Package: bioacoustics (via r-universe)

October 28, 2026

Type Package

Title Analyse Audio Recordings and Automatically Extract Animal Vocalizations

Version 0.2.10

Maintainer Jean Marchal <jean.marchal@wavx.ca>

Description Contains all the necessary tools to process audio recordings of various formats (e.g., WAV, WAC, MP3, ZC), filter noisy files, display audio signals, detect and extract automatically acoustic features for further analysis such as classification.

License GPL-3

Encoding UTF-8

LazyData true

SystemRequirements fftw3, GNU make

Depends R (>= 3.3.0)

LinkingTo Rcpp

Imports htmltools, graphics, grDevices, methods, moments, Rcpp (>= 0.12.13), stringr, tools, tuneR (>= 1.3.0)

Suggests knitr, markdown, rmarkdown

URL https://github.com/wavx/bioacoustics/

BugReports https://github.com/wavx/bioacoustics/issues/

NeedsCompilation yes

RoxygenNote 7.3.3

VignetteBuilder knitr

Biarch TRUE

Author Jean Marchal [aut, cre], Francois Fabianek [aut], Christopher Scott [aut], Chris Corben [ctb, cph] (Read ZC files, original C code), David Riggs [ctb, cph] (Read GUANO metadata, original R code), Peter Wilson [ctb, cph] (Read ZC files, original R code), Wildlife Acoustics, inc. [ctb, cph] (Read WAC files, original C code), Jordan Biserkov [ctb], WavX, inc. [cph]

Repository https://cran.r-universe.dev

Date/Publication 2025-11-13 06:10:18 UTC

RemoteUrl https://github.com/cran/bioacoustics

RemoteRef HEAD

RemoteSha bb91ba7029f67c5d33a967cd008d518916a6149b

Contents

bioacoustics-package 2
blob_detection 3
fspec 6
guano_md 7
metadata 7
mp3_to_wav 8
myotis 9
plot_zc 9
read_audio 10
read_mp3 11
read_wac 12
read_wav 12
read_zc 13
spectro 14
threshold_detection 15
write_zc 18

Index **19**

`bioacoustics-package` *bioacoustics: detect and extract automatically acoustic features in Zero-Crossing files and audio recordings*

Description

bioacoustics contains all the necessary functions to read Zero-Crossing files and audio recordings of various formats, filter noisy files, display audio signals, detect and extract automatically acoustic features for further analysis such as species identification based on classification of animal vocalizations.

Details

bioacoustics is subdivided into three main components:

- Read, write and manipulate acoustic recordings.
- Display what's inside acoustic recordings, whether to plot or just extract metadata.
- Analyse audio recordings in batch in search of specific vocalizations and extract acoustic features.

To learn more about bioacoustics, start with the introduction vignette: ‘vignette("introduction", package = "bioacoustics")‘

Author(s)

Maintainer: Jean Marchal <jean.marchal@wavx.ca>

Authors:

- Francois Fabianek <francois.fabianek@wavx.ca>
- Christopher Scott

Other contributors:

- Chris Corben <chris@hoarybat.com> (Read ZC files, original C code) [contributor, copyright holder]
- David Riggs <driggs@myotisoft.com> (Read GUANO metadata, original R code) [contributor, copyright holder]
- Peter Wilson <peter@peterwilson.id.au> (Read ZC files, original R code) [contributor, copyright holder]
- Wildlife Acoustics, inc. (Read WAC files, original C code) [contributor, copyright holder]
- Jordan Biserkov [contributor]
- WavX, inc. [copyright holder]

See Also

Useful links:

- https://github.com/wavx/bioacoustics/
- Report bugs at https://github.com/wavx/bioacoustics/issues/

`blob_detection` *Blob detection of a region of interest into a spectrographic representation of the recording*

Description

This function is a modified version of the Bat classify software developed by Christopher Scott (2014). It combines several algorithms for detection, filtering and audio feature extraction.

