals between species to discriminate between these pulse types.

172 Some song types were not discriminated well, such as *D. erecta* sine 1 vs sine 2 and *D. persimilis*  
173 sine 1 vs sine 2. In both cases, the alternative labels were assigned during manual annotation prior to  
174 the availability of *SongExplorer*. *Post hoc* examination of songs within the *SongExplorer* interface  
175 revealed that sine song is rare in both species and there is no compelling evidence for multiple sine  
176 types, suggesting that the classifier correctly failed to discriminate between sine song subtypes in these  
177 species because the species do not produce multiple subtypes. Alternatively, it is possible that the  
178 classifier failed to discriminate multiple sine song types because these songs are rare. However, we  
179 found weak dependence of classifier accuracy and precision on song event sample size (Figure 3C) and  
180 sample sizes above 100 had similar accuracies.

181

#### 182 A trained classifier can provide rapid prediction of song events for closed-loop assays

183

184 Rapid prediction of song could be valuable for closed-loop experiments. We therefore examined  
185 whether our classifier could be implemented, in principle, in closed-loop scenarios. We examined  
186 classifier accuracy when the 204.8 ms context window (the position of the sampled window relative to  
187 the target song event) was shifted earlier in time (Figure 4B). We call the time between the predicted  
188 event and the end of the context window the latency. This latency corresponds to “information latency”  
189 and is a lower bound on the true latency, which will be higher and depend on the efficiency of the

190 software and hardware implementation, which we do not consider here. The classifier had precision of  
191 88.6% and recall of 76.6%, on average, in predicting *D. melanogaster* pulse events with a latency as low  
192 as 12.8 ms (Figure 4B-F), compared with precision and recall of 91.5% and 89%, respectively, with a  
193 latency of 102.4 ms. Accuracy was lower for isolated pulses and the first pulse in a pulse train (Figure  
194 4D-G), suggesting that the classifier may have used past pulses in a pulse train to improve the accuracy  
195 of detecting future pulses. Thus, in principle, this neural network classifier can identify many song events  
196 on the order of fast neuronal spiking in *Drosophila* (~200Hz).

197

198 The learned feature representations of the neural network exhibit latent structure about song types and  
199 allow efficient discovery of new songs

200

201         Investigators studying animal sounds typically spend considerable time listening to recordings  
202 and examining song traces and video to identify and categorize sounds. This work is not only time  
203 consuming and tedious, but it is also subjective and can frustrate identification of rare sounds. We  
204 therefore sought a method to rapidly identify both common and rare sounds. We found that the  
205 activities of the hidden-layer neurons of a trained network exhibit considerable structure about song  
206 types (Figure 5) and allow rapid identification of novel song types. We illustrate two modes of this  
207 discovery approach.

208         First, we trained a network on manually labelled male *Drosophila melanogaster* pulse and sine  
209 song. We also included automatically generated labels for inter-pulse interval and ambient noise  
210 samples in the training. We visualized the patterns of hidden layer neural activation in UMAP space  
211 (Figure 5A-D). These representations revealed that the input layer showed some structure, with pulse  
212 song mainly occupying two domains of the UMAP space and sine song a third domain (Figure 5B).

213 However, points were distributed diffusely and multiple song types were intermingled. The two domains  
214 of pulse song in the input layer reflect the two phases of pulse song resulting from different positions of  
215 the fly relative to the directional microphone (Figure 5B inset). Through successive layers of the  
216 network, however, we noticed that each of the song types coalesced into nearly distinct clouds  
217 corresponding to the trained label classes of pulse song, sine song, inter-pulse interval, and ambient  
218 noise (Figure 5C-E). The network apparently correctly learned that the positive and negative deflecting  
219 phases of pulse song were an artifact and not distinguishable features of pulse song.

220 To explore whether other types of songs might have been missed in *D. melanogaster* recordings,  
221 we selected a different set of recordings, which had not been manually annotated, and detected all  
222 sounds using time- and frequency-domain thresholds. These sounds were projected through the neural  
223 network and the neural activation values were embedded in a common UMAP space defined by  
224 annotated and detected song events (Figure 5F). This embedding revealed a new density of points that  
225 was obvious in the hidden layers (Figure 5F-H), but not in the input layer (Figure 5E). Manual  
226 examination of these events (a subset of which are highlighted in red in Figure 5G-J) revealed that they  
227 are examples of a recently described female copulation song (Kerwin et al., 2020). Thus, this approach  
228 to discovering animal sounds allows rapid discovery of known and previously unknown song types in  
229 large audio datasets.

230 To further explore the utility of this kind of approach to song discovery, we trained a classifier to  
231 distinguish samples from 21 species (Figure 6A), providing as training data only sound events detected  
232 by time- and frequency-domain thresholding. Although the classifier was trained only to recognize the  
233 species of origin for a song event, the hidden layer activations contained considerable latent structure  
234 that facilitated discovery of song types (Figure 6B-D). That is, without providing *any* manually labelled  
235 training data, *SongExplorer* revealed the multiple song types produced by each species (Figure 6E-F).  
236 This allowed discovery of new song types in species of the *Drosophila nasuta* species group. One

237 previous paper has reported song types from some of these species (Hongguang et al., 1997). Notably,  
238 we identified multiple song types in several species that were not identified in the earlier study (Figure  
239 S6). We estimate that using *SongExplorer* we discovered many or all song types for each species within  
240 approximately 20 minutes per species of exploring songs.

241

## 242 **DISCUSSION**

243

244 *SongExplorer* provides the first interactive graphical interface to allow exploration, discovery,  
245 and segmentation of animal sounds using deep learning tools. While we have characterized  
246 *SongExplorer* and a particular instantiation of a neural network model using *Drosophila* courtship song,  
247 many kinds of animal sounds can be accurately classified using deep learning models (Chesmore and  
248 Ohya, 2004; Coffey et al., 2019; Ivanenko et al., 2018; Koumura and Okanoya, 2016; Parsons, 2001;  
249 Sattar et al., 2016; Steinfath et al., n.d.). The neural network classifier accuracy is only weakly dependent  
250 on sample size and the classifier attained greater than 90% accuracy at detecting *D. melanogaster* pulse  
251 song with just 100 labelled events.

252 Many researchers will be interested in generating discrete ethograms from the probabilistic  
253 output generated by *SongExplorer*. There are many ways to discretize the *SongExplorer* output and for  
254 this study we employed a threshold derived from the precision-recall curve. It is also possible to employ  
255 heuristic thresholds, or combinations of heuristics to filter *SongExplorer* output for any particular  
256 purpose.

257 *SongExplorer* provides an intuitive interface for efficient discovery of new song types,  
258 segmentation of large song datasets, and reannotation of songs to correct false negative and false

259 positive labels. In contrast to the size of deep learning models used in computer vision, our deep  
260 network for acoustic signals is light-weight and runs acceptably on a CPU without the need for  
261 specialized GPU hardware. We therefore anticipate that *SongExplorer* may be valuable to a wide range  
262 of biologists studying animal sounds.

263

## 264 **Materials and Methods**

### 265 *SongExplorer* software program

266 The *SongExplorer* user-interface is web browser-based and is implemented in Python using the  
267 Bokeh library (<https://bokeh.org/>). It is cross-platform and has been tested to run on Mac OS X,  
268 Windows, and Linux computers. The deep learning components use Keras and Tensorflow and benefit  
269 modestly from access to a CUDA-compatible Nvidia GPU.

270

### 271 Software availability

272 We provide the source code as well as Singularity and Docker containers of *SongExplorer* for  
273 Linux, Microsoft Windows, and Apple Macintosh platforms  
274 (<https://github.com/JaneliaSciComp/SongExplorer>). Extensive tutorials on using *SongExplorer* are  
275 available on YouTube (  
276 <https://www.youtube.com/playlist?list=PLYXyXDkMwZip8x78RAyN6ee9NK42WBbKb>).

277

### 278 Annotating training data:

279 We obtained training data from two sources. First, Kyriacou et al. (Kyriacou et al., 2017)  
280 manually annotated 52,417 *D. melanogaster* pulse song events from songs that we had recorded  
281 previously (Stern, 2014). We downloaded their manual annotations from  
282 <https://doi.org/10.5061/dryad.80c1f>.

