







A practical guide for generating unsupervised, spectrogram-based latent space representations of animal vocalizations

Mara Thomas^{1,2}  | Frants H. Jensen^{3,4}  | Baptiste Averly^{1,2}  | Vlad Demartsev^{1,2}  |
Marta B. Manser^{5,6}  | Tim Sainburg⁷  | Marie A. Roch⁸  |
Ariana Strandburg-Peshkin^{1,2,5,9} 

¹Department for the Ecology of Animal Societies, Max Planck Institute of Animal Behavior, Constance, Germany; ²Department of Biology, University of Konstanz, Constance, Germany; ³Department of Biology, Woods Hole Oceanographic Institution, Woods Hole, MA, USA; ⁴Department of Biology, Syracuse University, Syracuse, NY, USA; ⁵Kalahari Meerkat Project, Kuruman River Reserve, Van Zylsrus, South Africa; ⁶Department of Evolutionary Biology and Environmental Studies, University of Zürich, Zürich, Switzerland; ⁷Department of Psychology, University of California San Diego, La Jolla, CA, USA; ⁸Department of Computer Science, San Diego State University, San Diego, CA, USA and ⁹Centre for the Advanced Study of Collective Behavior, University of Konstanz, Constance, Germany

Correspondence

Mara Thomas

Email: mthomas@ab.mpg.de

Funding information

Alexander von Humboldt-Stiftung; Deutsche Forschungsgemeinschaft, Grant/Award Number: EXC 2117 - 422037984; Gips-Schüle-Stiftung; Human Frontiers Science Program, Grant/Award Number: RGP0051/2019; Minerva Foundation; University Konstanz Zukunftskolleg

Handling Editor: Veronica Zamora-Gutierrez

Abstract

1. Background: The manual detection, analysis and classification of animal vocalizations in acoustic recordings is laborious and requires expert knowledge. Hence, there is a need for objective, generalizable methods that detect underlying patterns in these data, categorize sounds into distinct groups and quantify similarities between them. Among all computational methods that have been proposed to accomplish this, neighbourhood-based dimensionality reduction of spectrograms to produce a latent space representation of calls stands out for its conceptual simplicity and effectiveness.
2. Goal of the study/what was done: Using a dataset of manually annotated meerkat *Suricata suricatta* vocalizations, we demonstrate how this method can be used to obtain meaningful latent space representations that reflect the established taxonomy of call types. We analyse strengths and weaknesses of the proposed approach, give recommendations for its usage and show application examples, such as the classification of ambiguous calls and the detection of mislabelled calls.
3. What this means: All analyses are accompanied by example code to help researchers realize the potential of this method for the study of animal vocalizations.

KEYWORDS

animal sounds, animal vocalizations, bioacoustics, call classification, dimensionality reduction, spectrogram, UMAP, unsupervised learning

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial](https://creativecommons.org/licenses/by-nc/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2022 The Authors. *Journal of Animal Ecology* published by John Wiley & Sons Ltd on behalf of British Ecological Society.

1 | INTRODUCTION

Unsupervised dimensionality reduction projects data into low-dimensional space with the aim of visualizing underlying structure and aiding its detection. In contrast to supervised methods, it is not designed to separate known classes, but to reveal previously unknown structure and patterns in unlabelled datasets. Enabling exploration of high-dimensional datasets in a purely data-driven way makes unsupervised dimensionality reduction applicable to many research fields, among them the study of animal vocalizations. While it is often known that there is some underlying structure in vocalization datasets (e.g. different types of calls), it can be difficult to categorize signals into distinct types or quantify similarities between them. If done manually, for example, by human listeners, there is often disagreement between annotators, partly because of a lack of clear-cut rules for categorization and partly because many vocal repertoires are graded to some extent (Kershenbaum et al., 2016). Computational methods that tackle these challenges in a more objective and quantifiable way are thus highly desirable.

Unsupervised dimensionality reduction can provide the basis for such computational tools by projecting entire datasets of vocalizations to 2D or 3D space, thus allowing one to visualize underlying structure and facilitating the identification of clusters of highly similar signals (i.e. call types). Such methods have been extensively used to study vocalizations, but have mostly been applied to acoustic feature vectors, a type of encoding where vocalizations are described by parameters such as their fundamental frequency, mean spectral entropy, bandwidth or cepstral peak prominence (reviewed in Priyadarshani et al., 2018). Disadvantages of this approach are that the resulting visualization in 2D or 3D space varies greatly with the choice of acoustic features and that feature extraction is often not trivial and requires expert knowledge.

Recently, Sainburg, Thielk, and Gentner (2020) proposed a method to generate meaningful latent space representations of animal vocalizations that works directly on spectrograms, thus holding the promise of providing a less biased, more objective and easier to implement method to study vocal repertoires within and across species. Using Uniform Manifold Approximation and Projection (UMAP), they directly mapped spectrograms into low-dimensional latent space. In addition to the identification of call type groups through clustering in latent space, they showed that the low-dimensional representations can be used to study vocal encoding of individual identity, make cross-species comparisons and analyse sequential organization of vocalizations.

To make this simple yet effective approach more accessible, we provide a tutorial for the generation of such representations using a dataset of meerkat *Suricata suricatta* calls as an example. The meerkat repertoire is an ideal example use case as it has been extensively studied (Manser, 1998), yet holds many of the challenges that are typical in the field of bioacoustics: There are a number of distinct and well-characterized call types, but also some degree of gradation, with calls falling in between those types. There is a disagreement between human labellers on correct categorization of these types

and there are sub-types which have not yet been fully described. By comparing the patterns resulting from unsupervised dimensionality reduction to the manual categorization of calls by human expert labellers, we show strengths and weaknesses of the UMAP approach by Sainburg, Thielk, and Gentner (2020). In addition, we discuss the choice of various pre-processing steps and dimensionality reduction hyperparameters and provide recommendations as well as application examples for researchers who wish to apply this method to their own data.

2 | EXPLANATION OF THE METHOD

The approach of Sainburg, Thielk, & Gentner (2020) is based on two core concepts: First, the encoding of animal vocalizations as row-wise concatenated spectrograms as opposed to vectors of acoustic features. Second, the unsupervised dimensionality reduction with UMAP, a method that is based on manifold learning and topological data analysis. Hence, each vocalization is first transformed into a spectrogram, a visual representation of the frequency content of the signal over time. Then, all rows of the spectrogram are concatenated to generate a long, numerical vector (or high-dimensional datapoint; Sainburg, Thielk, & Gentner, 2020). Each field (or dimension) of the vector contains one 'pixel' of the original spectrogram, that is, the signal magnitude at a certain point in frequency and time. The Euclidean (or other) distance between these spectrogram vectors can then be used as a measure of acoustic similarity, which is essentially based on the element-wise comparison of spectrograms (similar to spectrogram cross-correlation; Clark et al., 1987). UMAP computes these distances and visualizes the structure of the data in a low-dimensional space (e.g. 2D, 3D) to facilitate the detection of clusters of similar vocalizations.

UMAP was first described in 2018 as a novel dimensionality reduction method that produced results similar to t-Stochastic Neighbour Embedding (t-SNE) (Maaten & Hinton, 2008), but is computationally faster, more scalable and based on a different mathematical framework (McInnes et al., 2018). In brief, UMAP constructs a weighted neighbourhood graph from high-dimensional data and finds a lower dimensional representation with similar topological properties through a stochastic learning process. This objective makes UMAP very similar to other neighbourhood graph-based algorithms (e.g. t-SNE (Maaten & Hinton, 2008) or LargeVis (Tang et al., 2016)), and differentiates it from algorithms that, for example, aim to preserve all pairwise distances (e.g. Sammon mapping (Sammon, 1969), Multi-Dimensional Scaling (Torgerson, 1958)) or variation in the data (e.g. principal component analysis (PCA); Hotelling, 1933). The core idea of UMAP and other neighbourhood graph methods is to emphasize the preservation of local over global structure in the inherently lossy process of dimensionality reduction (as it is generally impossible to maintain the exact distance structure of high-dimensional data in lower dimensional space). Hence, given a dataset of different types of animal vocalizations, an embedding generated with UMAP (or t-SNE, LargeVis, etc.) will more accurately

reflect the closeness of similar vocalizations in space than distances between dissimilar vocalization types. Furthermore, relative local density of the data is not preserved, meaning that dense or more loose point clouds in latent space do not necessarily reflect density of the original data (however, a density-preserving version for UMAP has recently been published; Narayan et al., 2020). While a detailed discussion of the UMAP algorithm is beyond the scope of this manuscript, it is important to keep its basic properties in mind when interpreting embeddings, as their consequences could otherwise be misinterpreted (McInnes, n.d. and McInnes et al., 2018 recommended for further reading).

3 | THINGS TO CONSIDER BEFORE USING THIS METHOD

3.1 | Input requirements

The approach requires a dataset of sound files, each containing a single vocalization or syllable as input. Note that it may be necessary to extract such vocalizations from acoustic recordings in a pre-processing step which is not covered here (e.g. use dynamic threshold segmentation (Sainburg, Thielk, & Gentner, 2020) or other acoustic event detection, see Lostanlen et al. (2019) for an overview). Ideally, the start and the end of the sound file correspond to start and end of the vocalization. If there are delays in the onset of the vocalizations, these should be the same for all sound files. Otherwise, varying onsets may make these vocalizations appear dissimilar. If it is not possible to mark the start times correctly, the pipeline can be adapted, for example, to allow for time-shifting (see Supporting Information).

3.2 | Considerations of sample size and constitution

While there is no definite minimum sample size for UMAP, it is not recommended to use UMAP on datasets with less than 100 samples in total. For our dataset, $N = 50$ vocalizations of each type ($N = 350$ in total) were sufficient to achieve the same degree of call type clustering as with any higher number of samples (see Supporting Information P3). Strong over- or underrepresentation of specific vocalizations in the dataset (e.g. class imbalance) was also unproblematic for our dataset (see Supporting Information P2). However, it may be advisable to downsample heavily overrepresented types to improve the readability of the visualizations.