Usage

```
blob_detection(
  wave,
  channel = "left",
  time_exp = 1,
  min_dur = 1.5,
  max_dur = 80,
  min_area = 40,
  min_TBE = 20,
  max_TBE = 1000,
  EDG = 0.9,
  LPF,
  HPF = 16000,
  FFT_size = 256,
  FFT_overlap = 0.875,
  blur = 2,
  bg_substract = 20,
  contrast_boost = 20,
  settings = FALSE,
  acoustic_feat = TRUE,
  metadata = FALSE,
  spectro_dir = NULL,
  time_scale = 0.1,
  ticks = TRUE
)
```

Arguments

wave	either a path to a file, or a Wave object. Audio files will be automatically decoded internally using the function read_audio.
channel	character. Channel to keep for analysis in a stereo recording: 'left' or 'right'. Do not need to be specified for mono recordings, recordings with more than two channels are not yet supported. Default setting is 'left'.
time_exp	integer. Time expansion factor of the recording. Set to 1 for real-time recording or above for time expanded recording. Default setting is 1.
min_dur	numeric. Minimum duration threshold in milliseconds (ms). Extracted audio events shorter than this threshold are ignored. Default setting is 1.5 ms.
max_dur	numeric. Maximum duration threshold in milliseconds (ms). Extracted audio events longer than this threshold are ignored. The default setting is 80 ms.
min_area	integer. Minimum area threshold in number of pixels. Extracted segments with an area shorter than this threshold are discarded. Default setting is 40 pixels.
min_TBE	numeric. Minimum time window between two audio events in milliseconds (ms). If the time interval between two successive audio events is shorter than this window, they are ignored. The default setting is 20 ms.
max_TBE	numeric. Maximum time window between two audio events in milliseconds (ms). If the time interval between two successive audio events is longer than this window, they are ignored. The default setting is 1000 ms.

EDG	numeric. Exponential Decay Gain from 0 to 1. Sets the degree of temporal masking at the end of each audio event. This filter avoids extracting noise or echoes at the end of the audio event. The default setting is 0.996.
LPF	integer. Low-Pass Filter (Hz). Frequencies above the cutoff are greatly attenuated. Default is set internally at the Nyquist frequency of the recording.
HPF	integer. High-Pass Filter (Hz). Frequencies below the cutoff are greatly attenuated. Default setting is 16000 Hz. A default of 1000 Hz is recommended for most bird vocalizations.
FFT_size	integer. Size of the Fast Fourrier Transform (FFT) window. Default setting is 256.
FFT_overlap	numeric. Percentage of overlap between two FFT windows (from 0 to 1). Default setting is 0.875.
blur	integer. Gaussian smoothing function for blurring the spectrogram of the audio event to reduce image noise. Default setting is 2.
bg_substract	integer. Foreground extraction with a mean filter applied on the spectrogram of the audio even for image denoising. Default setting is 20.
contrast_boost	integer. Edge contrast enhancement filter of the spectrogram of the audio event to improve its apparent sharpness. Default setting is 20.
settings	logical. TRUE or FALSE. Save on a list the parameters set with the threshold_detection function. Default setting is FALSE.
acoustic_feat	logical. TRUE or FALSE. Extracts the acoustic and signal quality parameters from each audio event in a data frame. The sequences of smoothed amplitude (dB) and frequency (Hz) bins of each audio event, temporal values (in ms) of the beginning and the end of each audio event are also extracted in separate lists. Default setting is TRUE.
metadata	logical. TRUE or FALSE. Extracts on a list the metadata embedded with the Wave file GUANO metadata extraction is not -yet- implemented. Default setting is FALSE.
spectro_dir	character (path) or NULL. Generate an HTML page with the spectrograms numbered by order of detection in the recording. Spectrograms are generated as individual .PNG files and stored in the 'spectro_dir/spectrograms' subdirectory. The R working directory is used if spectro_dir is NULL. spectro_dir is set to NULL by default.
time_scale	numeric. Time resolution of the spectrogram in milliseconds (ms) per pixel (px). Default setting is 0.1 ms for bat echolocation calls. A default of 2 ms/px is recommended for most bird vocalizations.
ticks	either logical or numeric. If TRUE tickmarks are drawn on the (frequency) y-axis and their positions are computed automatically. If numeric, sets the lower and upper limits of the tickmarks and their interval (in Hz). Default setting is TRUE.

Examples

```
data(myotis)
Output <- blob_detection(myotis, time_exp = 10, contrast_boost = 30, bg_substract = 30)
Output$data
```

`fspec` *Generate spectrograms*

Description

This function returns the spectrographic representation of a time wave in the absolute scale or in decibels (dB) using the Fast Fourier transform (FFT).