283 Second, before we had completed the *SongExplorer* interface, we employed *Tempo*  
284 (<https://github.com/JaneliaSciComp/tempo>) to manually annotate additional song events. *Tempo* allows  
285 simultaneous examination of synced audio and video recordings and manual annotation of song  
286 recordings with user-defined song types. We generated dense annotation  
287 of 6,770 pulse events of new *D. melanogaster* recordings for the congruence assays. We also  
288 generated sparse annotations of 589 sine song events and 172 copulation events from the same *D.*  
289 *melanogaster* recordings annotated by Kyriacou et al. We also generated 15,192 sparse pulse  
290 song annotations and 7,537 sparse sine song annotation from five additional strains of *D.*  
291 *melanogaster*. Finally, we annotated 74,065 song events from recordings of *D. erecta*, *D. mauritiana*,  
292 *D. persimilis*, *D. pseudoobscura*, *D. santomea*, *D. simulans*, *D. teissieri*, *D. willistoni* and *D. yakuba*.

293 Sine song was annotated as a range of time points and pulse events were marked by a single  
294 time point per pulse. Pulse annotations were automatically aligned to the largest peak in the full-wave  
295 rectified waveform within 5 ms of the manual annotation to reduce variability in annotator pulse  
296 placement. Time periods between pulse events that were within the same pulse train were  
297 automatically given the annotation of inter-pulse interval (IPI) if the interval was within a median  
298 absolute deviation of the species-specific IPI.

299  
300 Detecting sound events with time and frequency domain thresholds

301 Sound events were detected using time domain and frequency domain criteria as follows. Time  
302 points with absolute magnitude exceeding a time-domain threshold of 6 median absolute deviations  
303 above the median were selected, and gaps between selected time points shorter than 6.4 ms were  
304 morphologically closed. A second set of sound events were detected using a frequency-domain  
305 threshold of  $p < 0.1$  (the FFT window as 25.6 ms and multi-taper settings were  $NW=4$ ,  $K=8$ ). Intervals  
306 shorter than 25.6 ms were morphologically opened and gaps shorter than 25.6 ms were morphologically  
307 closed.

308

#### 309 Recordings of *Drosophila* song:

310 To explore the ability of the deep learning framework to classify diverse song  
311 events we recorded male courtship song from seventeen *Drosophila* species using a previously  
312 described *Drosophila* courtship song recording apparatus (Arthur et al., 2013). The following fly stocks  
313 were employed and samples from the *Drosophila* Species Stock Center  
314 (<http://blogs.cornell.edu/drosophila/>) and Ehime stock center (<https://kyotofly.kit.jp/cgi-bin/ehime/>)  
315 are indicated with the prefixes DSSC and EH: *D. albostrigata* (Kandy, SriLanka, 15112-  
316 1811.08); *D. bilimbata* (Guam Island, DSSC 15112-1821.08); *D. elegans* (DSSC 14027-  
317 0461.03); *D. erecta* (DSSC 14021-0224.01), *D. gunungcola* (gift of Jonathan Massey); *D. kepulauan*  
318 (Sarawak, Borneo Island, DSSC 15112-1761.01); *D. kohkoa* (Sarawak, Borneo Island, DSSC 1511-  
319 1771.01); *D. mauritiana* (DSSC 14021-0241.01); *D. melanogaster* (Canton-S); *D. nasuta* (Mombasa,  
320 Kenya, DSSC 151121781.06); *D. niveifrons* (Lae, PapuaNewGuinea, EH LAE-221); *D. persimilis* (DSSC  
321 14011-0111.50); *D. pseudoobscura* (DSSC 14022-0121.94); *D. pulau* (Sarawak, Borneo Island, 15112-  
322 1801.00); *D. santomea* (STO-CAGO 1482); *D. simulans* (*sim5*, gift of Peter  
323 Andolfatto); *D. sulfurigaster* (Kavieng, New Ireland, DSSC 15112-1831.01); *D. taxonf* (Sarawak, Borneo

324 Island, EH B-208); *D. teissieri* (DSSC 14021-0257.01); *D. willistoni* (DSSC 14030-0791.00);  
325 and *D. yakuba* (DSSC 14021-0261-01).

326  
327 Design and training of the deep neural network classifier

328         The *SongExplorer* interface enables users to customize the classifier architecture and  
329 hyperparameter values to their needs. The classifier tested in this paper is a deep 1D time-domain  
330 convolutional neural network implemented in Keras and Tensorflow (<https://www.tensorflow.org/>) that  
331 operates directly on the 5000 Hz waveform. The architecture consists of four convolutional layers with  
332 ReLU activation functions. Dropout layers (dropout probability=0.5) were added following each  
333 convolution and activation layer. The number of outputs per time point in the output layer  
334 corresponded to the number of labels. Eight feature maps were used for the three-word pulse classifier  
335 models, and 128 for the others. The last three convolutional layers have a stride of two, which has the  
336 effect of reducing the temporal sampling rate of the output predictions by 8-fold. The classifier is trained  
337 with cross entropy loss using the Adam optimizer (Kingma and Ba, 2014) with a batch size of 32 and  
338 learning rate of 1-e6 for a million training steps over half a day. The learning rate was set such that  
339 accuracy on the validation data set had not plateaued until after at least a full epoch of the training data  
340 had been sampled. Eight-fold cross-validation of the smallest model (batch=32 features=8) can be done  
341 simultaneously on a single Nvidia 1080Ti GPU with only a 29% slow down compared to training a single  
342 model. Using seven CPU cores instead of a GPU is only 6% slower for a single model.

343  
344 Training and valuating *D. melanogaster* pulse classifier accuracy and inter-annotator variability with  
345 dense annotation

346 The *D. melanogaster* pulse classifier was trained using 20 densely annotated recordings from  
347 Kyriacou et al. (Kyriacou et al., 2017), and the remaining 5 recordings were withheld for validation and  
348 used to estimate the best threshold for the classifier probabilities. Each time point was originally  
349 annotated with two classes, “pulse” vs “not-pulse”. We added an additional automatically defined class  
350 “other pulse” which were pulse-like events originally labeled as “not-pulse”. Events originally labeled  
351 “not-pulse” which passed the time domain sound detection threshold but not the frequency domain  
352 threshold were given the label of “other pulse”.

353 An interval of 2 ms was defined around each ground truth pulse. Probabilities predicted by the  
354 pulse classifier were thresholded and any contiguous time interval of pulse predictions which  
355 overlapped with the 2 ms interval around each ground truth pulse constituted a correct detection or a  
356 hit. Predicted pulse intervals that did not overlap with any ground truth intervals constituted false  
357 positives, and ground truth intervals that did not overlap any predicted intervals were counted as false  
358 negatives or misses. Inter-annotator differences were calculated similarly by defining 2 ms intervals  
359 around each annotated pulse and computing overlaps between intervals annotated by different  
360 humans. Overlapping intervals were counted as annotations agreed upon by both humans. Unmatched  
361 annotations were then labeled “only Person1” or “only Person2.” Precision-recall curves were calculated  
362 for both the validation and test set by sweeping the value at which the pulse probability was  
363 thresholded. Accuracies were reported for the densely annotated test dataset using threshold values of  
364 equal precision and recall on the densely annotated validation set.

365

#### 366 Training and evaluating multi-species song classification from sparse annotations

367 Multi-species song classification was performed by assigning each time point to the class with  
368 the maximum predicted probability. With sparsely annotated ground truth, only annotated time points

369 were considered when evaluating the accuracy of the multi-species classifier. For pulse annotations, a 2  
370 ms window around each pulse was excluded from consideration. The middle half of the interval  
371 between two adjacent pulses was automatically annotated with the “IPI” label. For example, if two  
372 pulses were annotated at 100 ms and 200 ms, the interval from 125 ms to 175 ms was assigned to the  
373 “IPI” class. The multi-species multi-song confusion matrix (Figure 3C) was calculated by counting the  
374 number (or percentage) of test dataset time points for which the class with the maximum predicted  
375 probability coincided with a ground truth class annotation. Confusion matrices for subsets of the classes  
376 (as in Figure 3I,L) were made by first re-normalizing the predicted class probabilities such that the  
377 probabilities of the subset of classes sum to 1, and then calculating the confusion matrix for those  
378 classes. Precision and recall values for each class in Figure 3D were calculated based on the maximum  
379 probability class assignments for each time point, rather than a fixed threshold as used for the pulse  
380 classifier.

381

### 382 Training the species classifier using automatically generated annotations for unsupervised discovery

383 Sound events from recordings of each species were detected using both time and frequency  
384 domain thresholds as described above, and all time points were labeled with the species class label.  
385 After training, the classifier was applied to the same detected sounds and the resulting hidden layer  
386 activations were used to generate low-dimensional embeddings for interactive discovery of song types.