3.3 | Constraints

The biological meaningfulness of distances in latent space, that is, how well they reflect similarity of vocalizations, depends mostly on the parameters for spectrogram generation and transformation

and the distance metric selected for UMAP. Therefore, these must be chosen with care and adapted for each dataset. We provide some recommendations for generating, denoising and transforming spectrograms and compare different distance metrics in sections 'Caveats and pitfalls' and Supporting Information P7. Lastly, it is important to keep in mind that distances in UMAP space do not faithfully reflect the distances in original space, as UMAP is designed to favour the preservation of local over global structure.

4 | WORKED EXAMPLES

We present all steps of the computational pipeline as proposed by Sainburg, Thielk, and Gentner (2020) using a dataset of $N = 6,428$ meerkat calls as an example (see Supporting Information P1 for a description of data collection and cleaning). Research with meerkats was conducted under the permission of the ethical committee of Pretoria University, South Africa (permit number: EC031-17) and permission to conduct the research was given by the Northern Cape Department of Environment and Nature Conservation, South Africa (FAUNA 1020/2016). The vocalizations in this dataset were between 50 and 500 ms long and had been manually labelled as one of seven call types: aggression (*agg*), alarm (*al*), close call (*cc*), lead (*ld*), move (*mo*), short note (*sn*) or social call (*soc*). Very noisy calls and ambiguous calls had been previously removed from the dataset (Supporting Information P1). All computations were performed with Python 3.8.

4.1 | Generation of spectrograms

Audio files contain time-series data (sound pressure over time) and the sampling rate sr (Hz, e.g. how many audio samples were acquired per second). To transform this data into a spectrogram, Short-time Fourier Transformation (STFT) can be used. Here, the audio data are divided into chunks, each chunk is decomposed into a vector of sound magnitude per frequency bin through Fast Fourier transformation (FFT) and all vectors (or FFT frames) are put side-by-side. The result is a spectrogram, that is, a two-dimensional matrix M , where $M[i,j]$ denotes the magnitude of the signal for a given frequency interval i at time point j . We used the STFT function from *librosa* v0.8.0 (McFee et al., 2015), but other packages provide the same functionality. Several hyperparameters define the spectrogram's resolution in time and frequency and prevent the occurrence of artefacts: The parameter n_fft determines how many audio datapoints go into one FFT frame and thus the frequency resolution of the spectrogram (frequency resolution in Hz = sr/n_fft). The parameter hop_length determines how many audio samples lie between adjacent FFT frames and is usually set smaller than n_fft so that there is an overlap between adjacent frames of the spectrogram. This improves the odds of FFT frames falling near the boundary of changes in the signal, and thus improves the visibility of signals. Furthermore, a window function is applied to the audio data of each FFT frame to prevent

spectral leakage (Harris, 1978). For our dataset, we set n_fft such that 30ms of audio data were used to generate one time frame (and a same-sized Hann window function) and set the hop length such that 3.75ms passed between adjacent FFT frames, resulting in an overlap of 87.5% between successive frames. We defined these parameters in seconds and calculated n_fft and hop_length for each audio file based on sampling frequency to ensure consistent temporal resolution of audio files (our dataset contained files with $sr = 48,000$ and $sr = 8,000$ Hz). As the highest frequency that can be detected without aliasing is at the Nyquist rate (one-half of the sampling rate), spectrograms of samples with higher sampling rates will have a larger frequency range than those with lower sampling rates (24,000Hz vs. 4,000Hz). Therefore, we only used frequency bins between 0 and 4,000Hz across all spectrograms. Alternatively, downsampling the 48,000Hz audio files to 8,000Hz prior to spectrogram generation could also be used to ensure that all spectrograms cover the same frequency range.

4.2 | Pre-processing of spectrograms

The spectrograms then undergo two modifications to emphasize biologically relevant features: (a) The frequency bins (Hz) are transformed to Mel bins based on the Mel-scale, a logarithmic, experimentally determined psycho-acoustic pitch scale (Stevens et al., 1937). Mel-transformation emphasizes differences between perceptually distinct calls by distorting the frequency axis to match the nonlinear hearing abilities of humans. Depending on the study species of interest and their hearing abilities, it may not always be advisable to apply (in this study, mel transformation with 20, 30, 40 or 50 mel coefficients provided better results than with 10 coefficients only or without any mel transformation, see Figure S8 for detailed analysis). We transformed all spectrograms using a Mel filterbank of 40 coefficients between 0–4,000Hz. (b) The energy content of the spectrogram is then transformed to a Decibel scale to reflect how the human auditory system perceives loudness logarithmically (Fletcher & Munson, 1933). As we used the maximal power of the spectrogram as reference, this step also provides a normalization for varying loudness of the audio files. Note that this may also be undesirable if you wish to distinguish vocalizations based on their loudness.

4.3 | Generating input vectors for UMAP

Next, each spectrogram is z-transformed to normalize for differences in overall intensity between calls, and padded with zeros up to the maximal call duration in the dataset (500ms) so that all spectrograms in the dataset are of equal length (UMAP requires a static number of attributes). All spectrograms are then row-wise concatenated to generate feature vectors (spectrogram vectors), which can be conceptualized as points in high-dimensional space.

5 | UMAP

Spectrogram vectors are mapped into low-dimensional space (2D and 3D) using UMAP from *umap-learn* (McInnes et al., 2018). UMAP builds an approximate nearest neighbour graph from the datapoints in original space (here, spectrograms of vocalizations) by computing a user-defined distance between input vectors (default: Euclidean) and then finds a low-dimensional representation that preserves the structure of the graph in an iterative optimization procedure. Even though many properties of the UMAP algorithm can be specified, the default values (with the exception of the parameter $min_dist = 0$, which is recommended for clustering) provided good results for our dataset and were also proposed in Sainburg, Thielk, and Gentner (2020) (see Supporting Information P7 for a more detailed analysis of the effects of different UMAP hyperparameters). Projecting the calls into 2D and 3D space ($n_components = 2$ or $n_components = 3$) and colouring the datapoints by their manual labels revealed structures with few distinct clusters (Figure 1), but clear separation of the manually annotated call types.

5.1 | Interactive visualization

To explore the 3D latent space representation in more detail, we developed an interactive visualization tool with audio playback (Figure 2, demonstration video and code tutorial in the provided code repository). Hovering over datapoints triggers the display of the respective spectrogram next to the plot, as well as a table containing metadata of the datapoint (e.g. meerkat identifier, sex and social status for our dataset), while a mouse-click on the datapoint triggers the audio playback of the respective call. Generally speaking, nearby calls can be interpreted as similar calls (i.e. Euclidean distance of their spectrograms is low), whereas far away calls can be interpreted as dissimilar. However, distances in latent space need to be interpreted with caution (please see section 'Caveats and Pitfalls' for a detailed discussion).

5.2 | Evaluation of the latent space representations

The development of metrics and methods for evaluating embedding quality is an open problem in the field of dimensionality reduction and the choice of embedding quality metrics is largely dependent on the experimenter's goals in embedding. Here, we discuss a set of embedding metrics that we consider relevant to the evaluation of vocal repertoire embeddings, both for completely unlabelled and partially labelled datasets. We define a good representation as one where similar vocalizations are close together and dissimilar ones are distant. Thus, the quality of the embedding depends on both (a) how well the distance metric in the original space reflects similarity between vocalizations and (b) how well the dimensionality reduction has preserved the structure of the data.

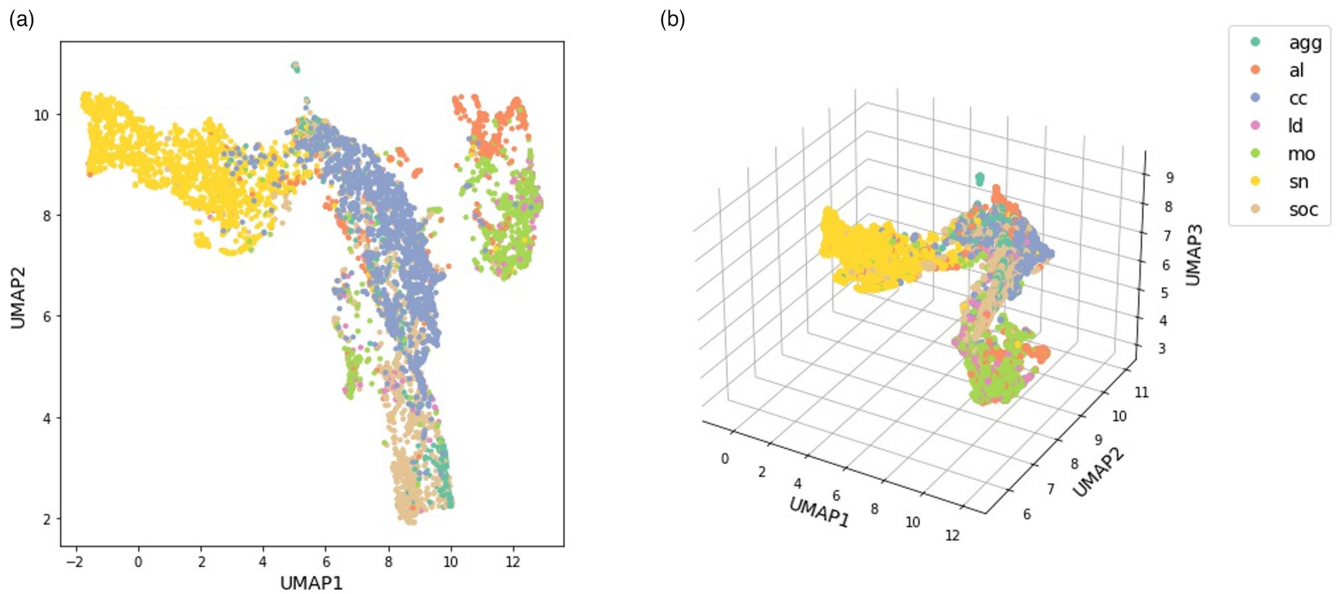


FIGURE 1 Latent space representations of meerkat vocalizations in (a) 2D and (b) 3D, colour-coded by manual call type labels