Usage

```
fspec(
  wave,
  channel = "left",
  FFT_size = 256,
  FFT_overlap = 0.875,
  FFT_win = "hann",
  LPF,
  HPF = 0,
  tlim = NULL,
  flim = NULL,
  rotate = FALSE,
  to_dB = TRUE
)
```

Arguments

`wave`	a Wave object.
`channel`	character. Channel to keep for analysis in a stereo recording: "left" or "right". Default setting is left.
`FFT_size`	integer. Size of the Fast Fourrier Transform (FFT) window. Default setting is 256.
`FFT_overlap`	numeric. Percentage of overlap between two FFT windows (from 0 to 1). Default setting is 0.875.
`FFT_win`	character. Specify the type of FFT window: "hann", "blackman4", or "blackman7". Default setting is "hann".
`LPF`	integer. Low-Pass Filter (Hz). Frequencies above the cutoff are greatly attenuated. Default setting is the Nyquist frequency of the recording.
`HPF`	integer. High-Pass Filter (Hz). Frequencies below the cutoff are greatly attenuated. Default setting is 0 Hz.
`tlim`	numeric. Specify the time limits on the X-axis in seconds (s). Default setting is `NULL`, i.e no time limits.
`flim`	numeric. Specify the frequency limits on the Y-axis in Hz. Default setting is `NULL`, i.e. frequency limits are equal to `c(0, LPF)`.
`rotate`	logical. Should the matrix be rotated 90° counter clockwise ? Default setting is `FALSE`.
`to_dB`	logical. Convert magnitude values to decibels (dB)? Default is `TRUE`.

Value

A matrix of amplitude or decibel (dB) values in the time / frequency domain.

Examples

```
data(myotis)
image(fspec(myotis, tlim = c(1, 2), rotate = TRUE))
```

---

guano_md *Read GUANO metadata in audio file*

Description

Read GUANO metadata in audio file

Usage

```
guano_md(file)
```

Arguments

file	Path to a wav file

Value

list of named metadata fields

---

metadata *Extract metadata*

Description

Extract metadata

Extract metadata from Zero-Crossing files

Extract metadata from a Wave object

Usage

```
metadata(x, ...)

## S3 method for class 'character'
metadata(x, file_type = c(file_type_guess(x), "wav", "zc"), ...)

## S3 method for class 'blob_detection'
metadata(x, ...)

## S3 method for class 'threshold_detection'
metadata(x, ...)

## S3 method for class 'zc'
metadata(x, ...)

## S3 method for class 'Wave'
metadata(x, ...)
```

Arguments

x	an object for which metadata will be extracted
...	further arguments passed to or from other methods.
file_type	type of file to read metadata from. Wav and Zero-Crossing files are currently supported.

---

mp3_to_wav *Convert MP3 to WAV*

Description

Convert an MP3 file to a Wave file

Usage

```
mp3_to_wav(file, output_dir = dirname(file), delete = FALSE)
```

Arguments

file	path to a MP3 file.
output_dir	where to save the converted Wave file. The Wave file is saved by default to the MP3 file location.
delete	delete the original MP3 file ?

myotis *Audio recording of myotis species from United-Kingdom*

Description

The myotis dataset is a Wave file of 19.73 seconds, 16 bits, mono, 10x time expanded recording with a sampling rate at 50000 Hz. It contains 20 echolocation calls of several species from the Myotis genus. The recording was made in United-Kingdom with a D500X bat detector from Pettersson Elektronik AB.

The zc dataset is a Zero-Crossing file of 16384 dots containing a sequence of 24 echolocation calls of a hoary bat (Lasiurus cinereus). This ZC recording was made in Gatineau Park, Quebec, eastern Canada, during the summer 2017 with a Walkabout bat detector from Titley Scientific.

Usage

```
myotis

zc
```

Format

Wave object

Zero-Crossing object

plot_zc *Generate spectrogram for Zero-Crossing files*

Description

Generate spectrogram for Zero-Crossing files.