## 387 Exploratory analysis and visualization using UMAP and t-SNE

388 *SongExplorer* can perform nonlinear dimensionality reduction of the high-dimensional acoustic  
389 signal for visualization in 2D or 3D using the UMAP (McInnes et al., 2018) and t-SNE (van der Maaten and  
390 Hinton, 2008) algorithms. Such visualizations can be generated from either the raw acoustic time series  
391 representation, a spectrogram representation, or from the intermediate feature representations of a  
392 trained deep network.

393 UMAP was implemented using <https://github.com/lmcinnes/umap> without any pre-processing  
394 of the high dimensional representation. t-SNE was implemented using the *scikit-learn* library,  
395 (<https://scikit-learn.org/>) and utilizes a user-selected initial linear dimensionality reduction using  
396 Principle Components Analysis (PCA) to increase algorithm efficiency.

397 Dimensionality reduction was applied to feature vectors corresponding to 204.8 ms windows of  
398 time. For the input layer, the feature vector corresponded to the raw waveform. For all hidden layers,  
399 the feature vector was constructed by concatenating the 1D time series of all the feature maps in the  
400 hidden layer. The dimensionality of this feature vector is thus given by the number of elements in each  
401 hidden layer, given in Figure S1.

402

### 403 **Acknowledgements**

404 We are grateful to Erik Snapp for supervising the Janelia-Loudoun County Public Schools  
405 Summer Program, which provided the initial opportunity for co-authors MS and FK to work on this  
406 project.

407

### 408 **Competing Interests**

409 The authors declare that they do not have any competing interests with the work described in this  
410 paper.

411 **References**

- 412 Arthur BJ, Sunayama-Morita T, Coen P, Murthy M, Stern DL. 2013. Multi-channel acoustic recording and  
 413 automated analysis of *Drosophila* courtship songs. *BMC Biol* **11**:11. doi:10.1186/1741-7007-11-  
 414 11
- 415 Berg S, Kutra D, Kroeger T, Straehle CN, Kausler BX, Haubold C, Schiegg M, Ales J, Beier T, Rudy M, Eren  
 416 K, Cervantes JI, Xu B, Beuttenmueller F, Wolny A, Zhang C, Koethe U, Hamprecht FA, Kreshuk A.  
 417 2019. Ilastik: Interactive Machine Learning for (Bio)Image Analysis. *Nature Methods* **16**:1226–  
 418 1232. doi:10.1038/s41592-019-0582-9
- 419 Chesmore ED, Ohya E. 2004. Automated identification of field-recorded songs of four British  
 420 grasshoppers using bioacoustic signal recognition. *Bulletin of Entomological Research* **94**:319–  
 421 330. doi:10.1079/ber2004306
- 422 Clemens J, Coen P, Roemschied FA, Pereira TD, Mazumder D, Aldarondo DE, Pacheco DA, Murthy M.  
 423 2018. Discovery of a New Song Mode in *Drosophila* Reveals Hidden Structure in the Sensory and  
 424 Neural Drivers of Behavior. *Current Biology* **28**:2400-2412.e6. doi:10.1016/j.cub.2018.06.011
- 425 Coffey KR, Marx RG, Neumaier JF. 2019. DeepSqueak: a deep learning-based system for detection and  
 426 analysis of ultrasonic vocalizations. *Neuropsychopharmacology* **44**:859–868.  
 427 doi:10.1038/s41386-018-0303-6
- 428 Ewing AW, Bennet-Clark HCC, Ewing AW, Bennet-Clark HCC. 1968. The courtship songs of *Drosophila*.  
 429 *Behaviour* **31**:288–301. doi:10.1163/156853968X00298
- 430 Greenspan RJ, Ferveur J-FF. 2000. Courtship in *Drosophila*. *Annu Rev Genet* **34**:205–232. doi:10.1146/annurev.genet.34.1.205  
 431 [pii]
- 432 Hill PSM. 2006. Vibration and Animal Communication: A Review1. *American Zoologist* **41**:1135–1142.  
 433 doi:10.1668/0003-1569(2001)041[1135:vaacar]2.0.co;2
- 434 Hongguang S, Dun L, Xianning Z, Haijing Y, Xia L, Dingliang Z, Yaohua Z, Zhancheng G. 1997. Study on the  
 435 recognition and evolutionary genetics of the courtship song of species in *Drosophila nasuta*  
 436 species subgroup. *Acta Genetica Sinica* **24**:311–321.
- 437 Ivanenko A, Watkins P, Gerven MAJ van, Hammerschmidt K, Englitz B. 2018. Classification of mouse  
 438 ultrasonic vocalizations using deep learning. *bioRxiv* 358143. doi:10.1101/358143
- 439 Kershenbaum A, Blumstein DT, Roch MA, Akçay Ç, Backus G, Bee MA, Bohn K, Cao Y, Carter G, Cäsar C,  
 440 Coen M, Deruiter SL, Doyle L, Edelman S, Ferrer-i-Cancho R, Freeberg TM, Garland EC, Gustison  
 441 M, Harley HE, Huetz C, Hughes M, Hyland Bruno J, Ilany A, Jin DZ, Johnson M, Ju C, Karnowski J,  
 442 Lohr B, Manser MB, Mccowan B, Mercado E, Narins PM, Piel A, Rice M, Salmi R, Sasahara K,  
 443 Sayigh L, Shiu Y, Taylor C, Vallejo EE, Waller S, Zamora-Gutierrez V, Iii EM, Narins PM, Piel A, Rice  
 444 M, Salmi R, Sasahara K, Sayigh L, Shiu Y, Taylor C, Vallejo EE, Waller S, Zamora-Gutierrez V. 2016.  
 445 Acoustic sequences in non-human animals: A tutorial review and prospectus. *Biological Reviews*  
 446 **91**:13–52. doi:10.1111/brv.12160
- 447 Kerwin P, Yuan J, von Philipsborn AC. 2020. Female copulation song is modulated by seminal fluid.  
 448 *Nature Communications* **11**:1430. doi:10.1038/s41467-020-15260-6
- 449 Kingma DP, Ba J. 2014. Adam: A Method for Stochastic Optimization. *undefined*.
- 450 Koumura T, Okanoya K. 2016. Automatic Recognition of Element Classes and Boundaries in the Birdsong  
 451 with Variable Sequences. *PLOS ONE* **11**:e0159188. doi:10.1371/journal.pone.0159188
- 452 Kyriacou CP, Green EW, Piffer A, Dowse HB, Takahashi JS. 2017. Failure to reproduce period-dependent  
 453 song cycles in *Drosophila* is due to poor automated pulse-detection and low-intensity courtship.  
 454 *PNAS*. doi:10.1073/pnas.1615198114
- 455 LaRue KM, Clemens J, Berman GJ, Murthy M. 2015. Acoustic duetting in *Drosophila virilis* relies on the  
 456 integration of auditory and tactile signals. *eLife* **4**:1–23. doi:10.7554/eLife.07277

457 McInnes L, Healy J, Melville J. 2018. UMAP: Uniform Manifold Approximation and Projection for  
458 Dimension Reduction.

459 Parsons S. 2001. Identification of New Zealand bats (*Chalinolobus tuberculatus* and *Mystacina*  
460 *tuberculata*) in flight from analysis of echolocation calls by artificial neural networks. *Journal of*  
461 *Zoology* **253**:447–456. doi:10.1017/S0952836901000413

462 Sattar F, Cullis-Suzuki S, Jin F. 2016. Identification of fish vocalizations from ocean acoustic data. *Applied*  
463 *Acoustics* **110**:248–255. doi:10.1016/j.apacoust.2016.03.025

464 Steinfath E, Palacios A, Rottschäfer J, Yuezak D, Clemens J. n.d. Fast and accurate annotation of acoustic  
465 signals with deep neural networks 30.

466 Stern DL. 2014. Reported *Drosophila* courtship song rhythms are artifacts of data analysis. *BMC Biology*.  
467 Stern DL, Clemens J, Coen P, Calhoun AJ, Hogenesch JB, Arthur BJ, Murthy M. 2017. Experimental and  
468 statistical reevaluation provides no evidence for *Drosophila* courtship song rhythms.  
469 *Proceedings of the National Academy of Sciences of the United States of America* **114**.  
470 doi:10.1073/pnas.1707471114

471 van der Maaten L, Hinton G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*  
472 **9**:2579–2605.