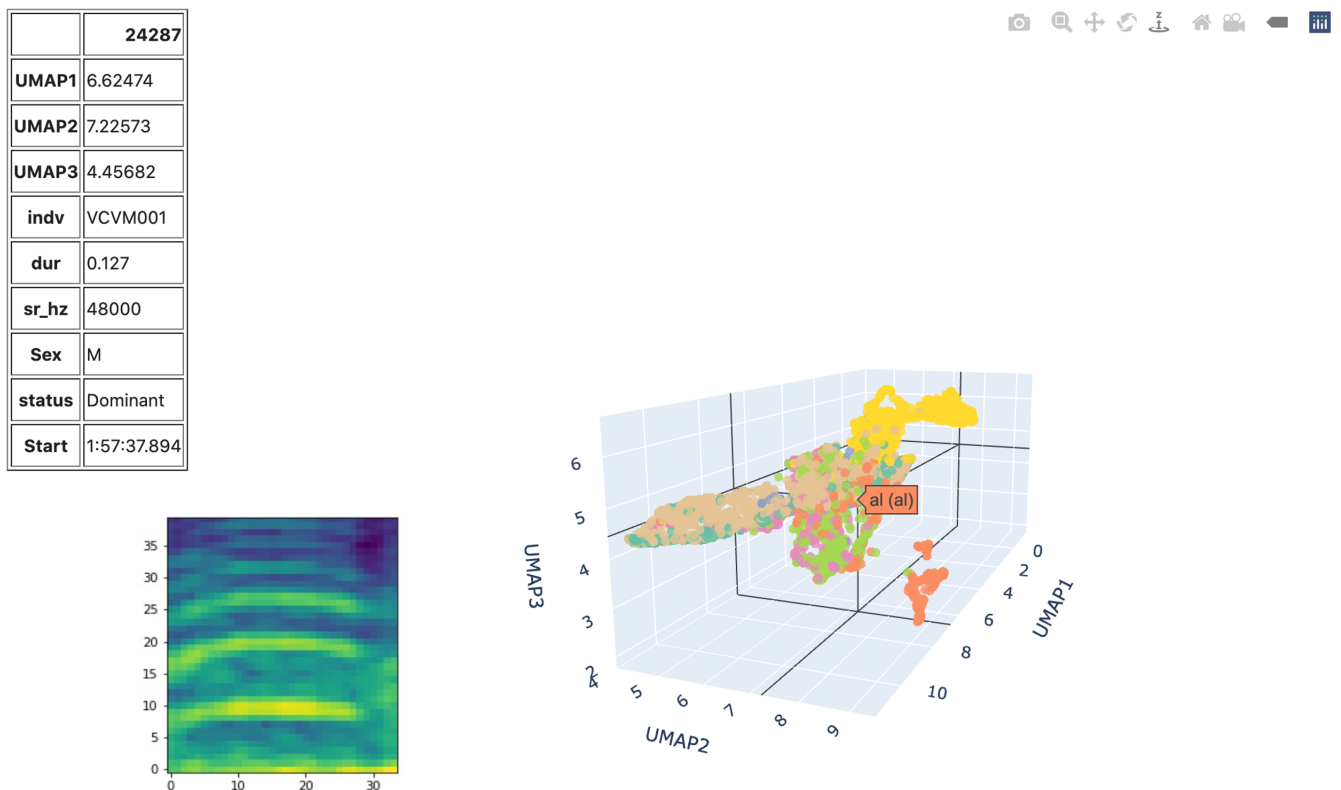


FIGURE 2 Screenshot of the interactive visualization tool, demonstrated with meerkat dataset ($N = 6,428$). Table on the left indicates the identifier of the meerkat (*indv*), call duration in s (*dur*), samplerate (*sr_hz*), sex and social status of the individual. The spectrogram of the respective call is displayed below. The plot on the right shows the representations in 3D UMAP space, coloured by manual call type label.

If no information on call types is available, the embedding quality can only be qualitatively assessed, for example, by exploring the space through the interactive visualization or by randomly pulling out example calls and their nearest neighbours. Hence, we randomly select calls from the dataset along with their k nearest neighbours,

display the spectrograms, visually assess their similarity (Figure 3) and/or play back the audio (see 'Section 5.1').

If some or all vocalizations in the dataset are labelled (as in our meerkat dataset), the clustering of call types groups can be quantitatively assessed (assuming that calls of the same type are more similar

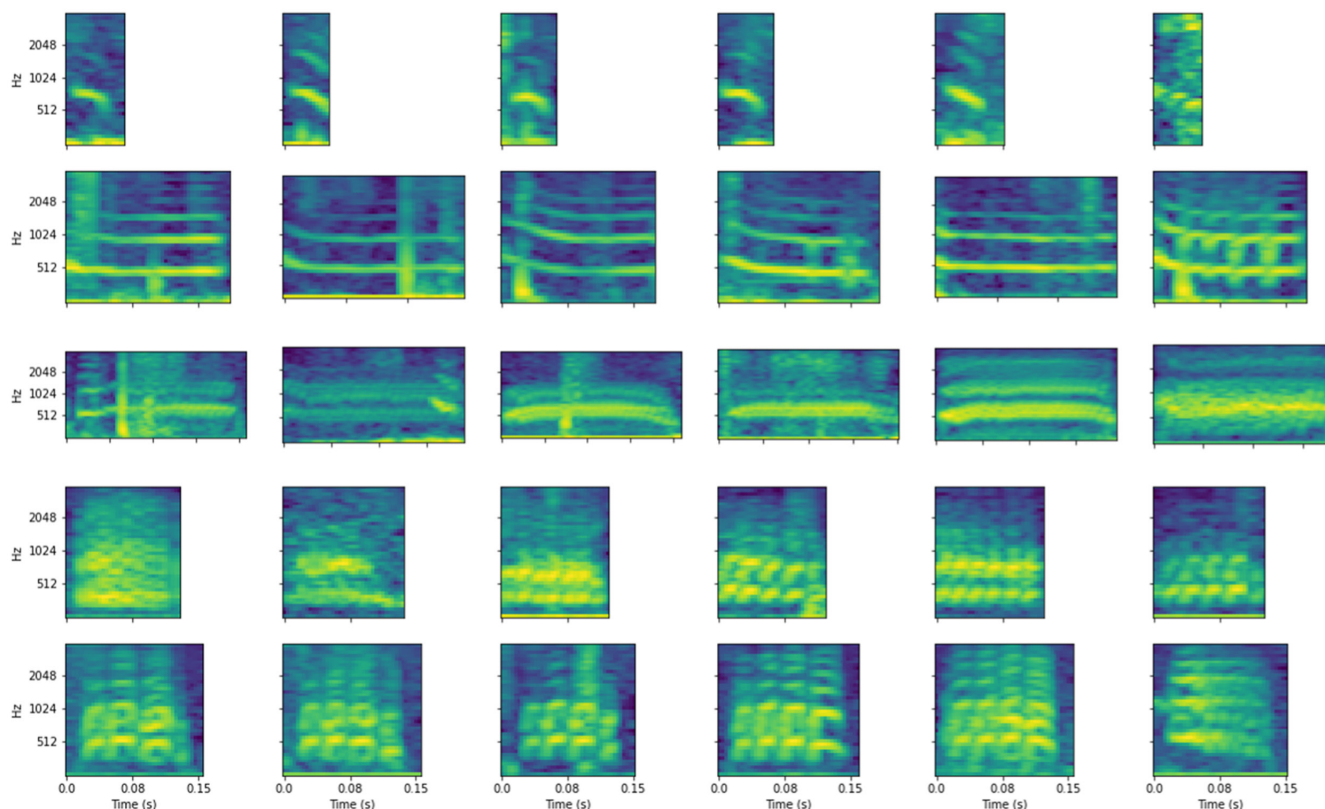


FIGURE 3 Spectrograms of five randomly selected calls from the dataset (first column) and their $k = 5$ nearest neighbours in latent space (columns 2–6)

than those of different types). Possible metrics are the Silhouette coefficient applied to the manual call type groups, the average percentage of calls of the same type in the local neighbourhood or the adjusted Rand index to compare the partition obtained from unsupervised clustering to the manual labels. In this tutorial, we also apply these metrics to the original, high-dimensional space to demonstrate the effects of dimensionality reduction. (The performance of UMAP, i.e. structure preservation, can be evaluated via nearest neighbour preservation and via the correlation of distances in low- and high-dimensional space, see Supporting Informations P5 and P6).

5.2.1 | Nearest neighbour metrics

To evaluate to what degree acoustically similar calls (e.g. with the same manual label) cluster in latent space, we assess the probability that a call is surrounded by calls of the same type in latent space. (This can also be performed for a labelled subset of the full dataset.) For a given call type label i , we select all calls of that label, identify their k nearest neighbours in latent space and note their type. We then analyse the composition of these neighbour labels and use the observed frequency as an estimate for the probability P of encountering calls of this particular type among the k nearest neighbours of calls of label i .

$$P(\text{neighbour label } j | \text{datapoint label } i) = \frac{\# \text{ of } knn_labels(i) \text{ with label } (j)}{\# \text{ of } knn_labels(i)}$$

with $knn_labels(i)$ being the list of labels of the k nearest neighbours of all datapoints with label i .

Doing this for manually labelled call types, we obtain a square evaluation matrix where each field $[i, j]$ represents the probability P (expressed in %) for a call of type i to have a neighbour of type j (Figure 4a,b).