Usage

```
plot_zc(
  x,
  LPF = 125000,
  HPF = 16000,
  tlim = c(0, Inf),
  flim = c(HPF, LPF),
  ybar = TRUE,
  ybar.lty = 2,
  ybar.col = "gray",
  dot.size = 0.3,
  dot.col = "red",
  ...
)
```

Arguments

x	an object of class 'zc'.
LPF	numeric. Low-Pass Filter (Hz). Frequencies above the cutoff are greatly attenuated. Default is set to 125000 Hz.
HPF	numeric. High-Pass Filter (Hz). Frequencies below the cutoff are greatly attenuated. Default setting is 16000 Hz.
tlim	numeric. Time limits of the plot in seconds (s). Default setting is set to `c(0, Inf)`.
flim	numeric. Frequency limits of plot in Hz. Default setting is set to `c(HPF, LPF)`
ybar	should horizontal scale bars be plotted. Default is `TRUE`.
ybar.lty	line type of the horizontal scale bars.
ybar.col	color of the horizontal scale bars.
dot.size	dot size.
dot.col	dot color.
...	not currently implemented.

Examples

```
data(zc)
plot_zc(zc)
```

---

read_audio *Decode audio files*

Description

Read audio files into a Wave object. WAV, WAC and MP3 files are currently supported.

Usage

```
read_audio(file, time_exp = 1, from = NULL, to = NULL)
```

Arguments

file	a Wave, WAC or MP3 recording containing animal vocalizations.
time_exp	integer. Time expansion factor of the recording. Set to 1 for real-time recording or above for time expanded recording. Default setting is 1.
from	optional. Numeric. Where to start reading the recording, in seconds (s).
to	optional. Numeric. Where to end reading the recording, in seconds (s).

Value

A Wave object.

Examples

```
filepath <- system.file("extdata", "recording.wav", package = "bioacoustics")
read_audio(filepath)
```

`read_mp3` *Read MP3 files*

Description

A thin wrapped around readMP3 from the package tuneR.

Usage

```
read_mp3(file, time_exp = 1, ...)
```

Arguments

`file`	a MP3 file.
`time_exp`	integer. Time expansion factor of the recording. Set to 1 for real-time recording or above for time expanded recording. Default setting is 1.
`...`	currently not implemented.

Value

A Wave object.

Examples

```
filepath <- system.file("extdata", "recording.mp3", package = "bioacoustics")
read_mp3(filepath)
```

read_wac *Read WAC files from Wildlife Acoustics recorders*

Description

Convert a Wildlife Acoustics' proprietary compressed WAC file into a Wave object

Usage

```
read_wac(file, time_exp = 1, write_wav = NULL, ...)
```

Arguments

file	a WAC file.
time_exp	integer. Time expansion factor of the recording. Set to 1 for real-time recording or above for time expanded recording. Default setting is 1.
write_wav	optional folder path where WAV files will be written.
...	currently not implemented.

Value

A Wave object.

Examples

```
filepath <- system.file("extdata", "recording_20170716_230503.wac", package = "bioacoustics")
read_wac(filepath)
```

read_wav *Read WAV files*

Description

A thin wrapped around readWave from the package tuneR.

Usage

```
read_wav(file, time_exp = 1, from = NULL, to = NULL)
```

Arguments

file	a WAV file.
time_exp	integer. Time expansion factor of the recording. Set to 1 for real-time recording or above for time expanded recording. Default setting is 1.
from	optional. Numeric. Where to start reading the recording, in seconds (s).
to	optional. Numeric. Where to end reading the recording, in seconds (s).

Value

A Wave object.

Examples

```
filepath <- system.file("extdata", "recording.wav", package = "bioacoustics")
read_wav(filepath)
```

---

read_zc *Read Zero-Crossing files*

Description

Read Zero-Crossing files (.zc, .#) from various bat recorders

Usage

```
read_zc(file)
```

Arguments

file	a Zero-Crossing file.

Value

an object of class 'zc'.

Examples

```
## Not run:
zc <- read_zc("file")

## End(Not run)
```

spectro *Plot a spectrogram*

Description

Plot a spectrogram

Usage

```
spectro(
  wave,
  channel = "left",
  FFT_size = 256,
  FFT_overlap = 0.875,
  FFT_win = "hann",
  LPF,
  HPF = 0,
  tlim = NULL,
  flim = NULL,
  ticks_y = NULL,
  col = gray.colors(25, 1, 0)
)
```