473 Zakon HH, Zwickl DJ, Lu Y, Hillis DM. 2008. Molecular evolution of communication signals in electric fish.  
474 *The Journal of experimental biology* **211**:1814–8. doi:10.1242/jeb.015982

475

476 **Figure Legends**

477

478 Figure 1 – *SongExplorer* web browser interface

479 (A) Screenshot of *SongExplorer* interface in a web browser window. Data and labelling tools are arrayed  
480 along the left side of wind. Analysis wizard buttons, file selectors, and parameter values are shown in  
481 the middle of the window. The right side of the window displays a detailed description of methods,  
482 including instructions for installation and data analysis.

483 (B-F) Several regions of the interface are shown in more detail. The analysis wizard buttons (B) guide  
484 users, from left to right, through analysis pipelines and highlight only those windows that are available  
485 for each step of an analysis. The low-dimension feature embedding window (C) displays a tSNE or UMAP  
486 projection of sound events. These projections can be interactively navigated in two or three dimensions  
487 and different projections can be displayed for different song types, species, or layers of the machine  
488 learning model. A subset of events can be examined in detail by selecting events with an interactive  
489 circle selector (shown in pink here). The events, or a subset of events if there are more than 50 in the  
490 region, are shown along with spectrograms for the events (D). Clicking on one of these events results in  
491 the display of this event in the larger context of a complete recording (E). This view of the song can be  
492 navigated and played as an audio clip. Events can be labelled (or unlabeled) in this window. If video was  
493 captured at the same time as audio, then the synchronized video can be loaded and displayed in a  
494 window below the audio recording (F).

495

496 Figure 2. Performance of the neural network classifier to detect one kind of song event

497 A – Outline of analysis pipeline employed to test accuracy of classifier to detect *D. melanogaster* pulse  
498 song. Dense annotations of pulse songs were combined with automated detection of other pulse-like  
499 sounds and ambient noise and a classifier was trained to recognize these three different kinds of  
500 sounds. Performance of the classifier was tested against dense human annotation.

501 B – An example of a non-obvious train of pulse song from a *D. melanogaster* recording is shown above  
502 and the probability of a pulse event over time assigned by the classifier is shown below. Vertical grey  
503 lines indicate human annotated pulse events.

504 C – To assess performance of the classifier, a dataset consisting of multiple recordings of many flies was  
505 annotated by an independent group that generated consensus labels of pulse events (“Kyriacou”) or  
506 segmented by our previously-developed classifier *FlySongSegmenter* (FSS). The neural network of  
507 *SongExplorer* was trained on 20 recordings and validated on the remaining five. 5433 pulses were  
508 labelled by all three methods (“Everyone”), 561 only by *SongExplorer* (*SongExplorer* False positives), 578  
509 only by Kyriacou (False negatives) and 51 only by *FlySongSegmenter* (FSS false positives). A very large  
510 proportion of true pulses were not detected by *FlySongSegmenter* (4139; “not FSS”) but were detected  
511 by *SongExplorer* , illustrating the considerable improvement of *SongExplorer* over *FlySongSegmenter*.

512 D – To determine how *SongExplorer*, trained on the Kyriacou consensus annotations, performs relative  
513 to individual humans, two authors with expertise in annotating fly song independently scored pulse  
514 events in the same random selections of fly song. *SongExplorer* and the two humans agreed on 2847  
515 pulse events. *SongExplorer* displayed similar levels of disagreement on the remaining pulse events as the  
516 two humans. Subsequent manual investigation revealed that disagreements related to low-amplitude or  
517 isolated pulse events about which two humans could easily disagree.

518

519 Figure 3 - Multi-species song type classification by *SongExplorer*

520 A – Analysis pipeline to assess ability of classifier to recognize songs from many *Drosophila* species. Song  
521 events from recordings of ten species were sparsely annotated and species-specific song-type labels  
522 were used to train the classifier. Classifier performance was tested by assessing the frequency with  
523 which the classifier correctly assigned a song type at manually annotated events.

524 B – Phylogeny of the ten *Drosophila* species used in this analysis, color coded by the song types shown in  
525 axis labels and the performance statistics shown in panels C and D.

526 C – Confusion matrix for 37 song types across ten *Drosophila* species color-coded by the species name  
527 given in B, plus “ambient” and non-song sounds (“other”). During manual annotation, we tentatively  
528 identified multiple types of similar songs for some species, which are indicated by numbers following the  
529 song types. Some of these alternative song types, such as *D. erecta* pulse types 1 and 2, are well  
530 discriminated by the classifier. Other types, such as *D. erecta* sine types 1 and 2 are not well  
531 discriminated, suggesting that they do not represent alternative song types. However, the sample sizes  
532 for alternative song types that are not well discriminated are low, suggesting that the failure to  
533 discriminate may have resulted from insufficient data for accurate training on these potentially different  
534 song types. Each colored square represents the fraction of time points that were annotated by humans  
535 as indicated by the row label and classified by the machine learning algorithm as indicated by the  
536 column label. The upper right triangles within each square sum to 100% within each row; the lower left  
537 sum within each column. Non-zero entries off the diagonal indicate false positive and false negatives in  
538 the lower left and upper right triangles, respectively. Each annotated time point was classified as the  
539 song type corresponding to the event with the highest probability.

540 D, E — Precision (D) and recall (E) of the classifier for all song types plotted against the number of  
541 annotations for each song type. Performance tends to improve with increasing sample size for each song  
542 type.

543 F – Example of classifier performance to discriminate two pulse types produced by *D. persimilis* males. A  
544 trace of approximately 1.5 sec of song is shown at top and the probabilities for pulse type 1 and 2 are  
545 shown below in red and blue, respectively. Vertical lines indicate human annotated pulse events color-  
546 coded by pulse type.

547 G – Magnified views of the *D. persimilis* pulse types 1 and 2, shown in blue and red, respectively.

548 H-M – The classifier discriminated similar pulse song events produced by two closely-related species, *D.*  
549 *mauritiana* and *D. simulans*, even though the differences are not obvious to humans. Song traces for *D.*  
550 *mauritiana* (top of G) and *D. simulans* (top of J) are shown with the classifier probabilities shown below  
551 each trace. Magnified views of example pulses from *D. mauritiana* (H) and *D. simulans* (K) do not reveal  
552 obvious differences between the pulses. Confusion matrices for *D. mauritiana* (I) and *D. simulans* (L)  
553 reveal that the classifier classifies pulse events to the correct species with greater than 90% accuracy.  
554 Vertical lines in (H) and (K) indicate human annotated pulse events color-coded by species, red for *D.*  
555 *mauritiana* and blue for *D. simulans*.

556

557 Figure 4 - *SongExplorer* allows fast prediction, which will facilitate closed-loop applications.

558 A – Outline of the analysis pipeline, which differs from the analysis pipeline shown in Figure 2A only by  
559 the use of multiple context windows, representing different latencies relative to the predicted event, for  
560 training.

561 B – Illustration of how the context window was shifted to test ability of neural network to predict event  
562 at the focal position, indicated by dotted vertical line. Example shows song trace containing ten pulse  
563 events. The context window, which is normally centered on the predicted event, was shifted earlier in

564 time to test the predictive ability of the classifier when the latency between the event and the  
565 prediction is shifted from 102.4 ms to 6.4 ms.

566 C – Performance of the classifier with different latencies was measured as congruence amongst human  
567 annotators and the deep learning network. Congruence amongst “Everyone” remained high for all  
568 latency durations except for 6.4 ms.

569 D-G – Classifier performance using different latencies relative to manual annotation by two humans.  
570 Performance was considerably lower for isolated pulses relative to pulses within trains. Classifier  
571 performance for the first and last pulse of each train were also lower than for pulses within trains.

572

573 Figure 5 – Dimensionality reduction reveals structure of hidden layers activation and facilitates rapid  
574 identification of new song types.

575 A – Analysis pipeline to illustrate how visualization of hidden layer activations reveals relatively discrete  
576 structure of single song types. Songs that were sparsely annotated for pulse, sine, inter-pulse interval  
577 and ambient sounds were used to train a classifier. Principle component analysis values of the the input  
578 and hidden layer activations were projected into UMAP space.

579 B-E – UMAP projections of input (B) and hidden layer (C-E) activations for *D. melanogaster* song. For  
580 input layer (B), pulse events form two loose clusters and sine song forms one loose cluster and all IPI  
581 events overlap with ambient sound. The two pulse event clusters represent the two phases of pulse  
582 song present in the data, which is an artifact of the position of the fly relative to the microphone at  
583 different times in the recording. At increasingly deeper layers in the model, the activations become  
584 increasingly differentiated (C-E), and clearly separate out most pulse, sine, IPI, and ambient events by  
585 hidden layer 3 (E).