Since this probability is not normalized to varying call type frequencies in the dataset (i.e. it is more likely to have a common call type in the neighbourhood than a rare one by chance alone), we divide it by the probability of encountering calls of this type by random chance alone (i.e. the frequency of this label in the dataset). The resulting score can then be interpreted as the fold increase or decrease in likelihood of observing this many neighbours over the random chance expectation. To make the score symmetric around zero, we apply a \log_2 transformation and obtain the normalized score P_{norm} (Figure 4c,d).

$$P_{\text{norm}}(\text{neighbour label } j | \text{datapoint label } i) = \log_2 \left[\frac{P(\text{neighbour label } j | \text{datapoint label } i)}{P(\text{neighbour label } j)} \right]$$

with $P(\text{neighbour label } j)$ being the probability of observing a neighbour with label j due to random chance alone (e.g. the frequency of label j in the dataset).

To capture the quality of an embedding in a single score, we calculate the unweighted average of P (or P_{norm}) over all classes (the diagonal of the evaluation matrix), thus obtaining the summarized

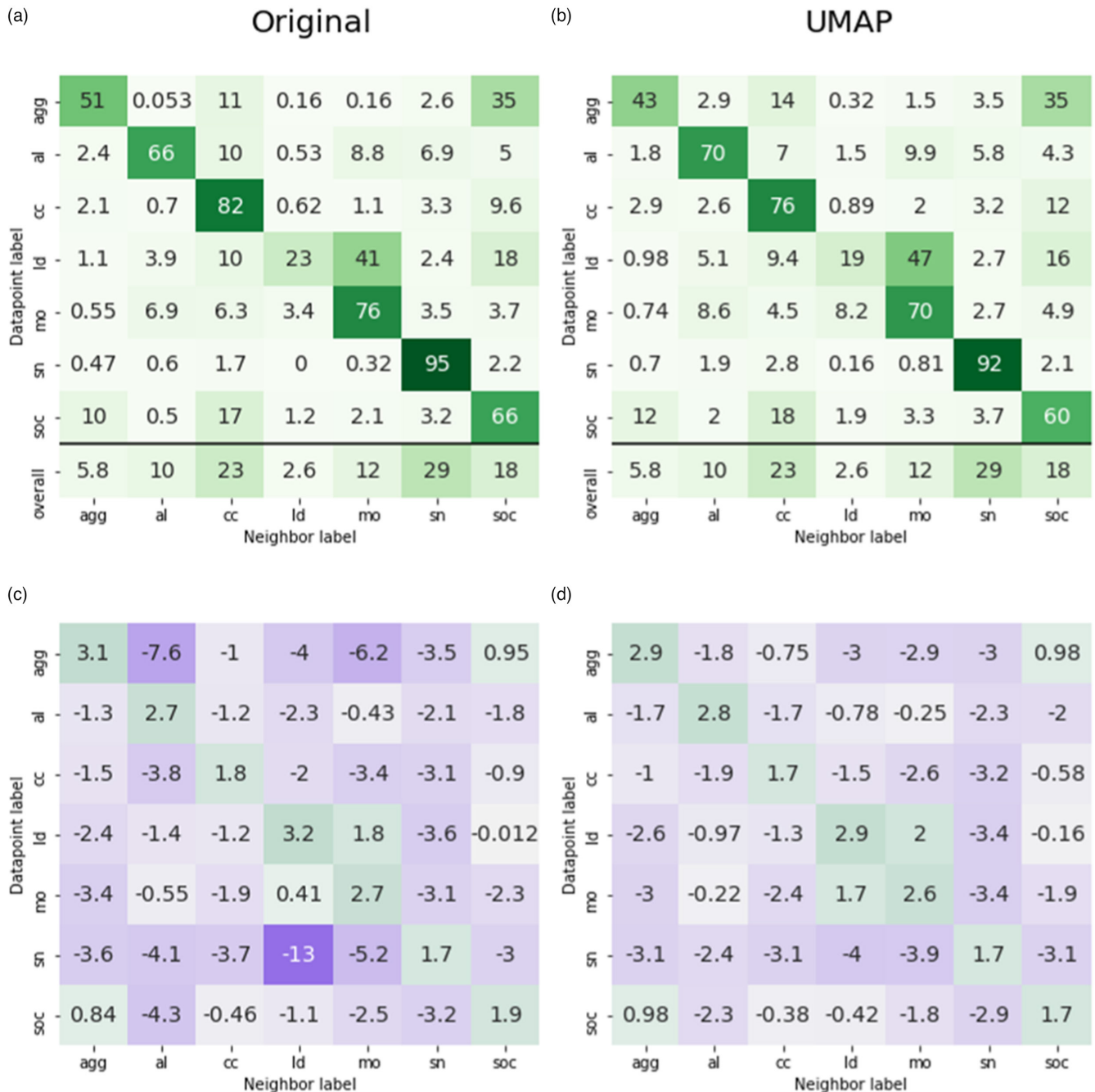


FIGURE 4 Evaluation matrix in original (a, c) and 3D UMAP (b, d) space. (a)–(b) show average frequency (%) of datapoints with label x among the nearest neighbours of a datapoint with label y . For ease of interpretation, the last row shows the random chance expectation of encountering a neighbour with this label in the dataset ('overall', e.g. frequency of this call type in the dataset). (c)–(d) show \log_2 -transformed ratio of observed frequency vs. frequency expected by chance. Colours are mapped on a violet-green scale from minimum to maximum value.

score S (or S_{norm}). We explicitly chose the unweighted average so that the same-class neighbour probabilities of each call type have equal weight in the final score and the final score is not biased towards the scores of the more frequent call types in the dataset. When comparing different embeddings of the same dataset, that is, with the same call type frequencies, we report the unnormalized score S , which can be interpreted as the average frequency of same-class labels among the k nearest neighbours of each call type.

S was 61.3% for our meerkat dataset, for example, for any given call type, on average 61.3% of the $k = 5$ nearest neighbours were of the same type. This indicates that calls of the same type were found much more often in close neighbourhoods than expected by random chance alone (random chance expectation: 14.7%). The percentage of same-class neighbours varied among call types, with the highest score for *sn* calls (92%), which was also the most frequent call type in the dataset (29%). When normalized to the random chance

expectation, *ld* and *agg* calls had the highest normalized probability of same-class neighbours ($P_{\text{norm}} = 2.9$, i.e. frequency of same-class neighbours was 7.5-fold [$2^{2.9}$] higher than expected by random chance) (Figure 4d). The normalized neighbour metrics indicate that move and lead calls, as well as social and aggressive calls, are often found in close vicinity of each other and these calls are indeed acoustically (and functionally) similar (see Figure S1). Altogether, we recommend inspecting the full evaluation matrices of both scores to get a comprehensive overview of the neighbourhood probabilities of different vocalization types.

To understand the effects of UMAP, we calculated the same evaluation matrices in original, high-dimensional space (i.e. spectrogram vector space). While the overall patterns between UMAP and original space were very similar, the original space had slightly higher quality scores ($S = 65.59$ vs. $S = 62.81$ and $S_{\text{norm}} = 2.45$ vs. $S_{\text{norm}} = 2.4$). These differences indicate that the local neighbourhood ($k = 5$) was not exactly preserved in UMAP. When re-calculating the quality score S for a different number of k nearest neighbours and thus investigating a more global neighbourhood, we found that within a larger neighbourhood ($k > 25$), S was higher in UMAP than in original space (Figure 5). This shows that while UMAP does not accurately preserve the closest nearest neighbours, it does improve the overall clustering of similar datapoints.

5.2.2 | Within- vs. between-call type distances

We also investigated the distribution of pairwise distances within a call type group vs. between calls of a different type. Again, we show results for original and UMAP space to demonstrate the effects of UMAP.

The average distance to datapoints of the same type was smaller than the average distance to datapoints of a different type for all call types in UMAP, but not in original space (Figure 6). This illustrates the effectiveness of UMAP in generating tighter clusters of similar datapoints, which facilitates the detection of patterns and

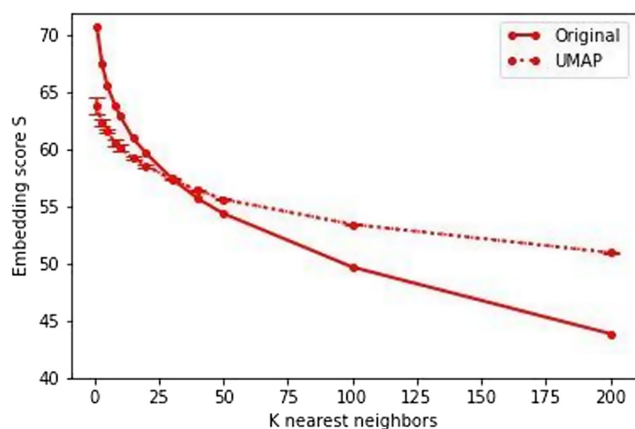


FIGURE 5 Comparison of embedding score S for different k nearest neighbours in original vs. 3D UMAP space. UMAP line represents mean and standard deviation of $n = 5$ UMAP runs.

structure in the data. In our dataset, the best separation of within- vs. between-call type distances was obtained for *mo* and *sn* calls. *Cc* and *al* calls were less separated from other call types, even though they had similarly high same-class nearest neighbour frequencies (*mo* 70%, *sn* 92%, *al* 70%, *cc* 76%), illustrating the different types of class separation that are captured by within- vs. between-class distances as opposed to the nearest neighbour metrics.

5.2.3 | Silhouette plot

As another means to quantify the global clustering by call type, we calculated the silhouette values of the manual label clusters in original and UMAP space using the implementation of *scikit-learn* (Pedregosa et al., 2011) and plotted the scores for all datapoints sorted by call type group (Figure 7). The silhouette value indicates how close datapoints are to their own cluster compared to other clusters and is defined as:

$$sil(x) = \frac{(b - a)}{\max(a, b)}$$

with a being the mean intra-cluster distance in the cluster of datapoint x and b the mean distance between x and the nearest neighbouring cluster. A positive score thus indicates that this datapoint is near elements of the same cluster, whereas a negative score indicates that it is closer to elements of another cluster.