Arguments

wave	a Wave object.
channel	character. Channel to keep for analysis in a stereo recording: "left" or "right". Default setting is left.
FFT_size	integer. Size of the Fast Fourrier Transform (FFT) window. Default setting is 256.
FFT_overlap	numeric. Percentage of overlap between two FFT windows (from 0 to 1). Default setting is 0.875.
FFT_win	character. Specify the type of FFT window: "hann", "blackman4", or "blackman7". Default setting is "hann".
LPF	integer. Low-Pass Filter (Hz). Frequencies above the cutoff are greatly attenuated. Default setting is the Nyquist frequency of the recording.
HPF	integer. High-Pass Filter (Hz). Frequencies below the cutoff are greatly attenuated. Default setting is 0 Hz.
tlim	numeric. Specify the time limits on the X-axis in seconds (s). Default setting is NULL, i.e no time limits.
flim	numeric. Specify the frequency limits on the Y-axis in Hz. Default setting is NULL, i.e. frequency limits are equal to c(0, LPF).
ticks_y	numeric. Whether tickmarks should be drawn on the frequency Y-axis or not. The lower and upper bounds of the tickmarks and their intervals (in Hz) has to be specified. Default setting is NULL.
col	set the colors for the amplitude scale (dB) of the spectrogram.

Examples

```
data(myotis)
spectro(myotis, tlim = c(1, 2))
```

`threshold_detection` *Amplitude threshold detector above Signal to Noise Ratio (SNR)*

Description

This function is a modified version of the Bat Bioacoustics freeware developed by Christopher Scott (2012). It combines several detection, filtering and audio feature extraction algorithms.

Usage

```
threshold_detection(
  wave,
  threshold = 14,
  channel = "left",
  time_exp = 1,
  min_dur = 1.5,
  max_dur = 80,
  min_TBE = 20,
  max_TBE = 1000,
  EDG = 0.996,
  LPF,
  HPF = 16000,
  FFT_size = 256,
  FFT_overlap = 0.875,
  start_thr = 40,
  end_thr = 20,
  SNR_thr = 10,
  angle_thr = 40,
  duration_thr = 80,
  NWS = 100,
  KPE = 1e-05,
  KME = 1e-05,
  settings = FALSE,
  acoustic_feat = TRUE,
  metadata = FALSE,
  spectro_dir = NULL,
  time_scale = 0.1,
  ticks = TRUE
)
```

Arguments

wave	either a path to a file, or a Wave object. Audio files will be automatically decoded internally using the function read_audio.
threshold	integer. Sensitivity of the audio event detection function (peak-picking algorithm) in dB. A threshold value of 14 dB above SNR is recommended. Higher values increase the risk of leaving audio events undetected (false negative). In a noisy recording (low SNR) this sensitivity threshold may be set at 12 dB, but a value below 10 dB is not recommended. Default setting is 14 dB above SNR.
channel	character. Channel to keep for analysis in a stereo recording: 'left' or 'right'. Do not need to be specified for mono recordings, recordings with more than two channels are not yet supported. Default setting is 'left'.
time_exp	integer. Time expansion factor of the recording. Set to 1 for real-time recording or above for time expanded recording. Default setting is 1.
min_dur	numeric. Minimum duration threshold in milliseconds (ms). Extracted audio events shorter than this threshold are ignored. Default setting is 1.5 ms.
max_dur	numeric. Maximum duration threshold in milliseconds (ms). Extracted audio events longer than this threshold are ignored. The default setting is 80 ms.
min_TBE	numeric. Minimum time window between two audio events in milliseconds (ms). If the time interval between two successive audio events is shorter than this window, they are ignored. The default setting is 20 ms.
max_TBE	numeric. Maximum time window between two audio events in milliseconds (ms). If the time interval between two successive audio events is longer than this window, they are ignored. The default setting is 1000 ms.
EDG	numeric. Exponential Decay Gain from 0 to 1. Sets the degree of temporal masking at the end of each audio event. This filter avoids extracting noise or echoes at the end of the audio event. The default setting is 0.996.
LPF	integer. Low-Pass Filter (Hz). Frequencies above the cutoff are greatly attenuated. Default is set internally at the Nyquist frequency of the recording.
HPF	integer. High-Pass Filter (Hz). Frequencies below the cutoff are greatly attenuated. Default setting is 16000 Hz. A default of 1000 Hz is recommended for most bird vocalizations.
FFT_size	integer. Size of the Fast Fourrier Transform (FFT) window. Default setting is 256.
FFT_overlap	numeric. Percentage of overlap between two FFT windows (from 0 to 1). Default setting is 0.875.
start_thr	integer. Right to left amplitude threshold (dB) for audio event extraction, from the audio event centroid. The last FFT where the amplitude level is equal or above this threshold is considered the start of the audio event. Default setting is 40 dB. 20 dB is recommended for extracting bird vocalizations.
end_thr	integer. Left to right amplitude threshold (dB) for audio event extraction, from the audio event centroid. The last FFT where the amplitude level is equal or above this threshold is considered the end of the audio event. Default setting is 20 dB. 30 dB is recommended for extracting bird vocalizations.