586 F – Analysis pipeline illustrates how new song types can be rapidly identified by clustering hidden layer  
587 activations. Sound events passing a simple amplitude or power spectrum threshold were projected  
588 through the model trained on annotated song (A-E) and the input layer and hidden-layer activations  
589 were projected into UMAP space.

590 G-K – One strong cluster of thresholded events was obvious in all hidden layers (H-J), but not obvious in  
591 the input layer (G). Most of these events corresponded to a novel song type, which has recently been  
592 reported to be the female copulation song. Three of these copulation song events are illustrated (K) and  
593 their locations in the input and hidden layers are indicated by three larger red dots.

594

595 Figure 6 – A classifier trained to recognize species sounds, rather than specific events, facilitates *de novo*  
596 discovery of song types.

597 A – The deep learning classifier was trained to recognize sounds automatically detected with time- and  
598 frequency-domain thresholds sampled from multiple species. The classifier was trained to recognize the  
599 species, not individual song types. Data represented in UMAP space were manually examined to identify  
600 previously discovered and new song types.

601 B – The 21 species used in the analysis. Two *D. melanogaster* recordings were labelled separately as a  
602 negative control.

603 C,D – UMAP projections of the input layer (C), which contained little structure, and hidden layer 3 (D),  
604 which exhibited extensive structure. The detected sound events from many species occupied distinct  
605 domains in the UMAP projection of hidden layer 3.

606 E – UMAP projections of hidden layer 3 for detected events from the two *D. melanogaster* recordings  
607 shows that that the classifier was unable to differentiate two sets of recordings from *D. melanogaster*.

608 F-H — UMAP projections of *D. teissieri*, *D. nasuta*, and *D. persimilis* illustrate that songs from different  
609 species occupy different locations within UMAP space and facilitated discovery novel song types.

610

611 Figure S1. Neural network architecture implemented in *SongExplorer*. Variable C indicates the number of  
612 feature maps used in each layer, and was 8 for the 3-word models and 128 for the others. Variable T  
613 indicates the number of output classes, which is equal to the number of labelled word classes. Other  
614 hyperparameters include a batch size of 32, the Adam optimizer, a dropout probability of 0.5, and a  
615 learning rate of 1e-6. A Keras summary of the model is shown below.

616

617 Figure S2. Use of precision-recall curves to select thresholds for converting event probabilities to  
618 ethograms. (A) Congruence between dense predictions by *SongExplorer* and dense annotations by a  
619 human was calculated for a range of thresholds. The chosen threshold (vertical purple line) was that for  
620 which an equal number of false negatives (orange line) and false positives (green line) was achieved, or  
621 whatever the user specified as the desired ratio. Compare with the threshold based on sparse  
622 predictions (vertical red line) at just those points which were annotated. (B) Same data as in (A) but  
623 plotted parametrically in threshold. The area under the curve is 0.97.

624

625 Figure S3. Effect of deep-learning network parameters on trained model accuracy. Maximum F1 is  
626 relatively insensitive to 8-fold changes in batch size and the number of feature maps. Larger batch sizes  
627 are more susceptible to overtraining, and fewer feature maps require longer training times and exhibit  
628 larger variability across repetitions. The learning rate was set such that accuracy on the validation data  
629 set had not plateaued until after at least a full epoch of the training data had been sampled (1e-6 for all  
630 models except 1e-7 for batch=32 features=64 and 1e-5 for batch=256 features=8). Eight-fold cross-

631 validation of the smallest model (batch=32 features=8) can be done simultaneously on a single Nvidia  
632 1080Ti GPU with only a 29% slow down compared to training a single model. Using seven CPU cores  
633 instead of a GPU is only 6% slower for a single model.

634

635 Figure S4. Recordings in different years display different noise characteristics. Recordings performed in  
636 2013 and 2019 were made using chambers of different sizes. We tested whether recordings made in  
637 2013 and 2019 differed systematically in their noise characteristics by training the deep learning  
638 network to recognize ambient sounds in recordings from these two years. UMAP projections of the  
639 ambient noise from each year mostly overlap in the input layer (A), but are strongly differentiated in the  
640 third hidden layer (B), implying that the classifier differentiated the two song types solely on the basis of  
641 ambient noise.

642

643 Figure S5. Dependence of classifier performance on the number of labelled song events used for  
644 training. As few as 100 labelled song events resulted in performance approximately as high as songs  
645 events labelled with many more events.

646

647 Figure S6. Song types discovered for nine species of the *D. nasuta* species group. Phylogeny of the  
648 species examined is shown on the left, with the samples used for song analysis in the same color font as  
649 the songs shown on the right. One previous study had found one song type each for 6 of the species we  
650 studied and none for two of the species we studied. In contrast, we identified song in all species we  
651 studied, and from two to seven apparently distinct song types in different species.

652

653 Figure S7 Video. Videos illustrating the *SongExplorer* workflow can be found at the following YouTube  
654 channel: <https://www.youtube.com/playlist?list=PLYXyXDkMwZip8x78RAyN6ee9NK42WBbKb>

655

656 Supplementary Audio Files: WAV files of the song types illustrated in Figure S5 are available on FigShare.

Figure 1

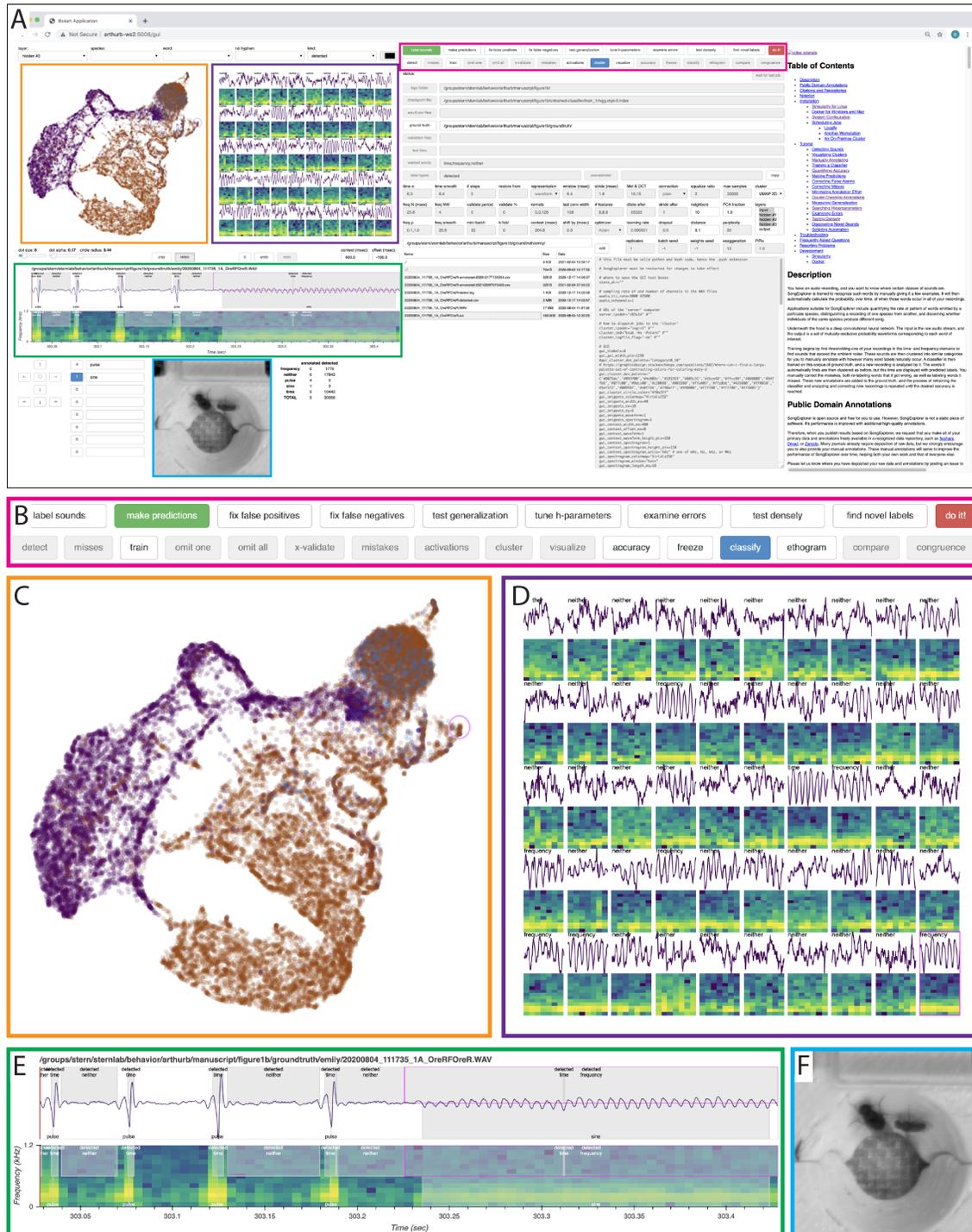
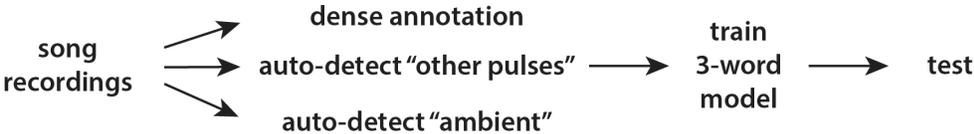
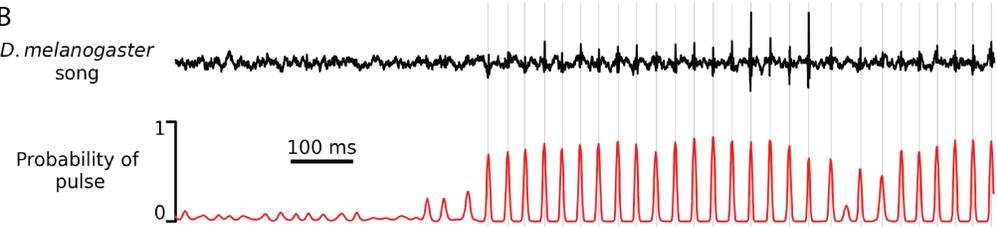