The comparison of the average silhouette value (= Silhouette coefficient, SIL) of manual label clusters in UMAP versus original space also confirms that dimensionality reduction improved the global clustering of call types in space (SIL = 0.03 for original, SIL = 0.20 for UMAP) (Figure 7). However, the scores are low in both original and UMAP space, indicating that when looking at overall distances, the call types are not tightly clustered in any of the spaces. These low values are also in line with the visual impression of few distinct clusters in the meerkat vocal repertoire (Figure 2).

Note that all presented evaluation methods can be used to study interesting biological questions beyond call type segregation, for example, to assess grouping of calls by individual, population, sex or social status, and our code repository provides easy access to analysing these questions.

5.3 | Validation of the latent space representations

To compare the latent space representations of spectrograms with those generated from the extraction of acoustic features, we extracted 99% energy duration, cepstral peak prominence, centroid frequency, peak frequency, root mean square (RMS) bandwidth, fundamental frequency (F0) mean, F0 start, F0 mid and F0 end from the meerkat calls. We generated an additional variable $\Delta F0$ to capture the change in fundamental frequency over time by subtracting

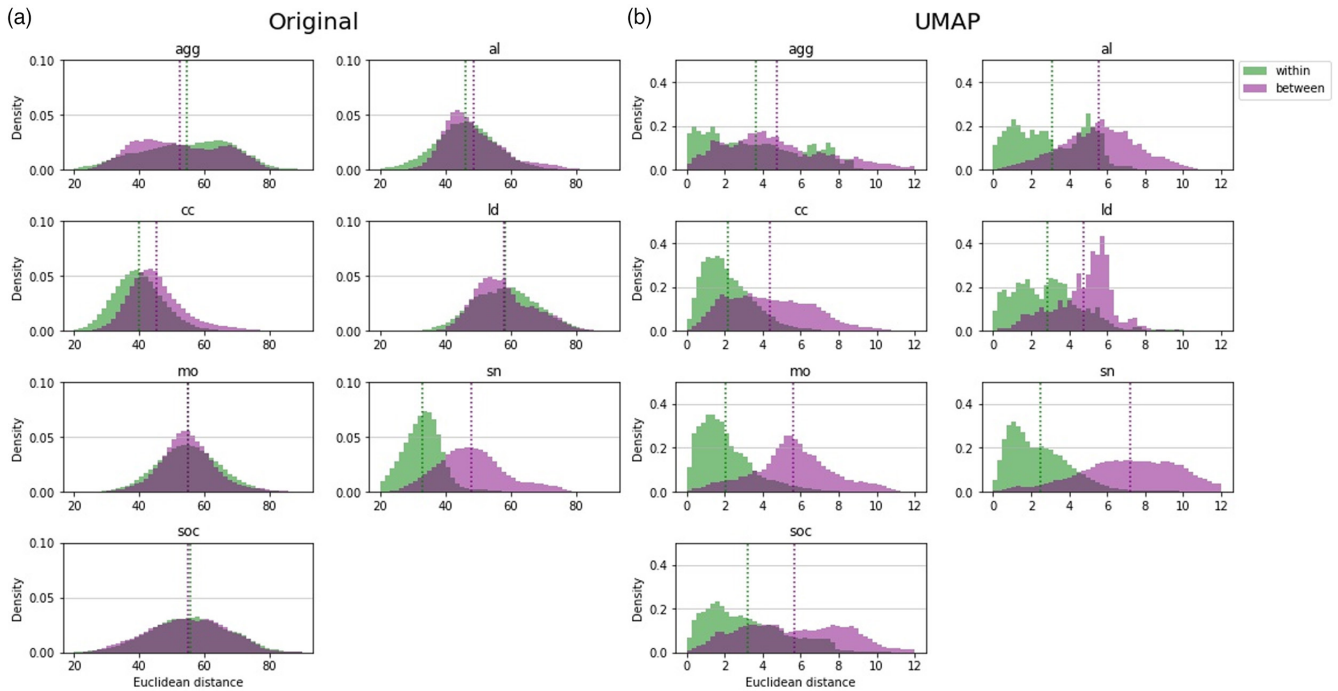


FIGURE 6 Distribution of pairwise distances between datapoints of the same call type (within) vs. between datapoints of different call types (between), shown for each call type separately and for (a) original space and (b) 3D UMAP space. Dotted, vertical lines show the mean. Note the different x- and y-scales for original vs. UMAP space.

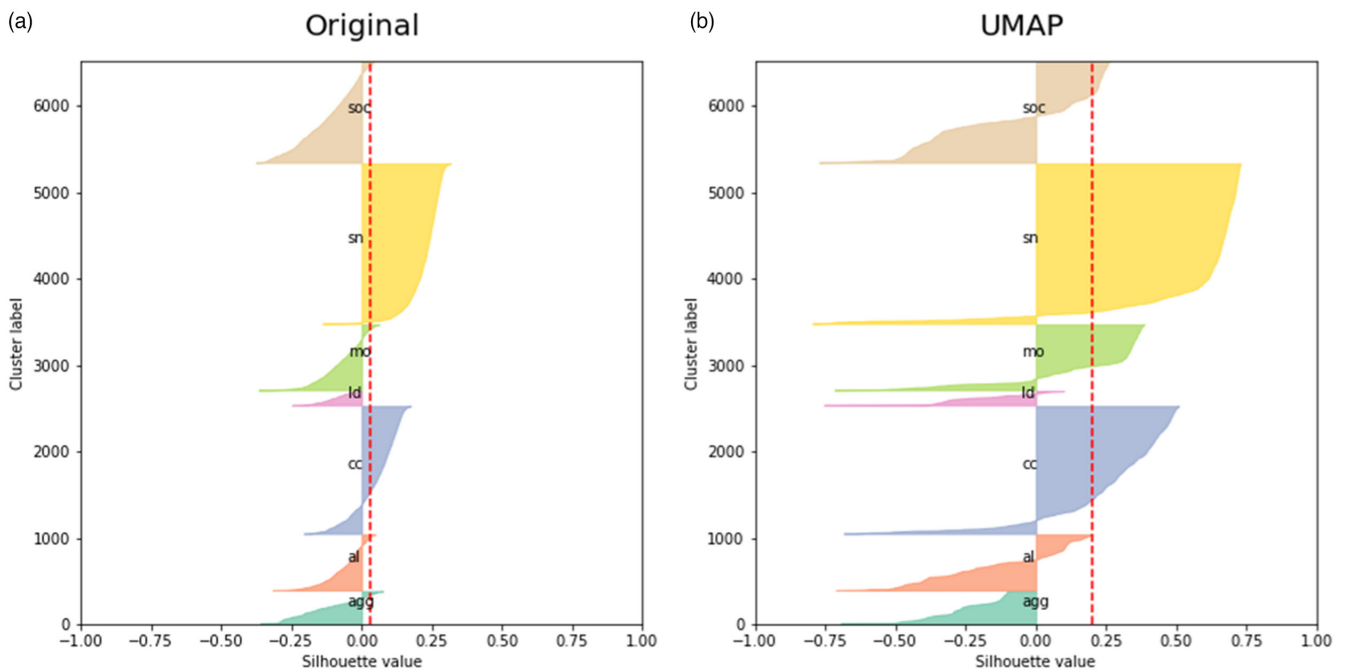


FIGURE 7 Silhouette plots for manual label clusters in (a) original and (b) UMAP space. Datapoints are sorted by label and silhouette values are displayed for each datapoint. Dotted red line indicates the average for all datapoints (SIL).

F0 start from F0 end. Signal-to-noise ratio (SNR) was not included in the analysis but was computed as a filtering step to discard weak calls (SNR < 10 dB, N = 933) for which features could not be reliably extracted. Noise level for SNR was calculated using the minimum of

the lowest RMS noise level in a 100ms window either preceding or following the signal. The remaining dataset of N = 5,495 calls represented by the eight acoustic features was z-score normalized across features and projected into 3D space using *umap-learn* (McInnes

et al., 2018) with default hyperparameters except $min_dist = 0$. For comparison, we also used the SNR-filtered dataset ($N = 5,495$) to generate the spectrogram-based embedding. Both embeddings were then evaluated based on $k = 5$ nearest neighbours and silhouette scores of manual call type labels.

Overall, the evaluation matrices of the spectrogram-based and acoustic feature-based embeddings were very similar. However, the embedding quality was higher for the spectrogram ($S = 63.38$, $S_{norm} = 2.37$) than for the acoustic feature approach ($S = 52.20$, $S_{norm} = 2.06$) (Figure 8).

When comparing the silhouette values of the manual label classes, the SIL (average over all datapoints) was higher in the spectrogram-based vs. the acoustic feature-based embeddings (SIL = 0.23 vs. SIL = 0.13). The differences in silhouette values were most apparent for *mo* and *cc* calls, which formed better clusters in the spectrogram UMAP space than in the acoustic feature UMAP space (Figure 9).

In summary, our manual selection and extraction of specific acoustic features based on expert knowledge did not lead to a better local or global clustering of call types in the feature space than

the simpler and less labour-intensive approach of using the spectrograms as feature vectors.

6 | TOOLS

The analyses presented here require a running installation of Python 3.8., Jupyter notebook and various core packages (Table 1):

For a full list of packages and dependencies, see conda environment file in Supporting Information.

6.1 | Try-it-yourself

Our example dataset of 6,428 meerkat calls (.wav files), together with a tutorial-like set of jupyter notebook files to generate UMAP representations, the interactive visualization, HDBSCAN clustering and all presented evaluations from any set of input sound files are provided in a public github repository at https://github.com/marathomas/tutorial_repo and at <https://doi.org/10.5281/zenodo.5767841>.