SNR_thr	integer. SNR threshold (dB) at which the extraction of the audio event stops. Default setting is 10 dB. 8 dB is recommended for bird vocalizations.
angle_thr	integer. Angle threshold (°) at which the audio event extraction stops. Default setting is 40°. 125° is recommended for extracting bird vocalizations.
duration_thr	integer. Maximum duration threshold in milliseconds (ms) after which the monitoring of the background noise is resumed. Default setting is 80 ms for bat echolocation calls. A higher threshold value is recommended for extracting bird vocalizations.
NWS	integer. Length of the time window used for background noise estimation in the recording (ms). A longer window size is less sensitive to local variations in the background noise. Default setting is 100 ms.
KPE	numeric. Set the Process Error parameter of the Kalman filter. Default setting is 1e-05.
KME	numeric. Set the Measurement Error parameter of the Kalman filter. Default setting is 1e-05.
settings	logical. TRUE or FALSE. Save on a list the parameters set with the threshold_detection function. Default setting is FALSE.
acoustic_feat	logical. TRUE or FALSE. Extracts the acoustic and signal quality parameters from each audio event in a data frame. The sequences of smoothed amplitude (dB) and frequency (Hz) bins of each audio event, temporal values (in ms) of the beginning and the end of each audio event are also extracted in separate lists. Default setting is TRUE.
metadata	logical. TRUE or FALSE. Extracts on a list the metadata embedded with the Wave file GUANO metadata extraction is not -yet- implemented. Default setting is FALSE.
spectro_dir	character (path) or NULL. Generate an HTML page with the spectrograms numbered by order of detection in the recording. Spectrograms are generated as individual .PNG files and stored in the 'spectro_dir/spectrograms' subdirectory. The R working directory is used if spectro_dir is NULL. spectro_dir is set to NULL by default.
time_scale	numeric. Time resolution of the spectrogram in milliseconds (ms) per pixel (px). Default setting is 0.1 ms for bat echolocation calls. A default of 2 ms/px is recommended for most bird vocalizations.
ticks	either logical or numeric. If TRUE tickmarks are drawn on the (frequency) y-axis and their positions are computed automatically. If numeric, sets the lower and upper limits of the tickmarks and their interval (in Hz). Default setting is TRUE.

Value

an object of class 'bioacoustics_output'.

Examples

```
data(myotis)
Output <- threshold_detection(myotis, time_exp = 10, HPF = 16000, LPF = 200000)
Output$data
```

write_zc *Write Zero-Crossing files*

Description

Write Zero-Crossing files (.zc, .#)

Usage

```
write_zc(zc, filename)
```

Arguments

zc	an object of class 'zc'.
filename	path or connection to write.

Examples

```
data(zc)
filename <- tempfile()
write_zc(zc, filename = filename)
```

Index

∗ **bat**
    myotis, 9
∗ **datasets**
    myotis, 9
∗ **echolocation calls**
    myotis, 9
∗ **hoary bat**
    myotis, 9
∗ **myotis**
    myotis, 9
∗ **zero-crossing**
    myotis, 9

bioacoustics *(*bioacoustics-package*)*, 2
bioacoustics-package, 2
blob_detection, 3

fspec, 6

guano_md, 7

metadata, 7
mp3_to_wav, 8
myotis, 9

plot_zc, 9

read_audio, *4*, 10, *16*
read_mp3, 11
read_wac, 12
read_wav, 12
read_zc, 13
readMP3, *11*
readWave, *12*

spectro, 14

threshold_detection, 15

Wave, *4*, *6*, *9–14*, *16*
write_zc, 18

zc *(*myotis*)*, 9