Figure 2

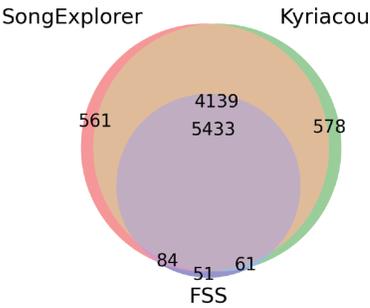
A



B



C



D

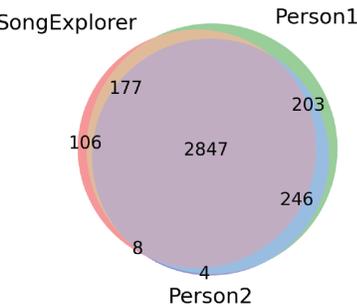


Figure 3

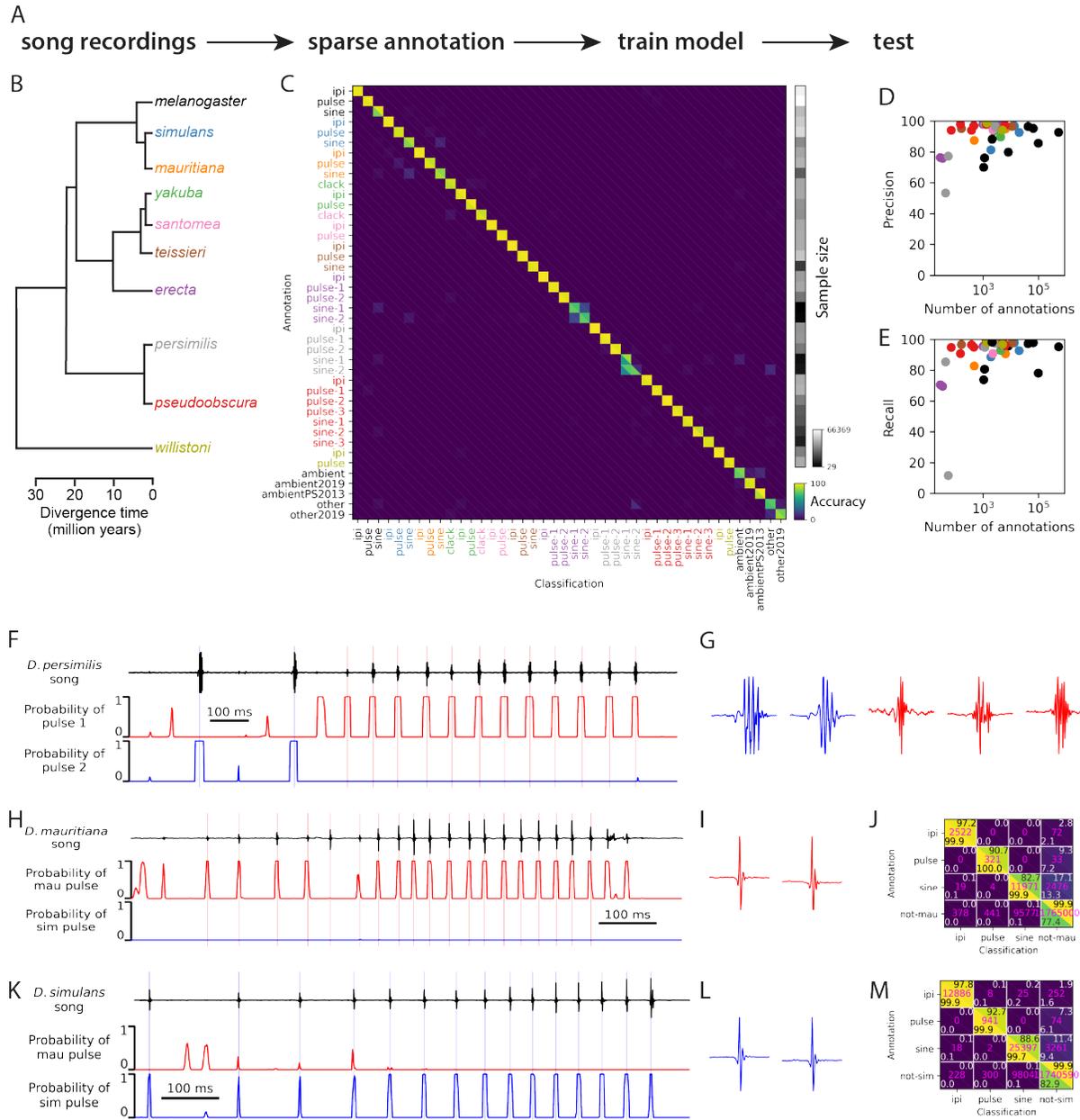


Figure 4

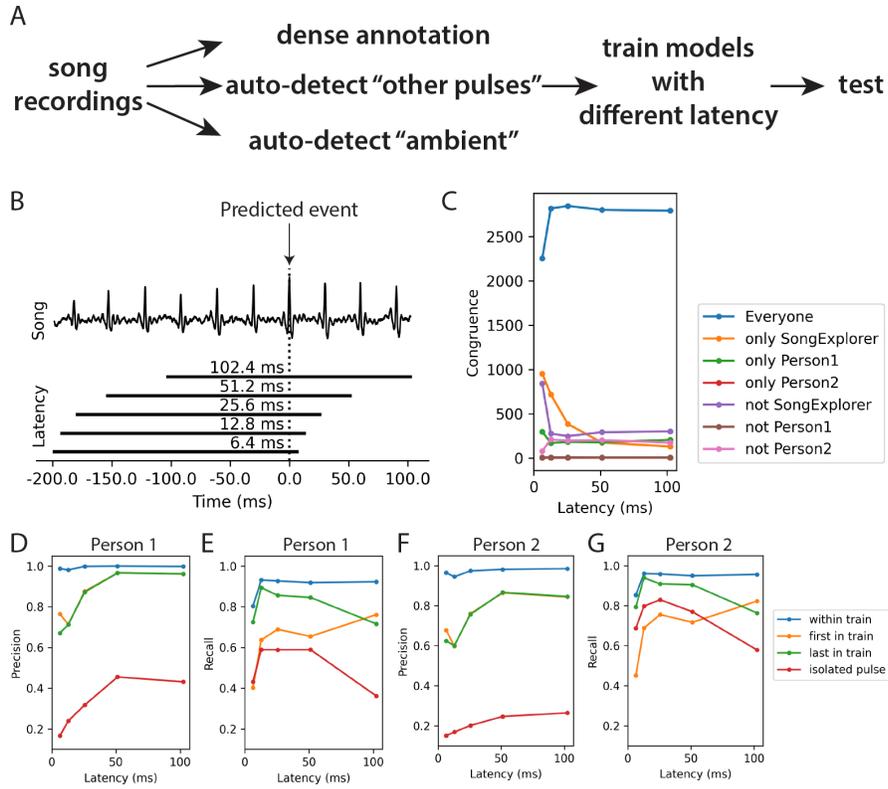


Figure 5

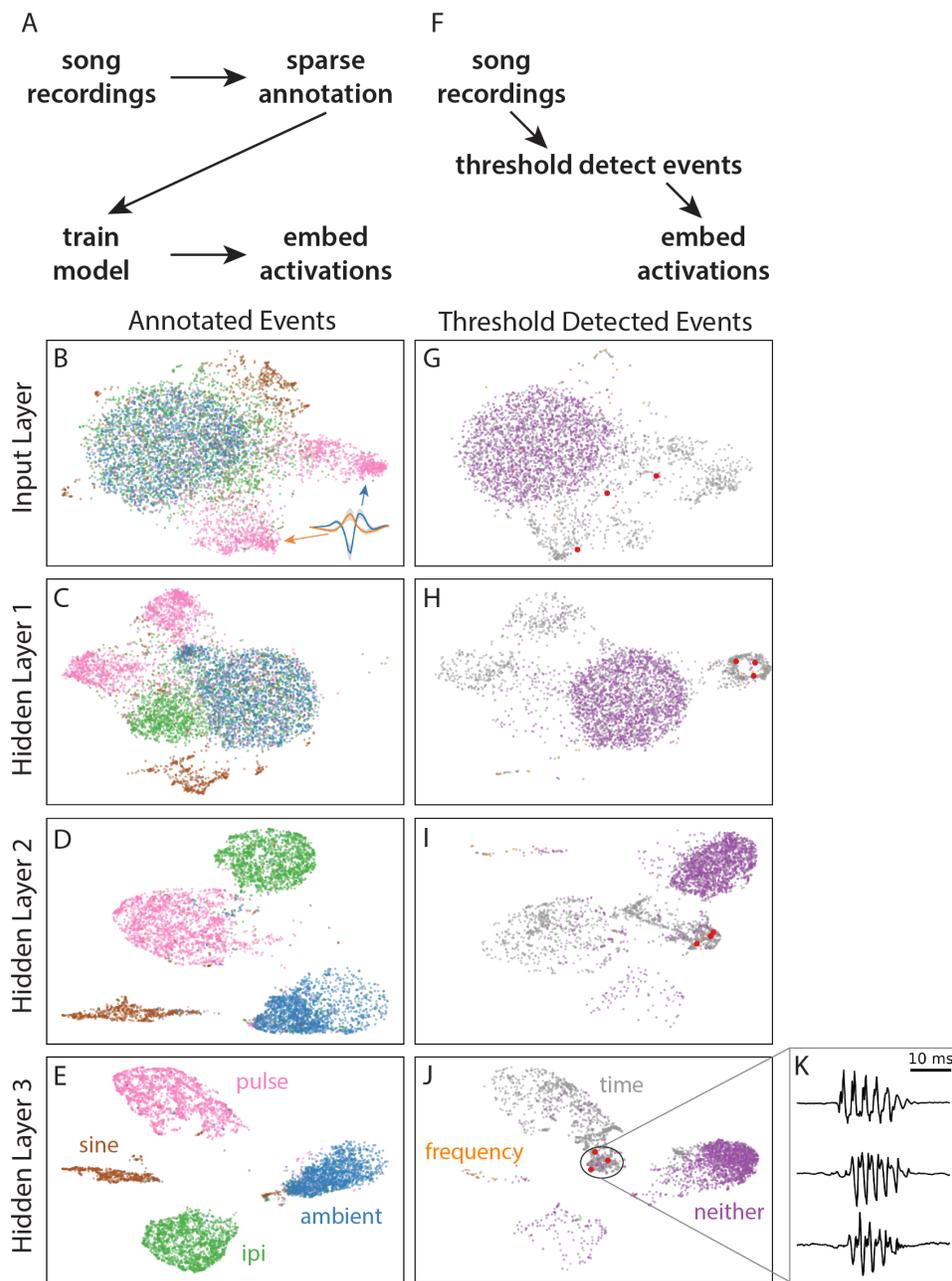


Figure 6



B

- D. albostrigata*
- D. bilimbata*
- D. elegans*
- D. erecta*
- D. taxon f*