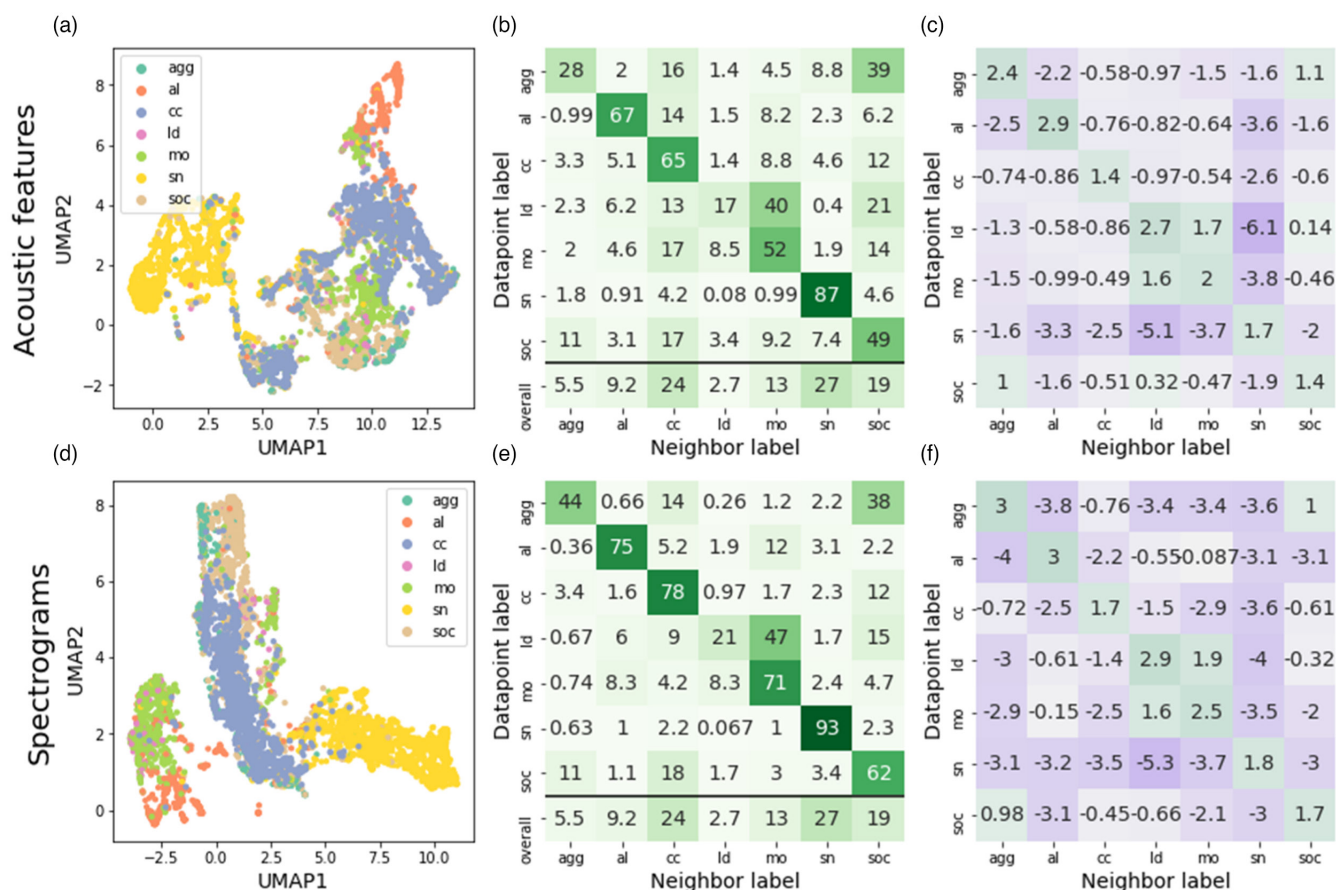


FIGURE 8 Comparison of UMAP embeddings generated with acoustic features (a–c) vs. spectrograms (d–f) as input. (a) and (d) are visualizations in 2D UMAP space. Evaluation matrices are based on $k = 5$ nearest neighbours in 3D UMAP space. (b) and (e) show the absolute probability (in percentage) of encountering a neighbour with label y within the nearest neighbours of a datapoint with label x . (c) and (f) display \log_2 -transformed ratio of that probability and the probability of encountering the neighbour label by chance. Analyses were performed with the reduced dataset, filtered for calls with $SNR > 10$ dB ($N = 5,495$). Colours are mapped on a violet-green scale from minimum to maximum value.

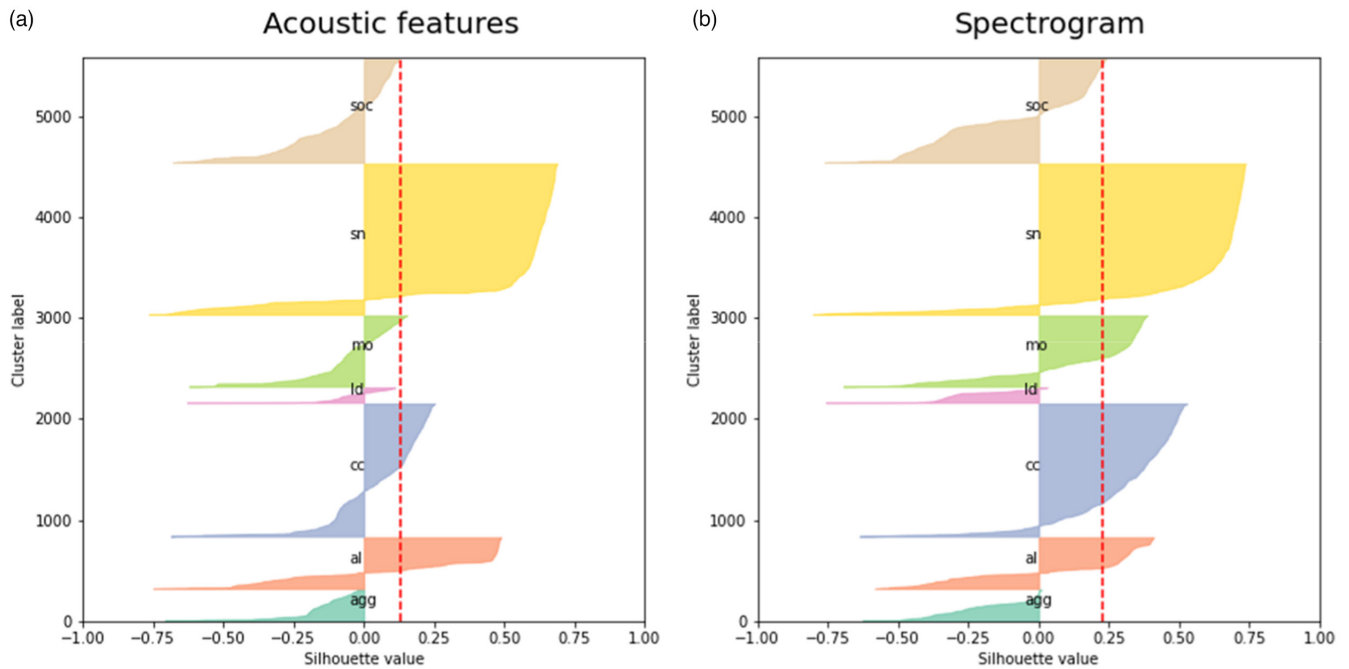


FIGURE 9 Silhouette plots for manual label classes in UMAP space generated with extracted acoustic features (a) and spectrograms (b). Silhouette values of all datapoints are represented as horizontal bars, grouped by call type label and sorted in descending order of silhouette values within each call type. Red dotted line shows average of all datapoints (SIL).

TABLE 1 Overview of the core packages needed for the analysis

Category	Name	Version	Purpose
Basic computational pipeline	pandas	1.2.4	Data handling
	numpy	1.20.1	Data handling, scientific computing
	pysoundfile	0.10.3	Reading audio data
	librosa	0.8.0	Generating spectrograms
	umap-learn	0.5.1	UMAP
	HDBSCAN	0.8.27	Clustering
	Latent space evaluation and adaptations of the basic pipeline	scikit-learn	0.24.1
pygraphviz		1.3	Neighbourhood graph
networkx		2.5	Neighbourhood graph
numba		0.53.1	Custom distance function implementation
Visualization	matplotlib	3.3.4	Plotting
	seaborn	0.11.1	Plotting
	plotly	4.14.3	Interactive visualization
	jupyter-notebook	6.1.4	Interactive visualization
	ipython	7.22.0	Interactive visualization
	ipywidgets	7.5.1	Interactive visualization
	widgetsnbextension	3.5.1	Interactive visualization
	voila	0.10.2	Interactive visualization

7 | OTHER POSSIBILITIES AND DEVELOPMENTS

The exemplary use case for unsupervised dimensionality reduction is clustering for the sake of detecting structure, for example, detecting call types in acoustic datasets. Indeed, clustering

on the UMAP representations led to high purity clusters (mostly composed of >80% of one particular call type) and can even identify biologically meaningful subtypes of calls (see Supporting Information P8). However, we found that the approach is also useful for several applications in labelled or partially labelled datasets.

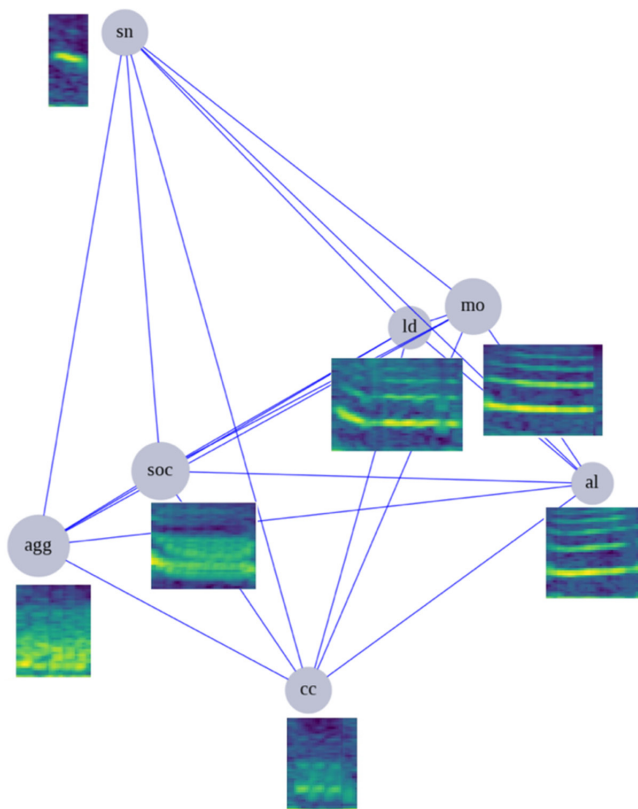


FIGURE 10 Call neighbourhood graph based on $k = 5$ nearest neighbours embedding evaluation. Call types connected by a shorter edge are more likely to be found in close vicinity to each other in the embedding. Example spectrograms are shown for each call type.

7.1 | Call type neighbourhood graph

To visualize the degree to which different call types are acoustically similar to one another, we constructed a neighbourhood graph where nodes represent call types and edges represent the probability of finding the connected call types among the $k = 5$ nearest neighbours in latent space (Figure 10). In more detail, we transformed the evaluation matrix of the normalized $k = 5$ nearest neighbour probabilities (P_{norm}) of the 3D UMAP embedding into a symmetric distance matrix, replaced each field with the average of itself and its diagonal counterpart ($M[i,j] = \text{mean}(M[i,j], M[j,i])$) and then $M[j,i] = M[i,j]$, multiplied the matrix by -1 and set the diagonal to zero. We then generated an approximation of a graph where edge length represents the distance values from the matrix using *network* (Hagberg et al., 2008) and *pygraphviz* (Hagberg & Renieris, 2004) (Figure 10). For the meerkat dataset, the resulting neighbourhood relations were in line with the perceived similarity of call types by human listeners and, to a certain extent, with the function of calls (*mo* and *ld* calls are both associated with group movement, *agg* and *soc* calls are given primarily in social interactions).

7.2 | Misclassification spotter

To test whether the local neighbourhood in latent space can be used to identify mislabelled calls, we identified calls whose $k = 5$ nearest

neighbours were all not of the same class, randomly selected $N = 100$ of these and asked two independent human experts to re-label the calls without providing any information on their previous assignment. In all, 80 of the 100 calls were indeed labelled differently than their previous assignment by at least one of the labellers and thus seem to be truly mislabelled or ambiguous. Only for a small fraction of these ($N = 21$), both labellers agreed on the new assignment ('clear cases') and 19 of these (90.48%) would also have been correctly re-assigned based on majority vote among their $k = 5$ nearest neighbours in UMAP space. For the remaining 59 calls, labellers either both agreed that these did not belong to any of the main call types ($N = 9$ calls labelled as hybrids, noise or unknown) or disagreed on their assignment ($N = 50$ calls), indicating that these calls are atypical and difficult to classify. For 12 of the 20 remaining 'false alarms' (neighbourhood in UMAP space indicated mislabelling, but both labellers agreed that their previous assignment had actually been correct), the labellers' comments indicated uncertainty about their assignment. In conclusion, while simply re-assigning call type labels based on nearest neighbour classification will likely introduce errors in the dataset, the local neighbourhood of calls can be used to identify groups of calls with a high probability of misclassification error and thus speed up the process of error detection and elimination in large datasets.

7.3 | Classification of ambiguous calls

To test the usefulness of the latent space representations for the classification of ambiguous calls, we used $N = 737$ vocal elements that had previously been excluded from the analysis because they could not be clearly assigned and had been labelled as hybrids between two types. We projected these ambiguous calls into the existing UMAP space using the *transform* function of *umap-learn* (McInnes et al., 2018) and assessed whether their $k = 5$ nearest neighbours matched both or any of the call types from which they were presumably composed.

The projected hybrid calls were, as expected, distributed across the entire latent space, as opposed to forming their own cluster (Figure 11a). In most cases, the average percentages of call types in the neighbourhood of hybrid calls were higher for those call types from which the hybrid call was composed (Figure 11b). Neighbours were also more likely to be acoustically similar to one of the hybrid labels. For example, *cc* calls were present among the nearest neighbours of *hyb:soc_agg* hybrid calls, and are acoustically similar to both types. When assigning a label to each hybrid call based on the majority call type among its nearest neighbours, 72.0% of calls were assigned to one of the designated hybrid labels, 25.8% to a completely different call type and 1.6% did not have a majority fraction (tie). When visually inspecting the 26.3% presumably mispositioned hybrid calls, the similarity between them and their nearest neighbours was evident (see Supporting Information P4). Thus, a majority vote against any of the hybrid labels does not necessarily mean the method has failed to position this call. Since the quality of nearest neighbour-based classification of novel calls depends on the quality of the labelled dataset, it is advisable to use a subset of the data

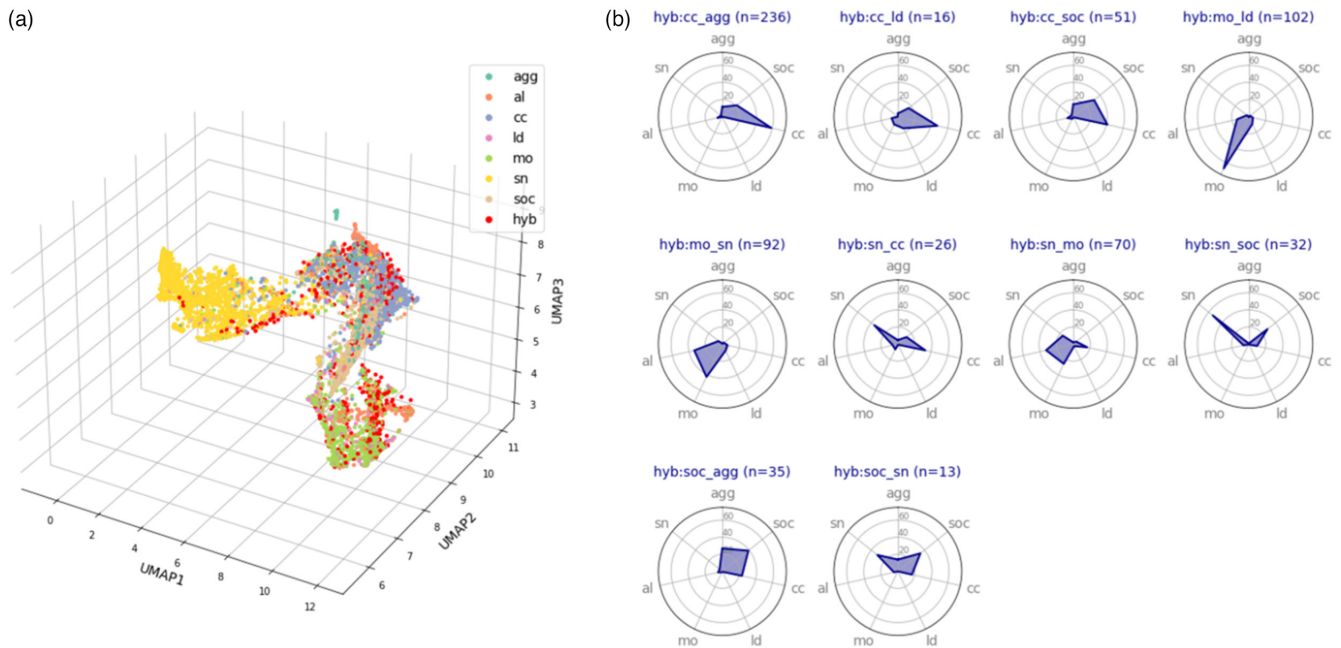


FIGURE 11 Classification of hybrid calls based on $k = 5$ nearest neighbours. (a) Shows 3D UMAP plus hybrid calls in red ($N = 993$). (b) Radar charts show average percentages of different call types present among the $k = 5$ nearest neighbours of hybrid calls. The order of call types on these charts was selected such that similar call types are next to each other. Chart titles indicate the two types that the hybrid call is presumably composed of. Data are shown only for hybrid labels with $N > 10$.

that contain only typical representatives of the classes, and/or set a threshold that only allows high-confidence assignments (e.g. 100% same-class neighbours). Note that even though specific call types are overrepresented in the dataset, normalization to the random-chance expectation is not necessary because the local neighbourhood of data points is non-random and a normalization would thus likely introduce more bias instead of reducing it.

It is important to note that all of the presented applications could also be performed on the high-dimensional dataset. In fact, this will provide better results for small k , as the nearest neighbours in original space have slightly higher same-class neighbour frequencies for the local neighbourhood ($k < 25$, Figure 5). Strictly speaking, these applications thus do not demonstrate the usefulness of UMAP, but of the idea of assessing vocalization similarity by computing the distance between spectrogram vectors. However, the differences between original and UMAP space are very minor and the use of UMAP space, while being less accurate, has the advantage that all steps can be visualized. In line with this, others have chosen to use UMAP for visualization, but performed computations in high-dimensional space (Kollmorgen et al., 2020).

8 | CAVEATS AND PITFALLS

8.1 | Meaningfulness of the distance metric

Altogether, the limitations of the latent space representations in visualizing acoustic similarity across a vocal repertoire are comprehensible considering how these representations were generated, that is, that

the approach is based on element-wise comparison of spectrograms, similar to spectrogram cross-correlation (Clark et al., 1987). This explains why certain types of signals appear similar in UMAP despite being perceived as acoustically distinct by animals and vice versa. For example, calls that are similar in shape, duration and tonality, but differ in continuity of the signal (e.g. pulsed vs. continuous signals in lead vs. move calls), will likely not be differentiated well, because they differ in only a few positions or elements of the spectrogram. In contrast, two signals of the exact same shape, but different intensity/loudness, would appear distant if the spectrograms were not normalized, which is why Decibel transformation and/or z-transformation are crucial pre-processing steps. Calls that have very similar shape but differ in duration or onset (especially calls with frequency modulation), calls with the same shape that are slightly different in frequency, calls with differences in the level and type of background noise or with different signal intensity can also appear very distant, even though they may in fact be acoustically similar. However, these issues are easy to anticipate and can be mitigated to some extent, for example, using dynamic time warping distance as distance metric or stretching all calls to the same length to account for differences in duration, by sliding spectrograms over one another to find the overlap position with minimum error (timeshift) and using fewer Mel or frequency bins to alleviate the separation of calls with small shifts in frequency. Background and impulse noise can be attenuated, and several transformations can account for varying signal intensity between recordings. While some of these adaptations require more effort (e.g. custom distance metrics for *umap* may need to be implemented), others can easily be added to the pipeline (e.g. denoising or different types of normalizations) and our provided code contains implementations of many of these

adaptations. Notably, all measures need to be carefully considered with regard to the aims of the analysis and the peculiarities of the dataset. For example, stretching or warping of calls should not be performed or be strictly constrained if duration is a biologically relevant acoustic feature. Similarly, tolerance limits for shifts in frequency depend on the vocal repertoire and the hearing abilities of the species of interest.