- D. gununcola*
- D. kepulauan*
- D. kohkoa*
- D. mauritiana*
- D. melanogaster 1*
- D. melanogaster 2*
- D. nasuta*
- D. niveifrons*
- D. persimilis*
- D. pseudoobscura*
- D. pulaua*
- D. santomea*
- D. simulans*
- D. sulfurigaster*
- D. teissieri*
- D. willistoni*
- D. yakuba*
- ambient

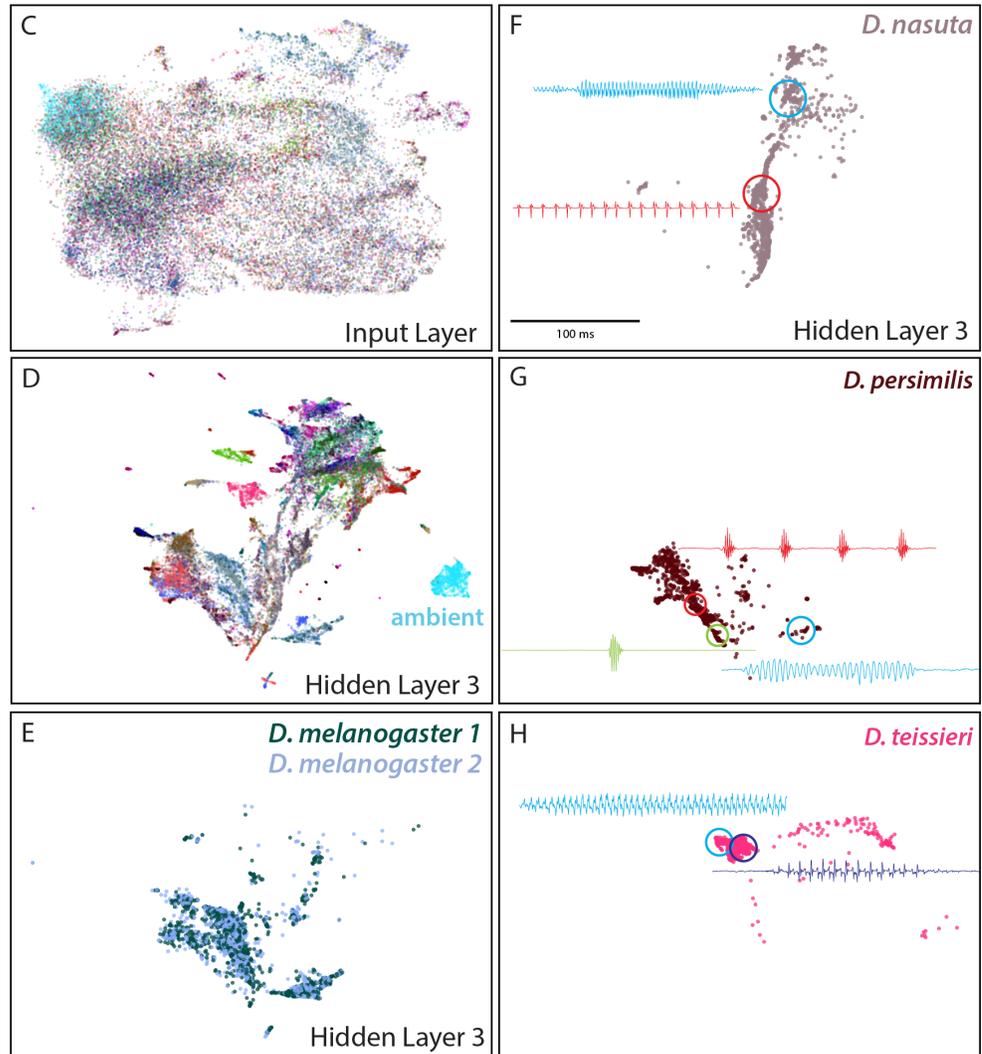
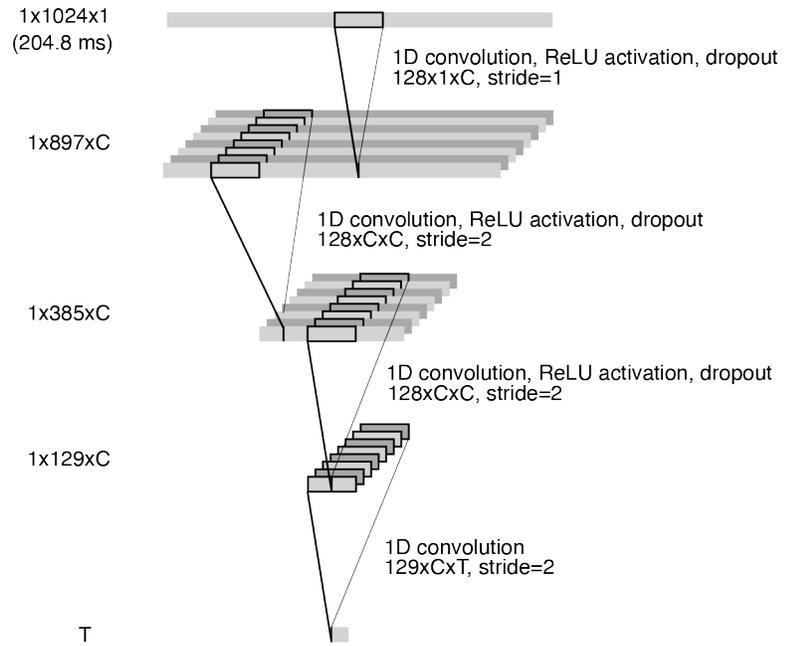