8.2 | Local and global structure preservation

Since UMAP favours preservation of local over global structure and is not designed to preserve pairwise distances, distance in latent space cannot be interpreted as a proxy for distance in original space, especially for moderate to large distances. Furthermore, even the local neighbourhood in high-dimensional space (i.e. the nearest neighbours) is not preserved exactly by UMAP (see Supporting Information P5) (McInnes et al., 2018). Hence, while the projection to low-dimensional space aids the visual and computational detection of clusters, many other downstream analyses (e.g. KNN classification) are more suitable to be performed in original, high-dimensional space (as done in Kollmorgen et al., 2020). However, preservation of global structure can be increased if desired, either by increasing the number of neighbours for UMAP's graph construction ($n_neighbours$) or by using Parametric UMAP (Sainburg, McInnes, & Gentner, 2020), a recent addition to UMAP that enables the variation of global and local structure preservation through a parametric balancing between UMAP loss and an additional global structure preservation loss function.

8.3 | Bias imposition through parameter tuning

The freedom in choosing hyperparameters for generating and pre-processing spectrograms, as well as for running UMAP provides both the opportunity and the risk of tuning the analysis to the researcher's needs. Assumptions about the relevance of specific properties of the signal (e.g. loudness or duration) will (and should) determine how spectrograms are generated, pre-processed and compared and thus define how acoustic similarity is assessed. It is important to be wary of this fact and not make the mistake of thinking that this (or any other) computational method can reveal 'truth' in the data without a user having defined what aspects constitute this 'truth'. In general, we recommend selecting all parameters prior to analysis and analysing the effect of specific parameters on the outcome. However, we found that our analysis was very robust to changes in pre-processing and run hyperparameters overall (see Supporting Information P7).

9 | ADDITIONAL PRACTICAL RESOURCES

In addition to our provided scripts, we recommend the original publication of the method (Sainburg, Thielk, & Gentner, 2020) and the

respective github repository: https://github.com/timsainb/avgn_paper. For information on UMAP beyond the original publication (McInnes et al., 2018), we recommend the official documentation (<https://umap-learn.readthedocs.io/en/latest/index.html>), as well as this tutorial by Andy Coenen and Adam Pearce (Google People + AI Research): <https://pair-code.github.io/understanding-umap/>.

10 | CONCLUSION

Altogether, UMAP of spectrograms can produce meaningful representations of vocalizations, which are not inferior to those generated from commonly used acoustic features and are useful for a range of downstream applications beyond visualization and clustering of call types. Due to the speed and simplicity of the approach, it can also be useful for quality control in large datasets, automated classification, as well as for answering biological questions. We recommend fine-tuning the general framework of this computational approach to the needs of the specific analysis, while being wary of imposing bias to the analysis.

AUTHORS' CONTRIBUTIONS

A.S.-P., B.A. and V.D. collected the data with support from M.B.M.; B.A., V.D. and A.S.-P. manually annotated the acoustic data with input from M.B.M. and with help from research assistants; M.T. carried out all analyses presented here with the exception of acoustic feature extraction, which was performed by F.H.J.; A.S.-P., M.A.R., F.H.J. and T.S. provided scientific input and guidance on the analyses; M.T. wrote the tutorial code and the original draft of the manuscript; A.S.-P. tested the tutorial code; A.S.-P. and B.A. performed the manual re-evaluation of potentially mislabelled calls. All authors contributed to the manuscript and gave final approval for publication.

ACKNOWLEDGEMENTS

We are grateful to the Kalahari Research Trust and Tim Clutton-Brock for the permission to work at the Kuruman River Reserve research site. We thank the Northern Cape Conservation Authority for research permission (FAUNA 1020/2016) and the managers and volunteers of the Kalahari Meerkat Project (KMP). This work was supported by HFSP Research Grant RGP0051/2019 to ASP, MBM and MAR, and funded by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy (EXC-2117-422037984). ASP received additional funding from the Gips-Schüle Stiftung, the Zukunftskolleg at the University of Konstanz and the Max-Planck-Institute of Animal Behaviour. VD was funded by the Minerva Stiftung and Alexander von Humboldt Foundation. We thank Gabriella Gall and Rebecca Schaefer for assistance with field work, as well as Richard Young, Silvan Spiess and Leonardos Leonardos for assistance with call labelling. Open Access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST

The authors have no conflict of interest to declare that are relevant to the content of this article.

DATA AVAILABILITY STATEMENT

Example data and code available from the Zenodo Repository <https://doi.org/10.5281/zenodo.5767841> (Thomas, 2021).

ORCID

Mara Thomas  <https://orcid.org/0000-0001-6713-7691>
 Frants H. Jensen  <https://orcid.org/0000-0001-8776-3606>
 Baptiste Averly  <https://orcid.org/0000-0001-7019-3213>
 Vlad Demartsev  <https://orcid.org/0000-0002-1456-9789>
 Marta B. Manser  <https://orcid.org/0000-0001-8787-5667>
 Tim Sainburg  <https://orcid.org/0000-0003-4223-2689>
 Marie A. Roch  <https://orcid.org/0000-0002-0687-2059>
 Ariana Strandburg-Peshkin  <https://orcid.org/0000-0003-2985-6788>

REFERENCES

- Clark, C. W., Marler, P., & Beeman, K. (1987). Quantitative analysis of animal vocal phonology: An application to swamp sparrow song. *Ethology*, 76(2), 101–115.
- Fletcher, H., & Munson, W. A. (1933). Loudness, its definition, measurement and calculation. *Bell System Technical Journal*, 12(4), 377–430.
- Hagberg, A., Swart, P., & Chult, D. S. (2008). *Exploring network structure, dynamics, and function using NetworkX*. Los Alamos National Lab. (LANL).
- Hagberg, A. S. D., & Renieris, M. (2004). Retrieved from <https://pygraphviz.github.io/>
- Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete Fourier transform. *Proceedings of the IEEE*, 66(1), 51–83.
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6), 417.
- Kershenbaum, A., Blumstein, D. T., Roch, M. A., Akçay, Ç., Backus, G., Bee, M. A., Bohn, K., Cao, Y., Carter, G., Căsar, C., Coen, M., DeRuiter, S. L., Doyle, L., Edelman, S., Ferrer-i-Cancho, R., Freeberg, T. M., Garland, E. C., Gustison, M., Harley, H. E., ... Zamora-Gutierrez, V. (2016). Acoustic sequences in non-human animals: A tutorial review and prospectus. *Biological Reviews of the Cambridge Philosophical Society*, 91(1), 13–52.
- Kollmorgen, S., Hahnloser, R. H., & Mante, V. (2020). Nearest neighbours reveal fast and slow components of motor learning. *Nature*, 577(7791), 526–530.
- Lostanlen, V., Salamon, J., Farnsworth, A., Kelling, S., & Bello, J. P. (2019). Robust sound event detection in bioacoustic sensor networks. *PLoS ONE*, 14(10), e0214168.
- Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov), 2579–2605.
- Manser, M. B. (1998). *The evolution of auditory communication in suricates, *Suricata suricatta**. University of Cambridge.
- McFee, B. R. C., Liang, D., Ellis, D. P. W., McVicar, M., Battenberg, E., & Nieto, O. (2015). *Librosa: Audio and music signal analysis in python*. Proceedings of the 14th Python in Science Conference, 18–25.
- McInnes, L. Retrieved from <https://umap-learn.readthedocs.io/en/latest/index.html>
- McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.
- Narayan, A., Berger, B., & Cho, H. (2020). Density-preserving data visualization unveils dynamic patterns of single-cell transcriptomic variability. *bioRxiv*.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *The Journal of machine Learning research*, 12, 2825–2830.
- Priyadarshani, N., Marsland, S., & Castro, I. (2018). Automated birdsong recognition in complex acoustic environments: A review. *Journal of Avian Biology*, 49(5), jav-01447.
- Sainburg, T., McInnes, L., & Gentner, T. Q. (2020). *Parametric UMAP: Learning embeddings with deep neural networks for representation and semi-supervised learning*.
- Sainburg, T., Thielk, M., & Gentner, T. Q. (2020). Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. *PLoS Computational Biology*, 16(10), e1008228.
- Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, 100(5), 401–409.
- Stevens, S., Volkman, J., & Newman, E. (1937). The mel scale equates the magnitude of perceived differences in pitch at different frequencies. *Journal of the Acoustical Society of America*, 8(3), 185–190.
- Tang, J., Liu, J., Zhang, M., & Mei, Q. (2016). Visualizing large-scale and high-dimensional data. In Proceedings of the 25th international conference on world wide web.
- Thomas, M. (2021). Supplement to https://github.com/marathomas/tutorial_repo/tree/v1.0. Zenodo.
- Torgerson, W. S. (1958). *Theory and methods of scaling*. John Wiley and Sons, Inc.

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Thomas, M., Jensen, F. H., Averly, B., Demartsev, V., Manser, M. B., Sainburg, T., Roch, M. A., & Strandburg-Peshkin, A. (2022). A practical guide for generating unsupervised, spectrogram-based latent space representations of animal vocalizations. *Journal of Animal Ecology*, 91, 1567–1581. <https://doi.org/10.1111/1365-2656.13754>