Figure S1



Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 897, 8)	1032
re_lu (ReLU)	(None, 897, 8)	0
dropout (Dropout)	(None, 897, 8)	0
conv1d_1 (Conv1D)	(None, 385, 8)	8200
re_lu_1 (ReLU)	(None, 385, 8)	0
dropout_1 (Dropout)	(None, 385, 8)	0
conv1d_2 (Conv1D)	(None, 129, 8)	8200
re_lu_2 (ReLU)	(None, 129, 8)	0
dropout_2 (Dropout)	(None, 129, 8)	0
conv1d_3 (Conv1D)	(None, 1, 3)	3099

Total params: 20,531  
 Trainable params: 20,531  
 Non-trainable params: 0

Figure S2

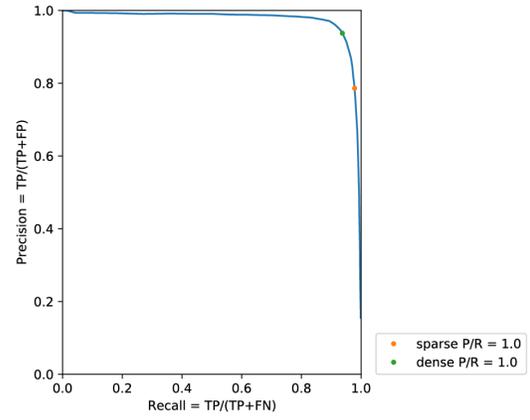
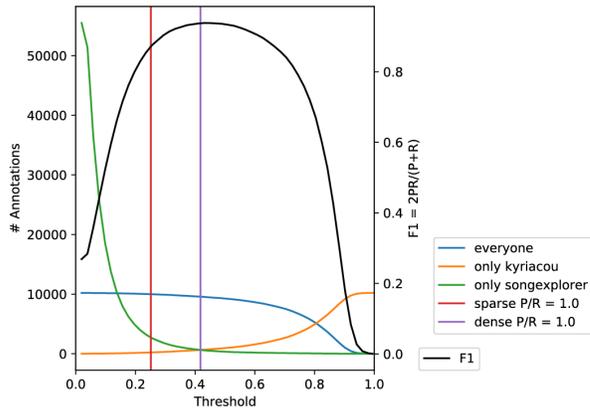
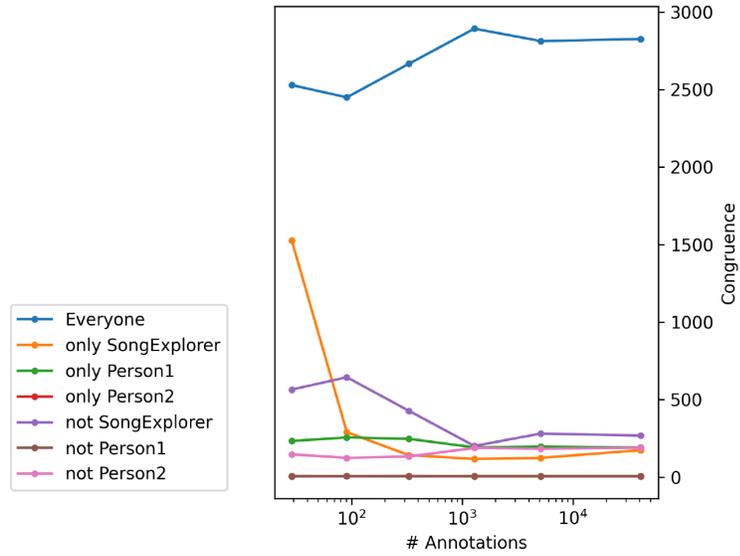


Figure S3

A



B

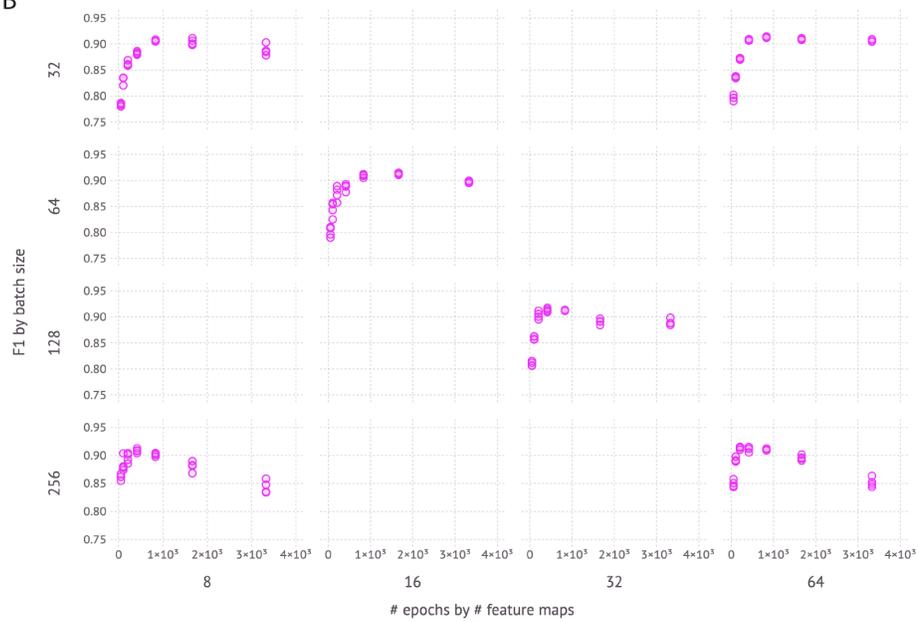


Figure S4

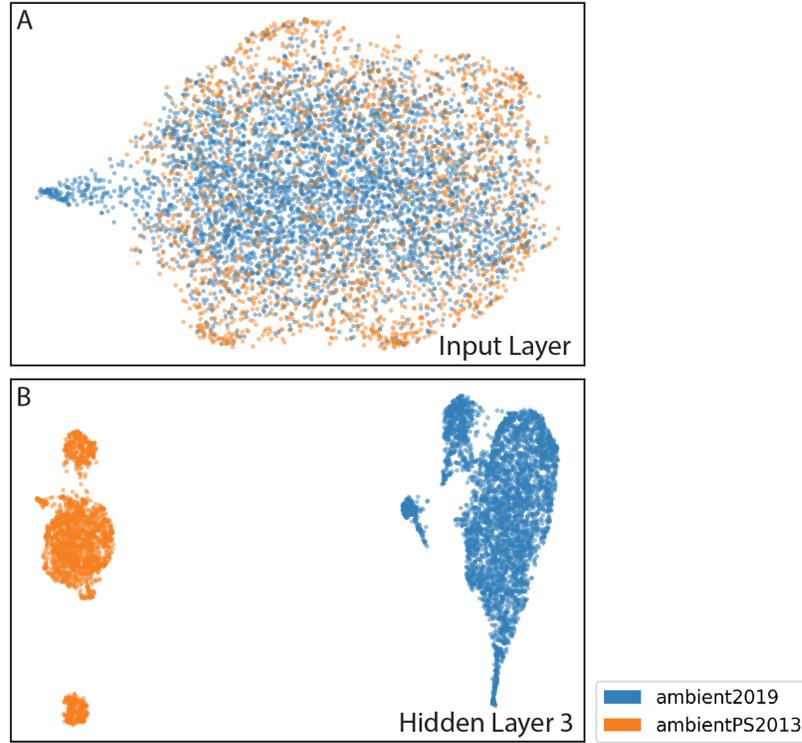


Figure S5

